TTK4551 Engineering Cybernetics - Specialization Project

Augmented Reality and Computer Vision for Subsea operations

Alexander Næsheim

Fall 2018

Foreword

This report is written as a specialization project in engineering cybernetics as a part of the TTK4551 course at NTNU. The report should therefore be evaluated to 7.5 credits. This specialization is done with the intention of continue the work in a master thesis. Therefore the main goal of this report is to build a solid foundation to further build upon in further work.

The project is also written as a part of the SEAVENTION project in collaboration with SINTEF. I want therefore to thank Aksel Transeth at SINTEF for his consulting role as well as Anastasios Lekkas at NTNU for his co-consulting part.

Summary

The project statement investigated in this project is looking at how implementation of AR and computer vision could help improve subsea IMR operations. These operations are mainly performed with remotely operated UIDs, where the operators generally only have basic 2D information at their disposal when maneuvering the drone. The solution proposed in this project will give depth of field information to the operator through an augmented operator interface on a regular 2D display. Some implementations is done to test the potential of realizing such an system. This is though not a trivial implementation as computer vision system experience a variety of challenges when submerged underwater compared to the more tolerable conditions introduced by air.

Contents

1 Introduction

As an introduction to this specialization project the theme of the paper will be presented together with the motivation for and contribution from the presented work.

1.1 Background

Subsea inspection, maintenance, and repair (IMR) operations are frequently carried out in the offshore oil and gas industry. The main purpose of these activities is to maintain a sustainable flow of oil and natural gas from subsea wells. These operations are normally performed with the help of remotely operated underwater intervention drones (UIDs) from a vessel, or now in recent years also from shore. The operators are operating them by looking at a monitor with a videostream coming from the UIDs camera. With recent technological innovations within augmented reality and computer vision, the potential of improving the information presented to the operator is now becoming more interesting. This is because Augmented reality or AR lets us add information layers on top of streams of digital images. This information could be operation status of equipment and valves as well as navigational information. The specter of possible information available to be presented is also widened through the help of computer vision, which can capture much important 3D information from the 2D camera and potentially be trained to understand the environment that it sees through the camera.

Because these IMR operations are frequently carried out and relays on much of the same technology as in the late 90s, the potential improvements with the help of today's technology are intriguing and could result in huge benefits for everyone with interests within the industry.

1.2 Motivation

The oil and gas industry has had to cope with a drastically changed oil price in recent years after its fall from 120 US dollars in 2014 to now what seems to be a stable level between 60-80 US dollars. For sure this change has had an influence on the decision on what operations to execute and which not to execute. Sadly, expensive maintenance operations are often first in line to get postponed when times are bad. Because maintenance is so important to ensure stable and redundant operations during expected lifetime or even longer, a reduction of maintenance could, therefore, increase the downtime of systems and even reduce their overall lifetime. This could over time often accumulate too much greater costs than the maintenance itself. So, a method to reduce costs and the "burden" of doing maintenance in subsea operations could potentially result in huge cost savings, longer operational time of equipment and increased safety.[1] The motivation for this thesis statement channels from the desire to change the fact that the oil and gas industry has been slow to adapt to the digital change and thus execute many operations in the same way, with much of the same equipment as they have been doing for the last 10-20 years. One should think that strict safety and robustness requirements could be the reason for this, however, Lloyd's Register summarized the current state of the industry in 2017 as follows: "Innovate or die. In today's Energy industry, the speed of adoption lags behind that of other industries; industries that are subject to the same rash of safety, legal, commercial and financial pressures faced by Energy companies, such as Aerospace." [2] It may be many reasons for the industry to be slow to evolve, adapt and see the potential in the digital era. The same consultant company, Lloyd's Register, have found that the general wealth of the oil and gas companies might be the problem, and formulates their continued statement like this: "Ironically, the reason oil companies have been slow to invest in more efficient practices is due to their wealth. High market prices and a steady influx of capital allowed them to hide inefficiencies. The focus was growth and ¡qreserves exploration. If they are killing it in the market, why should they change? There is an old-guard that doesn't necessarily understand how technologies can drive bigger business value."[2] From a tough couple of years, it seems as if the industry has realized that they now need greater investment in new digital technology. AkerBP is an example with their great investment in digitization, proving that they want to take advantage of new technologies to increase their business value. [3] Thus, this project thesis is looking into the possibilities to utilized digital technology to help improve and make IMR operations, i.e maintenance operations, more convenient for operators. In a time where the industry is starting to understand the need for digital aids to maintain a strong place in the marked among their competitors.

