OC2020 A-033 - Open

Report

Data capture and real-time data quality analysis

Author(s)
Eleni Kelasidi
Espen Moen, Christian Schellewald, Mauhing Yip, Bjørnar Moe Remmen

SINTEF Ocean AS

2020-03-16

SINTEF Ocean AS

Address:
Postboks 4762 Torgarden
NO-7465 Trondheim
NORWAY

Switchboard: +47 46415000

Enterprise /VAT No:
NO 937 357 370 MVA

Report

Data capture and real-time data quality analysis

REPORT NO.	**PROJECT NO.**	**VERSION**	**DATE**
OC2020 A-033	302002547	1.0	2020-03-16

KEYWORDS:
[Keywords]

AUTHOR(S)
Eleni Kelasidi
Espen Moen, Christian Schellewald, Mauhing Yip, Bjørnar Moe Remmen

CLIENT(S)
WaterLinked AS

CLIENT'S REF.
[Clients ref]

NUMBER OF PAGES/APPENDICES:
39 + Appendices

CLASSIFICATION
Open

CLASSIFICATION THIS PAGE
Open

ISBN
978-82-7174-377-2

ABSTRACT

This report presents results obtained in the CageReporter project regarding the development of a 3D vision system to be used for data capture in fish cages. The developed system enables to obtain high-quality data with the overall goal to identify fish conditions and perform cage inspections during daily operations, as well as the robotic vision for an underwater vehicle during the adaptive operation planning in the cage. A compact and robust sensor with optical components and lighting system was developed. In addition, this activity presents development of methods to evaluate the quality of the captured data. Based on defined quality criteria associated with fish conditions and cage inspection operations, algorithms have been developed to evaluate whether the quality criteria are met. The algorithms have been validated using image data obtained from 24/7 video streams from a full-scale fish cage. The work furthermore includes the development of image processing algorithms to estimate the distance and orientation relative to the inspected object of interest, such as the fish or the net. The developed algorithms have been validated based on vision data obtained during tests both in lab- and full scale.

PREPARED BY	**CHECKED BY**	**APPROVED BY**
Eleni Kelasidi	Nina Bloecher	Leif Magne Sunde

Document History

VERSION	DATE	VERSION DESCRIPTION
1.0	2020-03-16	Final version

Table of Contents

1 Background

The CageReporter project adapts the use of autonomous and tetherless underwater vehicles as a carrier of sensor systems for data acquisition, where the data are transferred from sea-based fish cages to a centralized land base (Figure 1). The vehicle will use active motion control and acquire data from the cage environment while exploring the fish cages. The main project objective was to develop technology for autonomous functionality for adaptive mission planning to achieve high quality data acquisition from the cage space. One of the most important capabilities within this context is to operate in a dynamically changing environment in interaction with the biomass (bio-interactive) and the aquaculture structures. The project addresses many challenges within the aquaculture industry related to poor accuracy and representative sampling of important variables from the whole volume of the cage. A successful project outcome will lead to new technology for collection of high-resolution data that could be utilized for assessment of the fish farm state, grouped within three main areas: A) fish, B) aquaculture structures and C) production environment. Examples of areas of applications are detection of abnormal fish behaviour, net inspection and mapping of water quality. CageReporter aims to provide a solution for continuous 24/7 inspection of the current situation and will be the mobile eyes of the fish farmer in the cage environment. The project idea is based on using low-cost technology for underwater communication, vehicle positioning, and camera systems for 3D vision.

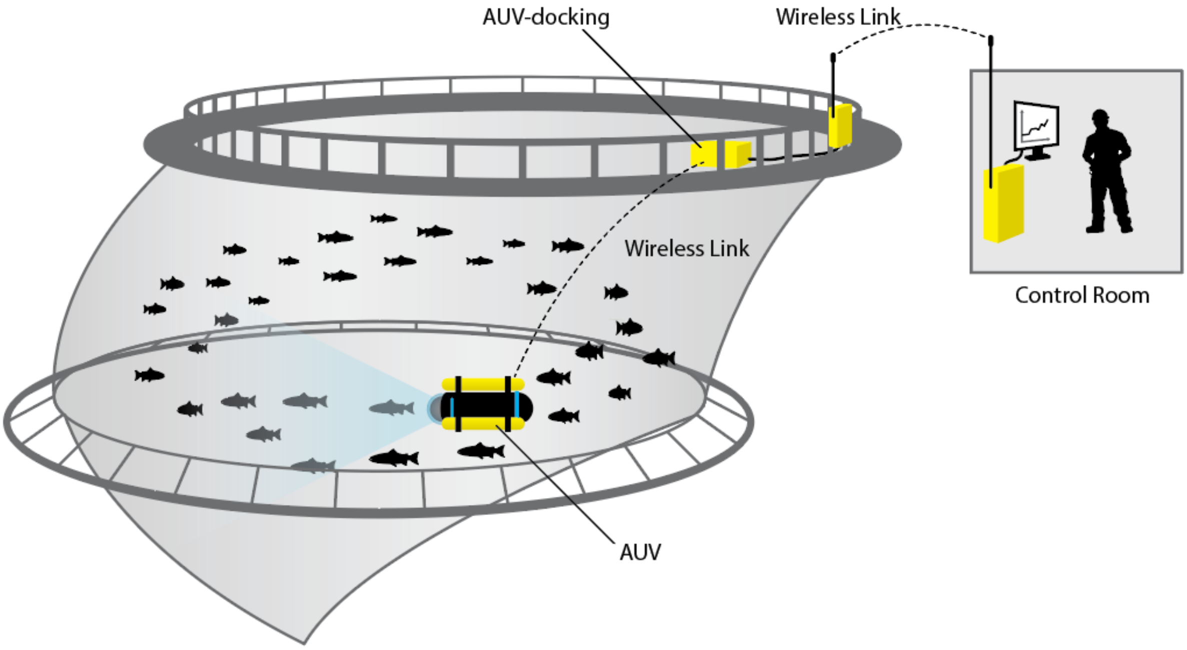


Figure 1 Resident (24/7), autonomous, non-tethered vehicle (AUV) for high quality data acquisition.

A key element of the project is to capture high-quality vision data from the cage. To obtain relevant high-quality vision data using the currently available state-of-the-art systems (e.g. systems based mostly on stationary sensors) is highly demanding process, and in many cases fails to obtain data describing the dynamic farming environment with sufficient resolution and accuracy. Biomass production at a single site can have up to 15.000 tonnes of salmon, in a water volume up to 50.000 m^3. In the future, these volumes are expected to increase even more, meaning that such large volumes cannot be considered as homogeneous environments, and thus it is not possible to obtain accurate and detailed information based on vision data collected using stationary sensor systems. The distribution of fish and variables related to the production environment vary in the cage, both through the day and with season. An autonomous underwater vehicle being equipped with a 3D vision system will be able to collect data from the whole volume of the cage. It is essential to develop a system that is able to capture data that describes the conditions of fish, cage net and production environment since this information can be used for a better mapping of environmental effects (escapes, feed lice, lice), improvement of fish welfare and economics. An important feature of the developed system is the real-time quality control of the obtained data in order to sort out data that does not meet objective quality criteria. Based on quality-assured data, a better decision support system can be developed for more objective decisions during operations in fish farms.

2 Sensor system for 3D vision

This section present results regarding the development of an underwater 3D vision system for use in fish cages, aiming to monitor the condition of the fish, inspect the fish cage facility as well as provide vision for a Remotely Operated Vehicle (ROV), which it will be mounted to. The R&D challenges to develop a 3D vision sensor system are related to the development of a camera and lighting systems that provide high-quality data under varying light conditions and visibility in the water (Figure 2). This is particularly demanding for high turbidity water, which provides optical dispersion and damping, limiting the observation volume. To prevent artificial lighting interfering with the fish, wavelengths invisible to the fish are assessed in combination with light-sensitive camera sensors. The project partner, SEALAB, has since its founding had ambitions to utilize and develop an underwater stereo vision system. Different applications have been tested earlier, including plenoptic cameras as well as rigs with two 2D cameras. This, SEALAB had knowledge and experience concerning stereo applications prior to this project. However, prioritizing other necessary work areas over stereo projects meant that the CageReporter project was essential to push this in a progressive direction. In this activity, SEALAB has provided hardware and software to capture and store the desired stereo data. This data has subsequently been used both by SINTEF and SEALAB for developing algorithms to achieve stereo vision.

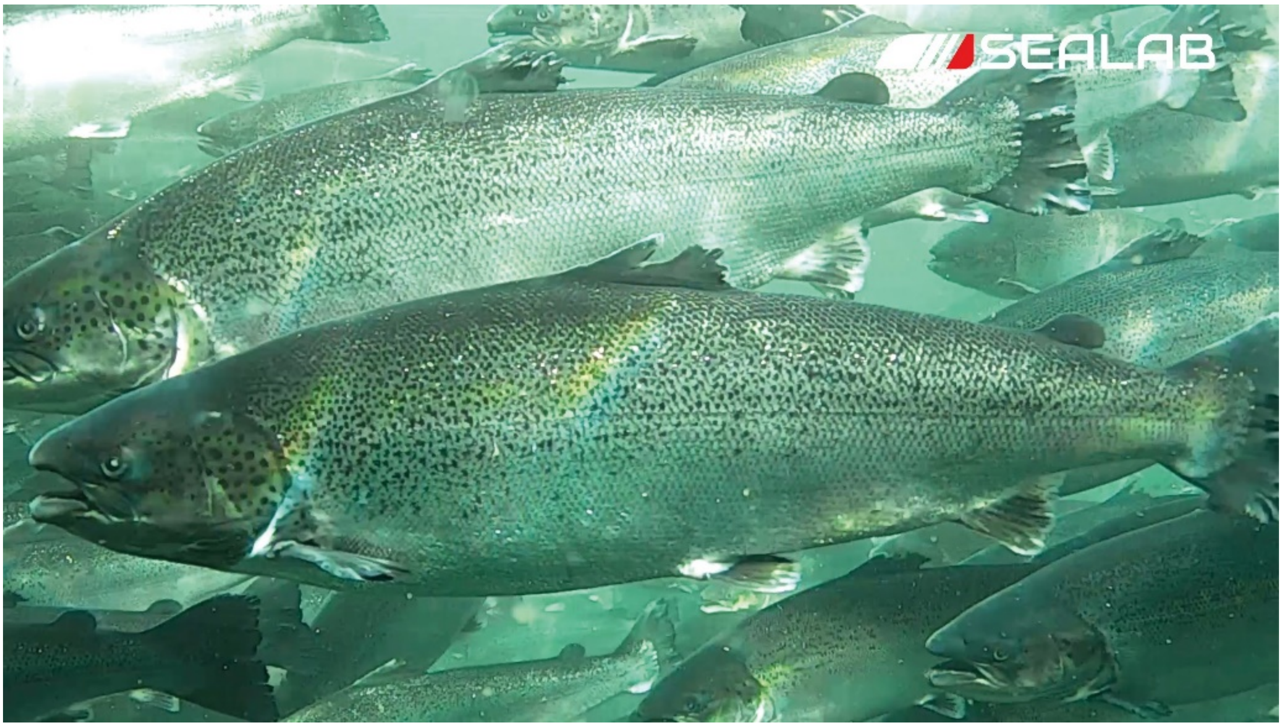
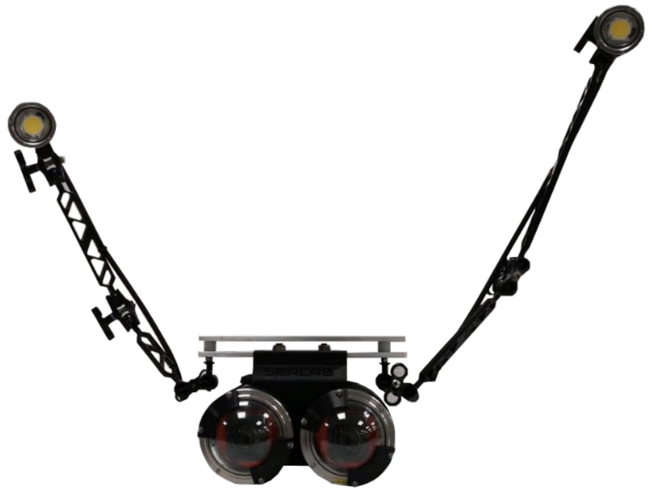


Figure 2 Frame from video of salmon with SEALAB camera system.

2.1 Sensor system with hardware and software

For the development of the 3D vision system (Table 1) two 4K cameras were mounted on a stereo rig to capture the left and right video stream representing the main components in a stereoscopic vision. The specifications of this camera are summarized in Table 1. This equipment has to be encapsulated in a waterproof casing and arranged in such a way that it can be mounted as payload on a ROV. Figure 3 shows a 3D rendering of the stereo camera setup, taken during development stages.

Table 1 Underwater camera used to develop the 3D vision system

3D stereo system with two 4K cameras and lights	**Camera specifications**
	**Sensor:** 1/2,5- type Exmor R CMOS **Video format:** 4K, 1080p, 720p, 480p **Optical zoom:** 20x **Video Output:** Y/Cb/Cr 4:2:2, R/G/B 4:4:4 **Dimensions:** **Length:** 250 mm **Diameter:** 125 mm **Weight in air:** 11 kg **Weight in water:** 2.3 kg

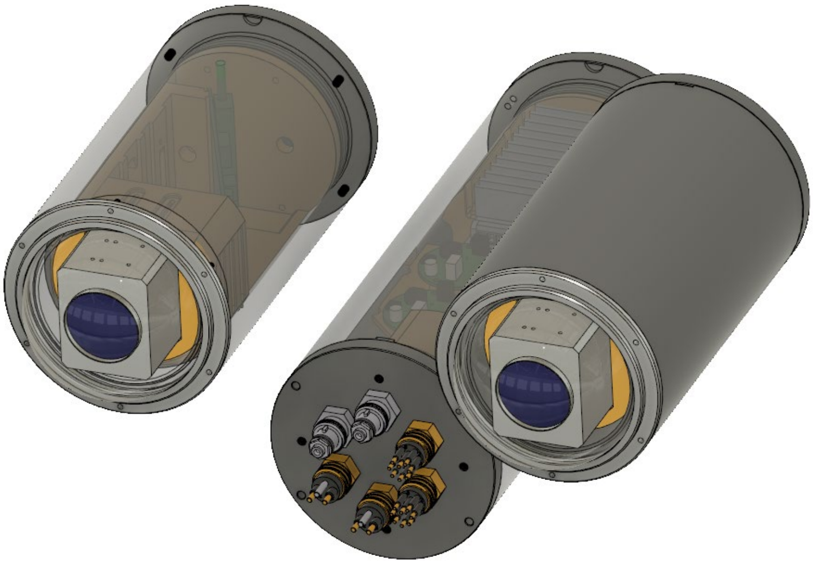


Figure 3 Render of the stereo camera.

Setup

As indicated above, two 4K cameras have been used to build the stereo vision system. **Error! Reference source not found.** shows an illustration of the system architecture. *Topside* refers to the location of the operator and represents the control center of this system as well as where the data is collected and image processing techniques will be executed. This topside was located on the boat MS *Torra* for full scale demonstrations in this project. The *Main Pod* connects all the components together and communicates with the *Camera Housings*, *Lights* and the *Topside*. The *Camera Housing* and *Light* are the components where the cameras and the lights are located. A figure of the lights and a summary of their specifications can be found in **Error! Reference source not found.**. This system was mounted to an underwater vehicle in this project, however, as the system is independent it could also be used independently. The umbilical from the *topside* to the *main pod* will be attached on the underwater vehicle's umbilical.

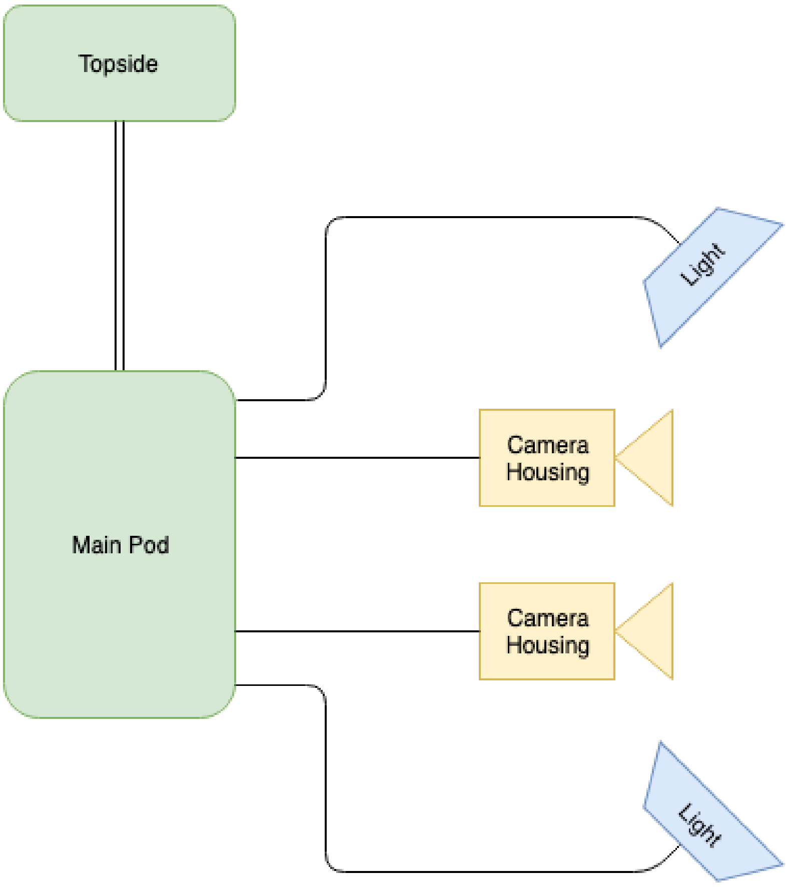
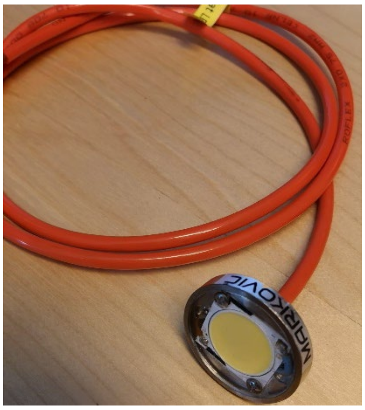


Figure 4 System architecture.

