Hovedoppgave

NTNU
Norges teknisk-naturvitenskapelige universitet

Matias Haugum

Designing a Flow-Through Particle Imaging System for Autonomous Surface Vehicles

Hovedoppgave i Elektronisk Systemdesign og Innovasjon
Januar 2022

NTNU
Kunnskap for en bedre verden

Matias Haugum

Designing a Flow-Through Particle Imaging System for Autonomous Surface Vehicles

Hovedoppgave i Elektronisk Systemdesign og Innovasjon
Januar 2022

Norges teknisk-naturvitenskapelige universitet

NTNU
Kunnskap for en bedre verden

Preface

I'd like to extend my thanks to everyone that helped me with this thesis. Thanks to my supervisor Milica Orlandić for all the feedback and assistance throughout the semester. I'd also like to thank my co-supervisors Tor Arne Johansen, Artur Zolich and Jo Arve Alfredsen for their ongoing support and allowing me to write about this particular subject. Finally, I'd like to extend my gratitude to Glaucia Moreira Fragoso for helping me with all the test-related activities at Trondheim Biological Station.

Abstract

Acquiring water samples is often done by hand, and the logistics surrounding the sample retrieval and analysis can be a time consuming process. In-situ particle imaging by the use of autonomous surface vehicles and flow-through imaging systems can significantly reduce the time to analysis, by forgoing the sample retrieval process. Many imaging systems are however not built for autonomous applications, and offer little room for customization, implementation, or come at high cost.

In this thesis, the design of a flow-through particle imaging system built for autonomous applications is presented. The design is based around a microscope, using a pump to move a water sample underneath a magnifying objective lens. A camera is used to take pictures of the sample, and a darkfield condenser offers increased contrast over conventional brightfield lighting used in other systems. Image manipulation methods are applied to remove unwanted data, and extract regions of interest in the data sets. The resulting system can sample at speeds up to 7.8mL/min and capture pictures of suspended particles with a resolution of 3.19μm/pixel. Testing showed that pumping at speed of 1.95mL/min using a channel slide of 0.8mm produced the best results. The system has a cost of approximately 1900€. Compared to the PlanktoScope with a resolution of 1.5μm/pixel, sample rate of 1.7mL/min, and cost of 500$, the system comes at a greater cost. However, with room for additional sensors coupled with a darkfield condenser, the system provides a more customizable platform catered to autonomous missions.

Sammendrag

Innsamling av vannprøver er ofte gjort for hånd, og logistikken rundt innhenting og analyse av prøver kan være en tidkrevende prosess. In-situ partikkelavbilding ved hjelp av autonome overflatefartøy og billedtakingsystemer kan redusere tiden til analyse betydelig, ved å utelate prøveinnhentingen. Mange slike billedtakingsystemer er derimot ikke designet for autonome bruksområder, og tillater lite rom for endring, implementasjon, eller har en høy pris.

I denne oppgaven blir designet for et partikkelbilledtaking-system for autonome bruksområder presentert. Designet baseres rundt et mikroskop, og en pumpe beveger en vannprøve under et forstørrende objektiv. Et kamera brukes til å ta bilder av vannprøven, og en mørkefelt-kondenser tilbyr økt kontrast sammenlignet med konvensjonelle lysfelt-kondensere. Billedmanipulasjonsmetoder benyttes for å fjerne overflødig data, og til å hente ut regioner av interesse i datasettene. Det endelige systemet kan prosessere vannprøver i en fart på opp til 7.8mL/min, og tar bilder med en oppløsning på 3.19μm/piksel. Testing viste at de beste resultatene ble oppnåd ved å pumpe vannprøven i en hastighet på 1.95mL/min, og benytte et lysbilde med 0.8mm dybde. Systemet koster ca 1900€. Sammenlignet med PlanktoScope som har en oppløsning på 1.5μm/piksel, samplingshastighet på 1.7mL/min og pris rundt ca 500$, koster systemet mer. Likevel, med rom for flere sensorer kombinert med en mørkefelt-kondenser, tilbyr systemet en mer tilpassbar platform for autonome oppdrag.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Autonomous Water Sampling

Acquiring in-situ water samples is, in many cases, necessary in many scientific disciplines. The logistics surrounding these activities often include time-consuming activities such as travelling and sample collection. If a laboratory analysis is required, the sample has to be collected from the point of interest and transported to the lab, adding additional time in transit. On top of this, expenses can add up as dedicated vessels or equipment may be required to conduct the field work. In urgent cases, such as during potential *harmful algal blooms* (HABs), a rapid, in-situ identification of the species present in these blooms is of utmost importance.

Algal blooms consist of a rapid proliferation of algal communities (including phytoplankton) that can, under certain conditions, provide detrimental effects to the water body. Such blooms can pose a major threat to marine life[1]. The detrimental effects include the production of toxins by some algae species, blockage of fish gills via grazing and/or underwater oxygen depletion due to the excess of biomass accumulated in the seafloor [2][3]. Toxins can be accumulated in the food-web and cause illness and possible death to many organisms, including humans when they consume directly the toxic seafood [4]. Such HABs can be a major inconvenience for the aquaculture industry, where large numbers of farmed fish can die in a short period of time. Microscopic identification and quantification of algal species is a tedious, time-consuming process. At the lab, the sample will have to be analysed by a competent taxonomist. In the case of a harmful algal bloom (HAB) close to a fish farm, costly measures must be undertaken in order to negate or minimize the loss of fish. This can include moving the fish elsewhere or harvesting it prematurely.

Work has been put into developing systems that can detect signs of algal blooms. Satellites equipped with hyperspectral imaging equipment such as ESAs Sentinel[5] satellites have proven capable of detecting blooms, and can estimate *chlorophyll* concentrations based on data from the hyperspectral sensor. Monitoring systems meant to be installed on ships called *FerryBoxes* can periodically sample water and measure any parameter as long as the appropriate sensor is installed. But even if these systems are capable of detecting signs of algal blooms, they are incapable of identifying the underlying species.

In this thesis, a design for a system that can automatically take pictures of microscopic[1] particles is presented. The design uses a pump to move a water sample underneath a magnifying microscope objective, and a camera takes pictures at high frame rates while the sample is being processed. Post-processing methods to remove unwanted data condense the data set into a smaller subset, containing only objects of interest. Additional image manipulation methods are presented to improve or extract specific features from the data set. The system is designed to work with minimal user input, with the intention that the system can be used for unmanned missions on autonomous vehicles.

The goal of the design is to enable for in-situ imaging of marine microscopic organisms such as *phytoplankton* and *zooplankton*. Quick verification of potential harmful algal blooms can be of great interest to the aquaculture industry, as they must commit to urgent and costly measures in the event of an HAB to prevent fish death. The images produced by the system could also form a basis for machine learning data sets.

[1] Particles that cannot be seen with the naked eye.

1.2 Background

Many systems for imaging suspended particles already exist, and come in a wide range of operation principles depending on the needs of the end user. These systems are often delivered as complete products and offer little room for customization and integration into other systems. Certain systems are manually operated, and require user input to process samples. Furthermore, the closed-source nature of these systems complicates the process of adding additional features or sensors to cater to specific missions. Depending on the application, certain requirements to size, weight, power consumption and system performance must be met to ensure a successful integration. For autonomous applications, restrictions to size and power consumption are typically imposed due to the limited space and energy available in these vehicles.

The imaging of microscopic organisms such as phytoplankton and zooplankton requires magnification, as the size of these organisms can range from less than $1\mu m$ to greater than $200\mu m$. A system built for this task should thus be able to properly image particles within this size range. For particles close to the lower end of this range, certain features may be hard to discern if the image resolution of the system is low. *Micrometers*[2] *per pixel* [μm/pixel] is a metric often used with systems capturing images under magnification, with a lower number correlating to a higher resolution. Objective lenses provide the needed magnification to image microscopic particles, but are dependent on being in the correct *working distance* from the object in order for it to appear in focus. During manual operation, the focus is easily adjusted by the user, but unmanned autonomous missions necessitates the capability of adjusting the focus automatically. Furthermore, a large amount of pictures may be generated during sampling, and depending on the particle concentration in the sample, many of these frames may be empty. These images can take up a lot of space on the systems storage device, and for autonomous missions, it's imperative to not waste power and bandwidth on transmitting useless data. For this reason, methods of trimming and removing unwanted data will be necessary for unmanned missions.

1.2.1 Integration into an Unmanned Surface Vehicle

The AutoNaut is an *unmanned surface vehicle* (USV). It is capable of operating autonomously and independent for long periods of time, with the limiting factors including biofouling of the hull, and to which degree the on-board batteries can be recharged, which is dependant on the weather and season. The weight is distributed in such a way that the AutoNaut is self-righting should it capsize, and the internal components are safely stored inside watertight compartments. These qualities make the AutoNaut a great platform for long, autonomous missions requiring minimal supervision. The AutoNaut is self-powered, utilizing the ocean waves for propulsion using spring-loaded struts. If the waves themselves cannot propel the AutoNaut, an electric thruster can be used instead. Solar panels on top of the vessel provide power to the on-board components, as well as the capability to charge the on-board batteries. The self-powered nature of the AutoNaut combined with its power-generating capabilities, makes it a great zero-emission, long endurance data collecting platform [6]. The AutoNaut is shown in figure 1.

[2] Also referred to as microns.

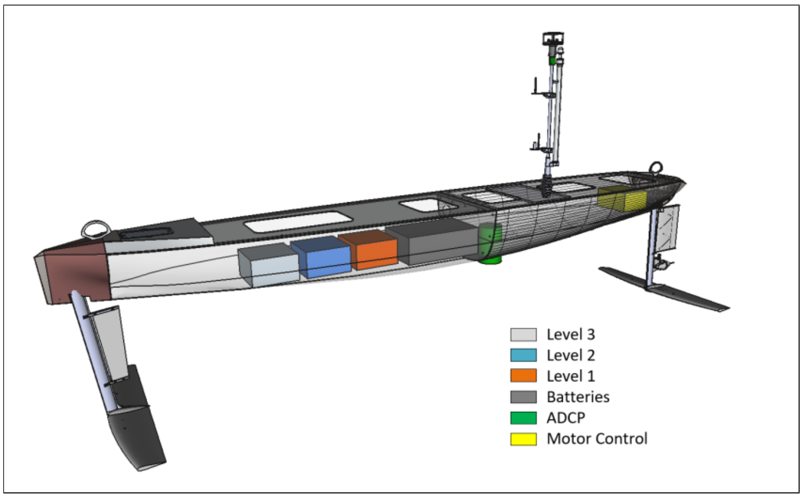


Figure 1: The AutoNaut and its internal components. Collected from [7].

The AutoNaut is considered a viable platform in which to install the system described in this thesis. It has a single board computer dedicated to scientific experiments, as well as radio communication systems to transmit data.

1.2.2 Existing Imaging Instruments

Imaging instruments existing today range from low-cost, citizen grade systems such as the *PlanktoScope*, to highly advanced industrial grade flow cytometers[3] such as *CytoSense*[8]. These vary in price, with the material cost for the PlanktoScope being around the $500 range, whereas the more expensive flow cytometers can cost upwards of $160 000. Examples are listed in table 1.

Instrument	Size range	Sample Volume	Condition of use	Method	Approximate cost
HIAC	$1.3 - 600\mu m$	25-100mL/min	On board/ laboratory	Optical	26,000€
Imaging Flow-CytoBot	$0.2 - 100\mu m$	$5mL(15ml/h)$	Laboratory	Optical	$158,000
FlowCam Nano	$300nm - 30\mu m$	$20\mu L/min$	On board/ laboratory	Imaging	$117,000
UVP-6LP	$60 - 20,000\mu m$	-	In situ	Optical/imaging	15,000€
ISIIS	$60\mu m - 130mm$	150L/s at 5 kts	In situ (towed)	Focused shadowgraph	$100,000
CytoSense	$10 - 80\mu m$	0.5-5ml	In situ/ on board/ laboratory	Optical	90,000-130,000€

Table 1: Other imaging systems with their mode of operation and price. Data collected from [9].

For this design, comparisons in price and performance are made to citizen grade instruments such as the PlanktoScope. The PlanktoScope is presented in section 1.2.3.

[3] Instruments that measures number and characteristic of cells

1.2.3 PlanktoScope

The **PlanktoScope** is an open-source imaging platform which enables sailors (citizen scientists) to look at small organisms without having to use a microscope. It is intended for both researchers and non-researchers, and is designed around readily available parts in order to be a low cost solution for anyone that wants to perform oceanography. Built around the affordable computer *Raspberry Pi* and assembled using laser-cut parts from whichever material is desired, the cost of the PlanktoScope is roughly $500 [10]. The PlanktoScope is shown in figure 2.

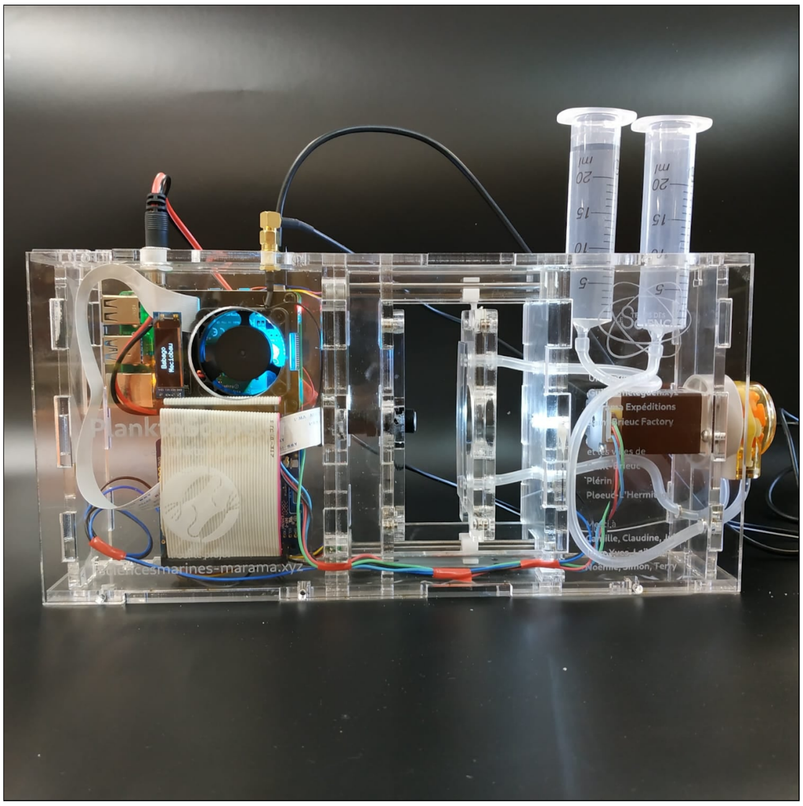
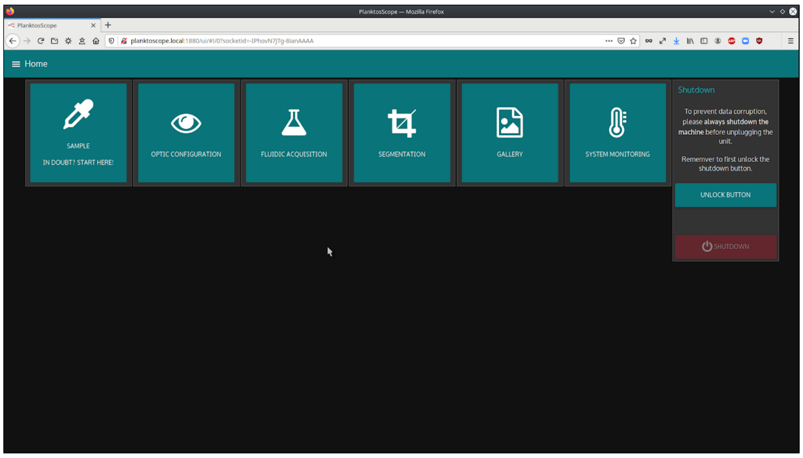


Figure 2: A PlanktoScope assembled in clear plastic, built by Guillaume Le Guen. Collected from [10].

The PlanktoScope offers a resolution of $1.5\mu m$ and is capable of imaging a volume of $1.7ml/min$ using a microscope slide with a depth of $500\mu m$. The system is built around the Picamera, which is developed specifically for the Raspberry Pi and connected using a ribbon cable. A lens with a low field of view (FOV) is used to provide magnification. The camera and lens can be seen in the center of figure 2. The lens is focused on a microscope slide mounted on two rails, which in turn are controlled by two stepper motors. Actuating these stepper motors changes the distance between the lens and the slide, adjusting the focus. The two syringes serve as an inlet and outlet for the water sample. A peristaltic pump provides control over the sample, and can pump the sample in both directions and at different speeds. A white LED is used to illuminate the sample. The Raspberry Pi isn't capable of powering and controlling all the steppers by itself, so HAT's[4] provide additional power and output pins for the components [11].

The PlanktoScope has its own dedicated operating system which makes sampling easier for the end user. In this interface, the user can adjust camera-related parameters, as well as the focus and control the pumps. The PlanktoScope broadcasts a WiFi-network to which the end user can connect, and the GUI can be accessed in a web browser. This GUI is shown below in figure 3.

[4] Hardware Attached on Top



To prevent data corruption, please **always shutdown the machine** before unplugging the unit.

Rememver to first unlock the shutdown button.

Figure 3: The graphical user interface of the PlanktoScope, which is accessible through a web browser. Collected from [10].

Additional features of the PlanktoScope includes labeling the data with GPS coordinates, as well as image segmentation built around the *MorphoCut* image segmentation library for Python [12]. The image segmentation software automatically extracts regions of interest from the images taken. The PlanktoScope is a low-cost and accessible way to perform microscope imaging. However, this puts a restriction on the image quality of the system. As it is built around the Picamera, there is not much that can be done in terms of the optical line and image quality. The white LED used for lightning provides grayscale-like pictures with low contrast. Changing the magnification involves changing the lens, and the hardware is not suited for sampling at higher frame-rates. This in turn reduces the amount of throughput, and can lead to image artifacts such as motion blur. The PlanktoScope is also developed in Python for the Raspberry Pi. Python is an easy programming language to develop software with, but is not considered as fast as other programming languages, such as C++. In addition, the Raspberry Pi is not a particularly powerful piece of hardware, and an increase in either resolution or frame rate would further strain the limited processing power of the computer.

1.3 System Requirements

The system needs to be able to image microscopic particles with sizes ranging from $10\mu m$ to $500\mu m$. To enable this, magnification is achieved using objective lenses, and a camera captures images of the particles. A pump moves the water sample from an inlet to the objective lens. This pump must be able to move the fluid at different speeds and directions. The use of a darkfield condenser makes objects in focus appear brightly lit against a dark background, increasing contrast as opposed to when using brightfield condensers. The increase in contrast can simplify the identification of species, due to their characteristic traits becoming more observable. As the system is meant for use on unmanned vehicles, it is desirable to control the focus of the objective, allowing for refocusing should the objective move. This means that the distance between the objective lens and the sample should be variable. Any physical peripherals such as the pump should be controlled by a microcontroller, allowing for the system to be easily integrated with other systems by the means of serial communication. Furthermore, the system should be customizable, being able to change magnifications and sensors based on the needs for the mission. Finally, to prevent the system from using unnecessary power and bandwidth, the system should extract features of interest from the data prior to storage and transmission.

2 Theory

2.1 Algae and Blooms

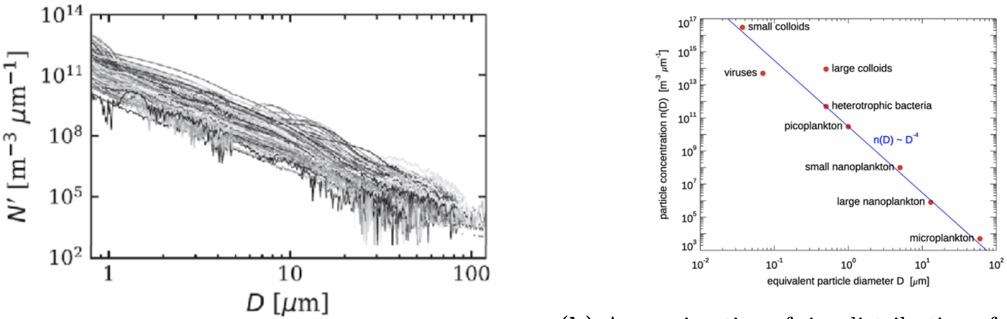
2.1.1 Different types of Algae

Plankton are often divided between *Phytoplankton* and *Zooplankton*. Most phytoplankton (for instance, diatoms) are *autotrophic*, meaning they produce their energy through photosynthesis. Other phytoplankton, such as dinoflagellates, are mixotrophic, meaning that they use both nutritional modes, autotrophic and heterotrophic. Zooplankton are *heterotrophic*; they can not produce their own energy and rely on consuming other plants or organisms to survive. Plankton form an important part of the marine ecosystem. Although they vary vastly in size, many phytoplankton and some zooplankton are typically microscopic and not visible to the naked eye. Mesozooplankton and macroplankton are typically larger parts of the plankton, and can feed on the smaller phytoplankton. Phytoplankton come in different groups, with the most common *taxa*[5] being *diatoms*, *cyanobacteria*, *dinoflagellates* and *coccolithophores*. These are also known as *phytoplankton functional types*, which means that they have distinct functional roles in the marine environment. Their distribution depend primarily on environmental factors such as temperature and nutrients, as well as predation and competition with other organisms [13]. The size of phytoplankton range from less than $1\mu m$ to larger than $200\mu m$, and they are divided into three groups: picoplankton ($0.2 - 2\mu m$), nanoplankton ($2 - 20\mu m$) and microphytoplankton($20 - 200\mu m$)[13]. Phytoplankton have an important role in the carbon and nutrient cycles of the ocean. The phytoplankton bind carbon into organic material, and are consumed by larger creatures such as zooplankton and fish [13]. Zooplankton are in general larger than phytoplankton. These can be divided into 4 groups: microzooplankton ($20-200\mu m$), mesozooplankton ($0.2-20mm$), macrozooplankton ($20-200mm$) and megazooplankton ($> 200mm$) [14]. As heterotrophic creatures, they feed on phytoplankton and other zooplankton and are an important link in the food chain.