1.3 Task description

The task looks at how augmented reality with the help of different sensors and computer vision can be used to aid operators during IMR operations with UIDs. More specifically the task looks at these questions

- Which IMR operations can benefit from AR and computer vision?
- How can this technology be used for decision support in IMR operations?
- What are the challenges in taking new technology such as AR and computer vision into use subsea?
- What kind of AR implementation is best applicable to help operators during operation?

In addition, the report is to answer the following thesis statement: How could the implementation of AR and Computer Vision help improve subsea IMR operations? This will be answered through a suggested system design for AR and computer vision implementation and some software implementation to check whether or not the suggested system would be realizable.

1.4 Scope

The technology field of Augmented reality and computer vision is wide and contains different levels of complexity. Because this is the author's first exposure to this kind of technology, C# programming, and the fact that this report is only evaluated to 7.5 credits, the complexity level of the implementations presented will be thereafter. We are therefore not looking into detailed implementation of SLAM or VO for computer vision tracking, however, the methods are briefly introduced as they provide the tracking that is the used standard in systems that relays on computer vision tracking, e.g unmanned autonomous areal vehicles. The project is started without any prior knowledge in either AR nor computer vision, other than through social media and articles read based on interest, and it will function as an introduction to the more comprehensive master thesis which will be a continuation of this work. Based on this, the main goal of this project will be quite ambitious. However, the methods used and approaches used in this project is based on the fact that this will function as a foundation to continue the work in a master thesis. Therefore, the scope is centered around the basics in computer vision/image processing and AR. Within this is also the desire to learn more about programming within one of the C languages to get used to the syntax, as this is likely something that's needed for the continued work in the master. Also, because of the limited scope of this project and resources when it comes to specific subsea IMR operation information, the project will not go into specifics when it comes to subsea operations.

In short, the scope of this project is to get a general introduction to AR, digital imaging and processing, and computer vision. With help of this information, it is desired to produce a suggested AR interface and investigate by implementation if this interface is realistic to realize. A method to set up the coupling between computer vision tracking and AR is without of this scope, and this problem will be solved with the help of pre-made developer kit

SKRIVE NOE OM KAMERA SENSORER BEGRENSINGER

1.5 Contribution

The contribution of this project thesis is a concept information layer laid on top of the UID videostream presented to the operator. This layer contains a target lock function, such that when a target is chosen, navigational information towards that target appears. A target being somewhere you want to go in order to perform the operation. In order for this target lock function to work an image segmentation by thresholding script is developed. In addition, to prove the potential of combining AR and computer vision, an application utilizing object tracking for navigational information is developed. Lastly, this project is also contributing to increasing the author's knowledge about the topics presented.

1.6 Outline of report

In the report, there are 7 sections, 6 of which are to follow with Theoretical background up next. In that section, the tools used to investigate the project statement is introduced and described. In addition, relevant theory for the presented results, implementation and discussion is also given here. The section is mainly divided into three subparts each looking deeper into Augmented Reality, Computer Vision and Subsea oil & gas fields, respectively.

Following the theoretical background, is section 3 where the methods used to investigate the project statement is listed. Next up is a description of how the implementations were executed in section 4. The results found from the implementation is given in section 5 and a discussion of these results and implementation is given in the following section. To finish off the report there is a conclusion and a paragraph with recommendations for further work.

2 Theoretical background

Facing a thesis statement which combines subsea industry with augmented reality (AR) and computer vision, a theoretical background is necessary to further down the road produce results and a discussion around these. In the following section, an introduction to AR and computer vision is presented together with a rather brief overview of subsea structures, IMR operations, and UIDs.

2.1 Introduction to augmented reality

This section will give a general introduction to AR through a definition, some history and brief about the state of the technology today.

2.1.1 What is augmented reality?

The Oxford Dictionary defines augment as "make something greater by adding to it" [4]. Therefore one could say that AR technology is integrating digital information with the user's environment in real time. This is different from other mixed reality technologies, such as VR which creates a totally artificial environment. AR, on the other hand, does not lock out our perspective of the real world. AR adds multiple layers of information to our vision of the real world to enhance our viewing perspective beyond the basic capabilities of human vision. These layers of information could be customized to fit different types of tasks, making AR potentially widely flexible in its area of application. [5, 6, 7]