Table 2 Light system used to develop the 3D vision system

<u>Light System Photo</u>	**<u>Specifications</u>**
	**Brightness:** 8000 Lumen **Control Interface:** RS485 **Electric Specifications:** 48VDC, 1.4A (Max) **Dimmable in 255 steps**

Topside

At the *Topside*, video and control signals from both camera housings as well as the control signals for the lights are transferred by fibre optic cables. This is achieved over three separate fibre optic cables, one for each camera housing and one for the light control. Together with these signal cables, a set of copper cables are required to supply the system with 48 VDC. Between the *Topside* and the *Main Pod,* a hybrid subsea umbilical is used consisting of 4 single mode fibre optic cables, 4 multi-mode fibre optic cables and 4 copper wires (Hybrid cable Type 3444; MacArtney). Table 3 shows some of the specifications of this cable. It was decided to use a Y-split at the end of the umbilical connected to the *Main Pod*, as they did not have a hybrid connector which fulfilled the requirements regarding the number of copper pins and optical fibre connections. Figure 5 shows the Y-Split, where the black connector is for the optical fibre and the red for the copper.

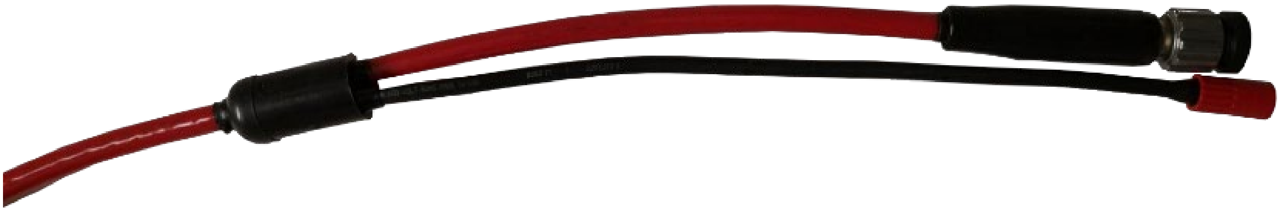
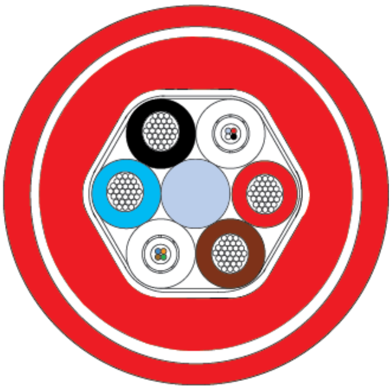


Figure 5 Y-Split.

Table 3 Hybrid cable used between topside and camera system

Umbillical Illustration	Specifications[1]
	**Diameter:** 15.50 mm ± 0.4 mm **Weight in air:** 235 kg/km nom **Weight in seawater:** 42 kg/km nom **Depth rating:** 5000 m **Min breaking strength:** 15 kN **Max attenuation (Fibre):** 0.40 dB/km at 1300 nm for single mode. 0.25 dB/km at 1550 nm for single mode.

The topside computer that was used was configured and installed for this specific purpose. Table 4 shows the specifications of this computer containing a *Pro Capture Dual HDMI 4K Plus LT.* This is a video capture card from *Magewell* which connects the camera inputs. This card has both Windows and Linux compatible drivers and has proven to be a good choice. In addition, the software used to capture video was OBS Studio v24.0.3 as well as FCB control software v6.1.0.0 for configurating the cameras.

[1] MacArtney Underwater Technology, "Hybrid cable, Kevlar - Type 3444", https://www.macartney.com/what-we-offer/systems-and-products/stock-cables/hybrid-cables/hybrid-cable-kevlar-type-3444/

Table 4 Topside computer specifications

CPU	INTEL Core i9-9900K
GPU	MSI GeForce RTX 2080 Ti VENTUS 11G
CAPTURE CARD	Pro Capture Dual HDMI 4K Plus LT
MEMORY	32GB DDR4
STORAGE	14 TB (4GB SSD + 10GB SATA)

Main Pod

Connecting the *Topside* with the cameras and the lights is the main functionality of the *Main Pod*. Figure 6 illustrates the hardware contained in the *Main Pod*. The *Power* block distributes power through the system with the corrector's voltage levels and the required capacity. The *Light Control System* sends commands from the *Topside* to the lights. The *Camera Synchronization Generator* supplies both cameras with synchronization signals, an essential component in stereo vision. Lastly, the *Fibre Optic Termination* block connects the fibre optic cables from the cameras to the *Topside* umbilical. This hardware is encapsulated in a waterproof container with connectors to the topside umbilical for both cameras and both lights. Table 5 shows a figure of the developed casing and some specifications.

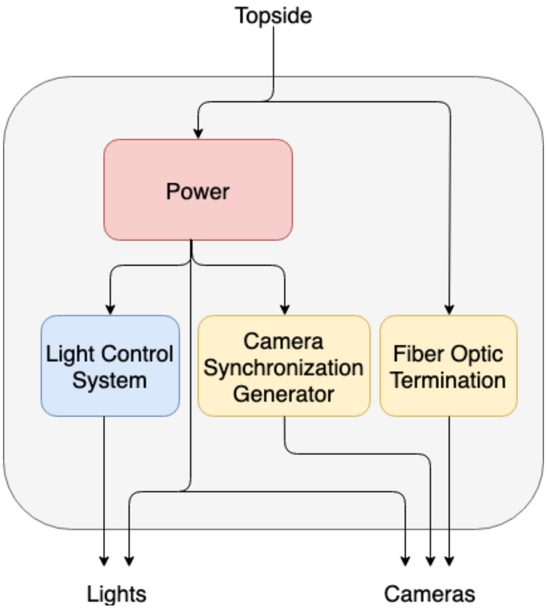


Figure 6 Main Pod.

Table 5 Main Pod specifications

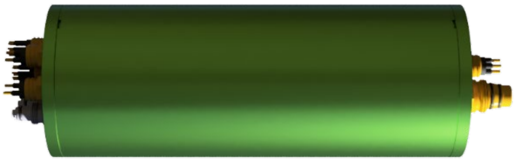
	**Specifications** Length: 320 mm Diameter: 125mm Weight in air: 7.5 kg Weight in water: 2.7 kg

Figure 7 shows the *Main pod* connector interface towards the topside umbilical. This is where the Y-Split is supposed to be connected. The left connector is for the optical fibre and the right connector for copper.

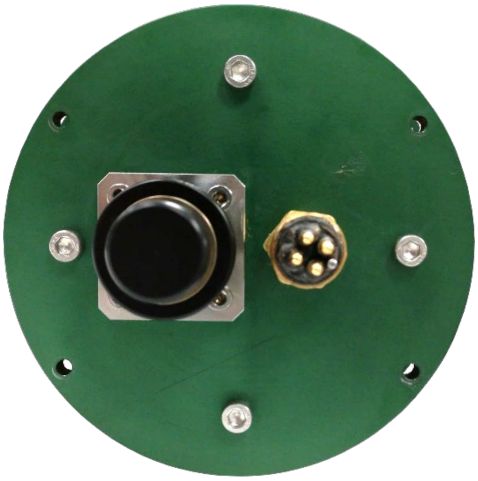


Figure 7 Main pod connector interface towards the topside umbilical.

Figure 8 shows the *Main pod* connector interface towards the cameras and the lights. The lights are connected to the two top connectors. The two connectors at the bottom are the optical fibre connectors for the cameras. Lastly, the two middle connectors are the copper connectors for providing power and sending steering-signals to the cameras.

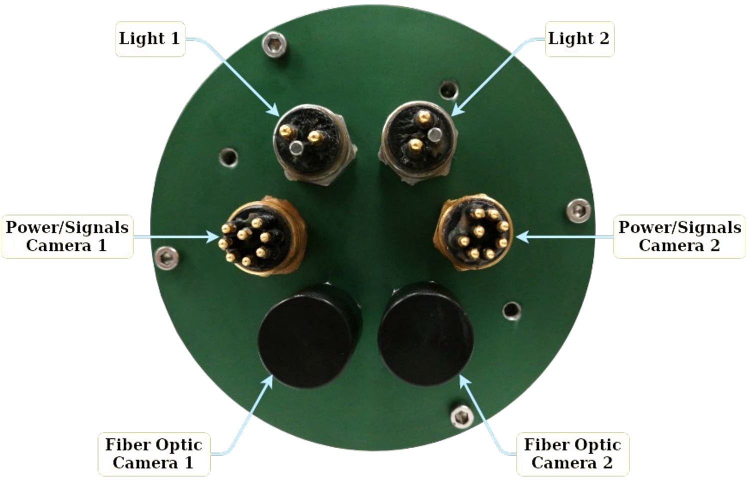


Figure 8 Main pod connector interface towards the cameras and the lights. Connectors are identified by labels.

Camera Housing

The *Camera Housing* is a waterproof encapsulation, which contains a camera and a video signal transmitter. The *Camera Housing* is connected to the *Main Pod* with two cables, a fibre optic cable for the video signals and a copper cable with 8 separate wires for power and synchronisation signals. The *Camera Housing* requires two connectors, both for the fibre cable and the copper cable. Figure 9 shows the connector interface of the stereo camera, consisting of two camera housings mounted together. The fibre connector used is *OptoLink single fibre BCR drybox* from *Macartney* and the copper connector is the *Macartney MCBH8M.*

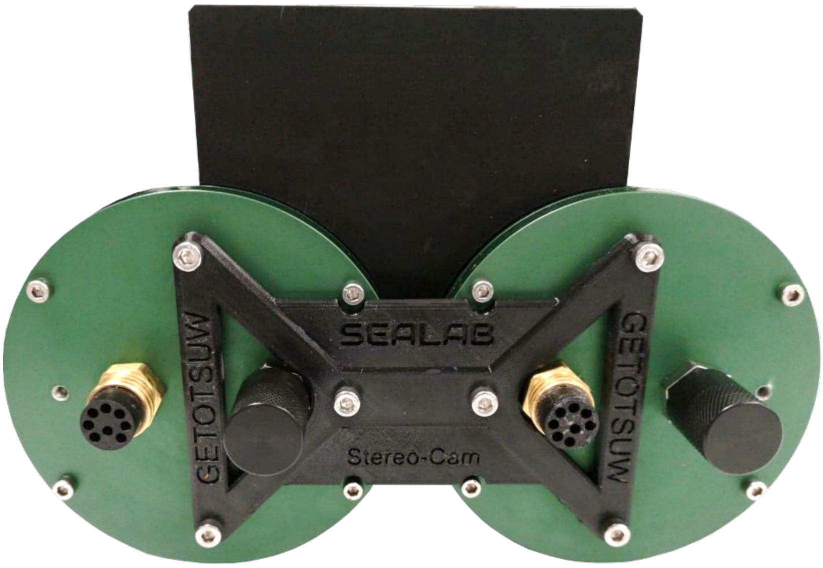


Figure 9 Connector interface for the stereo camera.

Light

The lights consist of a LED-chip encapsulated in a waterproof housing. Two of these are connected to the *Main Pod,* each using a subsea cable from *Macartney* with the connector *MCIL2F*. These LEDs are able to produce a luminous flux of up to 8000 lumen per chip, they are dimmable and easy to integrate using a RS485 interface. Figure 10 shows a single frame from a video recording at night-time within a fish cage. This light allows monitoring of salmon when daylight is absent, thus collecting information about the Salmon at night time, but also in winter time when daylight is limited.

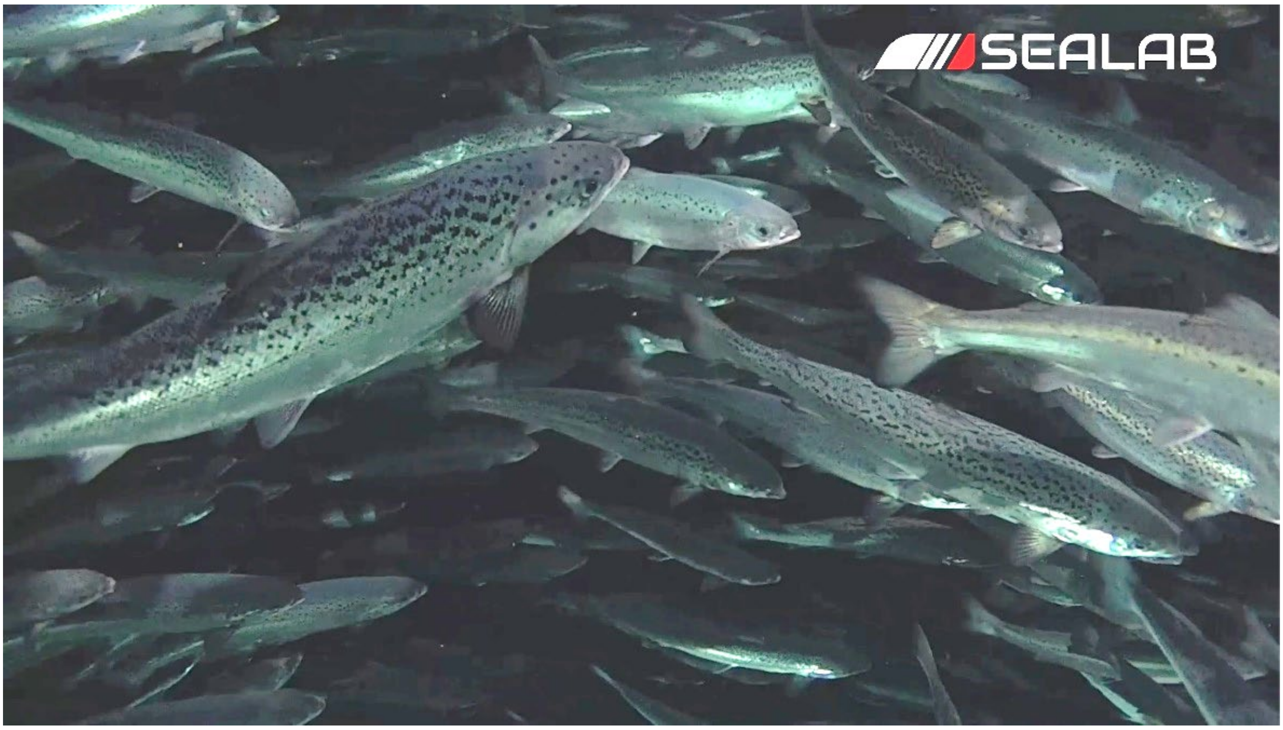


Figure 10 Night-time video recording using SEALAB lights.

Interfacing the camera

To be able to control the camera and lights from the topside, the system provides an interface to the user. Figure 11 illustrates the *Topside Interface*. Four fibre optic cables are available. These are distinguished by colours; green, blue, brown and orange. Green and blue are the stereo video channels. The brown cable is for the light control and the orange cable is currently not in use but available for additional functionality. To fetch frames from the camera channels, an optical fibre to HDMI converter was used. The HDMI outputs from this card is then connected to a grabber card inside a computer. This made the video streams available as devices in /dev/video0 and /dev/video1 at the Linux operating system. Thus the user may access the streams as required by the application. It can be seen from Figure 11 that the computer is connected with an ethernet cable to the *Brown* fibre cable. By connecting to the light control system located in the main pod via SSH, the user can control the lights.

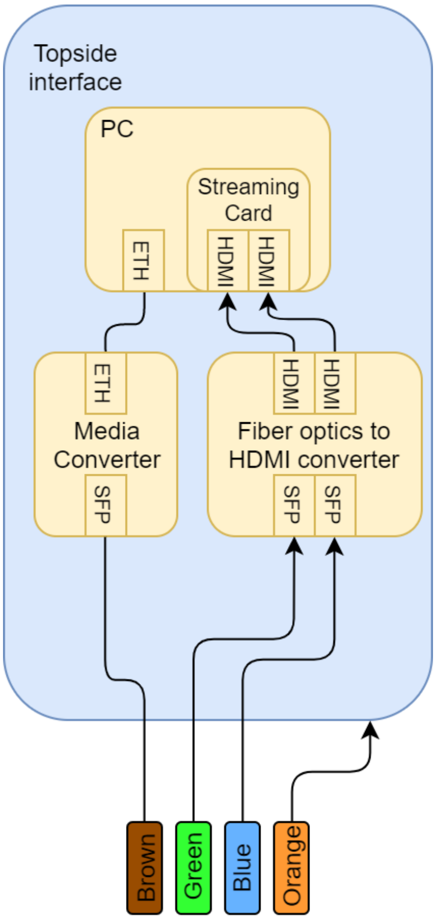


Figure 11 Topside Interface.

2.2 Integration and testing

A test of the cameras was performed to ensure its correct functionality. This included verification that both cameras were synchronised, and to see that the system did not produce any unwanted effects. Figure 12 shows the result from this testing. The two upper images are the left and right camera frames recorded at the same time. The lower images are from the left and right camera at the next frame. It can be seen that the flashlight of the mobile phone turns on when comparing the second to the first frame. Both cameras captured this transition of light, confirming that they are synchronised. No artefacts or unwanted effects were discovered confirming the functionality of the system.

Afterwards, all of the individual components were connected and tested together to confirm correct functionality. Figure 13 shows the stereo camera system integrated on the ROV. During the field tests conducted at the SINTEF ACE facility Rataren on autonomous navigation control concepts, one of the fibre cables between the *Main Pod* and the *Camera* got damaged and malfunctioned. Therefore, it was not possible to do recordings with this setup during the field trials. Since all of the parts are customised and expensive which means long production time and thus full-scale validation of the system had to be postponed. However, to ensure the results of the project, a stereo setup with two Gopro cameras was used during the full-scale trials to obtain the images necessary for the validation of the developed image processing algorithms reported in the following sections.

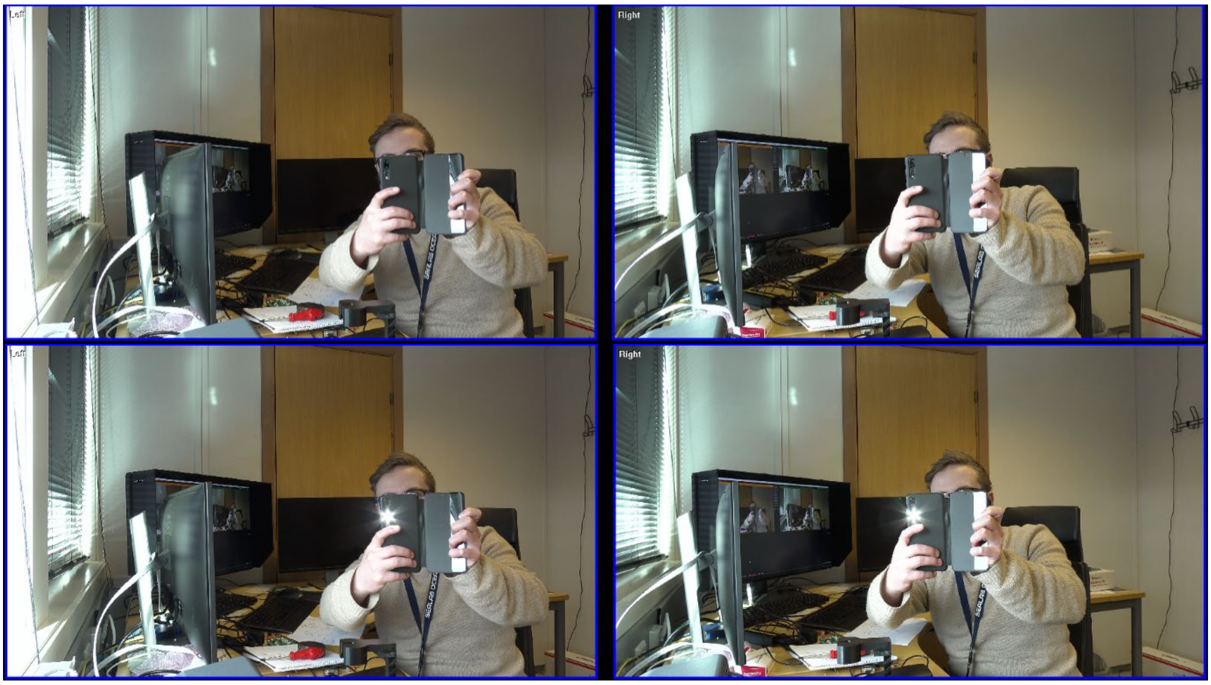


Figure 12 Stereo camera test, showing two subsequent frames recoded by the left and right camera simultaneously.