An important observation regarding size of both phytoplankton and zooplankton is that the smaller the organism is, the more abundant it is. This relation follows the *power law*, shown below in equation (1).

$$n_d(d_p) = Ad_p^m \tag{1}$$

In equation (1) the number of particles n_d as a function of particle size d_p, is given by the power law where A and m are constants. For $m = -4$, this distribution is called the *Junge-distribution*[15]. This relation is also applicable to marine species and particles, as shown below in figure 4a. This distribution is often plotted on logarithmic axes, such that the slope of the curve is equal to m. Figure 4b shows an approximate distribution of marine particles.



(a) Particle number N' as a function of particle diameter D, from arctic waters. Collected from [16].

(b) Approximation of size distribution of marine organisms. Figure collected from [15], with data originating from [17].

Figure 4: *Junge*-curves depicting particle size in marine environments.

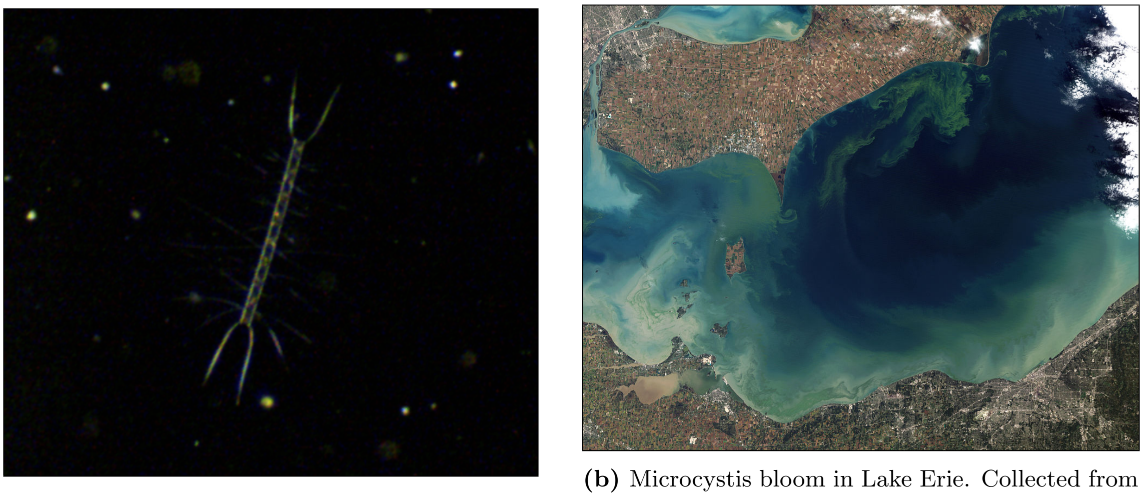
[5] plural of taxon, unit used in the science of classification, taxonomy.

2.1.2 Algal Blooms

An algal bloom is an aquatic phenomenon in which phytoplankton increase in numbers at a very fast rate. A bloom can occur as a result of several ecological factors, with examples being runoff from agriculture, which provides nutrients for phytoplankton to grow, increased sunlight due to calm weather, or a combination thereof [18]. These events create ideal growth conditions for the algae. Excessive use of land fertilizers can contribute to eutrophication, which happens when these nutrients are runoff and end up in bodies of water or the sea [19]. This runoff in addition with sewage being discharged into the sea, introduces a lot of inorganic nutrients (ammonium, nitrate and phosphate) to the water which the algae takes up. An increase in sunlight promotes more photosynthesis, and a calm weather promotes a shallow mixing water layer, proving more sunlight for the algae to accumulate at the surface.

Depending on the species, algal blooms can be detrimental to marine life. These blooms are called Harmful algal blooms, or HABs in short. Certain species have physical characteristics that are harmful to fish. As an example, certain species of the *Chaetoceros* diatoms (figure 5a) have physical characteristics that can cause harm to fish. The *Chaetoceros* diatoms have long setae; small hair- or bristle-like features that may cause irritation to the gills of the fish. This can lead to bacterial infections, and an increased mortality rate in the fish population [3]. The size of the *Chaetoceros* algae can vary from less than $10\mu m$ up to 50 μm.

Others produce toxins that can do harm in great concentrations. These toxins can also accumulate in filter feeders such as mussels, and cause harm to anyone that ingests them [4]. These HABs can be of great threat to marine life, and have proven to be a major and expensive nuisance for the aquaculture industry. In 2001, 680 tons worth of farmed fish died in in southern Norway due to an algal bloom [20]. The fish died quickly and could not be harvested due to the rapid onset of the bloom. Despite sometimes being observable from orbit (figure 5b), it's difficult to determine whether an algal bloom is harmful or not. As previously stated, whether a bloom is a threat or not depends on which species of algae (or cyanobacteria) the bloom consists of.



(a) Chaetoceros diatom.

(b) Microcystis bloom in Lake Erie. Collected from [21].

Figure 5

Determining whether an approaching bloom is harmful or not requires identification of the species. This is done by biologists (many taxonomy experts) looking at a sample underneath a microscope. In-situ identification could speed up the quantification time dramatically, and an automatic sampling-and segmenting system can provide a good foundation to generate datasets out of water samples. These datasets may be used to train neural networks, such that the identification can be performed without human supervision. Other HAB-warning methods include orbital imagery using hyperspectral cameras. These cameras can be used to estimate the concentration of chlorophyll-a, which can give a sound estimate of the algae concentration. A FerryBox is a system which periodically samples water and measures specific parameters using sensors. Such a system may be used to preemptively detect algal blooms, measuring physical parameters such as temperature, oxygen content and pH. Direct measurement of chlorophyll-a is also possible. However, like the orbital imagery, the FerryBoxes are not capable of identifying the underlying species.

2.2 Color Space Transformations and Image Manipulation Techniques

2.2.1 HSV Color Space

A *color space*, or color model, is a way of representing colors. The most common and widely known color space is the RGB-color space, often used for displays such as TV monitors and computer screens. In this color space, each color is represented as a sum of three basis colors; red, green, and blue. RGB is an *additive* color space, which means that any color can be represented as a sum of these three base colors. The absence of any color will represent black, whereas every color being fully present represents the color white. The number of presentable colors depends on how many discrete values are used to represent each of the basis colors. When storing images on a computer, a common image format is the *RGB888*-format. In this format, 8 bytes are used to represent each of the basis colors for each pixel. This means that every basis color, or *channel*, can have a value between 0 and 255. With three channels, the total amount of discrete colors presentable is equal to $256^3 = 16777216$ colors. The RGB color space is well suited for displaying images to humans. In RGB screens, pixels are commonly composed of three *subpixels*, one subpixel representing each of the basis colors. As they are closely grouped together, the subpixels appear as a single color to any human that is perceiving them.

The RGB color space is great for displaying images on screens, but less suited for task related to computer vision. In the RGB color space, the *luma*, or brightness of each pixel, is represented as the sum of each of the channels. This is a problem if we would like the computer to detect anything that appears 'red'. We could specify a certain range of the red channel that we would like to classify as our color, but would also have to take the other channels into account. If the lightning would change, the RGB-values could fall out of our desired range. Color spaces such as the HSV-color space is made to alleviate this issue. HSV stands for Hue, Saturation and Value. In this color space, the **hue** represents a specific color, and the colors of the rainbow are distributed in the hue-interval. The advantage of this is that a color range (such as every color between orange and green) can easily be represented as an interval. Using the HSV-color space thus separates the color-information from the brightness-information. The **saturation** is a merit of how "pure" the color appears. A saturation value of 100% would mean that a color would appear brilliant with vivid and rich colors. Decreasing the saturation is comparable with mixing more white into the color, until the color is completely white at a saturation value of 0%. Finally, the **value** of the color depicts how dark it would appear. A value of 0% is a completely black color, and a value of 100% is a brightly lit color. The HSV color space can be represented as an upside-down cone shown in figure 6. In figure 6, the hue is represented as an angle along the top of the cone. The red color is found at 0°, green colors are found around 120°, blue colors at 240°, until wrapping back around to red at 0°. The saturation is the radius of this circle, and the value is illustrated as the height of the cone.

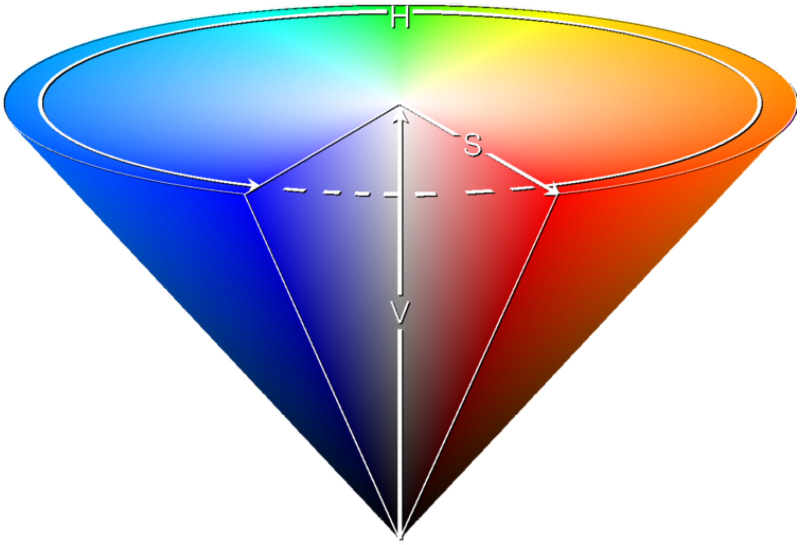


Figure 6: HSV Color Model illustrated as a upside-down cone. Collected from [22]

By converting to the HSV color space, the darkfield condenser in the system can be utilized to its full extent. The darkfield condenser provides more contrast compared to a brightfield condenser, which means that colors are easily discerned from each other. As stated earlier, when transforming to the HSV color space, color ranges can be expressed as hue-intervals. Species can thus be distinguished from each other based on their color. Phytoplankton that perform photosynthesis such as certain species of the *Dunaliella* genus have a bright green appearance. Larger zooplankton can have a orange-ish hue, and certain species have carapace with a transparent, white look.

Image processing libraries such as *OpenCV* [23] comes with functions that can convert between color spaces, as well as locate any pixel within a certain hue, saturation or value range. From these pixels, a *mask* can be created to simplify operations based on these colors. Inversely, any colors deemed "not natural" can be removed from the picture. An example of this can be artificially colored objects such as small strands of polyester rope. Ropes like these are used everywhere in marine environments such as kelp farms, fish nets, for mooring boats, and so on.

2.2.2 Averaging Frames to Remove Stuck Particles

With samples being continuously pumped trough the microscope slide, particles or specimen run the risk of getting stuck to the microscope slide. When recording at a frame rate of up to 300 images per second, this can lead to the same stuck specimen being photographed multiple times. A suggested solution to this issue is to subtract the average of the last n frames from the current frame. This is done by storing the last n-frames in an array, calculating the average frame from this array, and subtracting the average frame from the most recent frame. This method will remove any pixels that are not in motion.

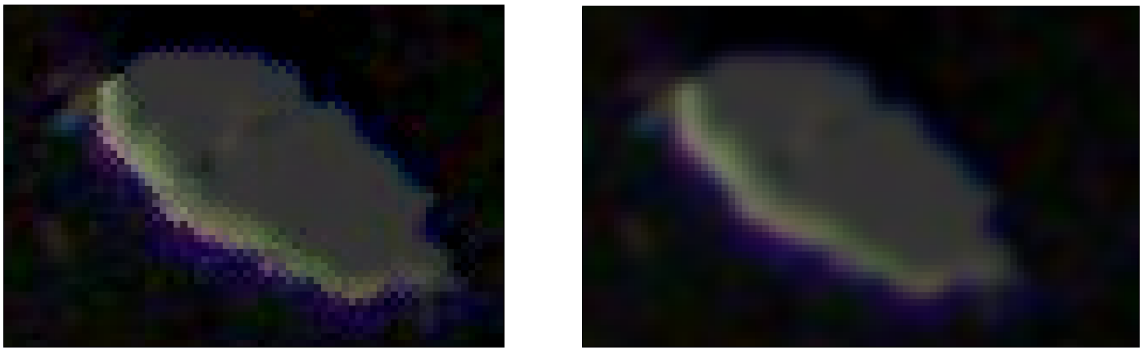
2.2.3 Removing Pixel Noise

When differentiating colors using a hue-interval, applying a *Gaussian blur* to the image may improve the the results by reducing *pixel noise* in the image, and average the hue of a pixel based on the hue of the surrounding pixels. This is done by performing convolution between the kernel and the frame. The kernel is a square matrix with an odd-numbered side length, such as 3 or 5. When using a Gaussian kernel, the values of every matrix element is distributed according to the Gaussian distribution, hence the name. An approximation of a Gaussian kernel of size 3 is shown in figure 7.

$$\frac{1}{12}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

Figure 7: Gaussian Kernel of size 3.

Convolution between the image and kernel will soften any edges in the picture, making the picture appear less sharp. This is shown below in 8.



(a) Microscope particle, unblurred. **(b)** Particle with Gaussian blur applied.

Figure 8: Image of a particle before and after a Gaussian blur filter is applied to the image. Notice the reduced amount of pixel noise in figure 8b. A kernel size of 5 is used.

2.2.4 Removing Empty Frames

Depending on the nature of the sample, a lot of frames may appear empty. This can occur due to a low concentration of specimen in the sample. Empty frames are of little interest, and filtering these out will decrease the post-processing time after a sample has been acquired. An simple and computational-efficient way to derive whether a frame is empty, is to calculate the *norm* of the frame. Calculating the norm returns a single value which can be interpreted as the "brightness" of the image. Calculation of the *L2-Norm* is shown in equation (2). The L2-norm of a frame is the square root of the sum of every array value (pixel) I squared.

$$\|frame\|_{L2} = \sqrt{\sum_{I} frame(I)^2} \tag{2}$$

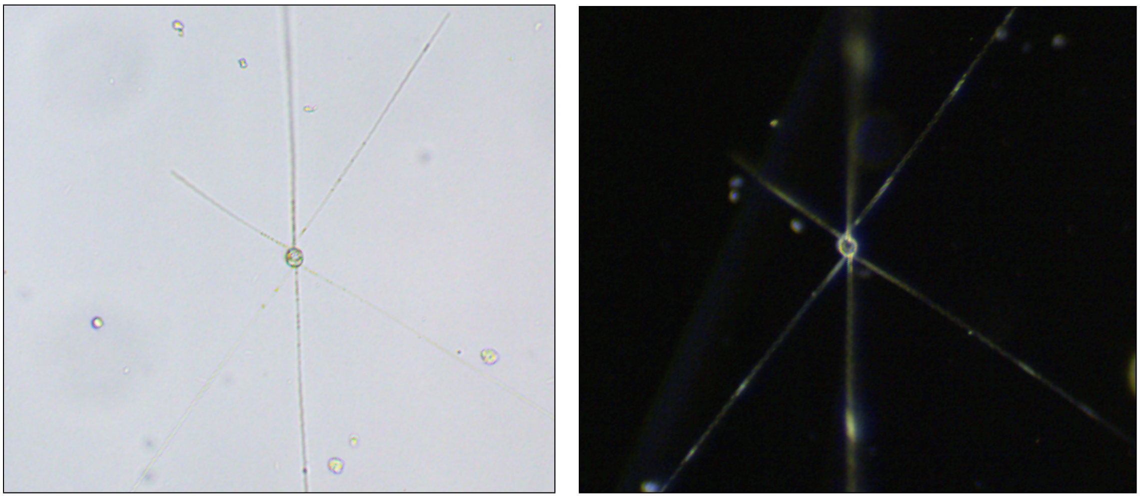
Using a darkfield condenser, a completely black background translates to an empty image, and a high relative norm would translate to a frame being populated by particles or specimen. With brightfield microscopy, the inverse is true. A low relative norm would mean that a lot of particles are casting shadows on the sensor, and the frame is thus populated. Despite there being meaning in removing empty frames straight away, particles accumulating in the microscope slide should be filtered away using the averaging-method mentioned above before the norm is calculated. This way, the norm of the frames doesn't increase as the sampling progresses and the slide gets dirtier.

3 Design

3.1 Proposed Improvements to the PlanktoScope

The PlanktoScope is a complete, viable product readily accessible for anyone who wants to do oceanography. However, extended functionality is limited by the choice of optics and camera. Should the end user desire a higher resolution, greater magnification, or higher frame-rate, changes would have to be made to the camera and optics. In addition, the PlanktoScope requires user input in order to pump, focus, and capture images. By using a camera that is able to capture video at high frame rates, data can be extracted from the frames while the sample is circulated continuously past the sensor.

Using a microscope as a platform is recommended. Most microscope cameras are designed so that they can easily be fixed to the oculars of the microscope. Some microscopes have an additional ocular dedicated for a camera[6]. The microscope provides a sturdy platform, as well as the possibility to change magnifications should the microscope be equipped with multiple objectives. The focus is easily manipulated using the coarse- and fine-adjustment knobs. Microscopes use *light condensers*, a part which focuses the incident light onto the specimen in focus. Normal condensers (brightfield) provide a brightly lit background, and the object is perceived as a dark shadow against a white background. Using a *darkfield*-condenser, the light is directed in such a way that it illuminates the object from the sides, while providing a black background. This provides additional contrast, and also retains more colour in the image. This is shown in figure 9.



(a) Using a brightfield-condenser.

(b) Using a darkfield-condenser.

Figure 9: A particle imaged with a brightfield and darkfield condenser, using a 4x scanning objective.

3.2 Flow Components

The flow-related components enable the water samples to move around in the system. These components should facilitate moving the samples in both directions, and at different flow rates. The samples are pumped from the sample-holders through flexible silicone tubing to the microscope channel slide, on which the microscope can easily focus onto. Since the samples are pumped through microscope channel slides whose depth can be as small as $200\mu m$, facilitating very low flow rates such as less than 1mL/min is desirable. At higher flow rates the objects of interest may move too quickly past the microscope objective, reducing the quality of the pictures. With this in mind, great care should go into deciding on which pump to acquire.

[6] Often called *Trinocular* microscopes.

3.2.1 Pump

When selecting a pump, there's a great variety to choose from. Pumps can come in several different operating principles, depending on the needs of the end user. These operating principles are usually divided between *reciprocating* and *rotary* pumps, each with their advantages and drawbacks. Reciprocating pumps operate by using a piston or a membrane that moves back and forth, and pumps water in a cyclic manner. Rotary pumps use rotating parts such as interlinked gears, which provide a steady flow of water compared to the reciprocating pumps. The gears in these pumps are however subjected to wear and tear, and one should avoid to run them without a fluid present. In addition, these pumps are often not *self-priming*, which means that they are unable to produce the suction required to lift the fluid into the pump, should the pump be mounted above the fluid.

In this design, the flow rate and degree of control over the pump should be prioritized. Dosing pumps[7] are pumps specifically built to deliver small and accurate amounts of fluid, often used in industry and medical laboratories. The flow rate of these pumps can be as low as in the μl-range, and depending on the motor configuration, can be run in both directions at different speeds. This is done by using *stepper motors* as drives. A conventional DC-motor can also be used, but does not offer the same level of control as a stepper motor. As the name implies, stepper motors are capable of rotating step by step in either direction. By varying the frequency of the step signal, the rotation speed is changed. This stepping manner gives one precise control over how much fluid is dispensed for a certain amount of steps. Certain stepper motor drivers also enable *microstepping*, which increases the resolution of the stepper motor.

A well suited rotary pump topology for this design is the peristaltic pump. The name originates from *peristalsis*; the series of muscle contractions that move food through the digestive tract. A peristaltic pump works in a similar manner by squeezing a flexible tube with rollers, pushing the fluid forward as it rotates. By changing the diameter of the tube inside the pump, one can facilitate a wide array of flow rates. A larger tube diameter corresponds to a larger flow rate. The operating principle of the peristaltic pump also ensures that the fluid is not in contact with the mechanical parts of the pump. It is also self-priming.