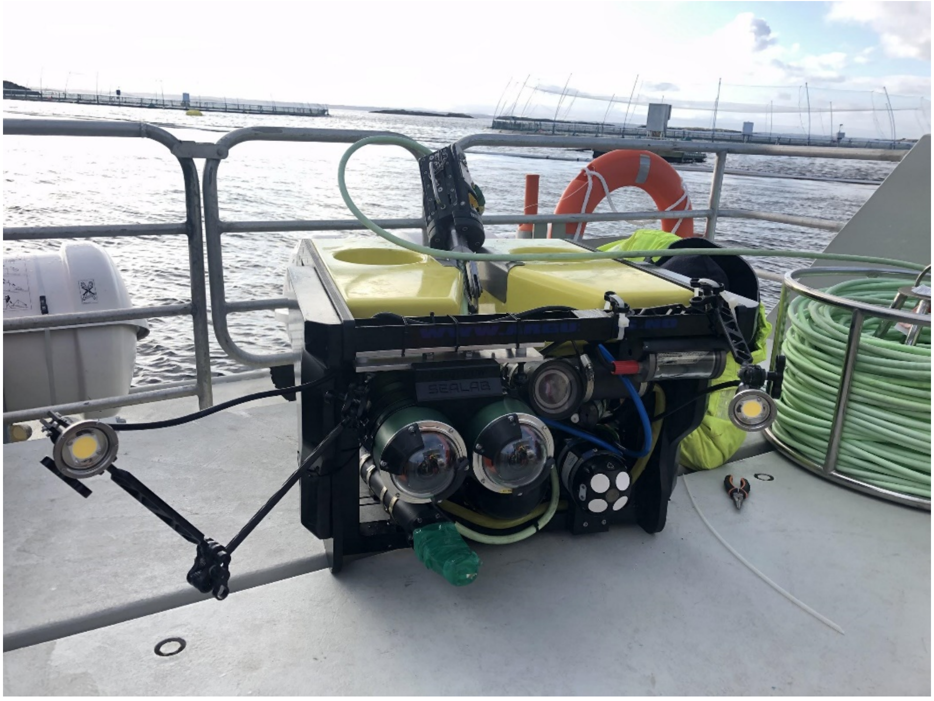


Figure 13 Stereo camera integrated on the ROV.

3 Real-time analysis of data capture quality

In order to develop a vision system that is able to obtain high-quality data, there are R&D challenges related to the data quality analysis. Data quality is here defined as a term, not only related to image quality, but also to the extent that the obtained data is relevant and thus can be used to assess fish, structure and environment conditions. A study has been carried out to define objective criteria for data quality, as well as to develop the algorithms that assess data quality. Note that this is particularly demanding for identification of fish conditions, where the objective criteria, in addition to the image quality of the fish, should ideally also assess the behaviour of the fish, including for example the flight response. For the structure conditions, it is vital to ensure that the images are of sufficient quality to enable, for example, the detection of holes in individual treads in the net.

Therefore, the task for this work package was to explore whether we are able to develop algorithms for analysing the quality of the recorded data from fish cages. This was performed in two stages: 1. the first stage consisted of a brief study to define the quality requirements of the data in relation to monitoring the condition of the fish, inspection of the cage facility and/or environment and 2. the second stage was to implement algorithms which evaluate the data towards the criteria set in stage one. In order to identify the quality requirements for videos for monitoring fish/structure and/or the environment within fish-cages, this study specifically provides insight into what minimal quality requirements are needed to record video-data that can be analysed automatically by classic computer vision algorithms and state-of-the-art machine learning algorithms. Generally, the quality analysis of captured video-data can be divided into two parts. The first part refers to the *technical aspects* of the image quality which depends mainly on the hardware, but also on some fixed camera parameters used during the recording, and the employed compression algorithms when sending the video-stream to any processing unit. The second part involves the *analysis of the quality* of the recorded video material itself. Note that as the interpretation of the content of the videos is beyond the scope of this project, our analysis focused on measures that were able to work on the pixel level of the images.

3.1 Technical image data aspects

The resolution of the provided video-stream is one key-variable to evaluate the capability of the cameras. A higher resolution generally indicates that we can see more detail within an image. Digital cameras often allow to select a specific resolution. Image/Video resolutions that can be considered to be of good quality (from today's perspective) include:

- HD [1280 × 720 progressive scan]
- Full HDi [1920 × 1080 two interlaced fields of 540 lines]
- Full HD [1920 × 1080 progressive scan]

The progressive scan (vs. interlaced), i.e. consecutive image pixel lines being recorded subsequently, has the advantage that the image can be used “as it is” for image processing and analysis. Interlaced recordings are performed by updating only every second line in the video-image at each timestep. This effectively represents a reduction of the resolution in *y* (vertical) direction and requires the images to be deinterlaced before processing. An example is shown in Figure 14.

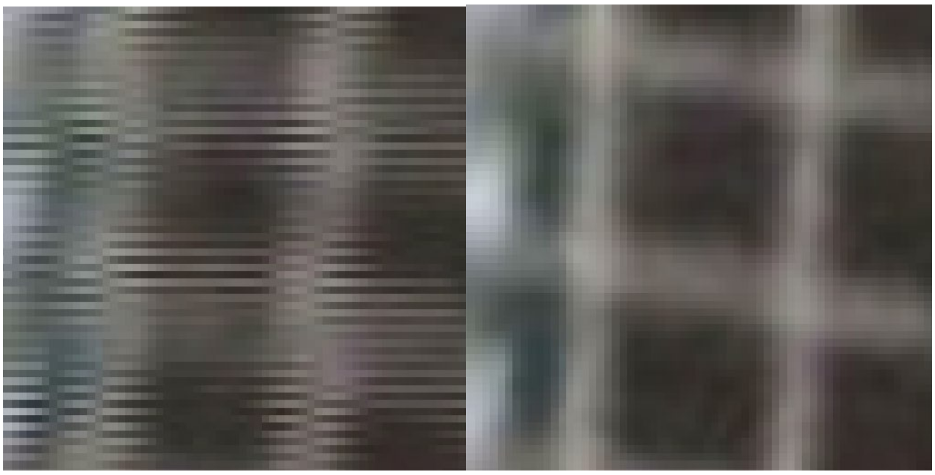


Figure 14 A small part of a net of a fish cage recorded in "interlaced" mode (left image). A deinterlacing is necessary before the image can/should be further processed. Deinterlacing the left image results in the image seen on the right.

Currently, many fish-farming companies still rely on gray-value-video-streams from the fish-cages that have a D1/DV PAL Wide-screen resolution (i.e. 720x576). Such a low resolution combined with an interlaced mode makes the automatic analysis of the data difficult, even if some aspects can be seen by a human. Examples are shown in Figure 15.

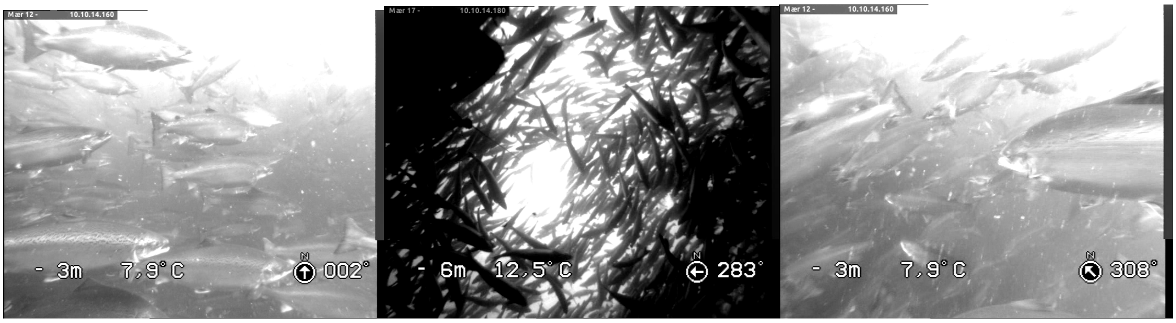


Figure 15 Example images from video-cameras commonly used for cage-observation. These often have a relatively low resolution (i.e 720x576) and a low dynamic range that quickly leads to overexposed areas in the image.

Objects that one wishes to identify should cover a minimal area of about 32x32 to 64x64 pixels in order to enable machine learning approaches to be trained with labelled regions of that size. For underwater fish cage observations, a typical video frame rate is about 25 fps (frames per second), which is sufficient for many applications (higher frame rates are desired when fast motions are to be analysed). A fixed focus defines the distance where the sharpness of an object is optimal. If the autofocus is switched on the intrinsic camera parameters may change. However, standard 3D reconstruction methods require/assume that the intrinsic camera parameters do not change and autofocus is usually avoided in these cases. The Aperture, often represented by *f (e.g., f2.8, f8.0 etc. where larger numbers correspond to smaller aperture openings)*, influences the amount of light that passes through the lens and is received by the image sensor. The aperture size also has an impact on the sharpness-range. Smaller openings lead to a larger range where objects appear sharp in the image. Lower light conditions generally require longer shutter times and lead to observable motion blur in the images. In addition, this is dependent on the sensitivity of the image sensor, with a higher sensitivity increasing the observable noise in the images. For all scenarios considered, including A) the State of the fish (behaviour/welfare) in a fish cage, B) Inspection of structures in cages and/or C) the production environment, one should aim to record images with the highest possible technical image quality. Note also that color cameras can provide additional information that is useful for special tasks (e.g. open wound detection). Size estimation, speed, distance and density related to the cases A-C may require underwater stereo imaging/3D cameras as these allow for metric measurements.

3.2 Common defects in digital video streams

To set criteria for video data recorded in fish cages, a summary of common defects in digital video streams is necessary. Figure 16 shows an overview of different compression artefacts that can be found in the literature. They are separated in two main branches, s*patial artefacts* and *temporal artefacts.* The former describes location-based artefacts while the latter describes time-based artefacts[2]. In this section three common compression artefacts will be presented in detail and with examples. Further, a assessment of the existing results with regard to how image quality affects Deep Neural Network applications will be performed.

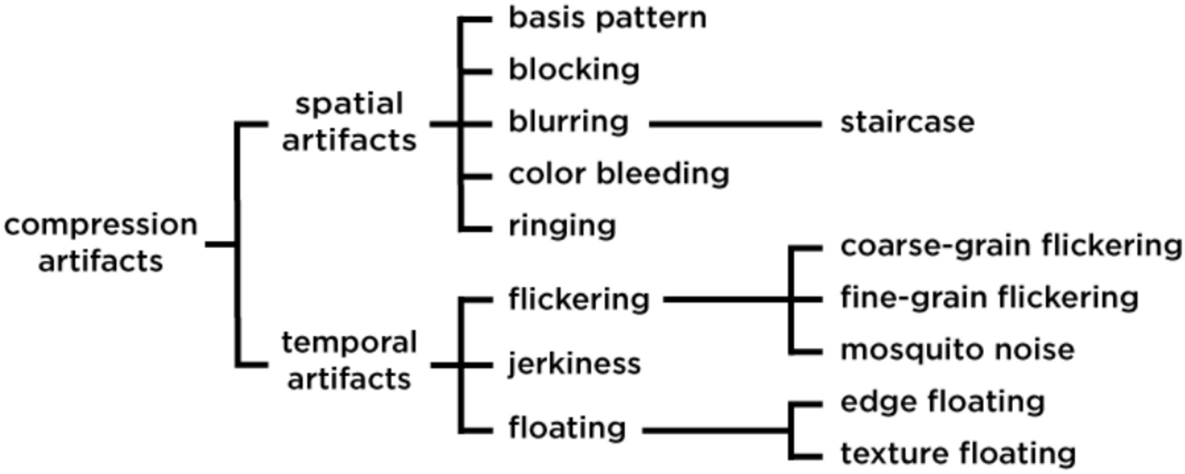


Figure 16 Overview of different compression artefacts.

Blocking

One of the most common video artefacts in real time video streams is blocking. This artefact is recognisable as small squares or blocks in the video image instead of smooth edges and detail. It can be seen in Figure 17 where the image has "square blocks" in the highlighted area within the black box. This can occur in small areas of the frame or be present in the whole frame. Often triggered by fast motion in the frame and when there is a lot of motion in the image sequence. The main reason for these artefacts is the compression of the video stream. Figure 17 shows blocking in the red highlighted squares.

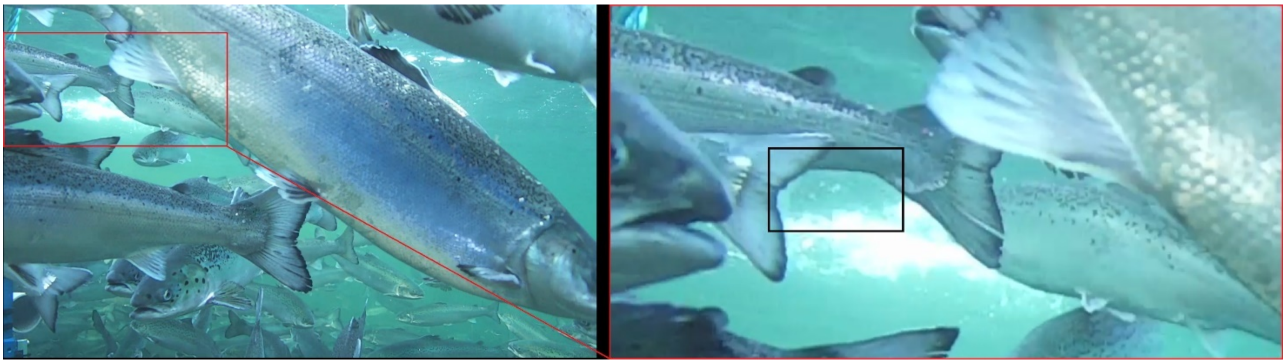


Figure 17 Illustration of blocking indicated by the black rectangle.

Pixelation error

A less common (compared to *blocking)* video artefact are *Pixelation Errors*. They typically occur when data is lost in transmission and the receiving end cannot correctly decode and recreate the correct pixel values.

[2] https://blog.biamp.com/understanding-video-compression-artifacts/

Figure 18 shows an example where a keyframe was lost from the data transmission during the decoding of a video stream. This is recognisable in the subsequent video frames as many areas show the wrong colour/gray values. Small transmission errors usually have a smaller effect but may still result in color-values that are off compared to the surroundings.

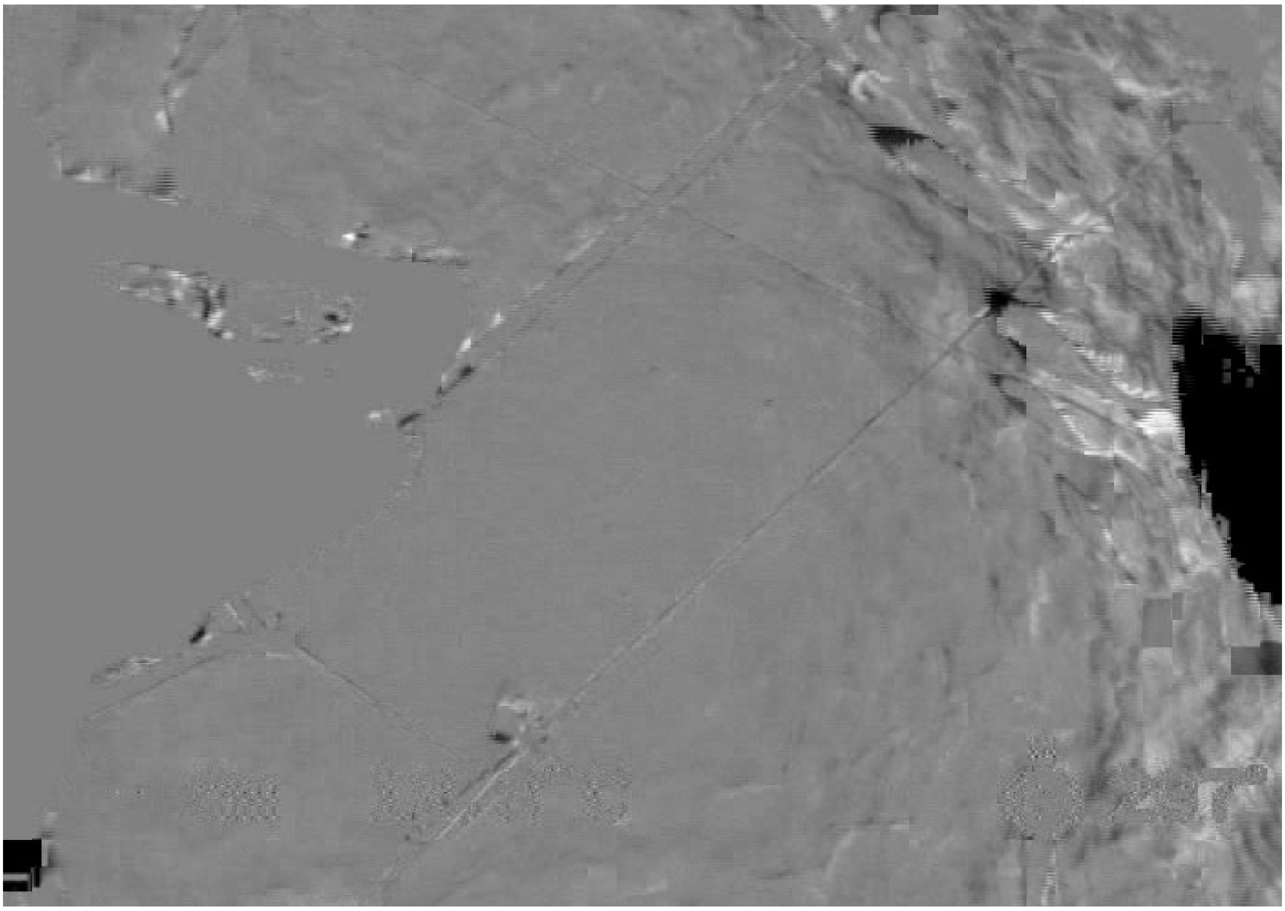


Figure 18 Illustration of a pixelation error. A missing key frame results in pixelation artifacts (snapshot from a feeding camera).

3.3 Analysis of the quality of captured (video)-data

The aim of determining the 'image quality' of videos in the context of aquaculture is to evaluate how suitable a particular image sequence is to provide information for a specific computer vision task. Here we consider application tasks where we wish to obtain the information related to A) the State of the fish (behaviour/welfare) in a cage, B) Inspection of cages and/or C) the production environment. Towards this aim we designed an approach to analyse video data based on their spatial spectra resulting in an algorithm that can distinguish whether one is seeing a net of a fish cage or if the regular net structure is not present. Knowing the camera-parameters and the mesh size of the net, an estimate for the distance can be computed. An example from a test-video recorded with an ROV in a fish cage during the test trials is shown in Figure 19. In addition to the specific net-inspection quality analysis, we also searched for and explored approaches that may serve as a more generic indicator for the quality of recorded video-sequences.

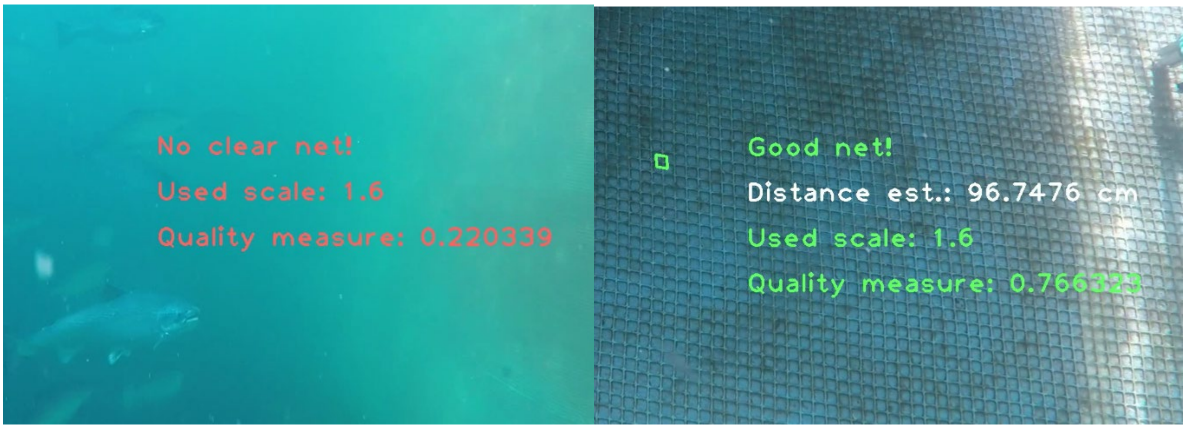

No clear net!
Used scale: 1.6
Quality measure: 0.220339
Good net!
Distance est.: 96.7476 cm
Used scale: 1.6
Quality measure: 0.766323

Figure 19 Analysis of an ROV video providing an indication whether a regular net-structure is visible or not.

3.4 Video sequence experiments

For an initial evaluation of algorithms providing low level quality indicators, we concatenated six image/video-sequences (with increasing quality [subjective opinion]) and evaluated some quality measurement candidate approaches on them. The six film parts (each part is 100 frames long, corresponding to a duration of ~4s) of the test video are illustrated in Figure 20. Note that we ordered the six video-segments based on our subjective opinion with increasing quality (i.e. the first very dark segment, represents the part with the lowest video-quality and the last segment corresponds to the part showing a high-quality underwater recording.)

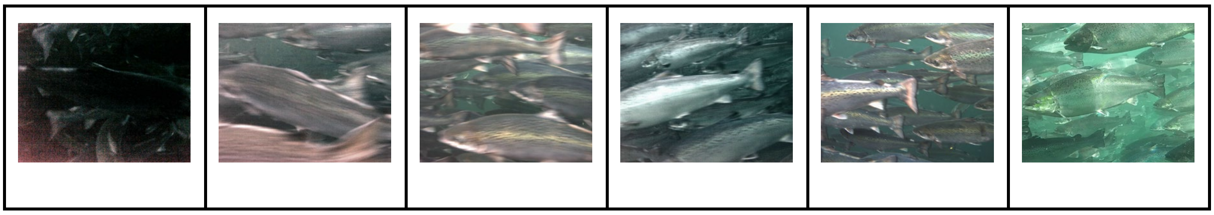


Figure 20 Illustration of six concatenated video-segments ordered according to increasing video quality (subjective opinion). Each film-segment has 100 frames and was evaluated by six video-quality measurements candidates.

The measurement criteria of video-quality we finally tested include the following six approaches:

- Fast Noise Variance estimation (Immerkær96)
- Modified Laplacian (Nayar89)
- Tenengrad: Sum/thresholded gradient measure (Tenenbaum70)
- Variance of Laplacian LAPV (Pech2000)
- Normalized Gray Level Variance (Santos97)

We applied the above measures to the test video sequence. Figure 21 shows the results we obtained for each of the six measurement criteria of video-quality applied to the concatenated video-sequences. Each sub-figure shows the measurement for a single criterion applied to the video-sequence with 600 frames. As we ordered the video sequences according to increasing quality, we searched for a criterion that would reflect this sequence. This means that the measure should increase (or decrease) for each of the subsequent video-segments. However, the experiments did not consistently reflect our subjective ordering of video-quality. Regarding the “expected” behaviour, the Tenengrad approach turned out to be closest to the desired outcome, but likely more advanced machine learning algorithms are necessary for mirroring a human quality assessment of such underwater videos.

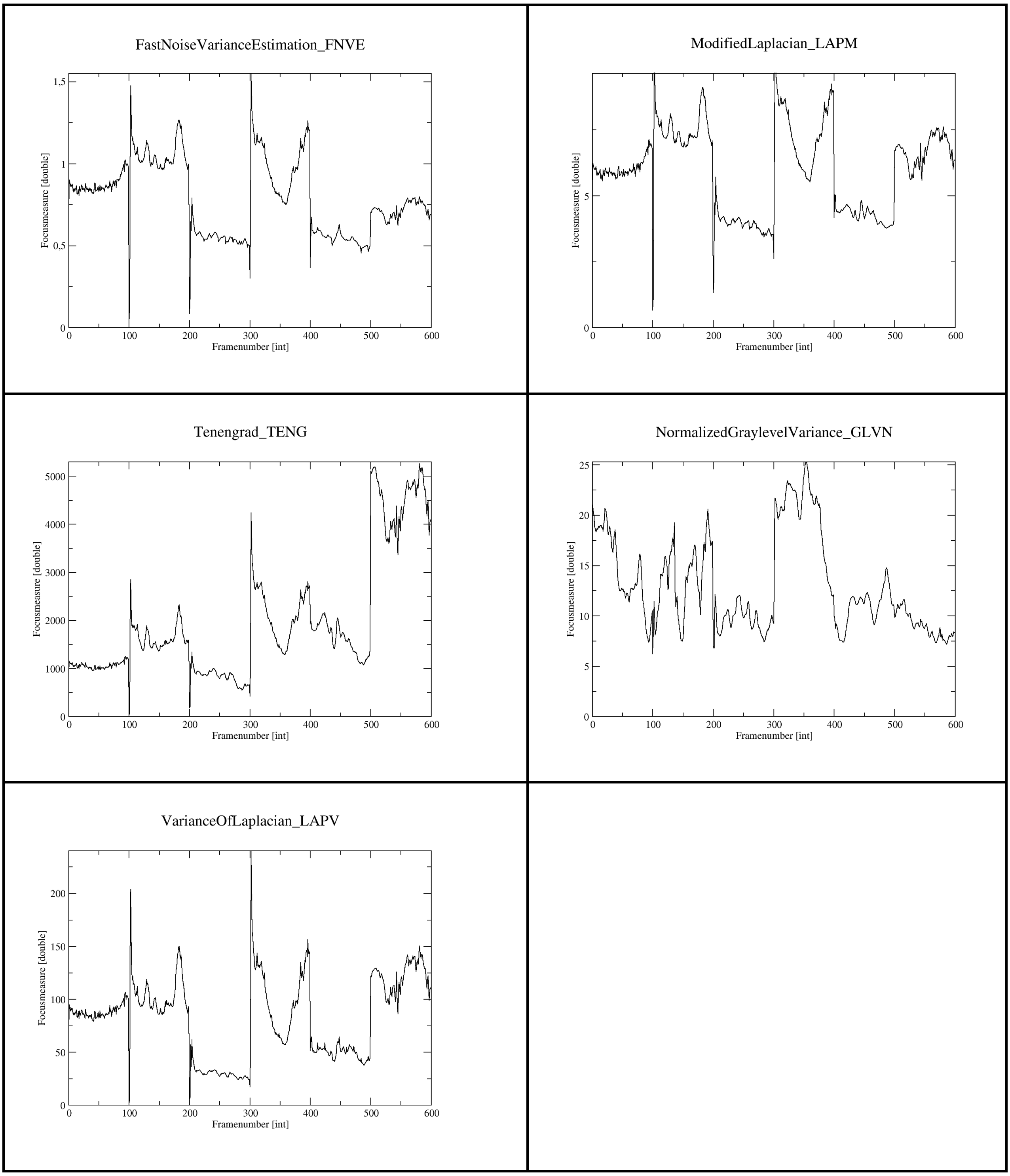


Figure 21 This figure shows the results of the five measurement criteria of video-quality applied to the concatenated video-sequences. None of the measures comply with an "expected" consistent increase (or decrease) of the measurement values.

3.5 Real-time considerations

In this section we briefly consider which of the explored algorithms can be applied in real time.

Algorithms that perform on video-streams with a processing-rate faster or equal to 24 frames per second are considered to run in “real-time”. Note that a video-stream may be downscaled -- still showing the essential details we wish to analyse – before a quality measurement algorithm is applied.

Below we list the processing speed we obtained on a desktop-PC indicating that all are able to work close to real-time (except for two that would need some code optimization):

A. “Fast Noise Variance Estimation:” //Realtime >=24fps
B. "Modified Laplacian: " // ~10fps
C. "Tenengrad: " // ~15FPS
D. "GLVN: " // Realtime >=24fps
E. "LAPV: " // Realtime >=24fps

4 Estimation of distance and orientation from the inspection object

4.1 Decisions on the 3D camera system

At the time of the project-application the most promising candidate for a 3D image acquisition system was the Raytrix camera (PRODUCER) that exploits the plenoptic camera technology. Due to the costs of a single camera-system and additional difficulties of getting depth images of sufficient quality in real fish cage environments, Sealab AS decided to explore other solutions. One option was the use of the ZED-camera (PRODUCER), but drawbacks such as the constraints resulting from the use of an USB 3.0 adaptor and the difficulties to perform the underwater-calibration properly finally resulted in the desicion to build a side-by-side underwater high-end stereo camera from scratch. Unfortunately, this led to a delay of a operational camera-system but finally resulted in likely the best 3D underwater-camera built for use in aquaculture (compare section 2).

4.2 Motivation

The underwater camera system that was developed to obtain high-quality data from fish cages, will be used to measure the distance and the physical dimensions of inspection objects, which is central for several operations in cages. In addition to the high-quality data capture, the camera system will be used as the 'eye' of an underwater vehicle in order to estimate the distance, orientation and relative speed from the inspection object. SEALAB has a vision to help fish farmers see and understand what happens under water. One of the problems fish farmers can have is the escape of fish from the cage due to damage of the net. Therefore, autonomous inspections of the net are a desired feature (Figure 22). One of the first problems to solve towards this aim is to estimate the distance to the net and the relative orientation to the two cameras placed on the Remotely Operated Vehicle (ROV). These inputs are crucial to the control system that enables the ROV to navigate autonomously inside the cage and inspect the whole net area. In a later stage when the steering is working robustly, the goal is to detect holes before the fish can escape, thereby decreasing the total amount of fish escapes. SEALAB is currently developing algorithms that can detect holes in the net.

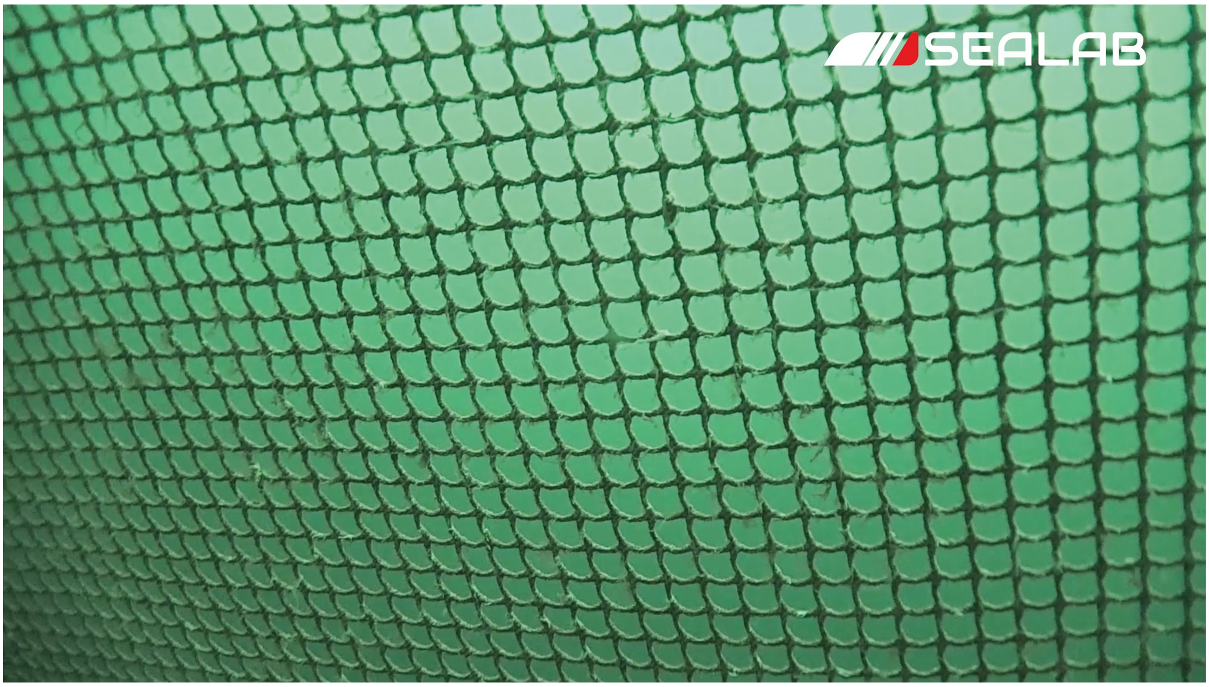


Figure 22 Picture of the net of a fish cage, recorded with a SEALAB AS camera.

4.3 Methods for estimation of the distance from and orientation of an inspection object

Multiple approaches are possible to estimate the distance from the ROV to the fish net. The approaches differ in hardware setup, e.g. single or multiple cameras. Because of the differences in the setup there are different assumptions that drive the use of different algorithms.

Distance estimation using a single camera

Initially, we investigated the potential of using a single camera to estimate the distance to the inspection object. We explored an algorithm based on a Fourier analysis to determine if a regular net structure is present in an image. Some of the strengths and limitations of the use of mono cameras are discussed here. The following two cases can be considered when using a single camera for estimating the distance to the net and its relative orientation to the camera:

1. Knowing the size of the nets mesh openings and the intrinsic camera parameters, it is possible to calculate the real distance to the net. This approach requires some assumptions including that a single mesh opening can be approximated by a flat rectangle/square.
2. By taking two pictures at slightly different time and knowing or tracking corresponding features on the net, it is possible to calculate the distance to the net by so called "structure from motion" algorithms.

Both cases are challenging with regard to a generic application in fish cages. The first one requires the knowledge of the mesh size which may vary from cage to cage. The second approach requires a reliable tracking of feature points which is especially challenging for repeating regular structures like a net. An advantage of using a mono camera is that processing is typically faster as the amount of data from a single camera is lower than the amount of data from two cameras in a stereo setup.

Distance estimation using stereo camera

Within this project we decided to use a stereo camera setup for the distance and orientation estimation of an inspection object (Figure 23). This enables to calculate metric distances from images more easily and more reliably without making many assumptions. The necessary extrinsic parameters can be obtained by calibrating the stereo camera and include the distance between the two cameras (baseline) and the relative orientation of the two cameras (Figure 24).

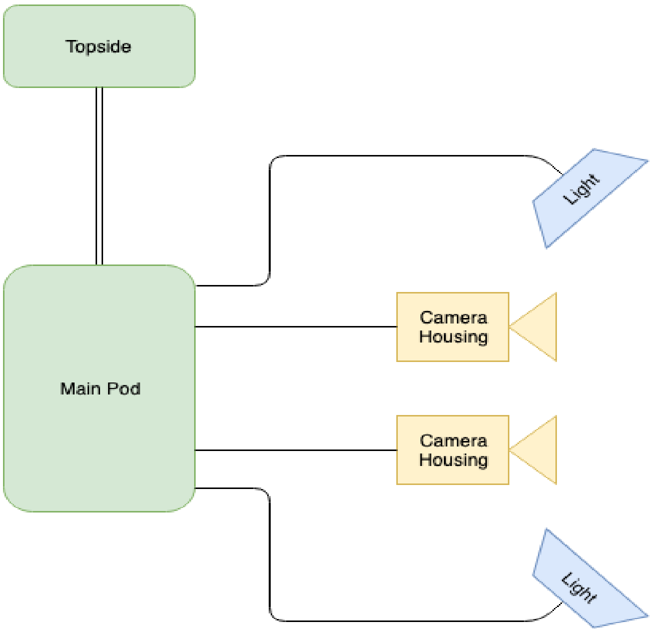


Figure 23 Illustration of the stereo camera architecture setup.