3.2.2 Microscope Channel Slide

To bring the sample into focus, a channel slide is required. The channel slide is installed underneath the magnifying objective and simplifies keeping the sample in focus. The slides are transparent, and have attachment points for silicone tubing. When picking a channel slide one may choose between several channel depths. Typically in the mm range, these depths often vary between 0.2mm and 1mm. Depending on the size of the specimen in the sample, the channel depth should be selected accordingly. The depth should be sufficiently large to accommodate the largest specimen flowing through it. A larger channel depth also allows more fluid to pass without increasing the speed. At the same time, having a narrow channel makes it easier to keep the objects of interest in focus, but will also increase the speed of the fluid.

[7] also called metering pumps

3.3 Optical Components

The following components are required to provide sufficient magnification and lightning of the specimen inside the microscope slide. Cameras are briefly discussed highlighting the desired qualities.

3.3.1 Objectives

The objectives come in a wide variety of magnifications, often ranging between 4x and 100x. The objectives are called *scanning objectives*, *low-power objectives* or *high-power objectives* depending on their magnification, with *scanning objectives* having the least magnification. The objectives also come in different configurations, typically referred to as *achromatic*, *semi-plan* and *plan* objectives. The main difference between the objectives is how much of the field of view is viewed as *flat*: that is, how much of the field of view appears in focus and not blurry [24].

- **Achromatic** objectives are the simplest and cheapest type of objectives. They bring red and blue wavelengths into a common focus, and correct for *spherical aberrations*[8] caused by the green wavelengths. They are well suited for black and white viewing, and provide a flat field of approximately 65% of the field of view.
- **Semi-plan** objectives are more complex that achromatic ones, due to additional stacked lenses inside the objective. This leads to better color correction, as well as a flat field of up to 80%.
- **Plan** objectives are even more complex than semi-plan objectives. They provide the best color correction, and can provide a flat field of up to 95%. They are however the most expensive.

The choice of which objective to use depends on the size of the specimen one is interested in observing, as well as which light condenser is used. Larger *zooplankton* are visible to the naked eye, and can be viewed in great detail using a scanning objective (4x magnification). Smaller *phytoplankton* may require additional magnification, and a low-power objective (10x magnification) can thus be more suitable. Magnification is however a loose term and for applications regarding digital images, the unit micrometers per pixel [μm/pixel] is better suited. This is further elaborated in section 3.3.3.

3.3.2 Light Condenser

The light condenser focuses light towards the sample in order to illuminate it. The most common type of light condenser is the *Abbe-condenser*, which provides a brightly lit background which any particle will appear dark against. This lightning can make it hard to discern between colors, due to the lack of contrast.

By using a **darkfield condenser** one can increase the contrast. In the darkfield condenser, the light is directed in such a way that the sample is illuminated from the sides, with the background appearing black.

3.3.3 Camera

When selecting a camera for the design, two qualities should be prioritized. A camera with a high resolution will yield a crisper picture, with a higher μm/pixel, which can help in identifying smaller organisms or objects of interest. Equally important is the ability to capture images at a high frame rate. When recording frames while simultaneously pumping water past the objective, motion blur can occur in the pictures if the cameras sensor is not exposed quickly enough. If this is the case

[8] ”Blurriness” at the edge of the field of view

the sample will have to be pumped in a stop-and-go manner, with the sensor being exposed as the sample stays still. Unfortunately, the relationship between frame rate and resolution in cameras is such that an increase in one usually leads to a decrease in the other. Cameras able to record both with a high resolution and at a high frame rate do exist, but often come at a high price or a bulky form factor. Cameras recording at high speeds are also designed so that the sensor is sufficiently exposed in a matter of milliseconds. This exposure time is needed to capture frames at a high frame rate, but can also introduce noise and produce ”grainy” pictures.

When selecting a camera it is also important to be able to change camera settings such as the exposure time. Many cameras come with an auto-exposure setting which functions well in cases where the scene is brightly lit, but may struggle with the opposite case such as when using a darkfield condenser. The auto-exposure mode will try to increase the brightness for the dark parts of the image, and thus overexpose the object in focus. In addition, the time used trying to increase the brightness means that the camera is unable to record at high frame rates.

Certain cameras come with a *C-mount*, a threaded exterior which means the camera can be attached to other objects, such as microscopes, telescopes, or anything sensible to attach a camera to. Cameras built for microscope applications always come with a C-mount or with adapters to cater for different standards. These are well suited cameras for the system, and come in different configurations depending on the end users need.

3.4 Hardware

This section presents hardware components that can simplify the construction of the system, and interface to mechanical parts such as the pump and focus drive stepper motors.

3.4.1 Microscope

Building the system around a microscope simplifies the construction substantially. The microscope provides a solid foundation, a known optical line, as well as knobs to adjust the focus and slide position. In addition, the microscope comes with one or several oculars (depending on the model) on which cameras can be attached. Many microscopes also provide a light source for the condenser, typically a LED powered by a battery or DC current from a mains adapter. Ideally, a battery is considered the best power supply. It is not affected by interference due to the AC-nature of the mains supply, with a frequency of 50Hz/60Hz depending on the location. Although not visible to the human eye, this interference can cause the LED to appear to flicker when recording at higher frame rates. This can cause inconsistent light levels between the recorded frames, which may affect the results.

A microscope offers a good foundation on which to build the system. The most obvious drawback is however the form-factor of the microscope. As they are built to be solid and precise, it's not uncommon that the base of the microscope is made of materials such as cast iron. This works well when the microscope is placed on a desk, with the sheer weight combined with rubber feet keeping the microscope in place. When installed on a smaller vessel however (such as the AutoNaut), the weight of the microscope must be accounted for such that the handling of the vessel isn't affected.

3.4.2 Microcontroller and CNC-Shield

In order to control hardware related components such as the pump and focus motors, a microcontroller is used. As previously mentioned, stepper motors need drivers in order to operate correctly. Depending on the hardware used, these drivers can be controlled directly from the main computer. However, this means that code to control the driver must be developed, as well as ensuring that the driver is supplied the correct signals.

The suggested approach is to control the drivers using a microcontroller, and communicate with the microcontroller through a USB-connection. Microcontrollers often come with many input and output pins, which are well suited to interface with the stepper drivers. Code can then be written such that simple instructions sent to the microcontroller translate to specific signals sent to the driver. This means that the pump and stepper motors can be actuated while the main computer is doing something else, such as grabbing frames from the camera or performing file operations. There also exists firmware for microcontrollers dedicated for CNC-tasks, such as machining or 3d-printing. An example of this is **GRBL**, which translates *G-code*[9] into movements of the stepper motors. Although GRBL is most commonly used for 3D-printing, it is well suited as a microcontroller firmware for this system. GRBL can be installed on an *Arduino Uno*, a very common and widely used microcontroller.

Stepper motor drivers are small integrated circuits that supply current to the phases of a stepper motor such that it rotates in a certain direction. The driver has input pins deciding which direction, angular displacement, and whether to enable or disable the motor. When a step-signal is sent to the driver, the driver energises the phases of the stepper motor, making it rotate. The power used to actuate the stepper motor is significantly greater than the power of the control signals, so the stepper driver has a power input dedicated to powering the stepper motor, often in a range between 12V-48V. A good driver often comes with a current limiter, to prevent the driver from overheating. Despite this, many drivers also require a heat-sink to dissipate the generated heat.

To incorporate a stepper driver into the system, a PCB or breadboard is necessary to connect the drivers to the microcontroller. Microcontroller *shields* are PCB-boards built specifically for mating

[9] A widely used programming language for telling machines what to do.

with microcontrollers, and are placed directly on top of the headers of the microcontroller. These shields simplify connections to peripherals or add additional features such as WiFi-connectivity or Bluetooth. Certain CNC-shields exist for the Arduino Uno, which has slots for inserting stepper drivers as well as headers to control micro-stepping. The shields are intended to be used for 3D print- and CNC-mill applications, but works well for this task. These shields are commonly compatible with the common A4988 or DRV8825 drivers, and often have a 12-36V input for powering the motors. Depending on the model, they are also compatible with GRBL. With the GRBL firmware installed, the motors are easily controlled by sending serial commands to the Arduino.

3.5 Software

This section lists suggested software or software development kits (SDK's) to interface with the different components of the system. It's not uncommon for producers to provide documentation or SDK's for their products, so that the end user can write custom code to control them. However, many different brands exist in the market, and for the sake of brevity, only selected brands and products are listed in this section.

3.5.1 Camera Control and Image Manipulation

When selecting a camera, it's ideal to chose one supplied with a comprehensive SDK. It is important to be able to control several features of the camera, including:

- The ability to take pictures or record videos at different frame rates. When pumping a sample at high speeds, it will be necessary to grab many frames per second in order to detect objects passing by. When pumping at slower speeds, the high frame rate can cause duplicate images to be stored, and bloat the storage device.
- Controlling settings related to exposure time, white balance, color balance, and so on. Many cameras have an auto-exposure mode which adjusts the exposure time automatically, often enabled as default. This can be fine for cases where you'd like to quickly have a look at the output of the camera, but will be detrimental with regards to uniformity between the individual frames. In the case of presence detection[10], counting the number of pixels above a brightness threshold can be a way to confirm whether an object of interest is in the image. If the brightness between the (empty) frames vary, problems can occur. It is also especially important to be able to modify the exposure time when using a darkfield condenser. If the auto-exposure mode is left on, the sensor will interpret the dark background as an underexposed image and try to increase the exposure time to no effect. This can lead to very long exposure times, throttling the frame rate.

There exist many different brands of cameras. Some of these are *white label brands*[11], with the camera producer behind the product being a company called **ToupTek**. ToupTek produces cameras for microscopes and telescopes with CMOS and CCD sensors. They offer an extensive SDK for programming languages such as Java, C++ and Python, as well as support for operating systems including Windows and Linux distributions. The ToupTek-SDK is well documented and includes examples on how to connect to and fetch data from the camera.

OpenCV is an open-source computer vision library for programming languages including Python and C++. It offers a wide variety of functions and algorithms to aid with computer vision, machine learning, and image manipulation. For the scope of this thesis, only image manipulation functions are used.

3.5.2 Stepper Motor Control

The previously mentioned **GRBL** firmware is well suited for this system. The firmware is installed onto compatible hardware, such as the Arduino Uno (or any microcontroller based on the ATmega 328 chip). The firmware is originally built for CNC purposes such as 3D-printers, CNC- mills, lathes, or any system where there's a need for moving something along the principal axes. GRBL has support for four axes: in the case of a 3D-printer, three stepper motors would control the X, Y, and Z axes, whereas the last stepper motor would control the feeding speed of the extruder[12]. For this application, the available stepper outputs could be used to control peripherals such as the focus, slide position, as well as the pump.

[10] Detecting whether something is present in the frame or not

[11] A product produced by a company, which is rebranded by other companies so that it appears that they made it.

[12] The part which deposits material onto the printer bed

To control the stepper motors, commands are fed serially through an USB-cable to the Arduino running the GRBL-firmware. This simplifies the control of the stepper motors significantly, and enables one to use any hardware as long as they come with a serial port. GRBL also allows for a lot of customization with regards to parameters relevant to stepper motors such as step pulse time, acceleration, axis travel resolution and so on. This is vital as the use of the stepper motor decides how fast it should go; the focus drive should rotate slowly with a great resolution, whereas the pump motor may rotate several hundred times per minute.

3.5.3 Morphocut Segmenter Library

Morphocut is a pipelined image processing library for Python, capable of extracting regions of interest in large amounts of medical and oceanographic image data. Morphocut works well for large amounts of files, and has a customizable pipeline with supporting image manipulation functions to preprocess images. These functions include reading from files, writing to files, and converting to grey-scale or other image formats. Morphocut detects regions of interest based on threshold values for brightness and object size. These regions are cropped, and subsequently stored in a zip-archive. This can condense hundreds of pictures down to a few pictures containing only objects of interest.

4 Implementation

This section lists the chosen components for the system, as well as their implementation.

4.1 Hardware

4.1.1 Pump and Driver

To cater to the low flow rate needed for this design, a **Boxer 29QQ Stepper Peristaltic Pump** with a **Pico Driver** is used. The pump can use three different sizes of inner tubing at 0.5mm, 1mm and 2mm (inner diameter). Depending on the inner tube, each revolution pumps $13\mu L$, $34\mu L$ or $100\mu L$, respectively. At a speed of 600rpm, these correspond to flow rates of 7.8ml/min, 20.4ml/min and 60ml/min. The pump and driver is shown in figure 10.

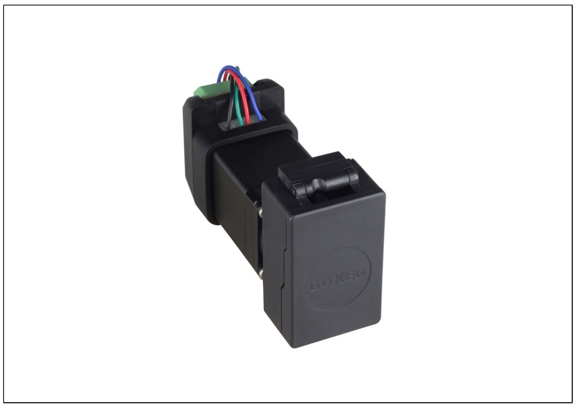


Figure 10: Boxer 29QQ Peristaltic pump with Pico Driver attached.

The optional Pico-Driver greatly simplifies controlling the pump. The driver operates at 24V, and has several customization jumpers that can be soldered to change the modes of the driver. This includes selecting between microstepping resolutions (1, 1/16 and 1/256), current limiting, or choosing between an analogue speed signal or a digital step signal. In order to control the pump using the Arduino, the driver is configured to use a digital step signal as well as using a 1/16 microstep resolution. From the data sheet of the driver it is found that the step angle for the stepper motor is 1.8°. This translates to 200 steps for a full rotation. For a microstep resolution of 1/16, one full rotation equals $200 \cdot 16 = 3200$ steps. To rotate at 600rpm, this corresponds to a step signal frequency of 32kHz. Instructions on how to configure GRBL to attain the correct step frequency is further elaborated in section 4.1.5.

4.1.2 Tubing and Flow Components

20*ml* syringe barrels serve as sample holders before and after the sample is pumped underneath the microscope objective. These are connected to the rubber tubing using appropriate adapters. The sample is led from the inlet through the pump to the microscope slide, until it ends up in the outlet. This is shown below in figure 11.

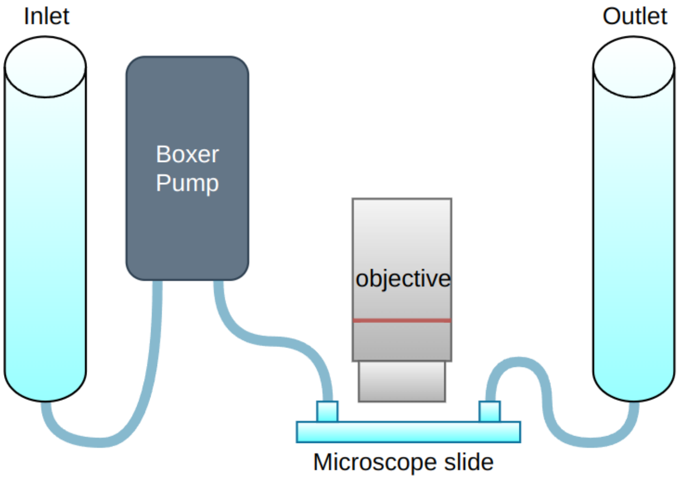
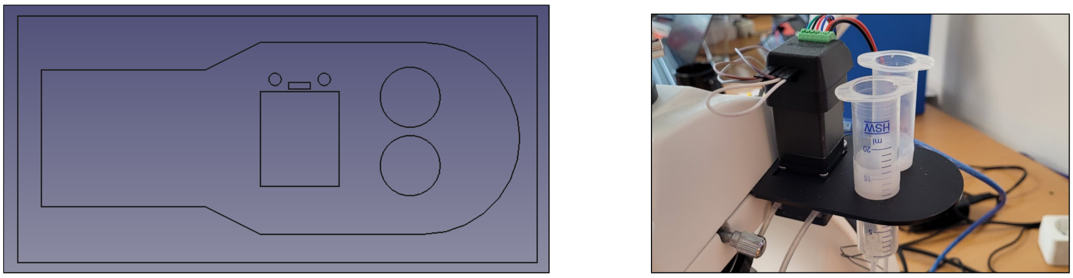


Figure 11: Tubing topology.

A small mount for the pump and sample syringes is designed in FreeCAD, shown in figure 12a, and cut out of black acrylic sheets. This mount has room for two syringe barrels, and has matching holes cut out to fit the pump. This is shown in figure 12b.



(a) Pump and syringe-holder design. Designed in FreeCAD.

(b) Pump and syringe-holder mounted on the microscope.

Figure 12: Sample and pump holder.

4.1.3 Camera

The camera that is chosen for this design is the **BRESSER MikroCam II 0.4 UHSP** microscope camera (figure 13). This CMOS-camera is capable of recording up to 300 frames per second, making it ideal for capturing the specimen flowing past the objective at high speeds. The camera is a *white-label* ToupTek camera, which means that it can interface with the ToupTek SDK. Since it has a global sensor, distortion due to motion is minimized. The camera has a resolution of 720x540 pixels and comes with a C-mount thread.

Figure 13: BRESSER Mikrocam II 0.4 UHSP. Collected from [25]

The MikroCam II is connected to the hardware using an USB3.0 cable. Image data is fetched using BRESSER's included software *MikroCamLab*, or by writing code to collect the frames using the ToupTek SDK. The MikroCamLab software has useful features to measure and determine parameters such as micrometers per pixel.

To determine the resolution of the camera in micrometers per pixel, the camera is fixed to the microscope and a micrometer slide is placed under the 4x scanning objective. An image is then taken, and the number of pixels between each hatch mark is counted. This is shown in figure 14.

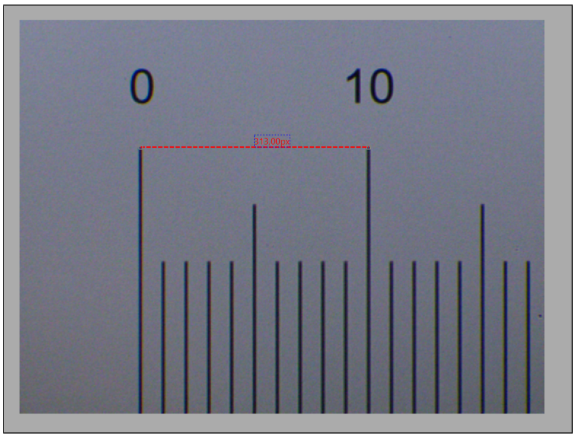


Figure 14: Determining the camera resolution using a micrometer slide and a measuring tool in the MikroCamLab software. The distance between the two hatch marks is found to be 313 pixels. Each div represents 100 μm.

The resolution is then found using the formula given in formula 3:

$$Resolution\left[\frac{\mu m}{\text{pixel}}\right] = \frac{1000\mu m}{313\text{pixels}} = 3.19\frac{\mu m}{\text{pixel}} \tag{3}$$

Using this result, the field of view can also be decided. As the resolution of the camera is 720 by 540 pixels, the horizontal field of view is given by equation 4.

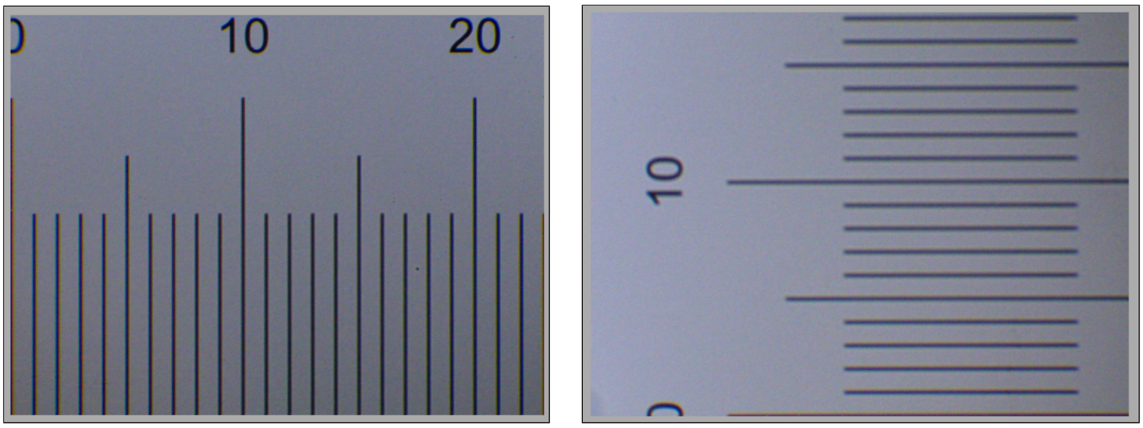
$$FOV_{width} = width_{\text{pixels}} \cdot \text{resolution} = 720\text{pixels} \cdot 3.19\frac{\mu m}{\text{pixel}} = 2296.8\mu m. \tag{4}$$

Since the aspect ratio (AR) of the camera is 4:3, the field of view in the height direction is found by equation 5:

$$FOV_{height} = \frac{FOV_{width}}{AR} = \frac{2296.8\mu m}{4/3} = 1722.6\mu m. \tag{5}$$

With a field of view of $2296.8\mu m$ by $1722.6\mu m$, the camera is able to image an area of $3.956mm^2$.

This is verified using the micrometer scale, and shown below in figure 15.