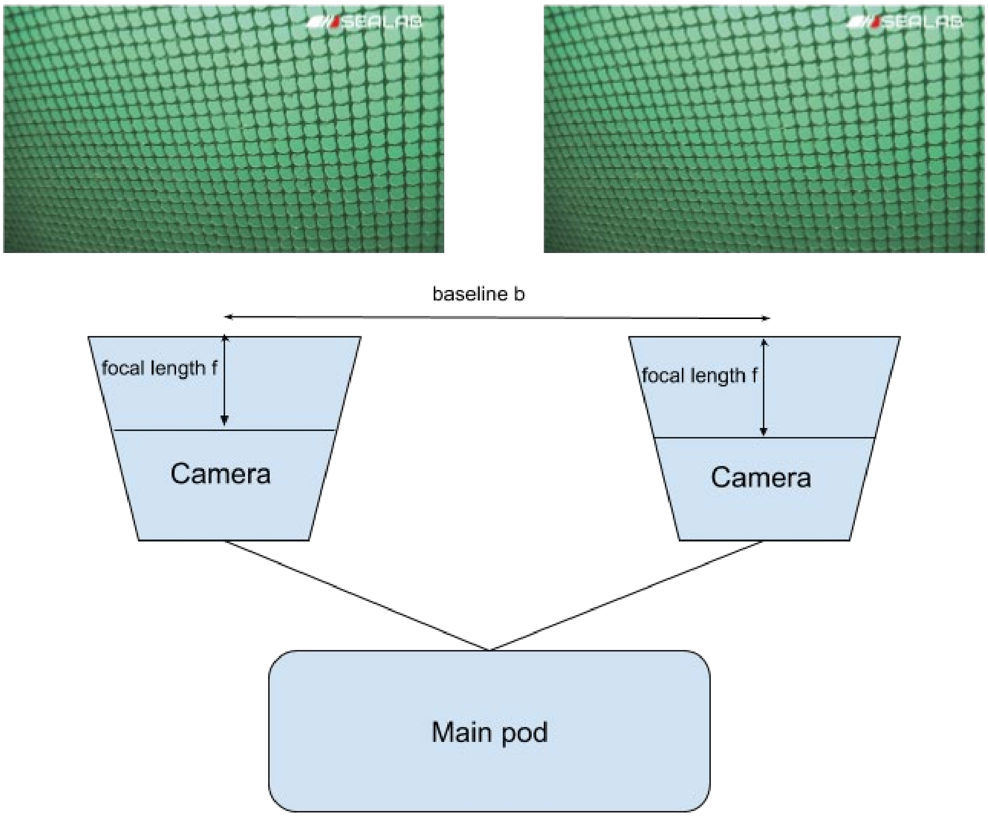


Figure 24 Illustration of the approach adapted to calculate the distance with a stereo setup.

Following, we provide an illustrative example how one can in principle calculate the distance of an object seen in both images of a stereo camera. For the computations we exploited the following variables:
Baseline, b: The distance between the two cameras used for distance calculation.

Pixel size, ps: The size of the individual pixels in an image sensor given in μm.

Note that the two cameras should have the same pixel size in the image sensor. The pixel size typically seen in most cameras is ranging between 6μm - 14μm.

Focal length, f: The distance between lens and image sensor.

***Pixel disparity, ds*:** This refers to the relative pixel difference between the two pictures, creating a map that shows the differences which in turn can be used to calculate the distances.

Depth calculation, *d*: The following equation can be used to calculate the distance to the object using the parameters above:

$$d = \frac{f * b}{ds * ps}$$

A simple example to for distance calculation[3].

Focal length *f* =4.3mm

Baseline *b* =60 mm

Disparity value *ds* =64

Pixel size *ps* = 0.006

$$d = \frac{4.3 * 60}{64 * 0.006} = 671.875mm \approx \boldsymbol{67cm}$$

This is a single calculation related to one point. However, if the calculation is repeated for an area in the two compared images, it will result in different measures for distance *d*.

To calculate a disparity map which shows the difference between the two pictures, it is possible to use a block matching algorithm from OpenCV. OpenCV is a library that can be used either from C++ or Python and has many computer vision algorithms readily available[4]. Note that the calculation of disparity maps generally requires more computational power than the computations with single image cameras, but it also depends on a number of other parameters such as image resolution, frame rate, etc.

4.4 Full-scale validation

As indicated earlier, due the technical issues the 3D vision system described in Section 2 was not functional during the full-scale field trials. However, we attached two GoPro cameras on the ROV as a backup plan. Even though the GoPro cameras show more motion blur than SEALAB's camera, the ad-hoc setup turned out to be sufficient for the aim of depth estimation and was able to show that the developed algorithms verify the underlying concept. The following procedure was adapted to estimate the depth using the recordings from the full-scale trials:

1. Stereo camera calibration
2. Rectification of stereo image exploiting the epipolar Geometry
3. Determine the disparity map and estimate 3D position

Stereo camera calibration

A chessboard of known size can be used to perform the stereo camera calibration under water when the relative position and relative orientation of the two cameras are fixed. In addition, the intrinsic camera calibration parameters are used to correct for image distortions.

After image distortion correction and stereo camera calibration, the baseline and relative orientation are known, and can be used to rectify the stereo image. Note that functions for

1. finding chessboard corner locations
2. single camera calibration
3. stereo camera calibration

[3] Vision-systems Depth Calculation - https://www.vision-systems.com/content/dam/VSD/NextGen/5-3D-2.pdf

[4] OpenCV documentation - https://docs.opencv.org/3.0-beta/index.html

were exploited to obtain the results in this report. The chessboard pattern was placed in front of the stereo camera set to obtain underwater video recordings. Afterwards, the images from these recordings were used to calibrate the 3D vision camera system. Some samples of images can be seen in Figure 25. The right image is suitable for calibration while the left image is less suitable due to motion blur. To filter suitable calibration images, we created a program to extract the frames from the stereo camera recordings while sorting out blurry images. We extracted 60 frame-pairs and a total of 120 images were used for calibration. Afterwards, the 60 images from left camera were used to correct the distortion of the left camera, and the 60 images of the right camera were used to correct the distortion of the right camera. Then, we used the image pairs to perform stereo calibration. For this purpose, one needs to know the size of the chessboard pattern. This also determines the unit of the measured metric distance results. In our case, the square length of our chessboard was 31.1 mm. Note that the quality of the calibration is crucial to the following estimation of depth.

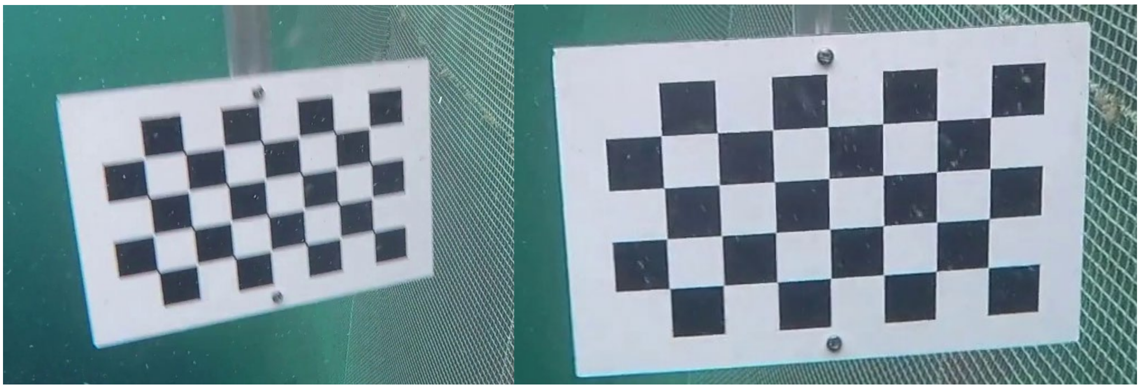


Figure 25 Samples of recordings of the chess board from the GoPro cameras.

Rectification and epipolar line correspondence:

After the rectification of the stereo images, epipolar lines are drawn parallel to the x-axis of the image and corresponding features should lie on the same horizontal line. Figure 26 shows an example of an undistorted and rectified stereo image pair.

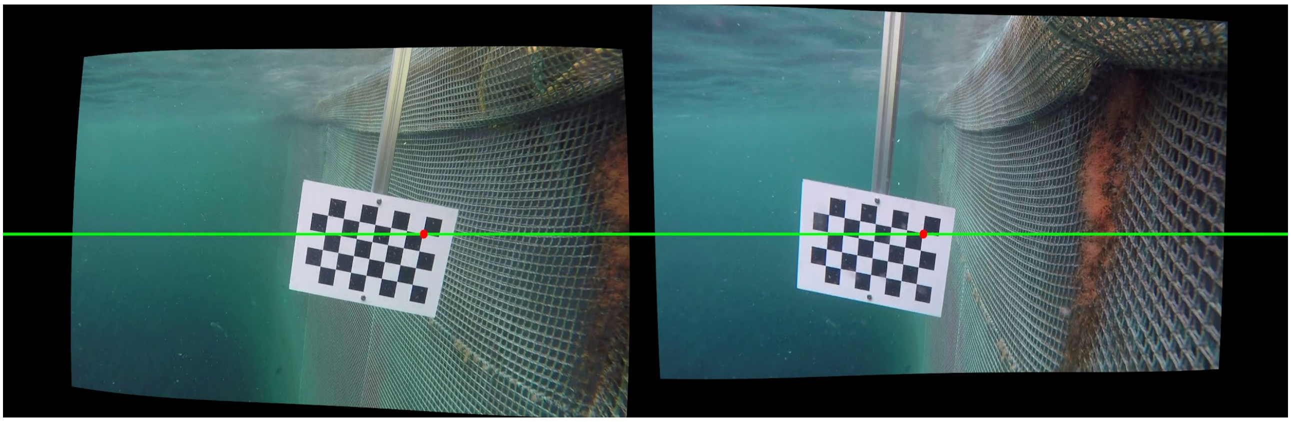


Figure 26 Visualization of the stereo image pair corrected for distortion and rectified. Corresponding features lie on the same horizontal epipolar line (green).

An example for a stereo recording in a fish cage is shown in Figure 27. For any point in the image of the left camera, the corresponding point can be found at the same horizontal axis in the right image and vice versa, except for occlusion. This is guaranteed by the "Epipolar Geometry". The displacement in the horizontal axis needs to be identified to calculate the depth.

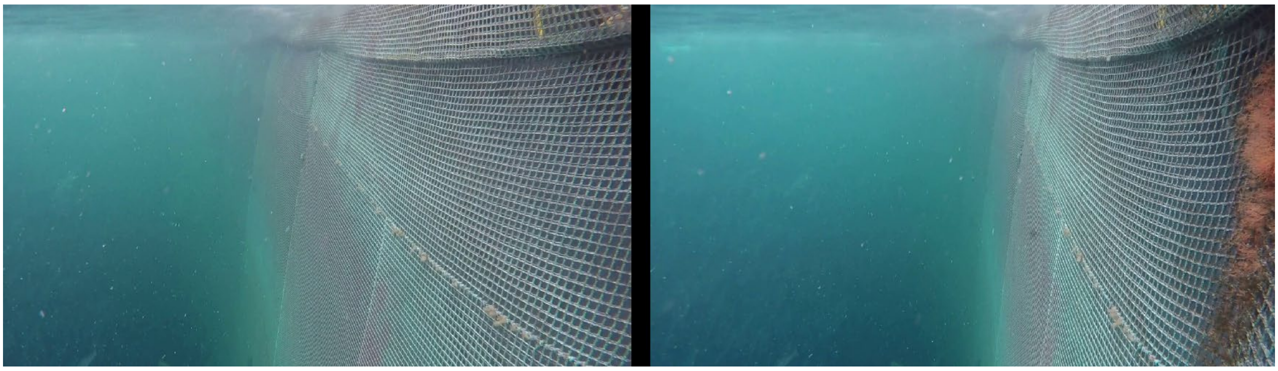


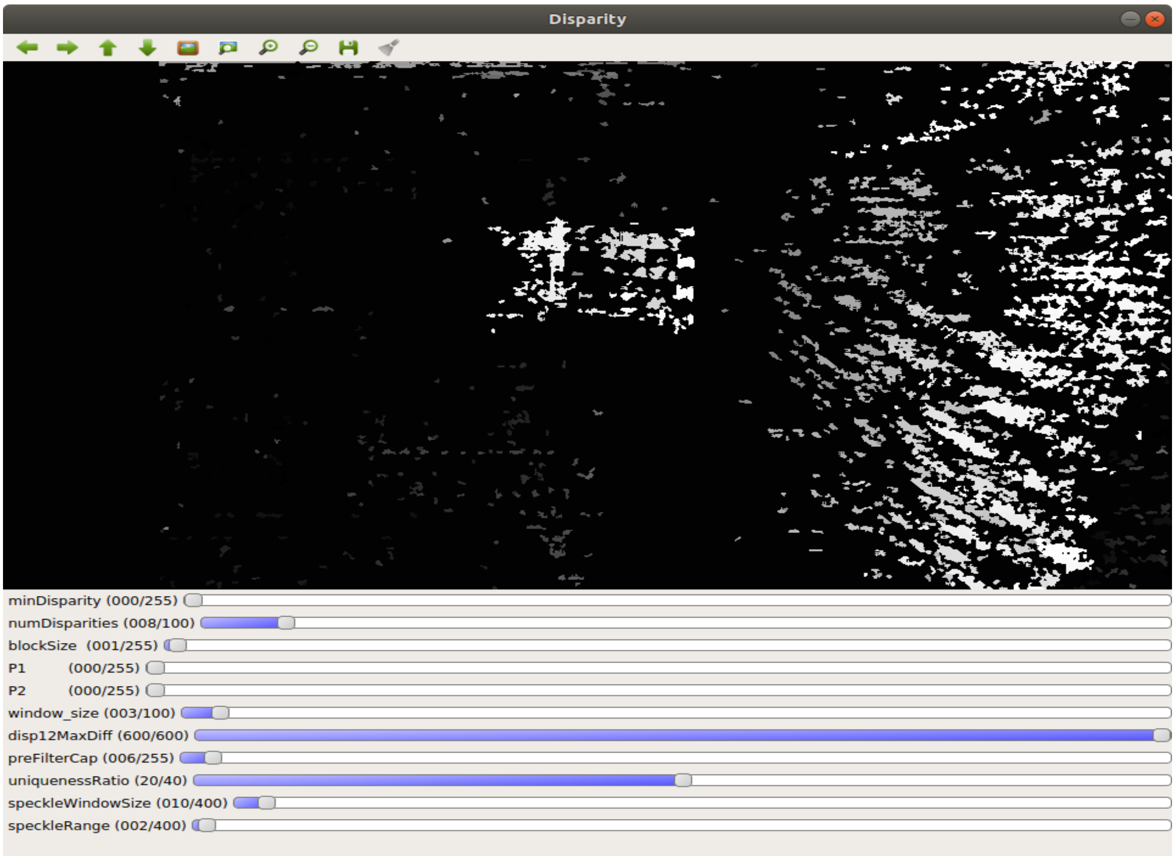
Figure 27. An example of a stereo recording close to the net of a fish cage.

Determining the disparity map

The stereo block matching was used to calculate the disparity map after rectifying the undistorted stereo images. OpenCV provides functions like “stereoBM” and “stereoSGBM” to do this. How well these algorithms perform is decided by visual inspection. Two methods were tested in this project: 1. Block matching and 2. Interactive matching.

The functions for block matching depend on many parameters (compare Figure 28) that all need to be optimized simultaneously. Automation is difficult to obtain and specific sets of parameters may work acceptably in specific lighting conditions. Even after tuning, it is still very hard to obtain a point cloud representing the cage net, as shown in Figure 29. However, the same set of parameters does not work equally well for other cases with different light conditions. In many net-related scenarios the net can appear very regular, which makes it difficult to find the correct correspondences in the images. Assuming the ground truth of the disparity is d_{true}, and the disparity $d_{false} = 2d_{true}$ gave us equally good visual results due to the spatial repeating pattern of the net structure. Subsequently, the distance to the net was estimated to be closer to the camera than it is (factor 1/2). In addition, noise due to ocean particles cause the block matching algorithm to ignore the net. Towards a more automated solution, a module for estimating the distance and orientation of an object was created. As input, this module required 3 corresponding stereo points. The 3D plane that is defined by these 3 points is used to compute the orientation of this plane. In a later step we plan to obtain these 3 points automatically.

To summarize the problems that need solving, the algorithms need to be more robust against lighting changes and noise originating from floating particle and the water turbidity. In addition, the disparity estimation needs to be more consistent when the ambiguity – due to a regular net structure – is high.



minDisparity (000/255)
numDisparities (008/100)
blockSize (001/255)
P1 (000/255)
P2 (000/255)
window_size (003/100)
disp12MaxDiff (600/600)
preFilterCap (006/255)
uniquenessRatio (20/40)
speckleWindowSize (010/400)
speckleRange (002/400)

Figure 28 Example of OpenCV stereo SGBM being used to produce the 3D point cloud.

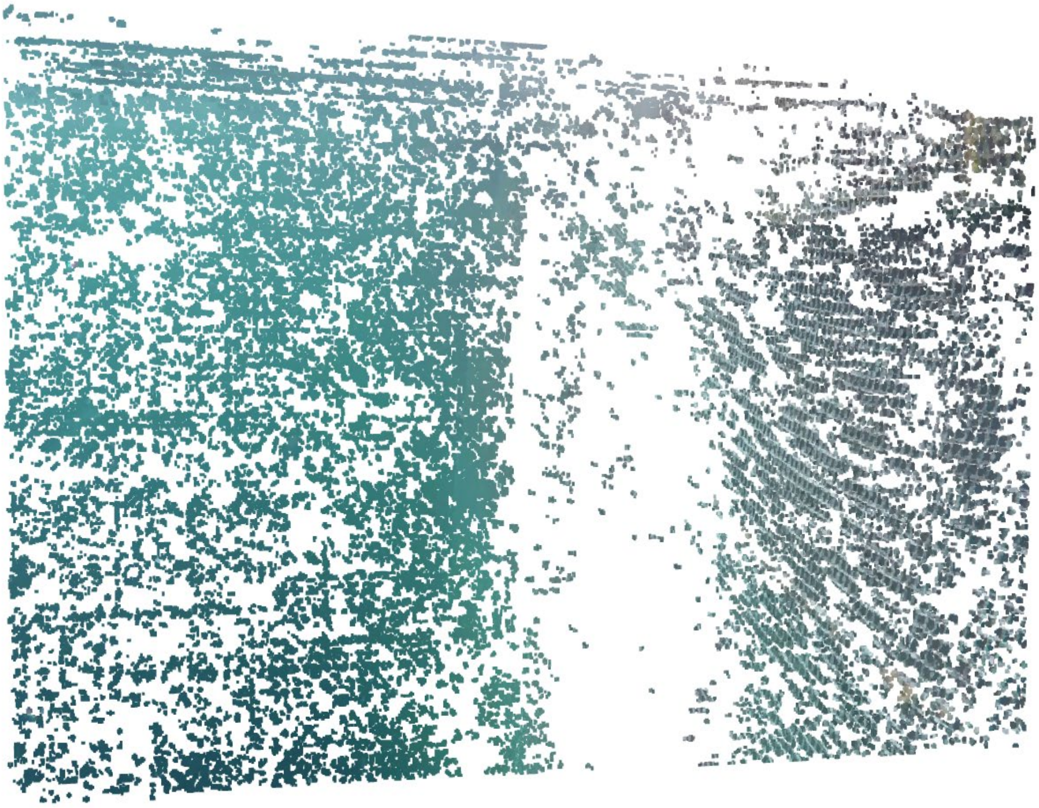


Figure 29 A point cloud of the fishnet using block matching.

The manual labelling allows a distance and orientation estimation of any object in the stereo images as long as we are able to find 3 corresponding features on the object. Note that corresponding features in the rectified stereo images lie on the same horizontal line (i.e. the green line in Figure 26). For an automated approach to determine the distance and orientation of a net, one needs to determine 3 unique features on the net. Fortunately, some net nodes have fouling organisms on them, and such easily identifiable and unique features help to avoid ambiguities resulting from the regularity of the net. Figure 30 demonstrates an example of the interactive interface that lets user label corresponding features in two mouse clicks.

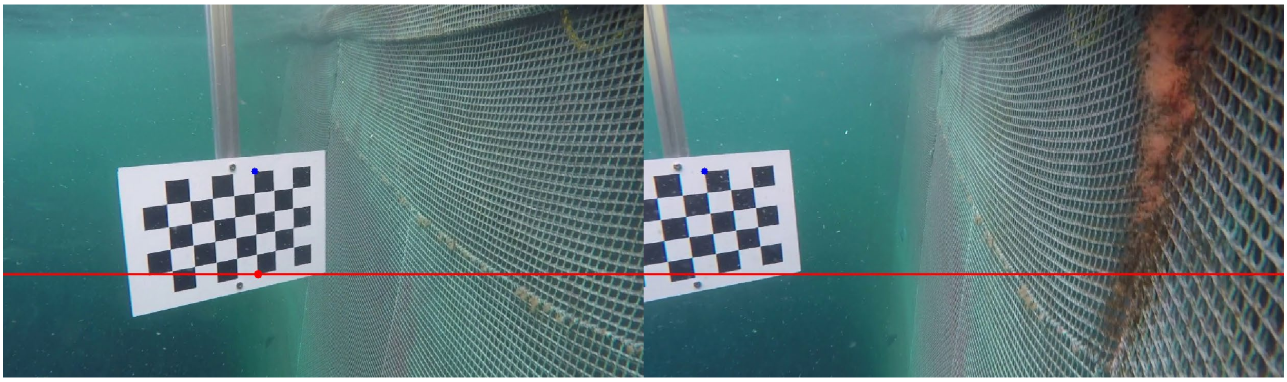


Figure 30 Demonstration of the interactive interface (the interface is waiting for the user to click on the corresponding feature in the right image after marking a feature (red dot) in the left image).

Once the labelling of three points of interest is finalized, we can define their 3D coordinate in units of the checkerboard measurement and estimate the distance and orientation of the triangle build by these three points. Based on optical physics and multiple view geometry we are able to use the following equation

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & -x_c \\ 0 & 1 & 0 & -y_c \\ 0 & 0 & 0 & f \\ 0 & -\frac{1}{T} & 0 & \frac{x_c - x_c'}{T} \end{bmatrix}}_{\text{Q}} \underbrace{\begin{bmatrix} x \\ y \\ d(x,y) \\ 1 \end{bmatrix}}_{\text{A}} = \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix}}_{\text{B}}$$

to calculate the real-world position of any point in the image pair that is visible in both images. Vector A includes $x, y, d(x, y)$, where x, y are the pixel-coordinates and d(x,y) is the disparity. Vector B has X, Y, Z, W parameters with the real-world coordinate of the object being $X/W, Y/W, Z/W$. Note that both A and B are provided in homogeneous coordinates. In the matrix Q, x_c and y_c represent the principal points[5] of the left image in pixel coordinates. f is the focal length, T is the base line and x'_c is the x-coordinate of the principal point in the right image. In our case, x'_c is equal to x_c. From this information, we are able to calculate the 3D real-world position of the object/point. We can obtain the matrix Q during the calibration process and x,y and d(x,y) is determined from the stereo-image pair.

Orientation requires three linear independent points on an object as such three points lie in a plane that can be described by two vectors in R^3 space. When these two vectors are linearly independent, it will span a plane in R^3 space. When we calculate the normal vector of the plane, this corresponds to the orientation (see Figure 31). In Figure 31 3 points of different colour are marked (blue, green and red). The determined 3D position is shown in the form p: $[x, y, z]$ and D is the computed distance (in millimetre). The x and y axes are shown in bright gray. The yellow plane is the triangle plane of BGR points. The orientation of this plane is illustrated by the purple arrow (The displayed number is normalized to 1) in Figure 31 where the 3D

[5] Pinhole camera model https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html#camera-calibration-and-3d-reconstruction

orientation vector is projected onto the x-y plane. Note that the orientation has two solutions and we choose the one which points forward to the camera (e.g. with z component being negative).

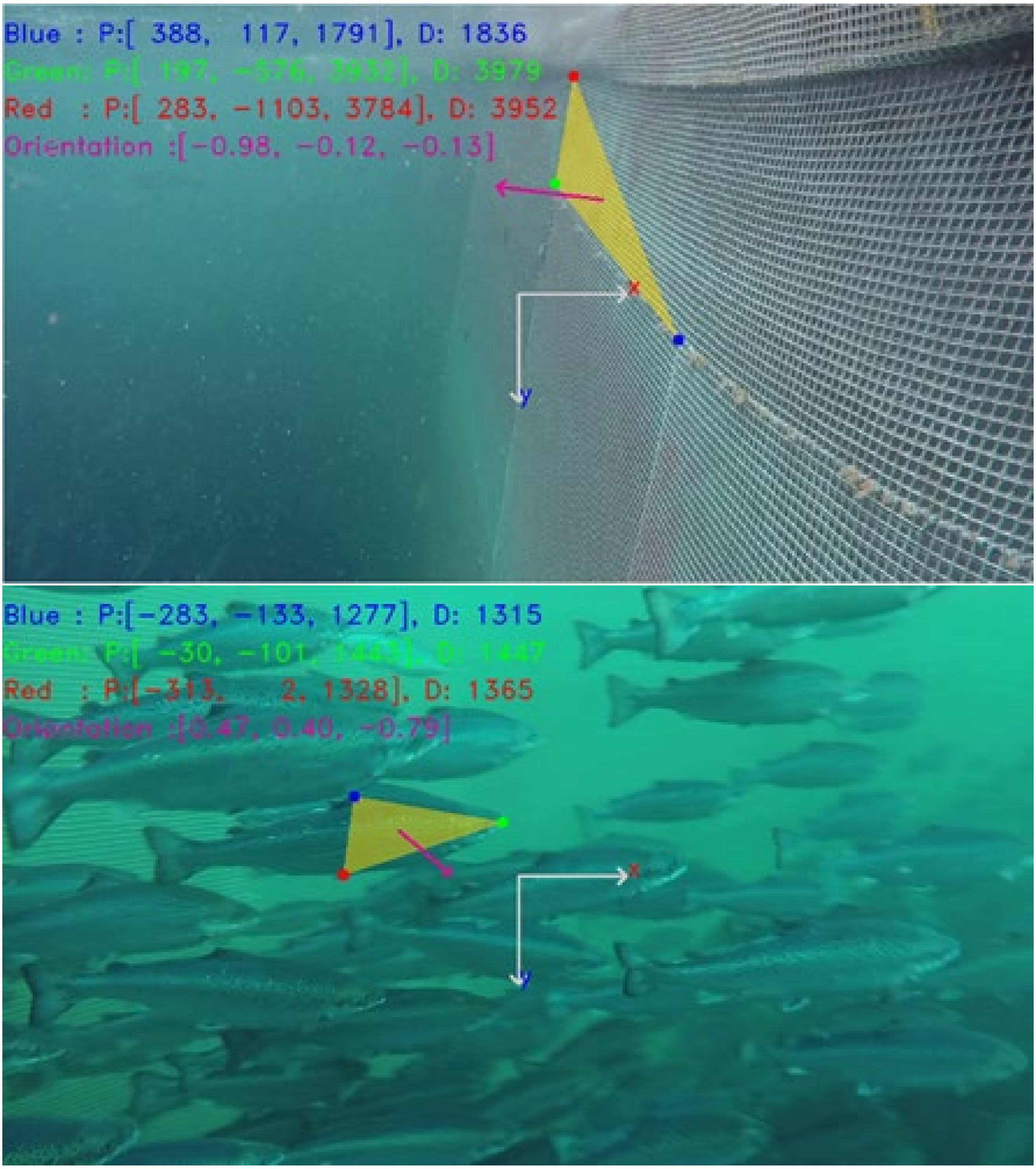

Blue : P:[388, 117, 1791], D: 1836
Green: P:[197, -576, 3932], D: 3979
Red : P:[283, -1103, 3784], D: 3952
Orientation :[-0.98, -0.12, -0.13]
x
y
Blue : P:[-283, -133, 1277], D: 1315
Green: P:[-30, -101, 1443], D: 1447
Red : P:[-313, 2, 1328], D: 1365
Orientation :[0.47, 0.40, -0.79]
x
y

Figure 31 Demonstration of orientation calculation using three points defined on an object.

Validation of the result

Even though the objects position, distance and orientation seem visually correct, we wish to verify these estimations. Based on the ground truth of the calibration board that identify the side length of a single square (e.g. 31.1 mm), we can select a plane (by defining 3 points with color "BGR") on the calibration board of known size and get an estimate of the accuracy of the used approach.

In the following, we selected three examples and computed the corresponding errors to validate the obtained results.

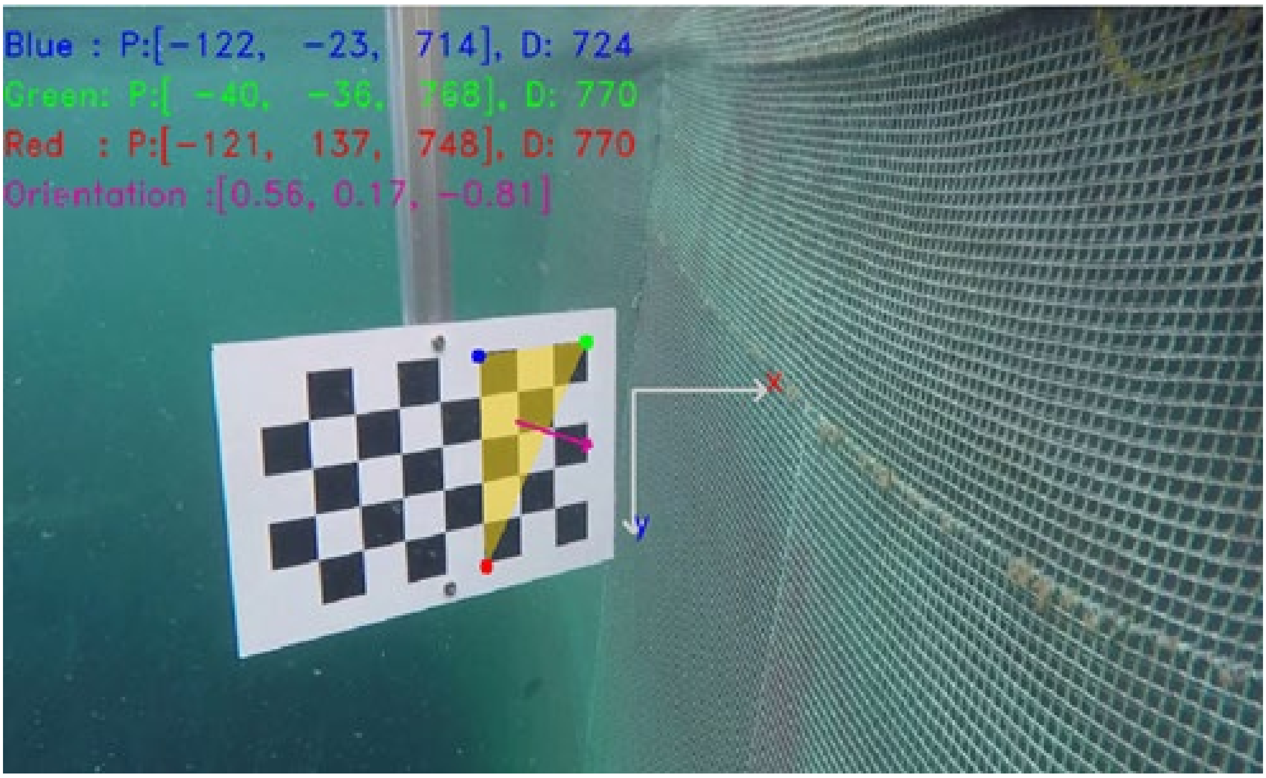
In Figure 32, the 3D vector formed by the blue and the green point is: $[82, -13, 54]$, and the vector formed by the blue to the red point is: $[1, 160, 34]$. The Euclidean norm for these vectors covering 3 and 5 calibration squares are 99 mm and 164 mm, respectively. Based on the square side length of 31.1 mm, one expects a ground truth of 93.3mm and 155.5 mm, respectively. The error in this case is 5.7%.

Two other measurements at longer distances and orientations to the calibration board are shown in Figure 33 and Figure 34 and resulted in increasing errors of 8.5% and 15%, respectively, with the depth estimation becoming more inaccurate with increasing distance. This is also understandable based on fact that the depth d is computed as value proportional to 1/disparity (zero disparity indicates that the point lies at infinity). In particular if we look at the matrix Q, we find that real-world coordinates (x, y, z) are inversely proportional to d(x,y). When an object is far away, a small error in disparity shift will increase the depth error as follows:

$$\Delta z = \sqrt{\left(\frac{-fT}{d^2}\Delta d\right)^2 + \ldots}$$

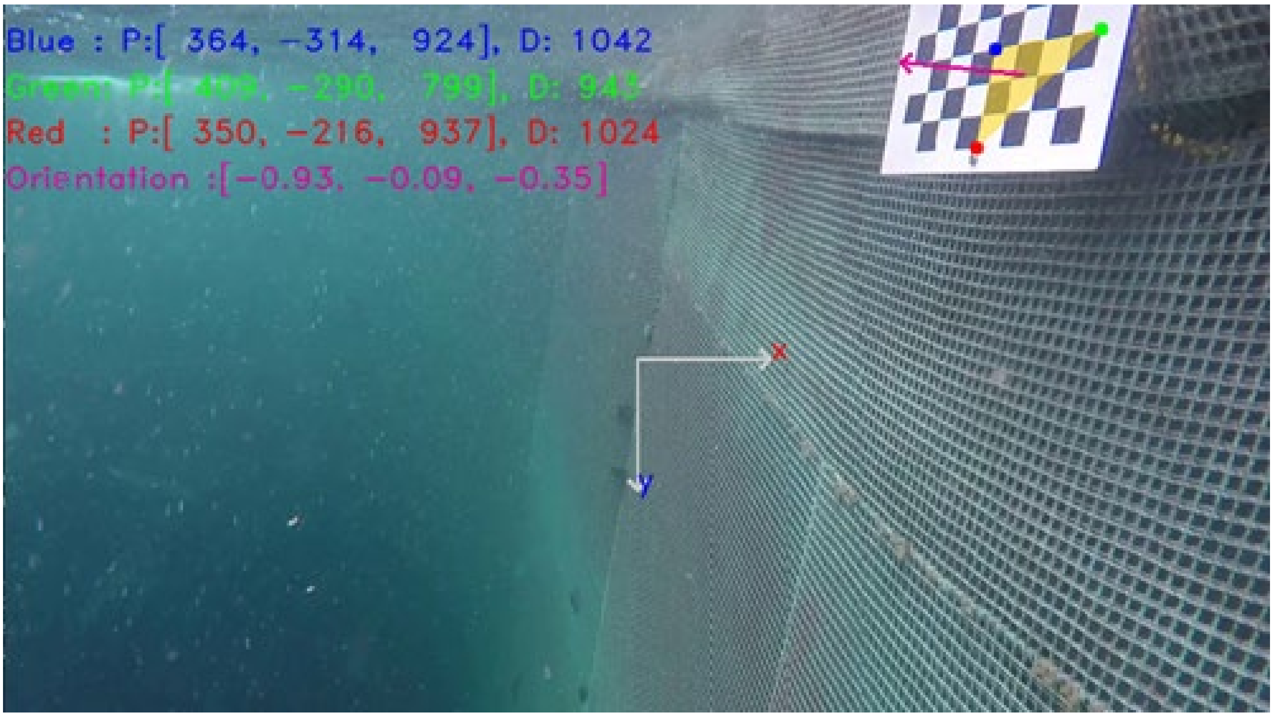
Here, we can see that with the same △d, a smaller d will result in a larger error △z.

Figure 35 illustrates the importance of an accurate stereo calibration. An inaccurate calibration quickly leads to a misalignment, resulting in different y-coordinates for corresponding points expected to lie on the same horizontal epipolar line. However, we note that the observed error is in a range that is still acceptable to guide autonomous underwater operations in a fish cage using robotic vehicles.



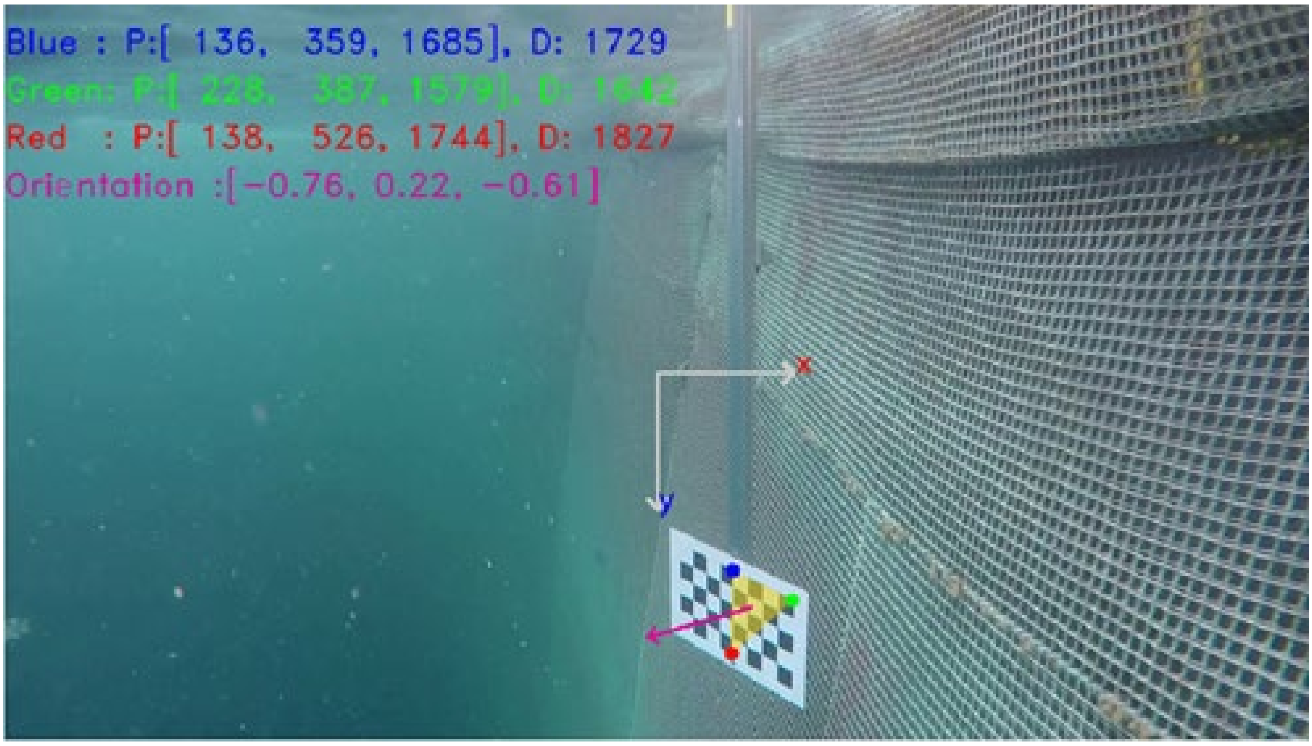
```
Blue : P:[-122,  -23,  714], D: 724
Green: P:[ -40,  -36,  768], D: 770
Red  : P:[-121,  137,  748], D: 770
Orientation :[0.56, 0.17, -0.81]
```

Figure 32 The errors per square length are 1.9 mm, and 1.7 mm (the error is around 6%).



```
Blue : P:[ 364, -314,  924], D: 1042
Green: P:[ 409, -290,  799], D: 943
Red  : P:[ 350, -216,  937], D: 1024
Orientation :[-0.93, -0.09, -0.35]
```

Figure 33 The errors per square length are 2.65 mm, and 1.9 mm (the error is around 8.5%).



```
Blue : P:[ 136,  359, 1685], D: 1729
Green: P:[ 228,  387, 1579], D: 1642
Red  : P:[ 138,  526, 1744], D: 1827
Orientation :[-0.76, 0.22, -0.61]
```

Figure 34 The errors per square length are 4.65 mm, and 4.1 mm (the error is around 15%).