(a) Field of view, width. Measured to be approximately $2.3mm$

(b) Field of view, height. Measured to be approximately $1.72mm$

Figure 15: Field of view using the BRESSER MicroCam II with a 4x scanning objective.

4.1.4 Microscope and Condenser

The **BRESSER Science TFM-301 Trino** microscope is chosen for this design. The microscope is well suited due to the additional ocular. This allows for up to three cameras to be attached to the microscope. The microscope has a revolving nosepiece equipped with 4 semi-plan objectives with 4x, 10x, 40x and 100x magnification. For illumination, the microscope has a built in dimmable LED. This is powered by a mains adapter, but can easily be connected to a battery. The microscope comes with a stock Abbe-condenser, and can also adopt a darkfield condenser. It has coaxial pinions to drive the stage, as well as coarse and fine focus knobs. The microscope is shown in figure 16a.

The microscope is compatible with the **BRESSER Darkfield Condenser Dry**. The condenser being 'dry' means that the condenser can be used without the need of having oil between the slide and objective, and it is suitable for the magnification range of the Science TFM-301. It is shown below in figure 16b.

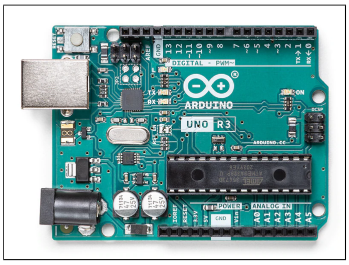
(a) BRESSER Researcher Trino microscope. Collected from [26].

(b) BRESSER Darkfield Condenser Dry. Collected from [27].

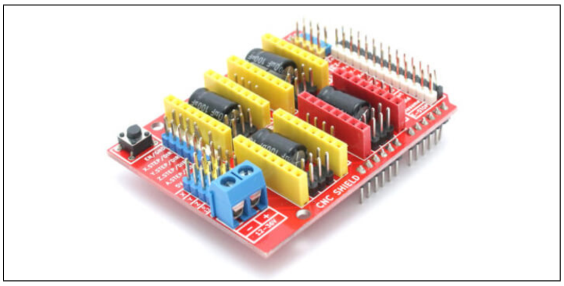
Figure 16: Microscope equipment

4.1.5 Microcontroller and CNC-shield

To interface with the pump and the focus motors, an Arduino Uno is used in tandem with a *CNC-Shield V.3*. The Arduino Uno is an open source, widely used microcontroller based on the ATmega328P microcontroller made by Microchip. The CNC-Shield V.3 is an open-source design and is sold by several manufacturers. These are shown below in figure 17.



(a) Arduino Uno Rev3 microcontroller. Collected from [28].



(b) Ooznest CNC Shield V.3. Collected from [29]

Figure 17: Microcontroller and CNC shield.

The Arduino microcontroller is connected to the computer using an USB cable, and is also supplied power from this cable. The CNC shield is powered by an additional 12V-36V, depending on which stepper driver is used. This power is supplied using a mains adapter, converting 230V AC to 24V DC. In this design, the **DRV8825** driver is used with the CNC-shield. It is rated for voltages ranging between 8.2V and 45V, and can supply a phase current up to 2.2A with a sufficient heat sink. It is capable of microstepping-resolutions down to 1/32.

In order to control the pump from section 4.1.1, certain changes must be made. First, the *Pico driver* attached to the pump is used instead of the DRV8825 drivers attached to the CNC board. The signals originally sent from the Arduino to the DRV8825 is routed to the Pico driver on the pump. The signals and configuration is shown in figure 18.

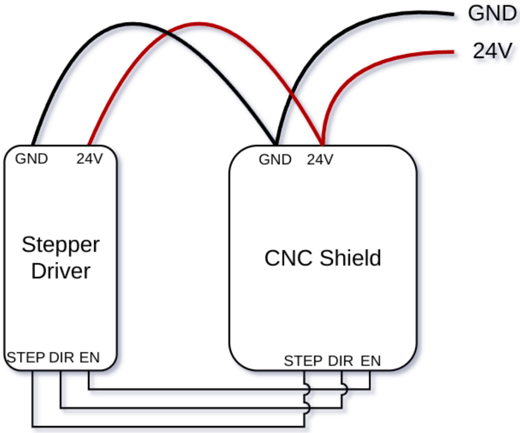
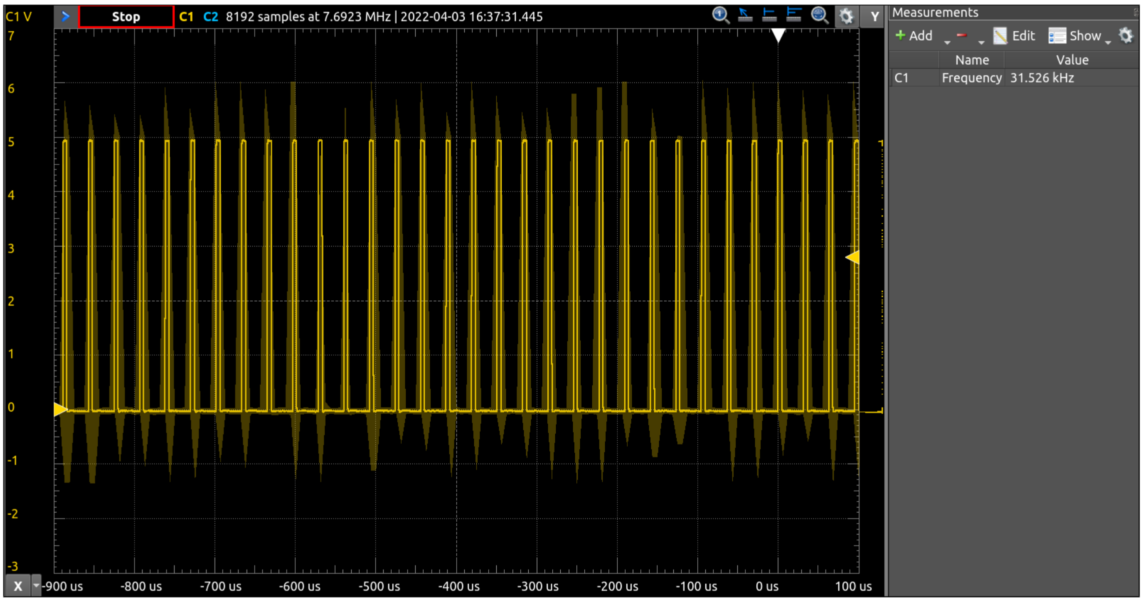


Figure 18: Schematic of how the Pico driver is connected to the CNC-shield.

The Arduino Uno is then flashed with GRBL version 0.9. When the Arduino is connected to a computer, the GRBL settings can be accessed by sending the command **$$** to the Arduino using a serial console[13]. From here, the settings worth noting is the step pulse length, the number of steps per millimeter, and the max rate given in mm/minute. These settings should be tweaked so that the desired step-signal frequency is attained. This is verified by performing a frequency measurement using a oscilloscope on the step-signal pin of the CNC-shield, shown in figure 19.

[13] An example of this is the serial monitor of the Arduino IDE, which allows serial data to be sent to the Arduino.



8192 samples at 7.6923 MHz | 2022-04-03 16:37:31.445

	Name	Value
C1	Frequency	31.526 kHz

Figure 19: Frequency measurement of step signal. Approximately 32kHz is attained.

4.2 Assembly

The system was then assembled as shown in figure 20. The camera was attached to the top ocular using the C-mount threads, and connected to a PC using the USB3.0 cable. The pump and syringe barrels were attached to the laser-cut plate and fixed between the top and bottom part of the microscope. Silicone tubing was then routed from the syringe barrels to the pump and channel slide. The pump driver was connected to the CNC shield and power supply as shown in figure 18.

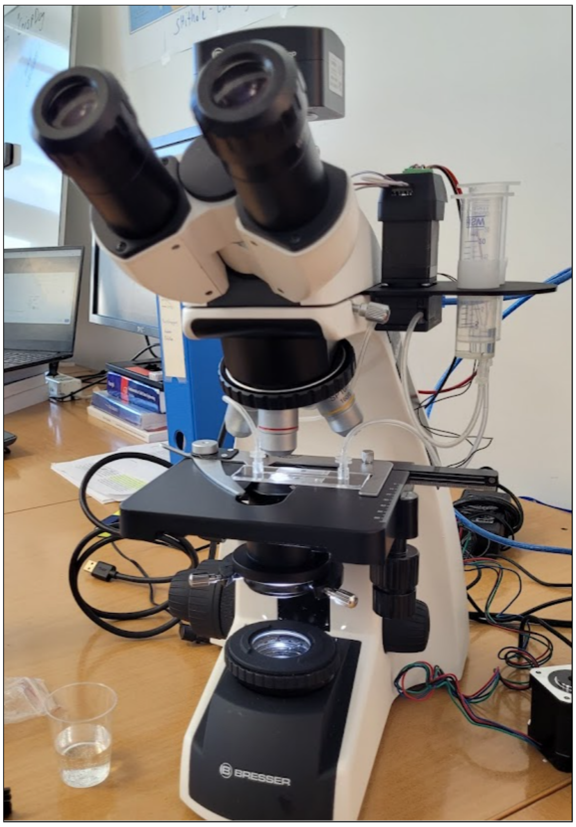


Figure 20: System assembled on the Bresser Trino TFM-301 microscope.

4.2.1 System Specifications

With a resolution of $3.19\mu m$/pixel, a field of view of $2296.8\mu m$ by $1722.6\mu m$ is achieved. Using microscope slides with a channel width of $5000\mu m$, the ratio between the FOV-width and the channel width is found to be $\frac{2296.8\mu m}{5000\mu m} \approx 0.46$. Since the camera can only see a part of the slide, the imaged volume is less than the actual pumped volume. This discussed further in section 7.

The cost of the components used is shown in table 2.

Component	**Seller**	**Price**
Science TFM-301 Microscope	BRESSER	759€
Darkfield Condenser Dry	BRESSER	172€
MikroCam II 0.4 UHSP	BRESSER	668€
29QQ Peristaltic Pump w/ driver	Boxer	243€
Arduino CNC - Kit	Amazon	59€
Sum		1901€

Table 2: Cost of the components in the system.

4.3 Software

4.3.1 Camera Software

The camera software is responsible for interfacing with the ToupTek camera and presenting the images to the end user. In this design, a python script is written in order to connect to the camera, pass settings to it, and stream image data back to the user for displaying and saving. To enable this, a powerful Python library called **PyQt** is used. PyQt is developed by Riverbank Computing and is licensed under the *GNU General Public License* which allows end users to freely run and modify the software. It is a powerful cross-platform GUI[14] toolkit with features that allow programmers to easily create complex GUI's. For the sake of brevity, all relevant code is made available on request. However, a brief description of the camera software is described here. Figure 21 below illustrates the data flow in between the different modules.

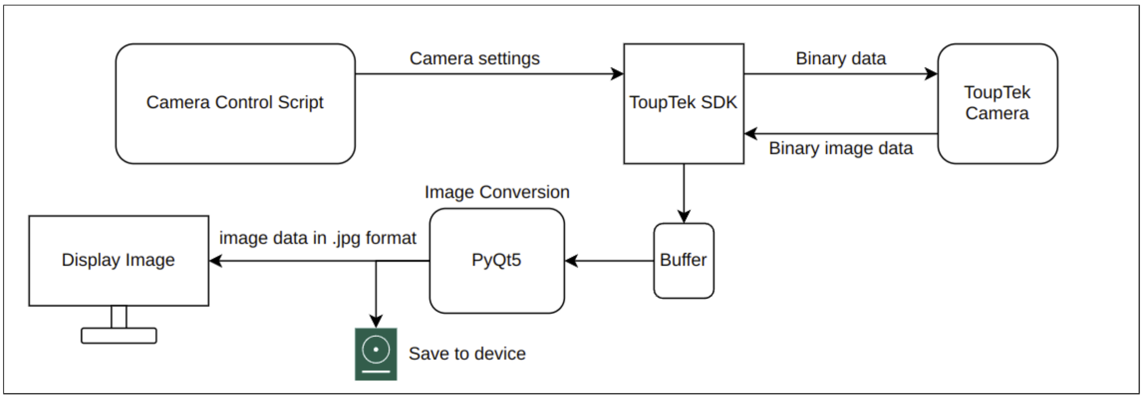


Figure 21: Data flow diagram, camera software.

In figure 21, the **Camera Control Script** takes user input for camera related settings such as the exposure time, camera sampling rate, and whether to start or stop the data acquisition. This data is then passed to the functions provided by the ToupTek SDK. The ToupTek SDK simplifies communication with the camera module, and translates the hexadecimal addresses of the camera to a human readable format. The camera settings are changed by writing data to these addresses. Important functions from the ToupTek SDK include functions such as `def Toupcam.EnumV2()`. This function locates connected cameras and assigns them to an ID. The camera is then accessed using the function `def Toupcam.Open(cameraID)`. `def Toupcam.StartPullModeWithCallback()` is then called to make the camera start recording. When the data from the camera is available, the function `def Toupcam.PullImageV2()` grabs the binary data in the camera and saves it in a buffer, shown in figure 21.

The data stored in the buffer is a one dimensional array containing all the values of the individual pixels in the image. The MikroCam II 0.4 UHSP stores images in a *RGB888*-format; where 8 bits are used for each color channel (red, green and blue). With a resolution of 720x540 and three color channels, this translates to $720 \cdot 540 \cdot 3 = 1166400$ individual pixel values with a value between 0 and 255. In order to display this data, *PyQt5* is used to convert the data to a *bitmap*. The PyQt function `def QImage(data,width,height,QImage.format)` takes the buffer data and image dimensions as an input, and returns a bitmap that can be saved or displayed on a screen. Writing the images to the disk is a CPU-intensive task, and may decrease the rate at which the frames can be displayed to the screen. For the same reason, the display is turned off when frames are recorded, to ensure that all the available processing power goes into grabbing the frames and saving them to the disk as fast as possible. As the camera is capable of recording up to 300 frames per second, such a frame rate might not be needed for low pump speeds. Duplicate images of specimen may occur when a high frame rate is used for a low pump speed. To prevent this, a selectable frame rate-divisor value is used to only grab the frame chosen. With a divisor value of 5, only every fifth frame is grabbed, thus effectively dividing the capture-frame rate by 5.

[14] Graphical User Interface

4.3.2 Serial Software

The serial software is responsible for taking user input and sending commands to the Arduino to actuate the pump and stepper motors. A data flow diagram is shown in figure 22.

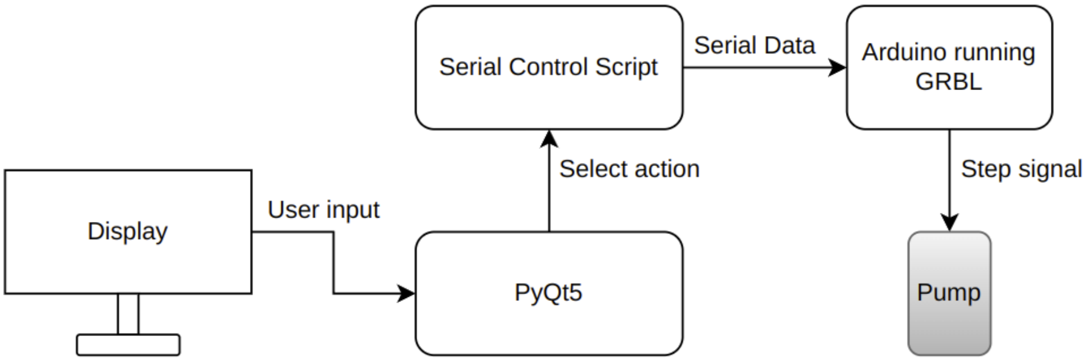


Figure 22: Data flow diagram, serial software.

The **Serial Control script** in figure 22 is a Python script that handles the communication with the Arduino. It connects to the Arduino and provides functions that sends serial GRBL commands that move the pump and stepper motors. An example of this is the `def step(axis, direction, length, speed)` function. Calling this function moves the stepper connected to *axis* at the desired speed and length. Writing 1 to the direction corresponds to a clockwise direction, whereas writing 0 corresponds to a counterclockwise rotation. PyQt5 is used to take user inputs in the form of button presses and combo-box selections. The user is presented with a GUI on which a selection of pump speeds can be chosen, as well as buttons to drive the pump in both directions as well as stopping it. This GUI is presented in section 4.3.3.

4.3.3 Complete Software Topology, GUI

A complete diagram of the software structure is shown in figure 23. PyQt5 is used to connect everything together, and provides a framework for providing a graphical user interface in which the image data is displayed, as well as taking user input through button presses and combo-boxes. All camera related settings such as exposure and divisor settings are passed to the Camera Control script, and the serial control script passes the correct data to the Arduino based on the input data through the GUI.

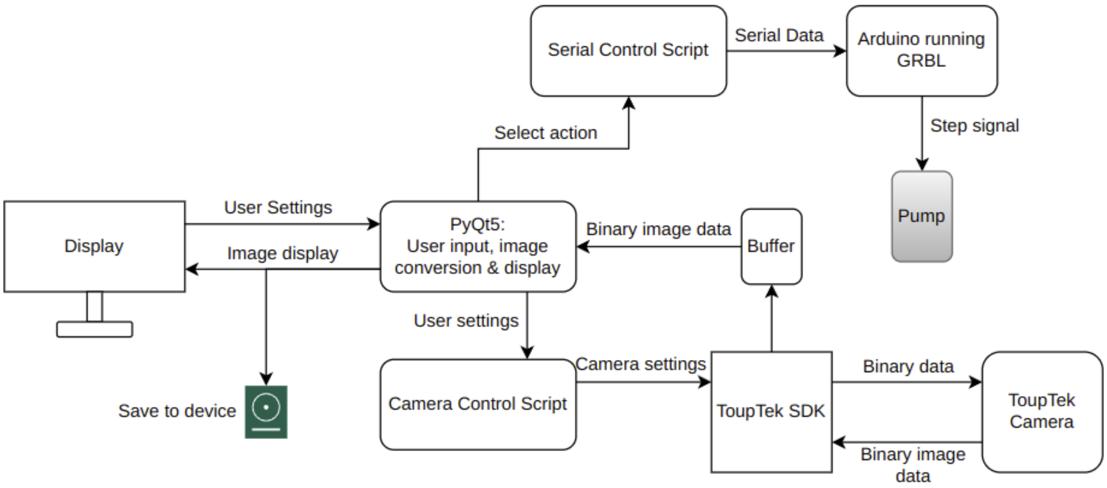
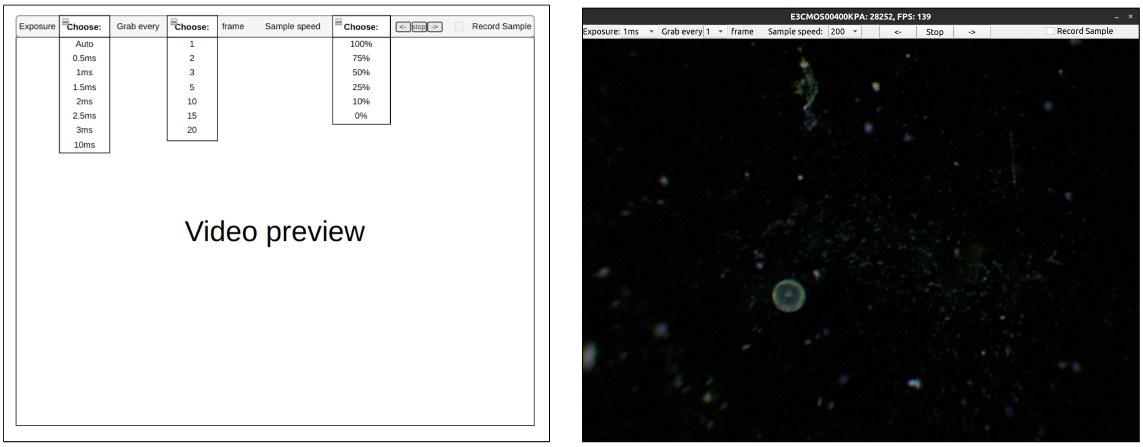


Figure 23: Data Flow Diagram, all software.

The GUI is shown below in figure 24. Figure 24a shows a model of the GUI, highlighting all the possible options presented to the user. Figure 24b is a screenshot of the GUI. The main window displays a live view of the image data from the camera. This simplifies bringing the sample in focus, and allows one to scan the slide without looking into the oculars. The video preview can have any size, as long as the aspect ratio is 4:3. Images saved to the device will retain the original 720x540 pixel image resolution[15]. In the upper left corner of figure 23, the user can select between exposure times of 0.5ms to 3ms using the combo-box, as well as auto-mode which will automatically select an appropriate exposure. Note that the auto-exposure will only work well with brightly lit backgrounds, such as the case when using a bright-field condenser.

Next to the exposure selection is the frame rate divisor selection box. Using this selection, the user can choose to grab every frame, every other frame, third, fifth or tenth frame and so on to divide the frame rate by the chosen amount. This is useful for cases where the pump speed is low, and a high frame rate would lead to duplicate images of specimen.



E3CMOS00400KPA: 28252, FPS: 139

Exposure: 1ms | Grab every 1 frame | Sample speed: 200 | <- | Stop | -> | □ Record Sample

(a) GUI-model.