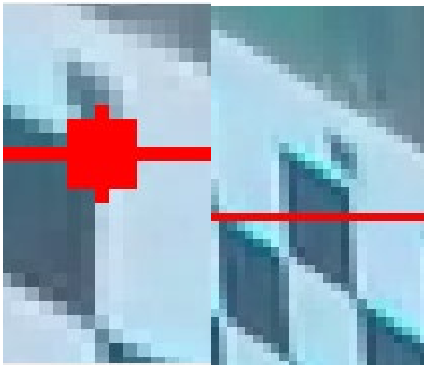


Figure 35 Downward epipolar line shift due to the imperfection in the calibration.

Feature tracking concept

If the labelled points have distinct features and do not move very fast (> 10 pixels per frame) between each frame, one can track them for a couple of frames. Sealab AS explored the performance of the feature tracking function. Figure 36 shows some results for a net image sequence where we were able to track the points on the net. The blue dots are automatically detected feature points, and the red dots in the second image are the feature points closest to our selected points in the first image. If the motion of the net relative to the camera is not large (< than the length of one mesh opening), we can follow the net junctions. In this way, we can track the positions of the triangle points, thus determining the distance and orientation of the triangle.

We have recorded different video sequences in order to test the efficacy of the methods by determining the number of frames we can track the selected points. The first video sample was recorded with 20 frames per second and has a total number of 50 frames (see **video**). For this video we were able to track a group of points reliable for 36 frames [Note, at frame 37, one of the points jumps to the neighbourhood net junction]. We observed that the tracking fails when the motion or the motion blur are too large. In this case, motion blur from the used GoPro cameras is the reason for discontinued tracking. We believe that using the SEALAB camera with very low motion blur would enable much longer tracking. In an additional test with a ZED[6] stereo camera we followed the same procedure as for the GoPros (i.e. calibration is included). In this test we moved the net very slowly, and were able to reduce the motion blur significantly (see **video**). Thus, we were able to reliably track features on the net for approximately 200 frames. We conclude that the reliable tracking of net-features required for an industrial fish cage inspection needs the development of dedicated software modules.

[6] Zed stereo Lab https://www.stereolabs.com/zed/

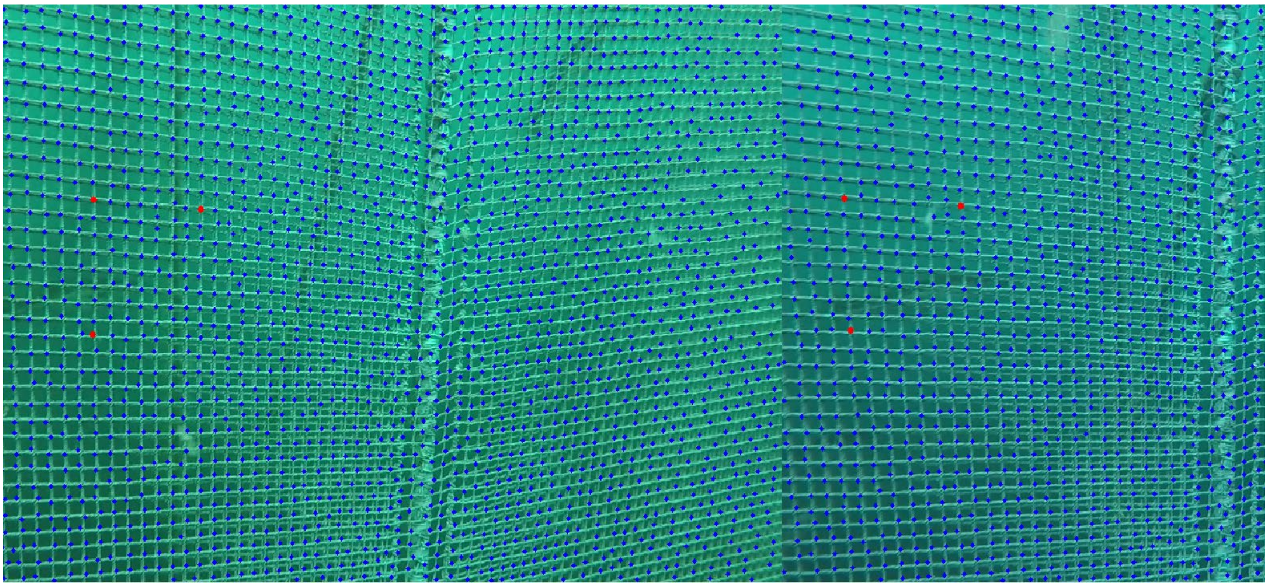
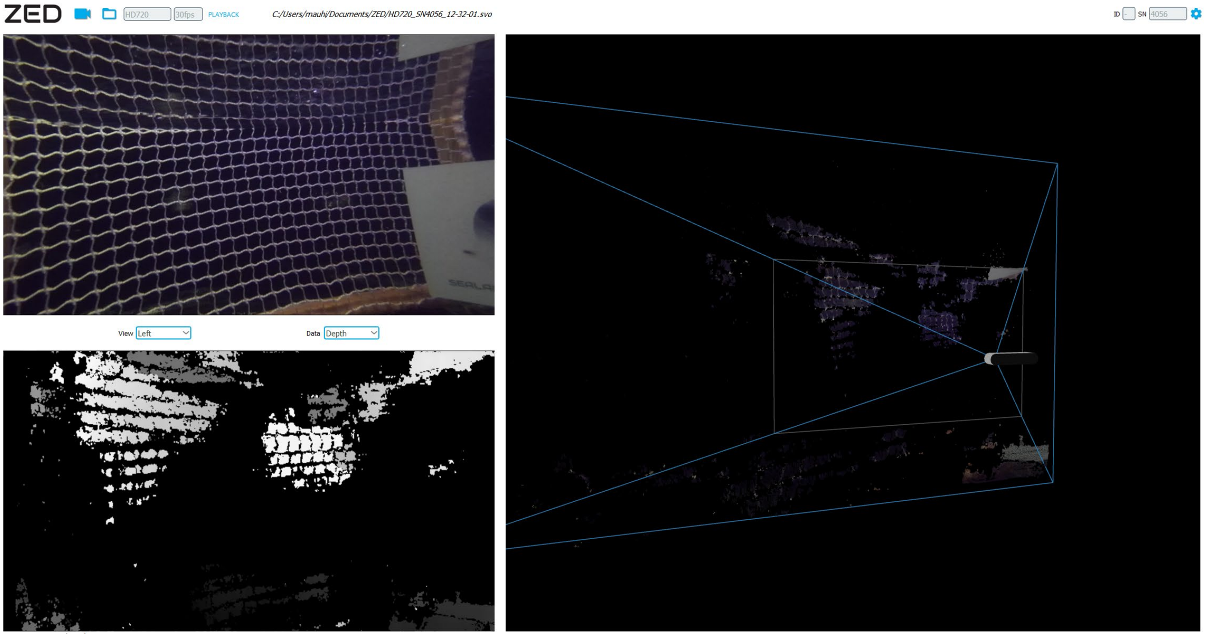


Figure 36 Feature tracking of cage net junctions (GoPro on ROV).

In our experiments we also observed that the orientation vector is 'jiggling' because the features tracking detects the corner of a net junction randomly in the upper left, upper right, lower left and lower right. This happens particularly when the net is too close, and the size of the net junction increases. However, the ZED stereo camera is a consumer stereo-camera that has its own dedicated stereo-matching algorithm and 3D point cloud viewer. Figure 37 shows results obtained with the ZED-camera when it is used for underwater recordings (a cage net placed in a smaller tank). The depth estimation from the ZED is not ideal and subsequently most of 3D structure of the net were missing. It appears that the ZED camera may not be ideal for underwater use and that the parameters are optimized for "in-air" recordings, and that a simple recalibration for underwater-conditions is not possible. And even in areas of the image where distortion appears to be small (in the middle), it is still hard for ZED's matching algorithm to find correct correspondences. In the future, the developed side-by-side stereo system discussed in Section 2 will be tested thoroughly and compared with results obtained using the GoPro-setup and ZED-camera system.



C:/Users/mauhi/Documents/ZED/HD720_SN4056_12-32-01.svo

ID - SN 4056

View Left

Data Depth

Figure 37 Disparity map and 3D point cloud from dedicated software of ZED.

Update: Sealab stereo camera

After repairs, some initial calibration tests could be conducted with the Sealab stereo cameras. The tests showed that a low shutter time, strong LEDs, and a 4K resolution were able to resolve any issues related to motion blur. As a consequence, all images of the chessboard show sharp corners, indicating successful calibration is possible.

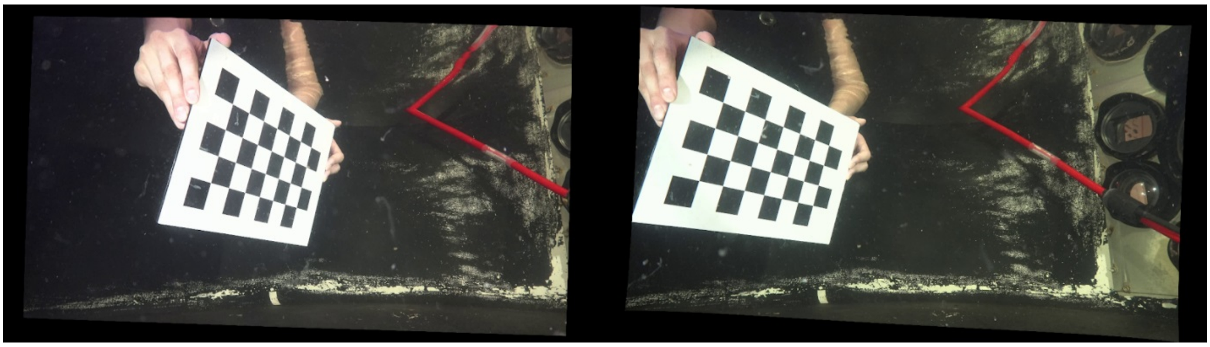
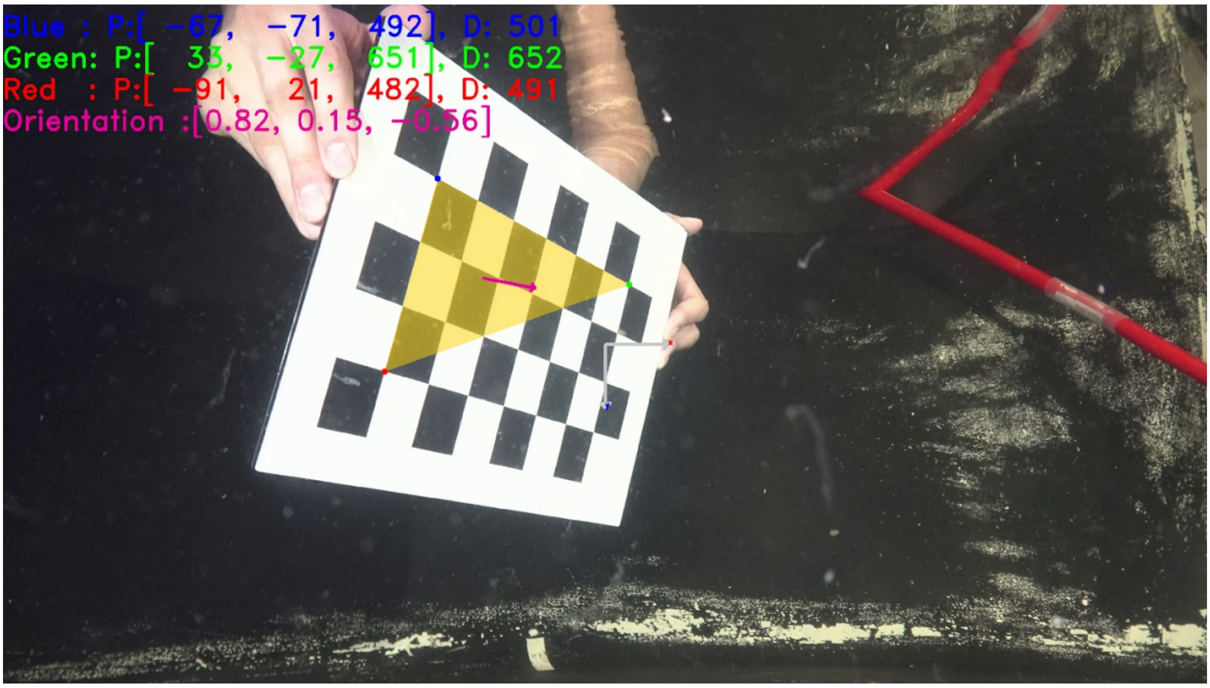


Figure 38 Calibration of the Sealab stereo cameras.

In Figure 38, the stereo image after calibration and correction is shown. However, the chessboard lines were almost straight lines already before correction. This indicates that the Sealab stereo cameras has a very low distortion underwater. Both left and right stereo images have an almost parallel orientation, indicated by the two optical axes being parallel to each other.



```
Blue : P:[ -67,  -71,  492], D: 501
Green: P:[  33,  -27,  651], D: 652
Red  : P:[ -91,   21,  482], D: 491
Orientation :[0.82, 0.15, -0.56]
```

Figure 39 Distance and orientation measured using the Sealab camera.

In Figure 39, the Euclidean norms are 193.5 mm and 95.8 mm, and the ground truths are 186.6 mm and 93.3 mm, respectively. The error per square are 1.15 mm and 0.83 mm. The error is around 3.6%.

Recall the fact that d(x,y) are constraint to integers in this case since the distance between two pixel positions is an integer. With the Sealab stereo camera having a 4K resolution, this will expand d(x,y) domain and consequently expand the range of the z- axis. This will decrease the error (and higher z-resolution) when estimating an object that is far away.

Discussion

In this section, the progress of the developed algorithms for distance and orientation calculation using a stereo setup was reported. Initially, we investigated the use of a stereo block matching algorithm to compute the disparity map that could then be used to generate a 3D point cloud. However, the method proved not to be suitable for the considered underwater environments and will need to be adapted significantly to provide meaningful result. Furthermore, it was difficult to tune all relevant parameters using the stereo block matching algorithm, even for just a single frame. Therefore, we decided to postpone the automatic selection of features on a particular object (e.g., the net) to future work. However, we successfully implemented a module for the distance and orientation computation with an interactive matching as input. This gives accurate result and it is mathematically rigorous. Therefore, we believe that we can estimate the distance and the orientation of any object given we are able to determine corresponding features in both images of the stereo camera. In frames containing nets, repetitive regular patterns are a problem for an automated net feature matching approach. This challenge may be overcome by selecting unique points such as biofouling organisms growing on the net or repaired net features causing irregularities to use as reference points.

Considering this challenge, the use of a laser could be a beneficial solution as it can produce a unique, recognizable point in the image. This will enable the algorithm to find and track correct correspondences also in areas with a very regular net structure and few unique features.

5 Results from five master theses related to CageReporter

Towards Underwater Biomass Estimation using Plenoptic Technology (Malin Kildal, June 2017: Supervisor: Annette Stahl, NTNU, Department of Engineering Cybernetics (ITK))

This thesis investigated the capabilities of 3D plenoptic camera technology to determine whether it can provide decent depth information of objects underwater. As there was no documentation of this technology working in underwater conditions, the underwater calibration procedure and metric measurements were performed. The plenoptic camera technology has been developed as a tool for 3D monitoring in stable and still environments and this thesis explored, by analysing the calibration process and by verifying measured depth points from the determined depth map, if the technology has the potential to be used in an ocean fish farm for measuring the biomass of several hundred thousand Atlantic Salmon.

Results from this thesis show that this technology must be further developed and tested before a complete biomass estimation system can be build, but the results also indicate that this technology indeed has potential for biomass estimation in fish farms. Figure 40 shows the Raytrix camera attached to an aquapod test rig. The best choice for an underwater housing for this system is a flat port. To obtain good results in applied underwater conditions, the fish should be close to the camera as fish farms produce a lot of noise in form of many particles in the water from food and excrement, which degrade the quality of the recorded depth map. A normal field-of-view lens is preferable, even if a narrow field-of-view lens provided a more accurate depth-map.