(b) GUI-screenshot.

Figure 24: Graphical user interface.

A combo box is also provided with a selection of pump speeds. The selected speed is used with the pump when pumping water using the buttons, or when sampling data.

[15] Depending on the camera model

To the right of the pump speed selection, three buttons are situated. These buttons are used to start and stop the pump for both directions. This allows the sample to be pumped in both directions at the speed specified in the previously mentioned combo box.

To the far right in figure 24a is the *record sample* checkbox. When ticking this checkbox, the pump will start running at the selected speed, and the image data will be saved to the device using the select divisor value. The sampling will continue until the box is unchecked. Note that the live display is temporarily stopped when the sampling is underway. This is to ensure that the most amount of processing power is used to save the image data to the device as quickly as possible. When the sampling is done, all the images are saved in a folder where the script is run from. The name of the folder will be in a date-time-speed-divisor format, and the images are named in the same order they were saved.

4.4 Image Manipulation Methods

The following methods are implemented in Python using image data from the camera and functions from the OpenCV image-manipulation library. A brief description of how the methods work is presented in this section.

4.4.1 Empty Frame Detection

For every frame, the L2-norm of the image array is calculated. This value is then stored together with the frame number in an array. Any frame whose norm is less than a specific value can then be discarded, or the array can be sorted by the norm values.

4.4.2 Removing Average of last n frames

To remove the average of the last n frames, an array of length n is created. As images are received from the camera, they are inserted at the beginning of this array. For every new frame, the frames are shifted back in the array, and the average of all n frames is calculated. This average frame is then subtracted from the most recent frame, yielding a manipulated frame. This is shown in figure 25.

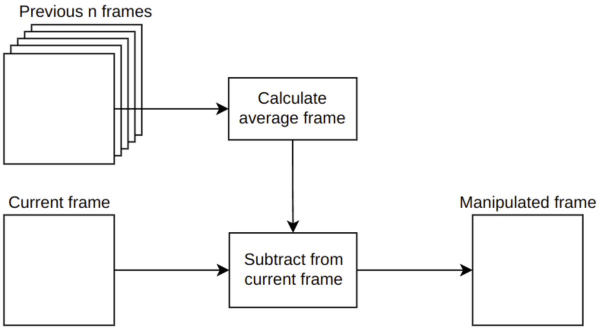


Figure 25: Flow chart depicting how the average of the last n frames is removed from the current frame.

4.4.3 Color Space Transform, Masking and Size Estimation

Frames are converted from RGB to HSV using OpenCV. A binary mask is then created from any pixel within a certain hue, saturation or value range. Boundary detection and size estimation is then applied to this binary mask. If the size of the enclosed area is larger than a certain threshold, a bounding box is drawn on the original frame using the binary mask as a reference. This is shown in figure 26.

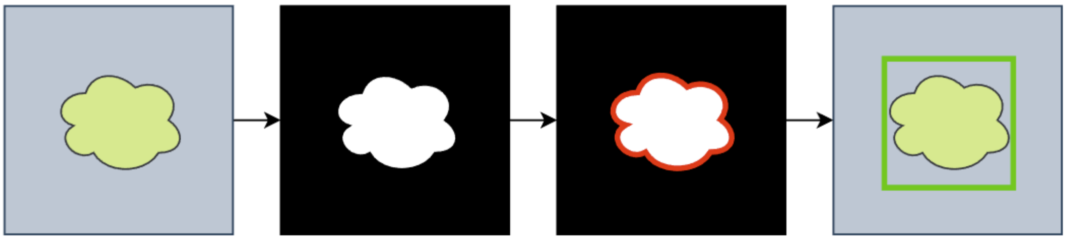


Figure 26: Process for how bounding boxes are applied to frames containing objects of interest.

5 Testing

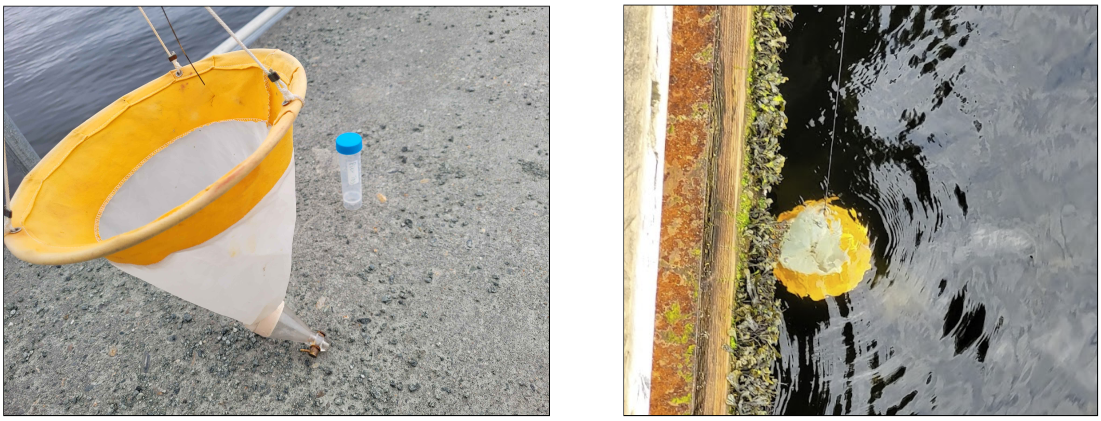
This section presents how the system is tested. The section presents how the samples for testing are acquired and how the test should be run to gauge the systems performance.

5.1 Acquiring Samples at Trondheim Biological Station

Once the system was built, a trip to Trondheim Biological Station (TBS) was made on may 16, 2022. TBS is situated directly by the Trondheimsfjord, and hosts several offices and laboratories for marine biological research. The objective of the trip was to collect seawater samples to test the performance of the system, as well as samples of specific species of algae. In addition to these, samples were also taken from the outdoor basin located at TBS. This pool is known to get dirty as the spring bloom progresses.

5.1.1 Test Equipment

A *plankton net* is used to collect the samples. The plankton net is a tapered nylon mesh with a collection cylinder at the end. Samples are acquired by dragging the net through a body of water, either horizontally or vertically. This is shown in figure 27b. The mesh size is chosen such that the size of the targeted particles are trapped in the collection cylinder, whereas smaller particles pass through the mesh. The collection cylinder has a valve to transfer the sample into another container. For this test a net with a mesh size of 20um was used. The plankton net and sample container is shown in figure 27a.



(a) Plankton net and sample container.

(b) The net is dragged through the water.

Figure 27: Plankton net being used.

As the plankton net traps any particles larger than the mesh size, particles too large can pose a problem within the system. Larger particles can get stuck within the microscope slide or the silicone tubing, and clog up the system. To get rid of the larger particles, a sieve was used to filter away any particles larger than $500\mu m$ in size. This ensured the size of the particles in the samples would be in the range $50\mu m - 500\mu m$. The sieve and filtered particles are shown in figure 28.

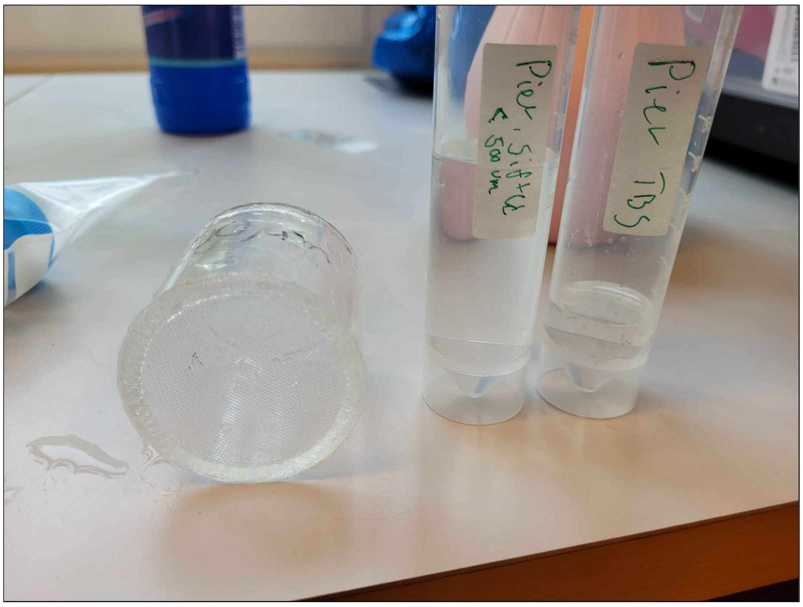


Figure 28: A sieve with a mesh size of $500\mu m$. All samples were filtered through this sieve such that the particle size would be in the range $50\mu m - 500\mu m$.

5.1.2 Sample location 1: Pier

The first sample was collected at the pier at TBS, shown below in figure 29. The site was chosen for its convenient location close to TBS as well as yielding a realistic distribution of marine particles in the epipelagic zone. $40ml$ of seawater was collected, and the sample was sieved so that the specimen size would be in the range of $50\mu m$ to $500\mu m$.

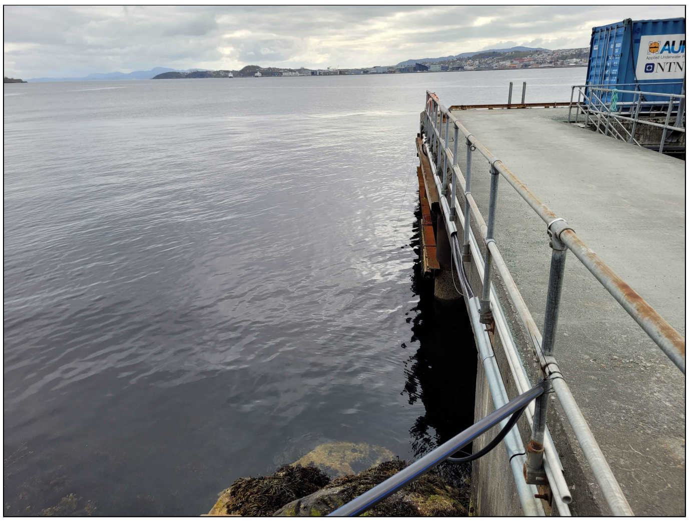


Figure 29: Sample 1 location: Pier at TBS.

5.1.3 Sample location 2: Pool

The second sample was collected from the pool at TBS, shown in figure 30. The pool consists of seawater from the Trondheimsfjord. The pool is known to get particularly dirty compared to the seawater from the fjord, due to its small size and the fact that algae and other organisms grow on the slanted walls. Based on this, $40ml$ of seawater from this location was also sampled and sieved.

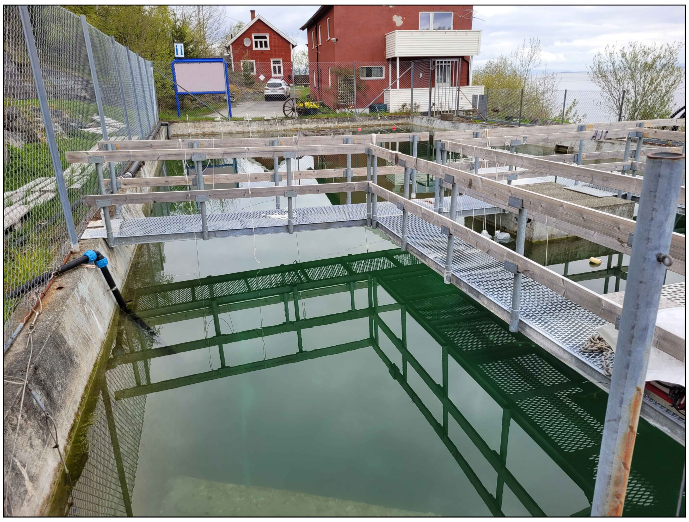


Figure 30: Sample 2 location: Pool at TBS.

5.1.4 Cultivated Dunaliella algae

In addition to the seawater samples, a small sample of lab-cultivated Dunaliella algae was also collected, shown in figure 31. Despite being a monoculture and at a concentration unlikely to be found in nature, the small size of the Dunaliella algae would prove a good test benchmark for the system as they are located in the lower range of the systems particle size range.

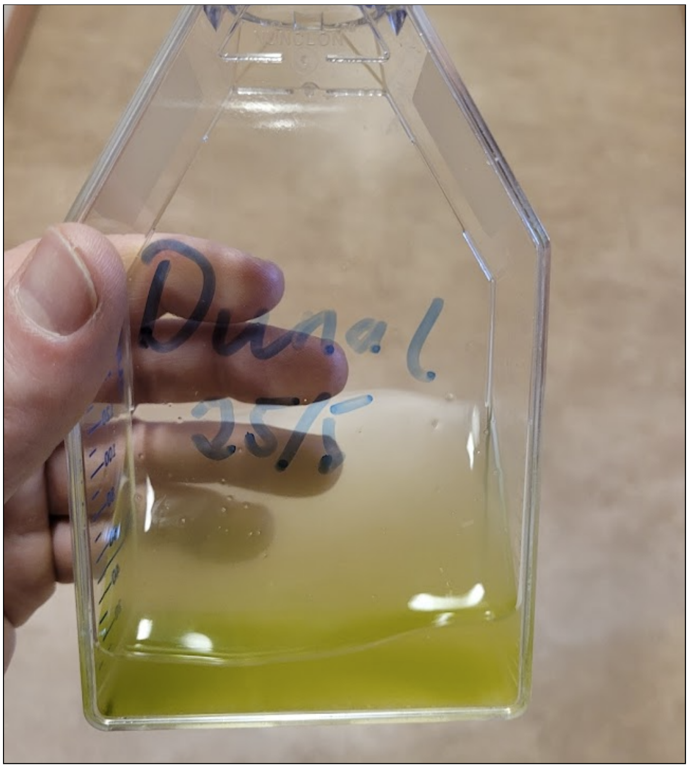


Figure 31: Lab-cultivated Dunaliella algae.

5.2 Test Scope and Motivation

Determining to which degree a system like this works is a difficult task. It's hard to define a single metric on which to gauge the quality of the images from the system. There are many factors to account for when testing, and the nature of the samples makes reproducing the tests very hard. This is due to several reasons, including the following:

- One cannot guarantee that the same specimen is imaged every time. When using the 4x scanning objective, a vertical field of view of approximately $2300\mu m$ is attained. The Ibidi-slides used in this system have a channel width of 5mm, which means that only 46% of the channel is captured by the camera. The particles may move past the camera outside its field of view. In addition, when using the deeper slides (0.6mm, 0.8mm), the particles may be positioned at a height which is out of focus for the camera, or the specimen may get stuck to the slide or silicone tubing. Ideally, the same specimen would be pumped past the objective at different speeds, exposure times, and frame rates.
- The particles position in the slide. The fluid speed isn't uniform through the channel, which can be modeled as a rectangular duct. Particles will move significantly slower along the edges compared to the center of the channel.
- The throughput. Shallower slides makes it easier to keep specimen in focus, but also significantly increases the speed in which they move past the objective. With the increase in speed, a great amount of motion blur would be expected in the frames.
- Contamination between tests, slides progressively getting more dirty.

In addition to these points are also the number of system parameters that can be varied between the the tests. These include the pump speed, camera exposure time, and which rate the frames are grabbed at. With these points in mind, the scope of the test should be to decide on which **channel slide depth** offers the best basis for future tests with regards to producing the best images while simultaneously maximizing throughput. In summary:

1. Identify the ideal channel slide depth which offers the best trade-off between throughput and object focus.
2. For this channel slide depth, identify the ideal pump speed and exposure time which maximises throughput while still retaining the best image quality (least amount of motion blur, pictures properly exposed).
3. The generated dataset should yield an ideal basis for testing image manipulation methods such as removing empty frames, particle size estimation, and finally segmentation using MorphoCut.

The images are rated based on a metric which is further elaborated in section 5.3.

5.3 Devising an Image Quality Metric

5.3.1 Subjective and Objective methods

There are many methods of determining the quality of an image, and they are often divided between *subjective* and *objective* methods. Objective methods calculate a score based on a specific algorithm, and will yield a score based on how distorted, blurry, sharp, or exposed an image appears. Subjective methods rely heavily on how individuals perceive an image, and a final score is based on how the individual perceives and rates the image. Subjective methods are further divided into *single*- and *double*-stimulus tests. In single-stimulus tests, the observer is only presented with a single test image and not a reference image. During double-stimulus tests, the observer is presented with both a test and reference image.

The nature of the dataset generated from testing implies a subjective, single-stimulus method of determining image quality. It is reasonable to expect that every particle is only imaged once, and that a correct identification of a potential species must be done with that particular image.

5.3.2 Suggested Image Quality Metric

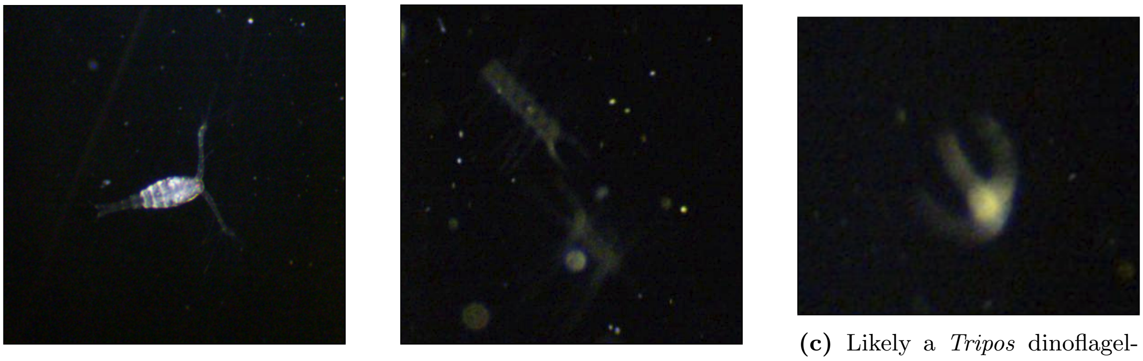
To gauge the quality of the specimen in the images, the following subjective metric is presented:

- Images are rated based on two factors: sharpness(A-C) and color (1-3).
- The sharpness score gauges how sharp an object appears. An object in focus without motion blur will receive a higher score. A blurry, out of focus object will receive a lower score. It can take three values: A = good, B = fair, C = bad.
- The color score gauges how vibrant the colors of the specimen appears. A high score means that the colors are easily distinguishable and prominent. A low score means that the object appears unsaturated or is affected by optical aberrations. It can take the following values: 1 = good, 2 = fair, 3 = bad.

Each object can thus be rated by the product of these factors, and a score matrix is shown in table 3. Examples of the scores applied to organisms is shown in figure 32.

	1	2	3
A	A1	A2	A3
B	B1	B2	B3
C	C1	C2	C3

Table 3: Score matrix. A green score is desirable and reflects the best pictures.



(a) A copepod. Score: A1

(b) Chaetoceros. Score: B2

(c) Likely a *Tripos* dinoflagellate. Score: C2

Figure 32: Scores applied to pictures of different organisms.

In figure 32a, a score of A1 is attained. The copepod appears in focus and its carapace is brightly lit and well exposed. The image appears sharp and the colors are pronounced. By looking carefully, one can even spot its red eye in the middle of its head.

The image of a (likely) *Chaetoceros* sp. in figure 32b receives a lower score of B2 due to the specimen being slightly out of focus. The species is however still identifiable, as its setae can be seen in the picture. The colors appear slightly washed out, but the green hue of the *Chaetoceros* is still recognizable.

The specimen in figure 32c receives an lower sharpness score, as the specimen appears out of focus. Still, its colors appear prominent and can aid in identifying the species. It receives a score of C2.

5.4 Test Setup

For these tests, the 4x scanning objective is used. Four different channel slides are tested. These have a depth of 0.2mm, 0.4mm, 0.6mm, and 0.8mm. For each channel slide, 4 pump speeds are tested. Full speed (100%, 7.8mL/min), half speed (50%, 3.9mL/min), quarter speed (25%, 1.95mL/min), and low speed (10%, 0.78mL/min). For each speed, an appropriate recording frame rate and exposure time is chosen. This will produce 16 data sets. Note that the number of frames between the data sets will not be consistent, due to the different frame rates images are captured at. For each data set, a subset of the frames are evaluated according to the metric defined in section 5.3.2. The data set(s) which produce the best images are then run through the image manipulation software to gauge its performance.

In addition to the above tests, samples of *Dunaliella* sp. are also tested.

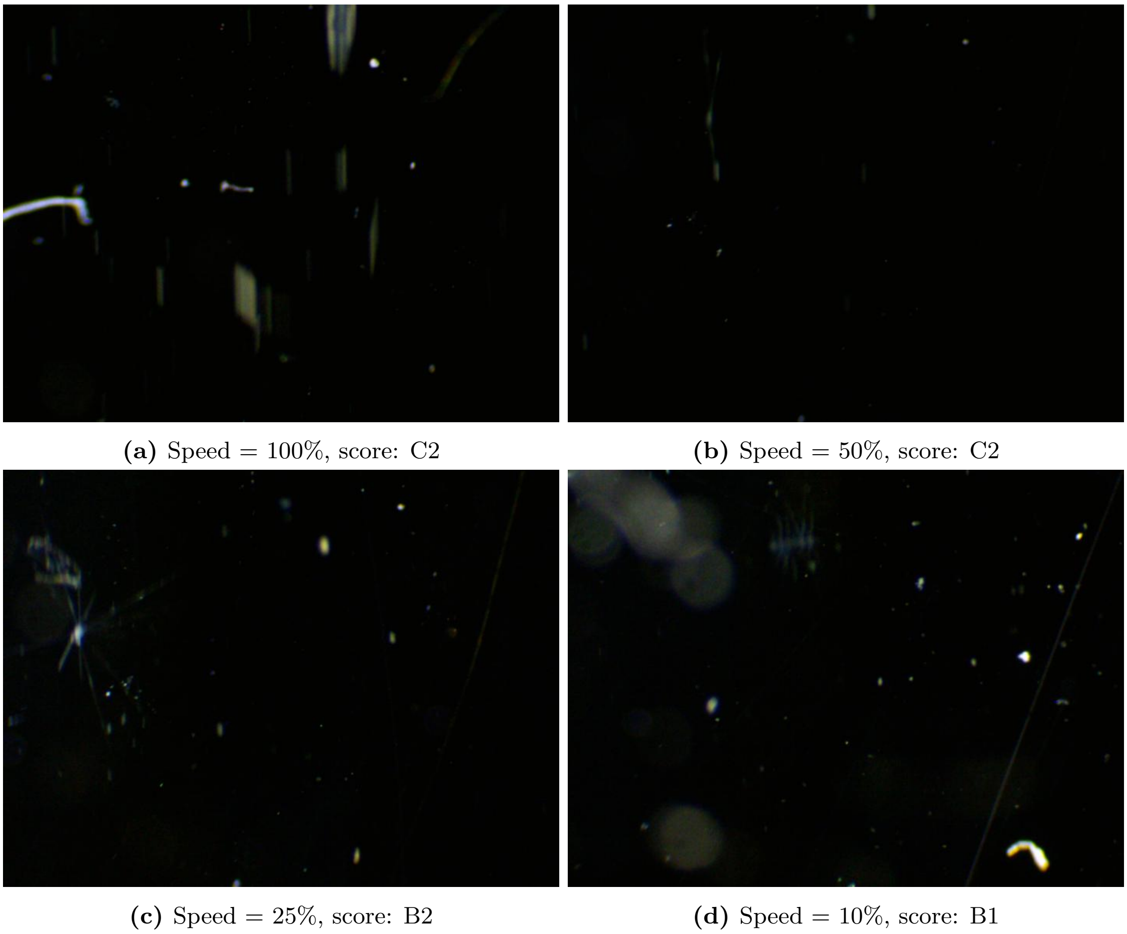
6 Results

6.1 Slide results

This section presents the results from the slide tests. For each slide used, example pictures are presented.

6.1.1 0.2mm Slide

The 0.2mm slide was the most shallow slide used under testing. 2203 pictures were taken, divided roughly over four speeds. The shallow slide caused the fluid to move quite fast past the objective, even at lower pump speeds. A low exposure time of 1ms was used to minimize the motion blur. Even at the lowest speed setting, specimen in focus had some degree of motion blur, and appeared dark due to the low exposure time of 1 ms. Pump speeds higher than 25% yielded poor results with scores reaching C2 at best, whereas lower speeds produced the best results yielding scores of B1 at 10% speed. Some excerpts are shown in figure 33.



(a) Speed = 100%, score: C2

(b) Speed = 50%, score: C2

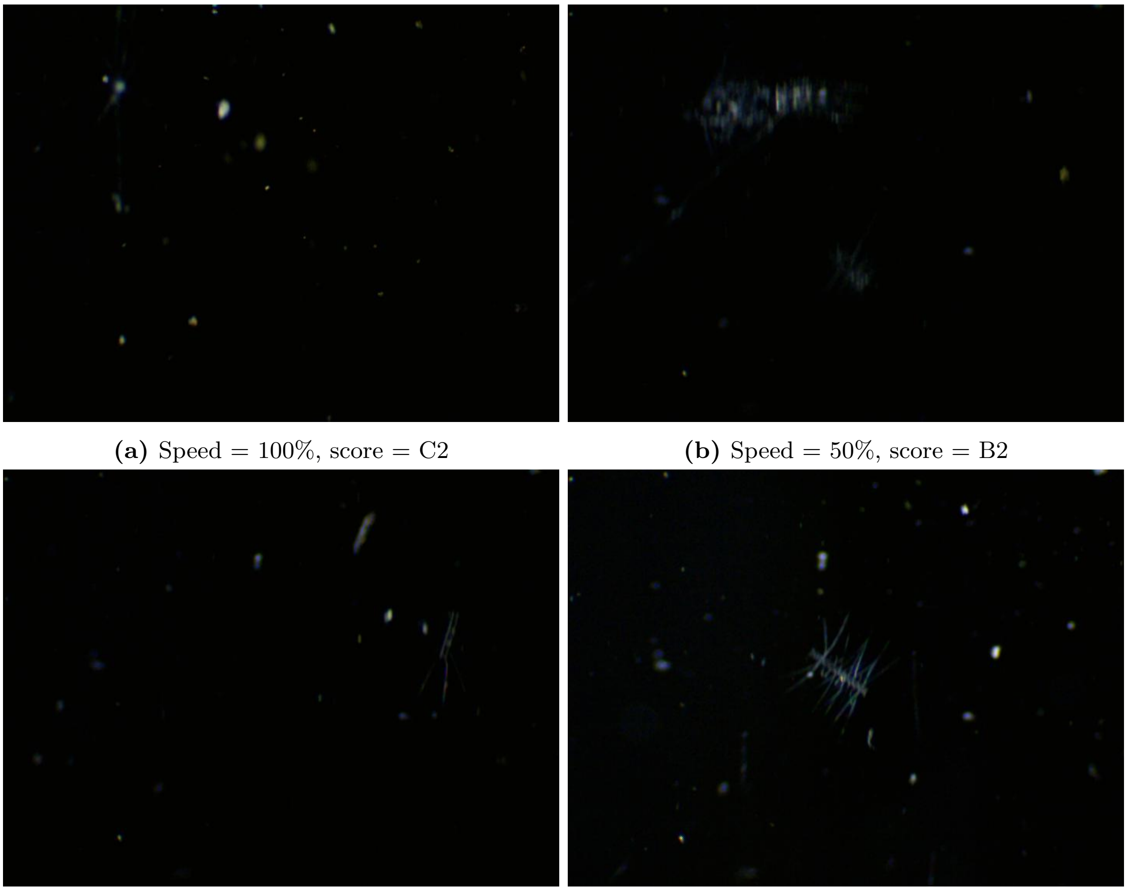
(c) Speed = 25%, score: B2

(d) Speed = 10%, score: B1

Figure 33: Results from 0.2mm slide.

6.1.2 0.4mm Slide

Testing with the 0.4mm slide produced 2309 images divided among 4 different speeds. As with the 0.2mm slide, viable images were only produced using lower speeds, from 25% and down. This speed produced pictures of A-level sharpness quality, and the low particle speed when pumping at 10% allowed the exposure time to be increased. For this speed, A1 picture quality was attained for most specimen in focus.



(a) Speed = 100%, score = C2

(b) Speed = 50%, score = B2

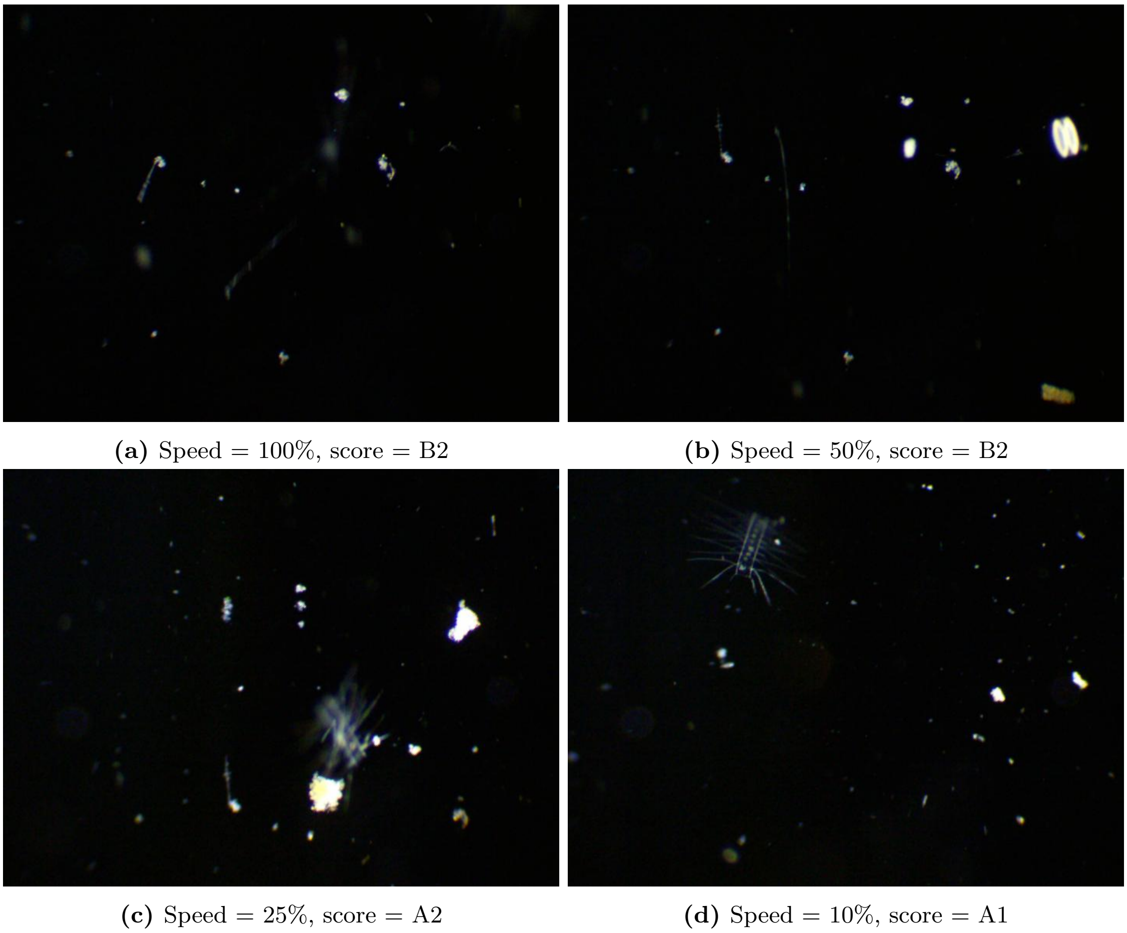
(c) Speed = 25%, A2

(d) Speed = 10%, A2

Figure 34: Results from 0.4mm slide.

6.1.3 0.6mm Slide

The data set produced from using the 0.6mm slide produced 2303 individual images. Speeds from 25% and down produced the best images, but certain frames at a speed of 50% can also be considered viable. Some particles were seen moving against the back of the slide. An example is shown in figure 35b, where a (likely) *Coscinodiscus* .sp is shown to gently roll along the bottom of the channel. Low particle speed allowed for a exposure time of 3ms for pump speeds of 25% and down. This produced great images with scores of A1 attained.



(a) Speed = 100%, score = B2

(b) Speed = 50%, score = B2

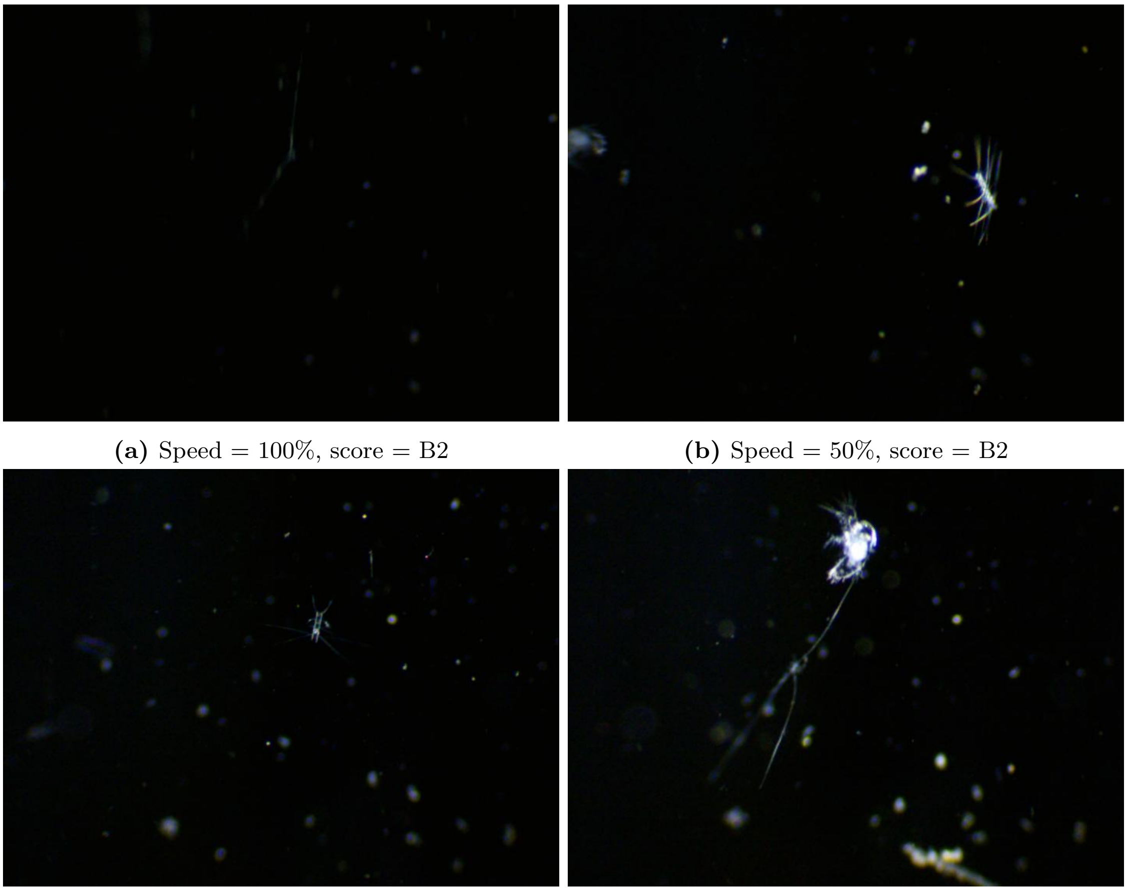
(c) Speed = 25%, score = A2

(d) Speed = 10%, score = A1

Figure 35: Results from 0.6mm slide.

6.1.4 0.8mm Slide

The additional depth of the 0.8mm slide meant that objects would move slower, compared to the previous slides. This meant that an exposure time of 3ms could be used. A speed of 100% produced some blurry images, whereas speed values of 50% and less produced the best results. A speed of 25% yielded a sufficiently slow particle speed and also an acceptable throughput. Due to the high frame rate used when grabbing the frames, particles were observed to be in the frame for up to 20 consecutive frames. Larger and (presumably) heavier particles were also observed to slide along the back of the slide when using the 0.8mm slide.



(a) Speed = 100%, score = B2

(b) Speed = 50%, score = B2

(c) Speed = 25%, score = A1

(d) Speed = 10%, score = A1

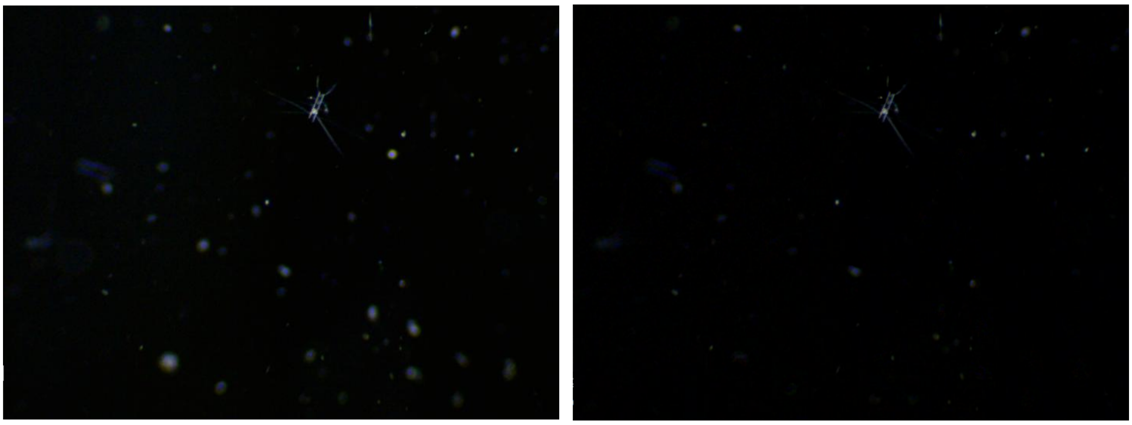
Figure 36: Results from 0.8mm slide.

6.2 Image Manipulation Methods

Based on the results from section 6.1, the data set generated using the 0.8mm-slide and a speed of 25% (1.95mL/min) was used when testing the image manipulation methods. For the method presented in section 6.2.3, an older data set known to contain copepods was also used.

6.2.1 Averaging Images

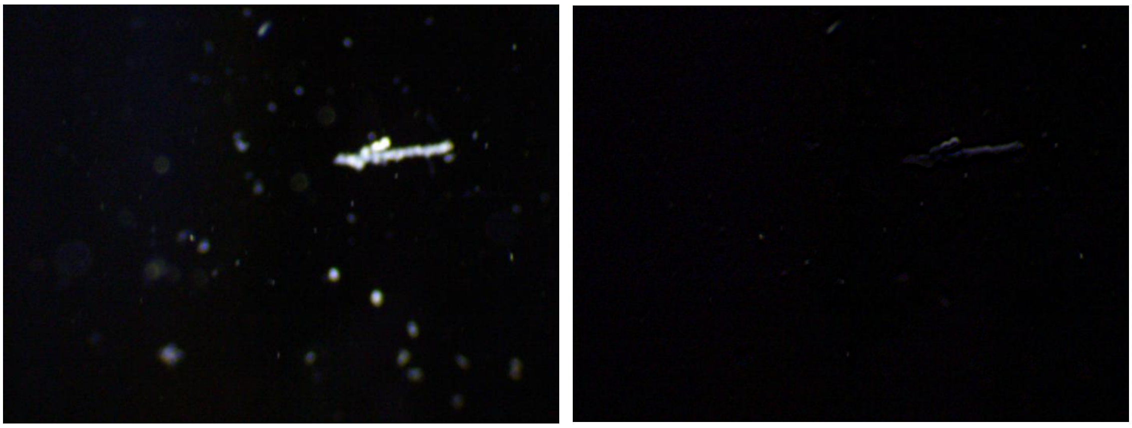
Subtracting the average of the last five frames from the current frame worked well to remove stuck particles and for correcting lighting anomalies. An example is shown in figure 37. When pumping at low speeds or sampling at higher frame rates, particles overlapping with the previous frames would have the overlapping parts removed. This is shown in figure 38.



(a) Frame before manipulation.

(b) Frame after manipulation

Figure 37: Results of subtracting the average of the last 5 frames from the current frame. After removing all stationary objects, only a *Chaetoceros* sp. and other small particles were visible.

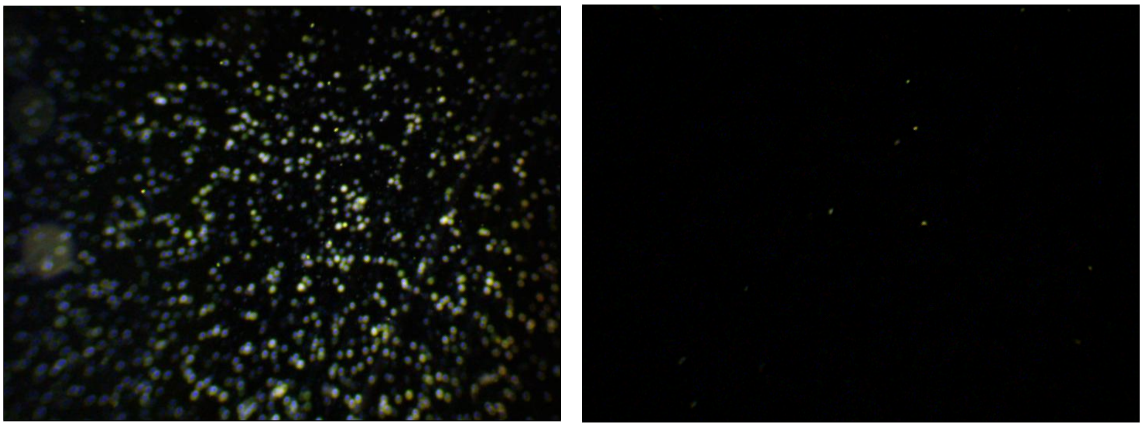


(a) Frame before manipulation.

(b) Frame after manipulation

Figure 38: Results of subtracting the average of the last 5 frames for a slow moving particle. The large white particle in figure 38a was overlapping with the previous frames, and was almost removed as shown in figure 38b.

The averaging method had great effect on the *Dunaliella* data set. Identifying the dunaliella while the pump was running was difficult due to their small size. They appeared to be avid swimmers however, and were easy to observe when the sample was standing still. Most of the specimen were stationary and presumed dead, but the ones alive were easy to spot especially after subtracting the averaged frames. This is shown in figure 39.