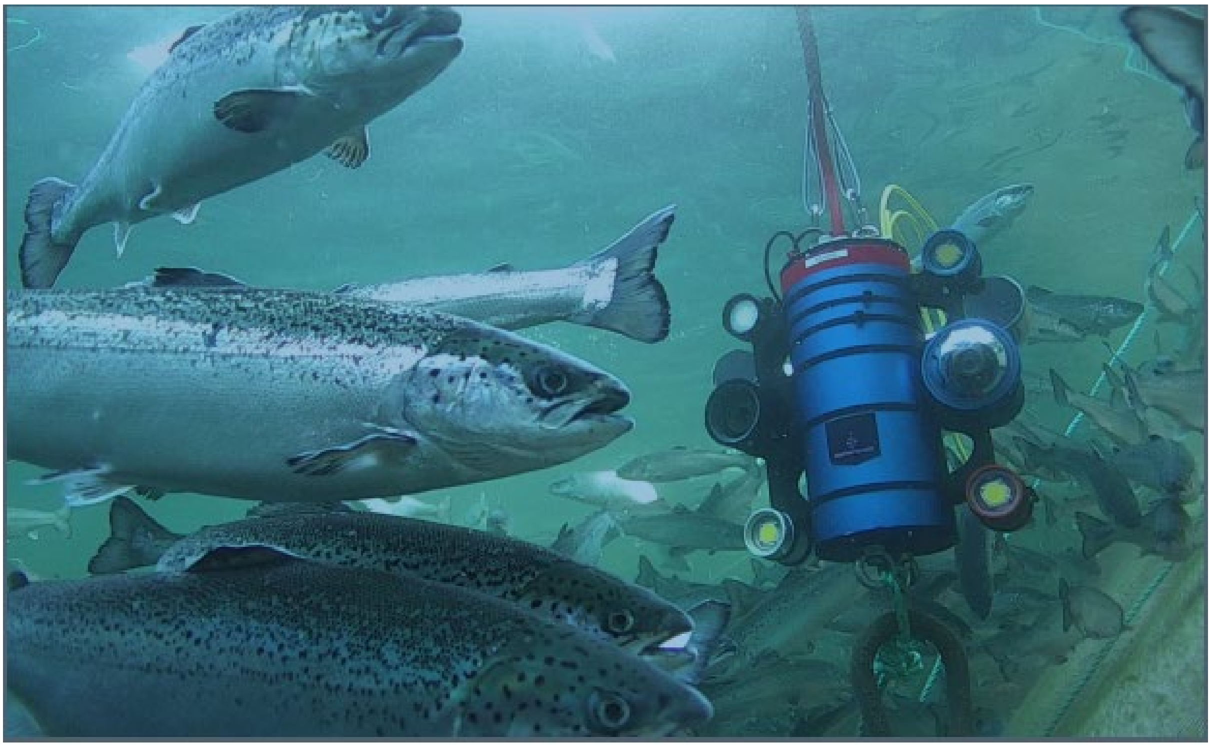


Figure 40 The SEALAB Aquapod with the Raytrix R42 camera attached at the Kåholmen test facility on Hitra. Photo: SEALAB AS.

Stabilization of an underwater camera (Thomas Norum Ur, June 2017: Supervisor: Annette Stahl, NTNU, Department of Engineering Cybernetics (ITK), Co-Supervisors: Per Rundtop and Christian Schellewald, SINTEF Ocean)

This thesis describes the development, implementation and testing of a full-scale underwater camera system for surveillance purposes in aquaculture. The mechanical development was carried out using Solidworks, and the software implementation was based on ROS (Robotic Operating System), in which several open source libraries have been incorporated. A mathematical model of the camera system has been derived as well as a simulation tool in Matlab for simulation. Suspended from a single rope, the camera system is equipped with a water jet propulsion system that allows the yaw (heading) to be controlled by the use of a PID controller (Figure 41). A gimbal inspired mechanism enables control of the camera pitch (tilt). Experiments at a full-scale fish farm facility yielded promising results for the yaw-control, whereas the pitch control needs to be further developed. The work presented in this thesis has been carried out during the spring of 2017 in collaboration with Sealab Ocean Group and SINTEF Ocean.

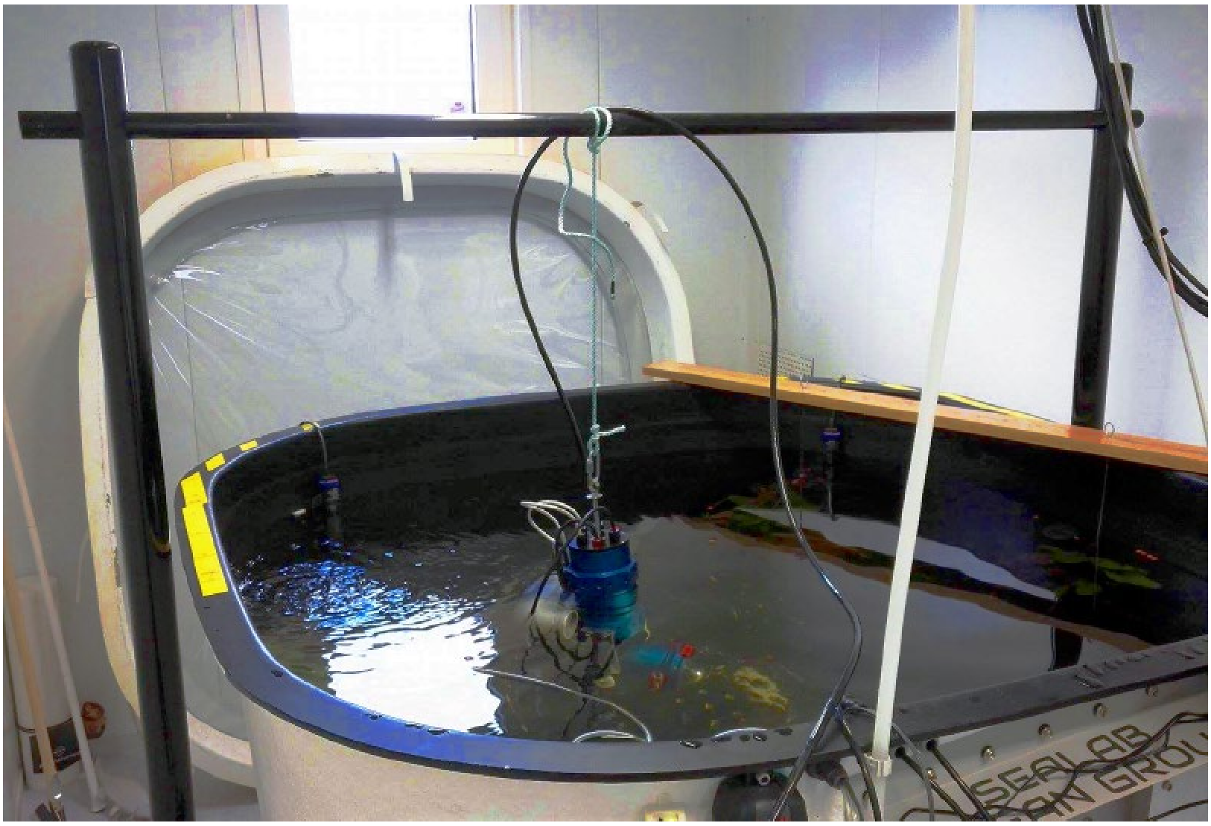


Figure 41 The developed stabilization rig with a camera attached in the SEALAB wet lab. Photo: SEALAB AS.

Classification of fish body parts in an underwater environment (Thorbjørn Sømod, July 2017: Supervisor: Annette Stahl, NTNU, Department of Engineering Cybernetics (ITK), Co-Supervisors: Per Rundtop and Christian Schellewald, SINTEF Ocean)

This master thesis investigated a possible approach to recognizing fish parts in a video stream from a camera system situated in an underwater environment. This task can be seen as the first part of a three-step scheme for implementing an automatic system for fish health assessment in the fish farming industry. This thesis describes the work done in setting up an interface to an IP camera that is situated in an underwater environment, collecting and labelling image material from the camera system for training and testing object classifiers, and training the object classifiers for multi-class object recognition based on image descriptors suitable for an underwater environment. Finally, a complete object recognition framework was implemented and performance tests were conducted based on the pre-trained classifiers. The results are analysed in Figure 42. The results of this thesis show that it is possible to create a system that is able to perform this classification by relying on Support Vector Machine (SVM) classifiers based on adaptations of the Local Binary Pattern (LBP) image patch descriptor. By using a linear SVM classifier, good results are achieved. Surprisingly, the non-linear SVM classifiers relying on the RBF kernel achieve much lower performance than most of the linear SVM classifiers.

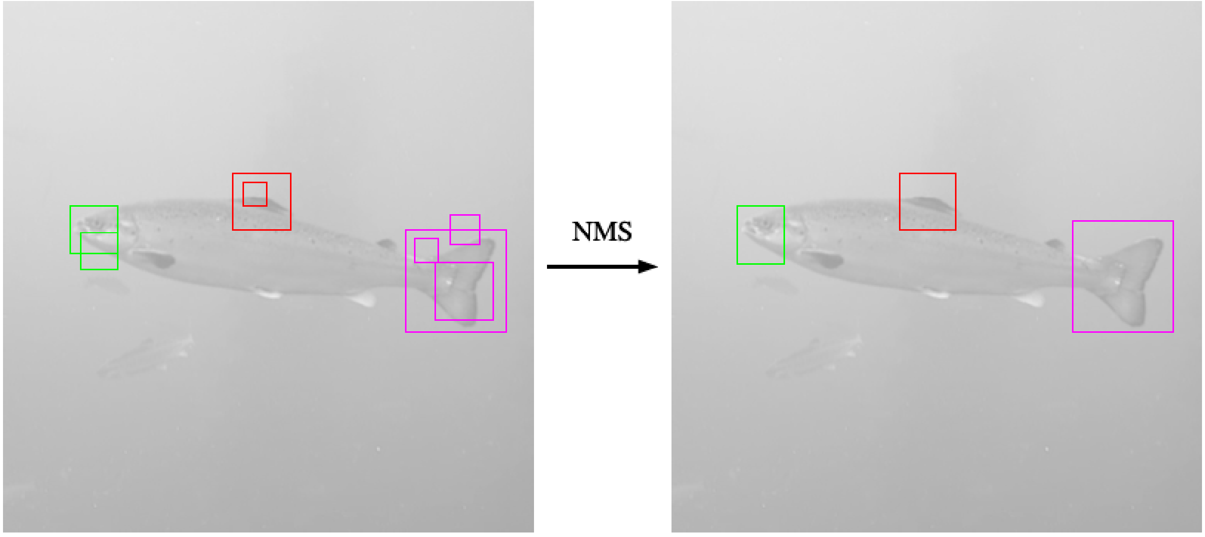


Figure 42 Exemplary results from the developed fish part detection algorithm. After a final non-maxima suppression (NMS) the desired parts of the fish can be detected.

The final goal of a complete system for the recognition of fish parts in a live video stream could not been reached with this classical computer vision and machine learning approach. The classifiers trained on the training images acquired at the test facility at Kåholmen on Hitra, Norway, were not able to classify image patches acquired from video streams taken at a later time. It is suspected that this is because the image patches used for training are not representative of a larger population. We note that state-of-the-art neural network-based approaches show more promising performance in detecting parts of fish. In particular, we refer to results achieved within the IPN project INDISAL.

Saliency based methods for camera orientation in aquaculture (Magnus Conrad Harr, June 2018: Supervisor: Annette Stahl, NTNU, Department of Engineering Cybernetics (ITK), Co-Supervisors: Christian Schellewald, SINTEF Ocean)

The aim of this thesis was to develop and provide insights into a saliency-based approach for automatic orientation of an underwater camera such that interesting/relevant regions are always captured. Existing algorithms for this purpose are not suitable for separating interesting and non-interesting objects in a sea-cage. Therefore, modifications/additions to these algorithms were implemented and tested. For the performance comparison, several saliency estimation techniques were used, combined with different extensions aiming specifically to work for aquaculture underwater recordings. The results are based on footage from an underwater camera system developed by Sealab (Figure 43). This project lays the foundations for future 24/7 surveillance in sea-cages using computer-vision algorithms. Such algorithms can also provide an image quality guarantee to operators with remote system access, even when the site is unmanned. The results presented in this thesis indicate that performing general camera orientation based on visual saliency in a sea-cage is difficult. It is expected that for a saliency-based orientation algorithm to function it will have to either be operated only when the camera is sufficiently far from the cage net or be used in tandem with a cage-net detector. As an alternative to a cage-net detector, one could implement a fish detector instead and use that as the basis for an automatic orientation. In conclusion, visual saliency can provide a basis for camera orientation. However, it is likely that other approaches based on machine learning (i.e. learning what is considered interesting or learning which objects should be looked at) would perform better.

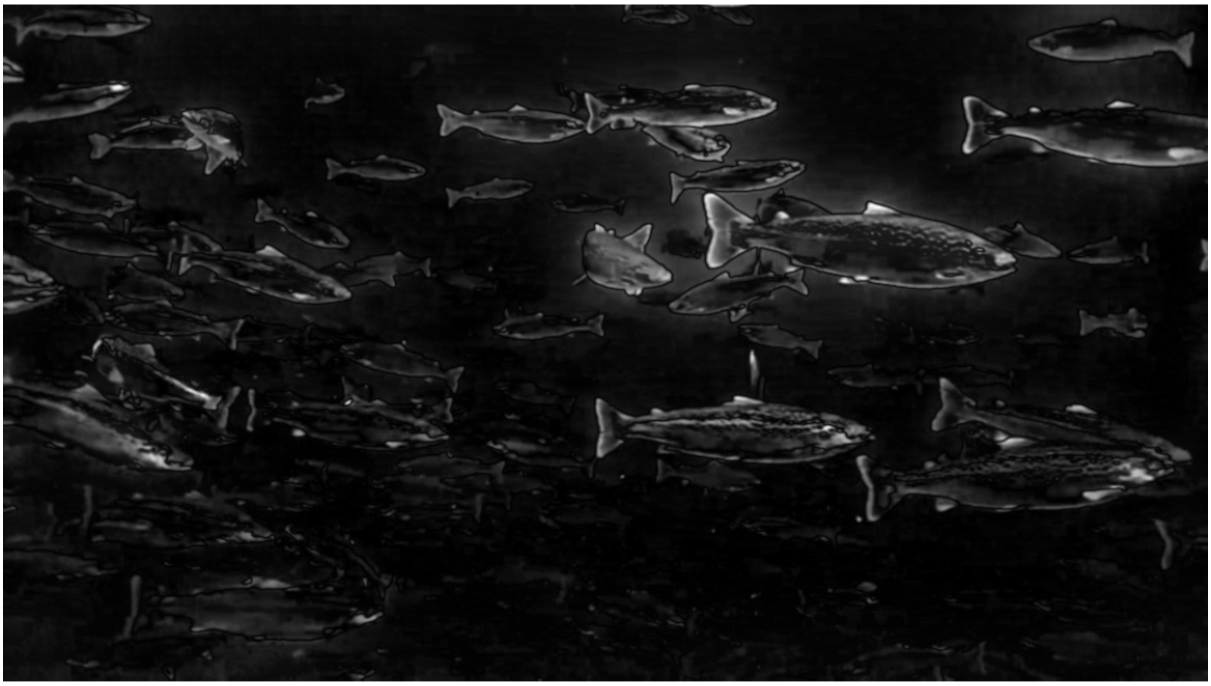


Figure 43 Example for a saliency detection algorithm applied to an underwater image containing salmon.

Unsupervised Learning of Motion Patterns for Object Classification in Aquaculture (Øyvind Rognerud Karlstad, June 2018: Supervisor: Annette Stahl, NTNU, Department of Engineering Cybernetics (ITK), Co-Supervisors: Christian Schellewald, SINTEF Ocean)

In this thesis the possibility of using unsupervised learning based on motion patterns to automatically classify the main groups of objects in a fish farm were investigated. The focus was on separating fish from feed. The approach is based on the hypothesis that fish and feed have distinct motion patterns that can serve as criteria to distinguish the two. The implemented approach is based on optical flow using KLT-tracking to estimate the motion in the image sequences. Similar motion patterns are automatically grouped together using cluster analysis. Mean shift and DBSCAN were chosen as the algorithms to be used in the experiments, based on a preliminary analysis of the motion data. Mean shift is centroid based, while DBSCAN is density based which provided a useful combination of differing properties to compare. Further, the effect of increasing object sizes to the clustering results was studied.

Results showed that automatically distinguishing fish and feed based on motion patterns is plausible under certain conditions (Figure 44). There are some requirements for the camera position that improve the classification accuracy. For instance, the clustering performance increases when numerous objects are simultaneous visible. Appropriately determining the clustering parameters is also necessary to avoid cluster merging. In cases where several clusters are merged together, valuable information about the objects gets lost. We found that the number of available data samples were too small to draw a conclusion, but for the tested image sequences we were able to distinguish visible motion patterns.

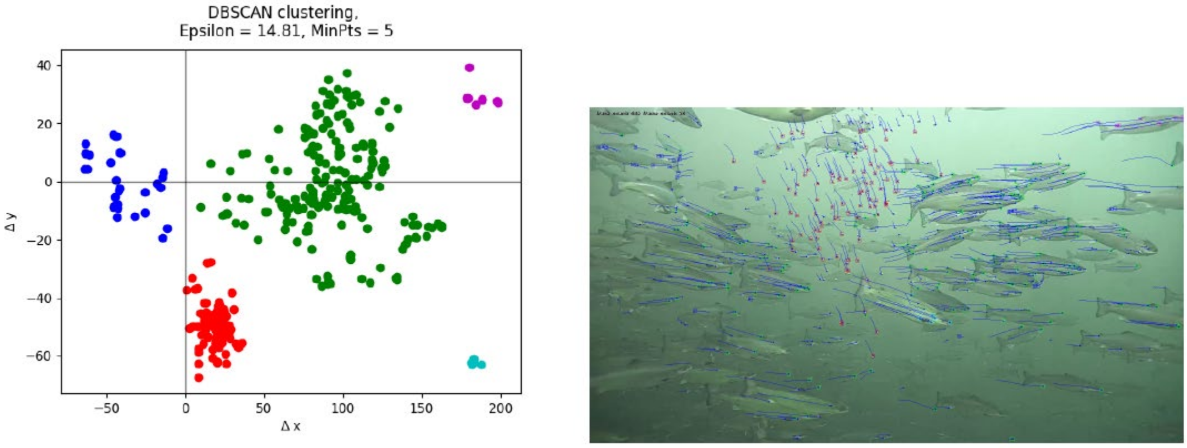


Figure 44 Example for automatic clustering of motion patterns visible in an underwater image-sequence containing salmon and feed pellets.

6 Conclusions

This report presents the development and validation of a 3D vision system to be used for data capture in fish cages. In particular, a compact and robust sensor with optical components and lighting system was developed to capture high-quality vision data. In addition, methods to evaluate the quality of the captured data were investigated and subsequently validated using vision data obtained from 24/7 video streams from a full-scale fish cage. This report furthermore includes the development of image processing algorithms to estimate the distance and orientation relative to the inspected object of interest, such as the fish or the net. The developed algorithms have been validated based on vision data obtained during laboratory and full-scale tests.

7 References

John Immerkær, Fast Noise Variance Estimation, Computer Vision and Image Understanding, Volume 64, Issue 2, 1996, Pages 300-302, ISSN 1077-3142, https://doi.org/10.1006/cviu.1996.0060.

Nayar, S. K., & Nakagawa, Y. (1994). Shape from focus. IEEE Transactions on Pattern analysis and machine intelligence, 16(8), 824-831.

Tenenbaum JM. Stanford University; Stanford, CA, USA: 1970. Accommodation in Computer Vision. PhD Thesis.

Pech-Pacheco, J.L.; Cristóbal, G.; Chamorro-Martinez, J.; Fernández-Valdivia, J. Diatom autofocusing in brightfield microscopy: A comparative study. In Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 3–7 September 2000; Volume 3, pp. 314–317.

A. Santos, C.O. de Solorzano, J.J. Vaquero, J.M. Pena, N. Mapica, F.D. Pozo,Evaluation of autofocus functions in molecular cytogenetic analysis, Journal of Microscopy 188 (1997) 264–272.