(a) Numerous *Dunaliella* sp.

(b) After manipulation.

Figure 39: *Dunaliella* sample before and after subtracting the average of the last five frames. The few particles in figure 39b were moving *Dunaliella* sp.

6.2.2 Empty Frame Removal

For every frame in the data set, the norm was calculated and subsequently plotted. This is shown in figure 40.

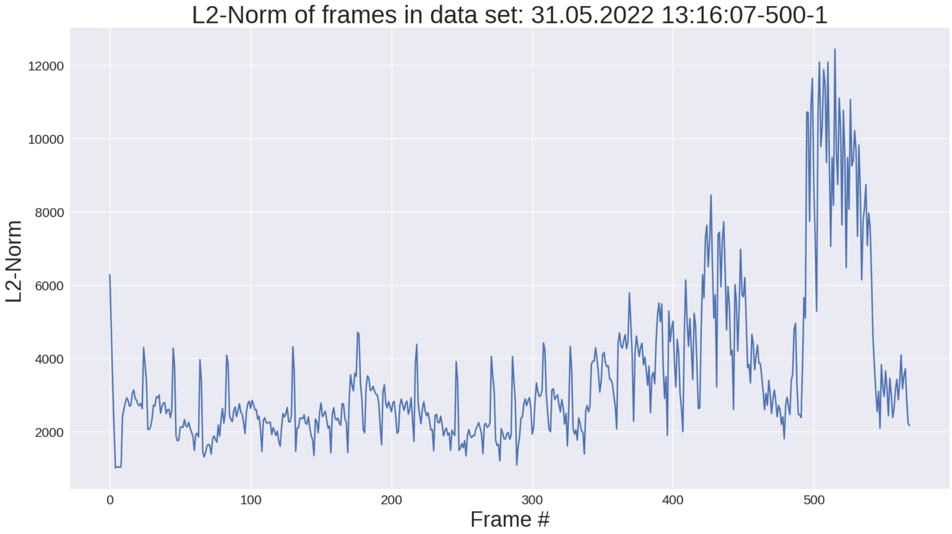


Figure 40: L2-norm plotted for every frame in the data set.

Numerous spikes were present in the data set. To verify that this was only a single occurrence with this data set, the norm was also calculated for another data set using the 0.6mm slide. This is shown in figure 41.

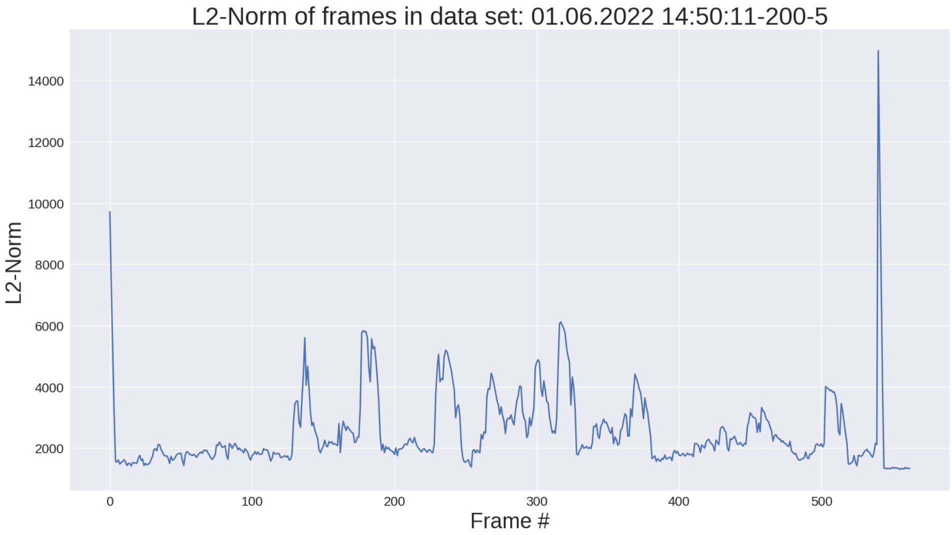
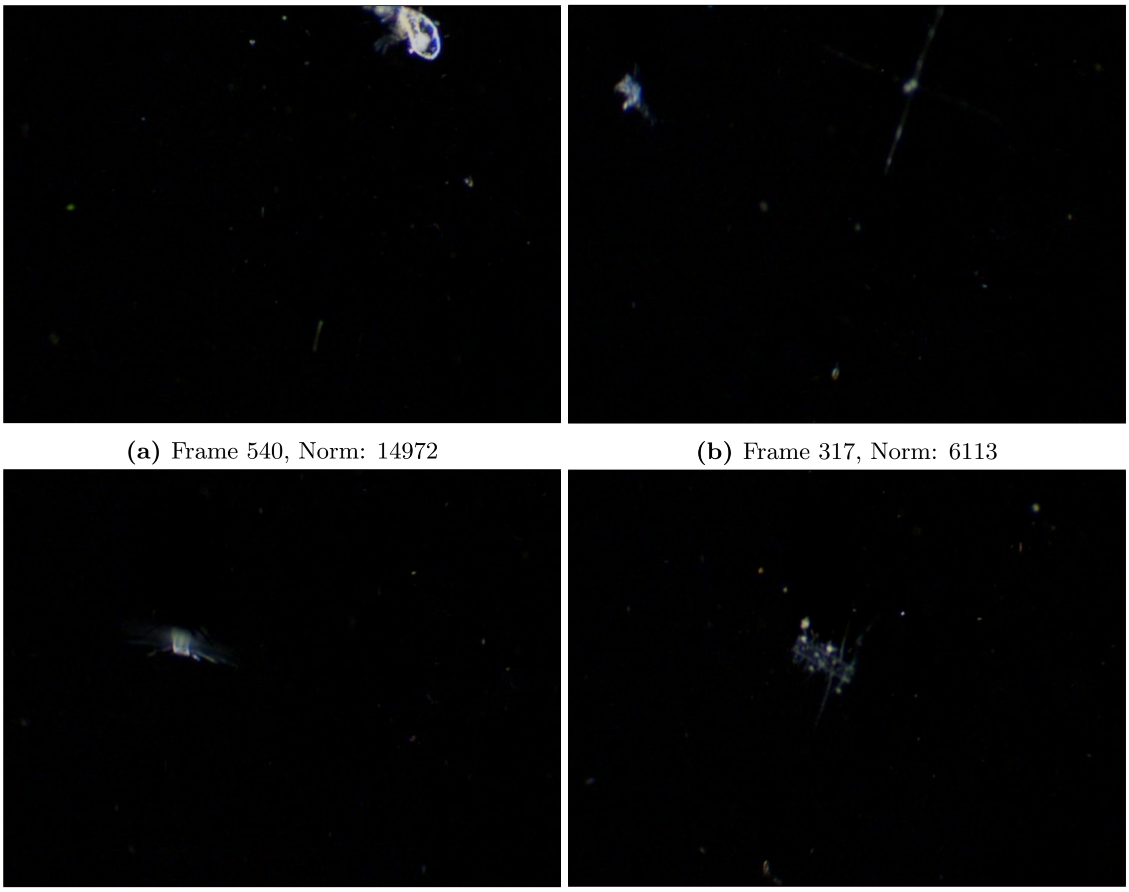


Figure 41: L2-norm plotted for every frame in the data set.

Significantly less noise was present in this data set compared to the first one. The list of norm values was then sorted in a descending manner, and select frames with their corresponding norm values are shown in figure 42.



(a) Frame 540, Norm: 14972

(b) Frame 317, Norm: 6113

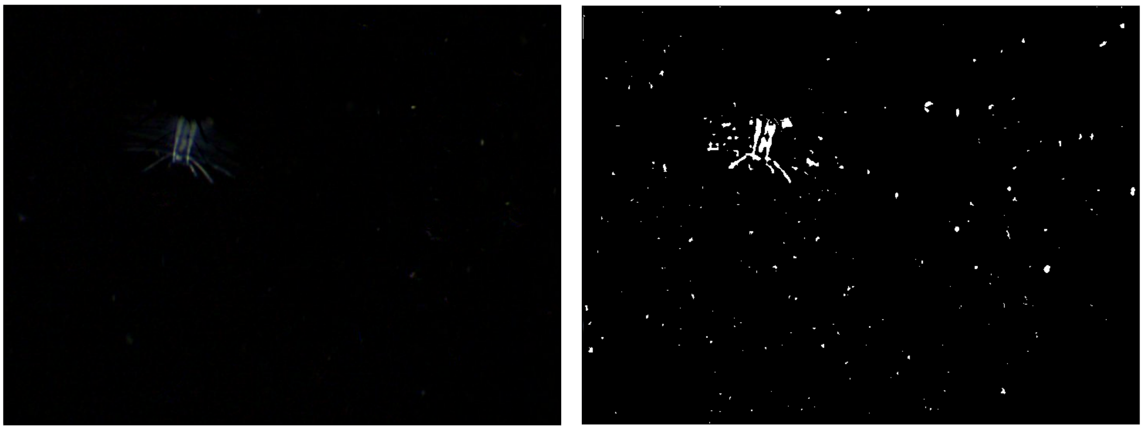
(c) Frame 137, Norm: 5596

(d) Frame 181, Norm: 5580

Figure 42: Select frames and their corresponding norm values.

6.2.3 Color Space Transformation and Color Masking

The images were converted to HSV color space using OpenCV, and a mask was created for any pixel within a certain hue, saturation and value range. The mask in figure 43b was created for any pixel within the following color range = $\begin{cases} \text{Hue} & \in [0, 179] \\ \text{Saturation} & \in [0, 90] \\ \text{Value} & \in [3, 255] \end{cases}$. This color range represented most ”bright” pixels, and included colors across the entire hue range.

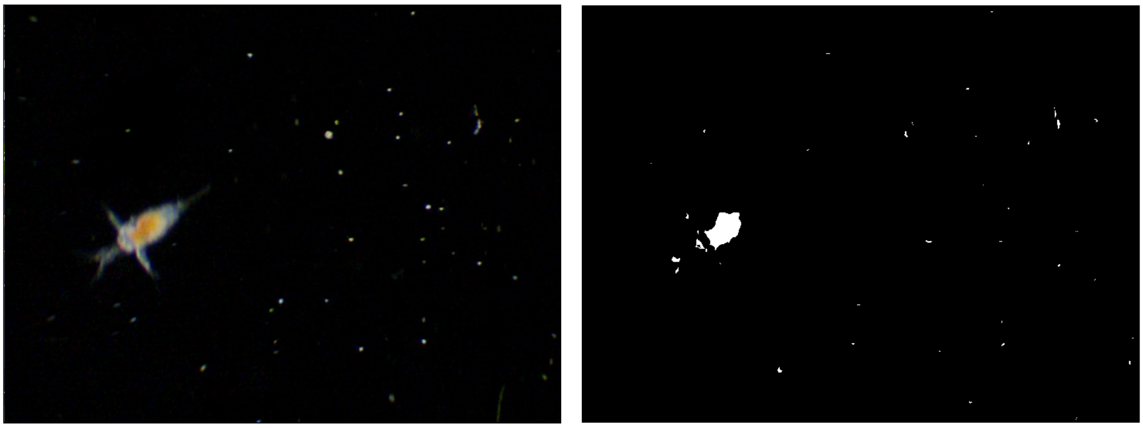


(a) Original image.

(b) Resulting mask created.

Figure 43: A mask (figure 43b) is created for colors within a certain range based on the image in figure 43a.

Masks for other colors were also created. In figure 44, a mask was created for any colors within the color range = $\begin{cases} \text{Hue} & \in [10, 30] \\ \text{Saturation} & \in [0, 255] \\ \text{Value} & \in [30, 255] \end{cases}$. This color range represented red-like colors, such as the orange body of the copepod in figure 44a.



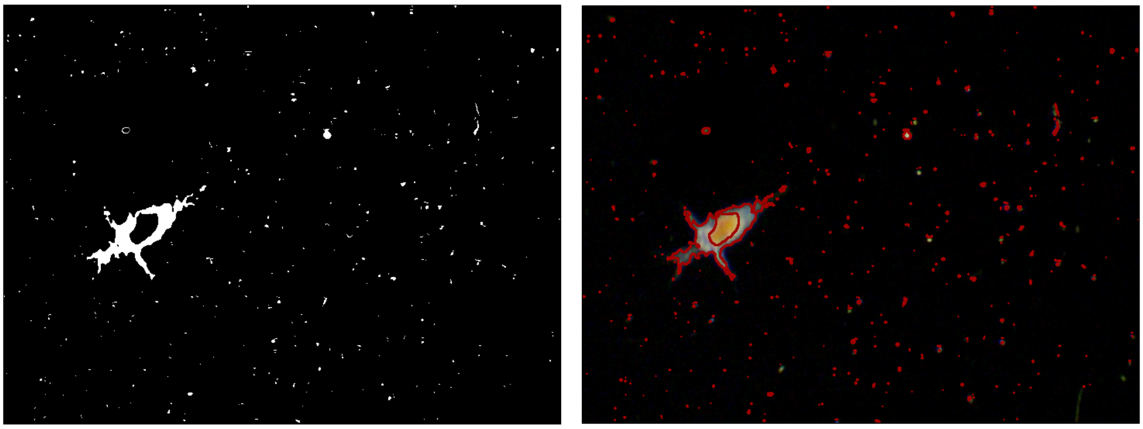
(a) A copepod.

(b) Mask created for red colors.

Figure 44: Creating a mask to detect features of certain organisms, such as the body of a copepod.

6.2.4 Size Estimation and Gaussian Blur

Size estimation of particles was done by applying boundary-detection algorithms using OpenCV on the masks from section 6.2.3. This is shown in figure 45.



(a) Mask created by filtering for the 'white' color values given in section 6.2.3.

(b) Boundary detection applied on the mask in figure 45a, and drawn over the original frame.

Figure 45: Particle size estimation was done by applying boundary detection methods to the binary mask.

The size was calculated from the encompassed area of the boundary. For any boundary with a size larger than a certain value, bounding boxes were drawn on top of the image. This is shown in figure 46.

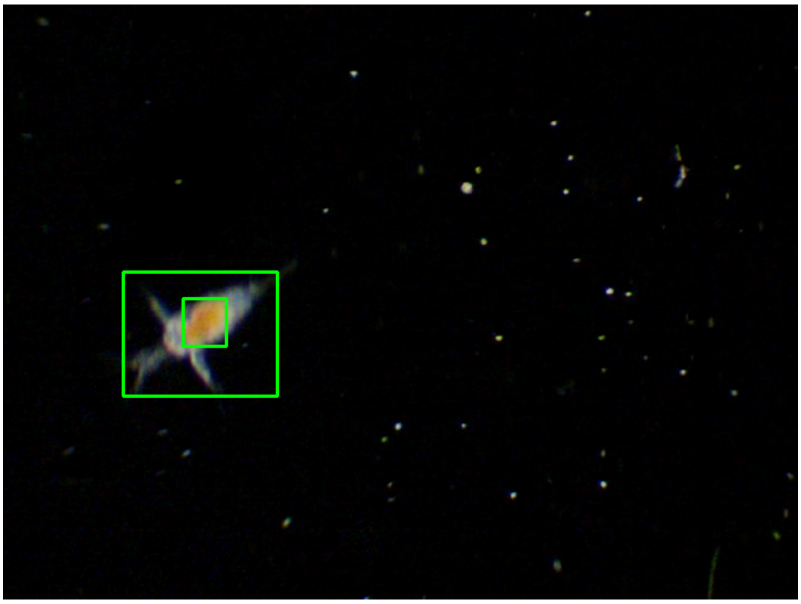
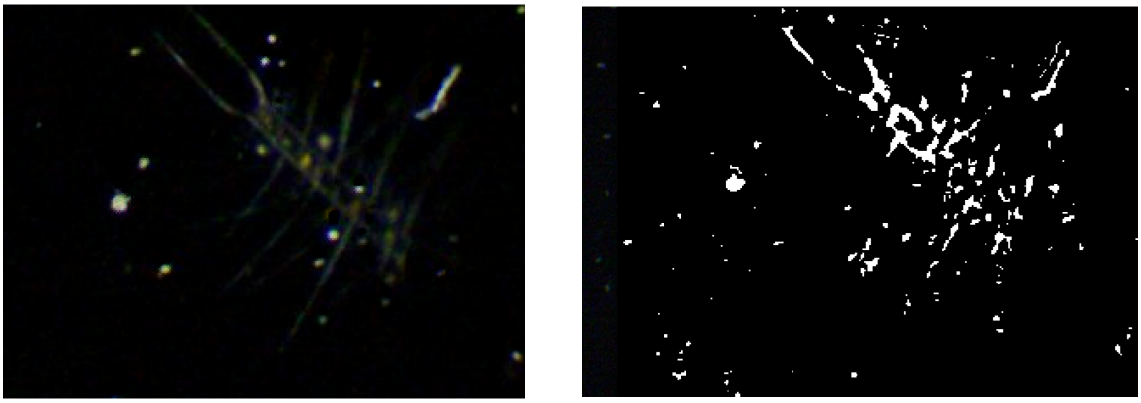


Figure 46: Bounding boxes drawn over masks whose area exceed a threshold. For this example, an area of 400 was chosen.

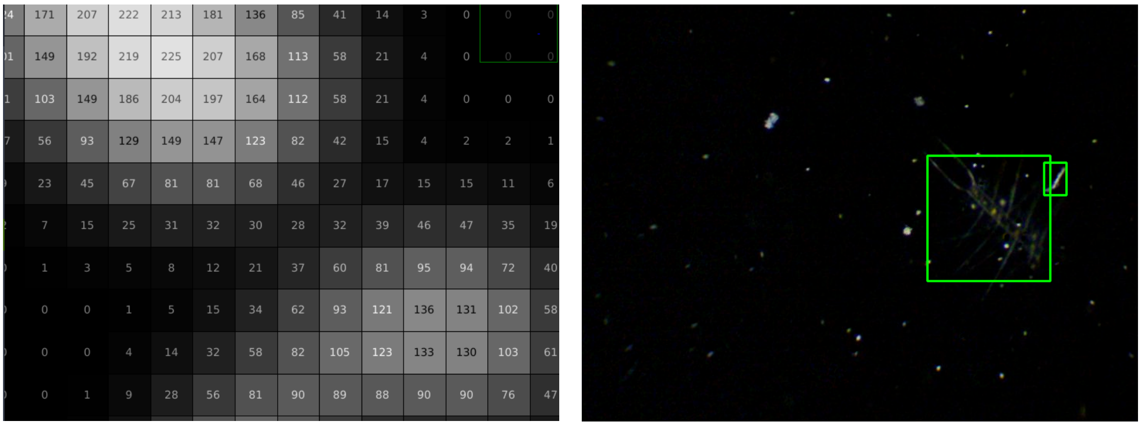
Particles whose mask appeared discontinuous and fragmented such as the specimen shown in figure 47b benefited from having a Gaussian blur filter applied to the mask. The blur would make the dark particles between the mask borders a non-zero value, thus bridging the gap and completing the boundary. This is shown in figure 48.



(a) Close up of a *Chaetoceros* sp.

(b) The resulting mask.

Figure 47: Certain particles had discontinuous mask that made boundary detection difficult.



4	171	207	222	213	181	136	85	41	14	3	0	0	0
1	149	192	219	225	207	168	113	58	21	4	0	0	0
1	103	149	186	204	197	164	112	58	21	4	0	0	0
7	56	93	129	149	147	123	82	42	15	4	2	2	1
	23	45	67	81	81	68	46	27	17	15	15	11	6
	7	15	25	31	32	30	28	32	39	46	47	35	19
	1	3	5	8	12	21	37	60	81	95	94	72	40
	0	0	1	5	15	34	62	93	121	136	131	102	58
	0	0	4	14	32	58	82	105	123	133	130	103	61
	0	1	9	28	56	81	90	89	88	90	90	76	47

(a) A closeup of figure 47b after a Gaussian blur with a kernel size of 7 was applied.

(b) Bounding boxes applied after applying a Gaussian filter.

Figure 48: A Gaussian filter was used to smooth out the mask such that the boundary became continuous and a bounding box could be applied.

Any particle larger than the area threshold was then cropped out of the image and saved as an individual image. This is shown in figure 49.

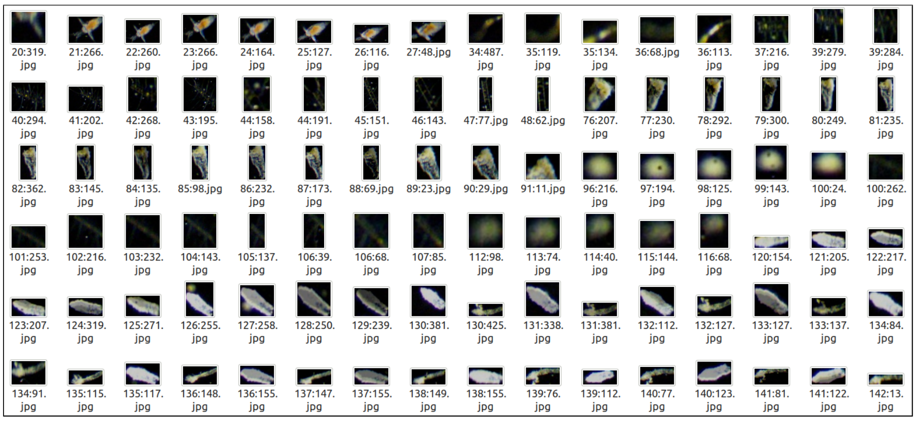


Figure 49: Cropped pictures of any particle with an area larger than the threshold value, saved to disk.

6.2.5 MorphoCut

The data set containing 2485 pictures taken from the TBS pier-sample was then run through the MorphoCut software. This data set was chosen due to its variety in species, and their sparsity in the data set. MorphoCut used 44 seconds to process the images and produced an archive containing 780 individual cropped images of detected objects. Examples of these are shown below in figure. The resulting images are in grayscale.

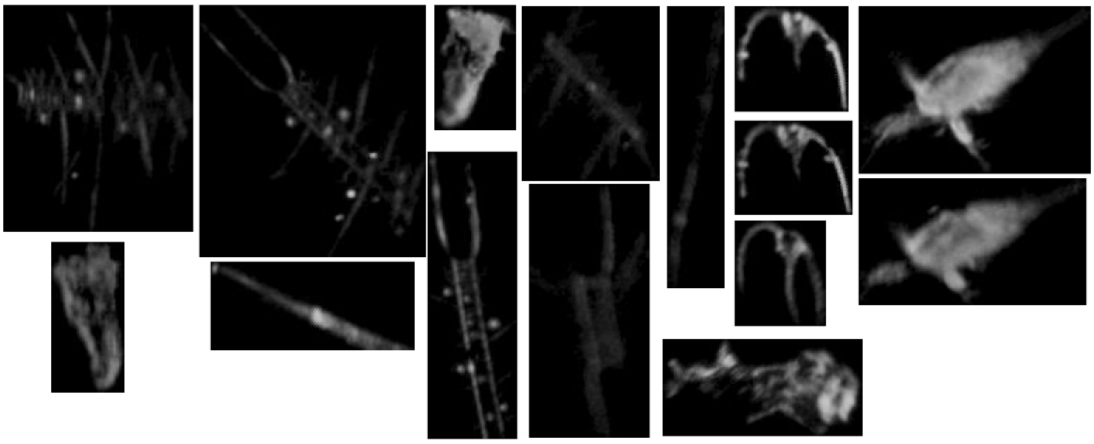


Figure 50: A subset of the images produced after processing the data set with MorphoCut.

MorphoCut was succesful in detecting specimen with solid boundaries such as the copepod to the far right in figure 50. It did however struggle detecting objects with long thin features, such as the specimen shown in figure 51.

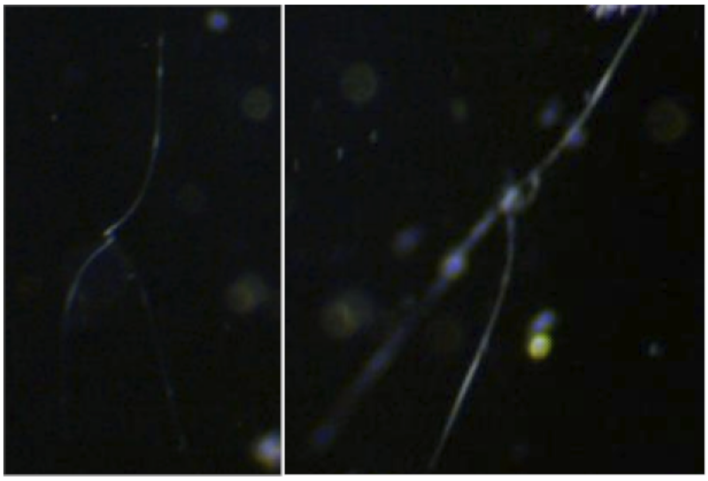


Figure 51: Particles with long, slender features which MorphoCut had difficulties in detecting.

7 Discussion

7.1 Recap

The goal of this thesis is to design and build a system that can image particles in a water sample under magnification. For the resulting data sets, image manipulation methods should be applied to filter away empty frames, remove irrelevant particles and detect any objects of potential interest.

The resulting system that was built is capable of imaging water samples at flow rates up to 7.8mL/min and capture upwards of 300 images per second. A microscope with a revolving nosepiece provides different magnifications through several objectives, ranging from 4x to 100x. When using a 4x scanning objective, a field of view of 2296.8μm by 1722.6μm is achieved. This setup provides a resolution of 3.19μm/pixel. The increased contrast due to the use of a darkfield condenser enables image-manipulation methods such as color space conversions and color masking to extract desired features that are unique for certain species.

7.2 Test Results

7.2.1 Sample age and organism decay

The samples used for testing were collected on may 16, 2022. Some tests were run the same day, but these were tested inconsistently and didn't follow a proper testing regime. With this in mind, it was decided to test the different slides as described in section 5.4. This was carried out on may 31, 2022. It is unclear to which degree organisms would die off and decay during this time period, and fewer larger organisms such as copepods were sighted in these data sets as compared to the ones generated on the day of collection.

7.2.2 Pump Speed vs Sampling Speed

Throughout this thesis, any flow rate mentioned has always referred to the sample flow rate in the system; i.e. the speed of which the pump has moved the sample. An argument can be made that the *sampling speed* of the system is less than this, due to the cameras field of view only being capable of observing 46% of the channel width. Thus, after processing a 5mL sample, only 2.3mL will have been imaged by the camera, assuming that the fluid speed is uniform in the slide. This is however a simplification, as fluid speeds are slower closer to the walls of the slide due to boundary conditions. With this in mind, the cameras field of view should be positioned in the middle of the slide as the highest sample flow rate would be found there.

7.2.3 Slide Results

Due to the nature of the samples, it was difficult to test the different slides in an objective manner. At the time of testing, only a subjective method of assessing the images was derived. Even if the particles were imaged more than once, it would be unlikely that they would be at the same position in the water column, or in the same orientation. The test results do however indicate that a deeper slide produces the best results, particularly the 0.8mm-slide. Despite the increased depth and possibility that a particle may be in an area out of focus, most particles would fall into the laminar and steady stream that was present in the middle of the slide. Heavier and larger objects such as copepods were observed to gently roll against the back of the slide, likely due to being pulled down by gravity and the slower fluid speed present there due to boundary conditions at the walls of the slide. Even though they were positioned against the back of the slide, the size of these particles ensured that certain features were in focus.

The shallower slides, particularly the 0.2mm- and 0.4mm slides, beat the deeper slides with regards to keeping particles in focus. When focusing at the middle of the slide channel, almost all particles were in focus regardless of their depth in the slide. Results did however show that the small cross section of the shallow slides increased the fluid speed, and images taken at speeds higher than 10% would appear blurry. This also reduced the exposure time that could be used, and as a result the data set appears darker. Based on these results, the 0.8mm slide is recommended for future testing. *ibidi*, the producer of the microscope slides, also produces custom slides based on customer requests. A slide that tapers inwards towards the middle in such a way that the field of view covers the entire water column would ensure that no particle could evade the camera. But again, this would increase the fluid speed past the objective, and an ideal depth would have to be found.

Ideally, the frames would be assessed using an *objective* method in addition to the subjective metric defined in section 5.3.2. This was not done during this test due to time constraints, but a method such as the one proposed by Marziliano et al[30] would yield an objective metric. In the mentioned article, a no-reference metric which determines the amount of blur is presented.

7.2.4 Image Manipulation Methods

Results showed that calculating the L2-norm of every frame in the data set proved to be an efficient and lightweight way of gauging whether a frame was populated with particles or not. The noticeable peak at the start of figure 40 and 41 likely stems from the first frames not having the average of the last 5 frames removed, and thus has more stationary particles included. It is unclear what caused the noise in the data set shown in figure 40, but the main suspects are a flickering LED light or camera related issues. Some problems with the LED were present during testing, in which the light intensity would periodically drop. This is inconsistent with the noise in figure 40, and the issue may originate from the camera.

Removing the average of the last five frames from the current frame had great effect on removing stuck particles and any form of glare or persistent light artifacts in the frame. Testing showed that slower moving particles would be partially removed as they were overlapping with the average image, and that when pumping on low speeds, the frame rate for recording should be adjusted accordingly. For smaller particles such as the *Dunaliella* sample, it was shown recording with the sample standing still as well as a high exposure time would easily show the organisms swimming around. This could also work in an seawater sample, due to smaller organisms being more numerous as described by the *Junge*-curves in section 2.1.1.

The color space transformations were considered a great approach as they unlocked the full capabilities of the darkfield condenser. Results showed that the binary masks made it easy to apply boundary detection algorithms, and subsequently size estimation methods. Coupled with a Gaussian blur filter, particles with discontinuous masks such as the *Chaetoceros* in figure 47 could be bounded, cropped and saved. This offered a simple alternative to the MorphoCut segmenter, and had the added benefit of saving images in RGB instead of grayscale such as MorphoCut.

Results also showed that creating masks for different colors can be used to pick out characteristics for certain objects, such as the copepod in figure 45. Additional masks can be created for different colors, which opens up the possibility to look for certain objects based on their hue value.

7.3 Physical System

Apart from certain issues with the microscope light source, the system integration process was one without major setbacks. It was suspected beforehand that the light source would flicker at the AC-power frequency of 50Hz. This was not the case however, as no results pointed to a 50Hz noise being present in the data sets. The noise in figure 40 is not consistent with the periodicity of the AC power. Regardless, when integrated into the AutoNaut the system would be powered by a DC-power source.

Despite not being integrated in this version, the stepper motors originally intended to be used for the focus adjusting could be considered vital should the system be deployed at sea. It is not unlikely that the rocking motion from the swells over time could knock the objective out of focus, rendering the images blurry. Suggested ways of refocusing includes remote control, but this would be dependant on a strong enough connectivity as a semi-live video feed would be required. A better suggestion would be to implement an auto-focus algorithm similar to the ones used in cameras. For every frame, a focus score would be calculated. The focus would then be adjusted slightly, and a new score would be calculated. This would continue until the maxima is found, and the camera would be in focus. This would yield great control of the focus point, allowing the objective to shift focus depth in the microscope slide.

It's also worth looking into an alternate light source for the system. For this thesis, the stock LED that came with the microscope was used. Its spectral distribution is unknown, which is not ideal for cases such as when a hyperspectral camera would be used. For hyperspectral applications, a light source providing an even intensity across the whole spectrum is required.

With regards to the microscope channel slides, testing showed that particles would accumulate and get stuck to the slide over time. Biofouling can also occur elsewhere in the system, such as in the silicone tubing. For long term operation, a form of cleaning procedure should be implemented between sampling runs. This could include pumping clean water at high speeds to dislodge stuck particles, and use a disinfectant solution such as bleach to kill off remaining organisms. In addition to the points above, a method of collecting and concentrating water samples on the AutoNaut should be looked into. During testing, plankton nets were used to collect the sample.

8 Future Work

With the discussion in mind, the following points are suggested for future work.

- Integrating additional stepper motors in the system. With three available stepper drivers on the CNC-shield, motors can be used to actuate the focus and slide position. Having control of the focus-knob enables for an auto-focus method to be implemented. A motor could also be used to rotate the revolving nosepiece, allowing for the magnification to be changed.

- Integration of the system into the AutoNaut. This would include designing a water inlet on the AutoNaut, and looking into a way to concentrate the sample prior to running it through the system. In addition, a form of cleaning system will be necessary after long deployments to prevent biofouling. Physical changes could also be explored, such as removing weight from the system.

- Accommodate for other sensors such as hyperspectral cameras. To facilitate these, a new light source for the system would be needed. Hyperspectral cameras are dependant on an even distribution of energy across the visible light spectrum, something a typical LED won't provide. As LEDs are in general more power efficient compared to a lightbulb, changing the light source may introduce the need for additional cooling.

- Tuning the parameters of MorphoCut and other image manipulation methods to improve segmentation. These methods were only briefly assessed during the testing, and a more thorough test could reveal ideal parameters for detection and segmentation. Finally, applying the objective test metric as described in section

- Testing a camera with higher resolution. Testing showed that pumping at a flow rate of 1.95ml/min provided the best pictures when using a 0.8mm slide. For this flow rate, the particles moved at a speed which meant that recording at high frame rates was not necessary. For this reason, a higher resolution camera should be tested to see if the resolution can be increased without a loss in image quality due to motion blur.

9 Conclusion

The goal of this thesis is to present a design for a system that can image particles in water samples under magnification, designed with an emphasis on operation in unmanned vessels. The proposed design takes images at 720x540pixels with a resolution of 3.19μm/pixel, and can image at flow rates up to 7.8mL/min. When used with a 4x scanning objective, a field of view of approximately 2296.8μm by 1722.6μm is achieved. The system is built around a microscope and readily available parts with a total cost of approximately 1900€. By designing the system around a darkfield condenser, increased contrast is achieved compared to systems using conventional brightfield lightning. The increase in contrast enables image manipulation methods to segment and extract objects of interest based on their color, brightness, or a combination thereof. Furthermore, methods to improve image quality and remove unwanted data is presented.

Testing with epipelagic water samples and lab cultivated algae cultures was carried out to gauge the systems performance. The test consisted of testing channel slides with different depths at different flow rates. This produced 16 data sets. Optimal results were achieved using microscope channel slides with a channel depth of 0.8mm, pumping at a flow rate of 1.95ml/min. This flow rate and slide combination yielded the clearest pictures with the least amount of motion blur. The image manipulation methods were tested on the data set which produced the best pictures. Calculating the L2-norm of every frame in a data set proved to be a simple and effective way of determining whether a frame was populated by particles or not. Color space conversions enabled the creation of masks for pixels within a certain color range. Boundary detection and size estimation applied to these masks implemented a way of extracting individual objects of interest from the frames.

Compared to other citizen grade imaging systems such as the PlanktoScope, the system has an increased cost, with this configuration costing around 1900€. The cost of the system is however dependent on the configuration, and cheaper components can bring the price down. Furthermore, fewer parts are needed and considerably less assembly is required. The system is also easily integrated into other embedded systems, as only two USB-ports are needed together with a 24V power supply. With a resolution of 3.19μm/pixel, the system is in its current configuration outperformed by the PlanktoScope at 1.5μm/pixel. However, the system can cater to a wide variety of cameras, and additional sensors such as hyperspectral cameras can be integrated. Coupled with other features such as a darkfield condenser, the system offers a customizable platform with ideal features for autonomous applications.

Bibliography

[1] 'Sublethal concentrations of the harmful diatoms, Chaetoceros concavicornis and C convolutes, increase mortality rates of penned Pacific salmon'. In: *Aquaculture* 117 (1993), pp. 215–225.

[2] Reidun Sirevåg. *blågrønnbakterier i Store norske leksikon på snl.no.* URL: https://snl.no/bl%C3%A5gr%C3%B8nnbakterier (visited on 6th Dec. 2021).

[3] 'Non-infectious gill disorders of marine salmonid fish'. In: *Reviews in Fish Biology and Fisheries* 21 (2011), pp. 423–440.

[4] *algetoksiner i Store norske leksikon på snl.no.* URL: https://snl.no/algetoksiner (visited on 6th June 2022).

[5] European Space Agency. *Sentinel Missions.* URL: https://sentinels.copernicus.eu/web/sentinel/missions (visited on 6th June 2022).

[6] Alberto Dallolio. *AutoNaut.* URL: https://autonaut.itk.ntnu.no/doku.php (visited on 14th Mar. 2022).

[7] Alberto Dallolio. *AutoNaut Documentation.* URL: https://autonaut.itk.ntnu.no/lib/exe/detail.php?id=start&media=autonaut_cover3.jpg (visited on 8th June 2022).

[8] Glaucia Moreira Fragoso et al. 'Trait-based analysis of subpolar North Atlantic phytoplankton and plastidic ciliate communities using automated flow cytometer'. In: *Limnology and Oceanography* 64.4 (2019), pp. 1763–1778. DOI: https://doi.org/10.1002/lno.11189. eprint: https://aslopubs.onlinelibrary.wiley.com/doi/pdf/10.1002/lno.11189. URL: https://aslopubs.onlinelibrary.wiley.com/doi/abs/10.1002/lno.11189.

[9] Fabien Lombard et al. 'Globally Consistent Quantitative Observations of Planktonic Ecosystems'. In: *Frontiers in Marine Science* 6 (2019). ISSN: 2296-7745. DOI: 10.3389/fmars.2019.00196. URL: https://www.frontiersin.org/article/10.3389/fmars.2019.00196.

[10] *PlanktoScope.* URL: https://www.planktoscope.org/discover (visited on 28th Feb. 2022).

[11] Thibaut Pollina et al. 'PlanktonScope: Affordable modular imaging platform for citizen oceanography'. In: *bioRxiv* (2020). DOI: 10.1101/2020.04.23.056978. eprint: https://www.biorxiv.org/content/early/2020/04/23/2020.04.23.056978.full.pdf. URL: https://www.biorxiv.org/content/early/2020/04/23/2020.04.23.056978.

[12] Simon Martin Schroeder. *the MorphoCut image processing library.* URL: https://morphocut.readthedocs.io/en/stable/index.html (visited on 6th June 2022).

[13] Nandini Menon, Monalisha Sundar and Minu Punathil. 'Phytoplankton functional types'. In: Oct. 2018, pp. 32–34. ISBN: ISBN-978-93-82263-18-0.

[14] Alistair Hobday et al. 'Impacts of climate change on Australian marine life'. In: (Oct. 2006), p. 20.

[15] Curtis Mobley. *Particle Size Distributions.* URL: https://www.oceanopticsbook.info/view/optical-constituents-of-the-ocean/level-2/particle-size-distributions (visited on 27th May 2022).

[16] Dariusz Stramski Hugh Runyan Rick A. Reynolds. 'Evaluation of Particle Size Distribution Metrics to Estimate the Relative Contributions of Different Size Fractions Based on Measurements in Arctic Waters'. In: *IGR Oceans* 125 (6 2020).

[17] D Stramski and D Petelski. 'Light scattering by microorganisms in the open ocean'. In: *Progress in Oceanography* 28 (6), pp. 343–383.

[18] Theodore J. Smayda. 'What is a bloom? A commentary'. In: *Limnology and Oceanography* 42.5part2 (1997), pp. 1132–1136. DOI: https://doi.org/10.4319/lo.1997.42.5_part_2.1132. eprint: https://aslopubs.onlinelibrary.wiley.com/doi/pdf/10.4319/lo.1997.42.5_part_2.1132. URL: https://aslopubs.onlinelibrary.wiley.com/doi/abs/10.4319/lo.1997.42.5_part_2.1132.

[19] Scott W. Nixon. 'Coastal marine eutrophication: A definition, social causes, and future concerns'. In: *Ophelia* 41.1 (1995), pp. 199–219. DOI: 10.1080/00785236.1995.10422044. eprint: https://doi.org/10.1080/00785236.1995.10422044. URL: https://doi.org/10.1080/00785236.1995.10422044.

[20] NRK. URL: https://www.nrk.no/sorlandet/680-tonn-fisk-drept-av-alger-1.336122 (visited on 7th June 2022).

[21] Jesse Allen and Robert Simmon. *Toxic Algae Bloom in Lake Erie*. URL: https://commons.wikimedia.org/wiki/File:Toxic_Algae_Bloom_in_Lake_Erie.jpg (visited on 7th June 2022).

[22] (3ucky(3all. *HSV cone*. URL: https://commons.wikimedia.org/wiki/File:HSV_cone.png (visited on 22nd May 2022).

[23] *OpenCV*. URL: https://opencv.org (visited on 23rd May 2022).

[24] EdmundOptics. *Infinity Corrected Objectives*. URL: https://www.edmundoptics.com/knowledge-center/frequently-asked-questions/microscopy/#:~:text=A%5C%20semi-plan,%5C%20or%5C%20semi,have%5C%20an%5C%2080%5C%25%5C%20flat%5C%20field (visited on 7th June 2022).

[25] *BRESSER MikroCam II UHSP 0.4*. URL: https://www.bresser.de/en/Microscopes-Magnifiers/Microscope-Cameras/USB-Cameras/BRESSER-MikroCam-II-0-4-UHSP-microscope-camera.html (visited on 2nd May 2022).

[26] *BRESSER Researcher Trino*. URL: https://www.bresser.de/out/pictures/master/product/1/267447b43119ad09be23110723263a9d_5750900_m_1_v0416.jpg (visited on 2nd May 2022).

[27] *BRESSER Darkfield Condenser Dry*. URL: https://www.bresser.de/out/pictures/generated/product/2/640_640_70/b7b33525de186b0f5e04ace3e90d8f6d_5942190_m_2.jpg (visited on 2nd May 2022).

[28] *Arduino Uno Rev3*. URL: https://store.arduino.cc/products/arduino-uno-rev3?selectedStore=eu (visited on 3rd May 2022).

[29] *Ooznest Arduino CNC Shield*. URL: https://ooznest.co.uk/product/arduino-cnc-shield/ (visited on 3rd May 2022).

[30] P. Marziliano et al. 'A no-reference perceptual blur metric'. In: *Proceedings. International Conference on Image Processing*. Vol. 3. 2002, pp. III–III. DOI: 10.1109/ICIP.2002.1038902.

NTNU

Kunnskap for en bedre verden