2.1.2 The history of AR

The first example of an attempt to add additional data to an experience was through a device called Sensorama. This was invented by a cinematographer named Morton Heiling in 1959. An illustration of the invention is shown in fig.(1). When seated inside this device the user could experience deep visuals, sounds, vibrations, and smells. Moving on, the first head-mounted mixed reality device was called "The Sword of Demacles" (see fig.(2)) and was invented by the Harvard professor and computer scientist Ivan Sutherland. This head-mounted device was so heavy that it had to be fixed to the ceiling. Through this device, the user would experience computer graphics with the intention to make the users feel like they were in an alternate reality. Further, Myron Krueger in 1975 invented a device called Videoplace, which was a combination of a projection system and video cameras that produced shadows on a screen. This gave the user a feeling as if they were in an interactive environment where they could manipulate and interact with virtual objects in real-time. However, it wasn't before in 1990 the term "Augmented Reality" was introduced by the Boeing researcher named Tom Caudell. Soon after this, in 1992, the first properly functioning AR system was developed at USAF Armstrong's Research Lab by Louis Rosenberg. The invention was an incredibly complex robotic system, called Virtual Fixtures, which managed to overlay sensor information in workers work-environment to improve human productivity on the production line. [8, 9, 10]

Since then there have been many other breakthroughs, some of the most notable includes:

- A mobile outdoor AR game called ARQuake developed by Bruce Thomas in the year of 2000

- The introduction of ARToolkit, a design tool made available in Adobe flash in 2009
- The introduction of the open beta "Google Glass" by Google in 2013
- Microsoft announcing their new augmented reality headset "HoloLens" and their exciting vision for AR in 2015.

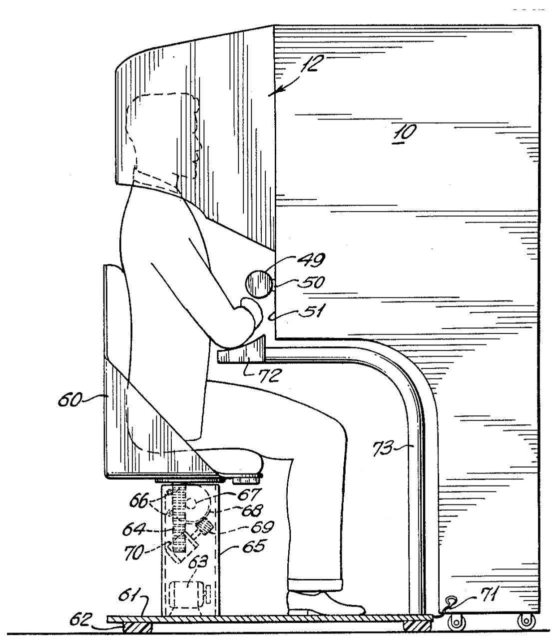

Figure 1: Illustration of the Sensorama device, invented by Morton Heiling

Figure 2: The Sword of Demacles head-mounted display

2.1.3 The state of augmented reality today

Due to big investment in augmented and virtual reality from highly valued digital innovative companies such as Facebook, Google, Sony, Microsoft and Apple in combination with great start-up environments, mixed reality technology is evolving day by day. It is though seen that AR is growing faster than VR, because of AR's adoption towards commercial and enterprise solutions. Industries from aerospace to logistics are all looking into ways to utilize AR in their day to day business to improve their position among competitors [11].

One of the reasons for the increased interest in AR in recent years is based on technological innovations bringing the necessary hardware and software for AR available to the masses. A great example is a smartphone which contains all the basic hardware needed for AR implementation. These components are the processor, display, camera, magnetometer, IMU and GPS and are all illustrated in fig.(3)[12].

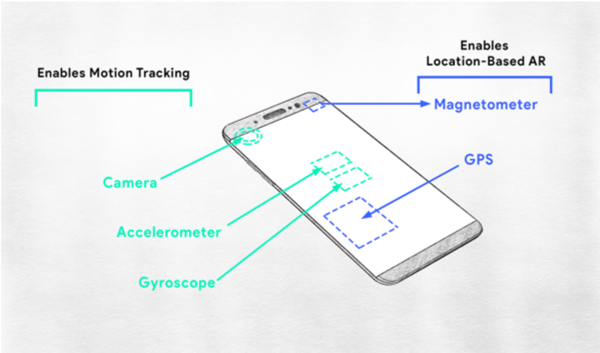

Figure 3: Hardware in smartphones that allows for portable AR implementation

Different types of AR

Very generally AR could be classified into four different types [13, 14]. These are:

- **Marker-Based AR** - This type of AR recognizes a marker or QR-code in the scene seen by a sensor (camera). When its recognized the detected area is overlaid with content or information. This type is also called Image Recognition or Recognition based AR. See fig.(4)

Figure 4: Illustration of marker-based AR

- **Marker-less AR** - Does not use any type of recognition system, but the environment is analyzed by using the camera sensor to harvest location information. When this data is collected, augmented objects can be added to the scene based on the 3D geometry in the observed location. This works great with smartphones as its performance is improved with the help of other sensors like the IMU and GPS. See fig.(5)

Figure 5: Illustration of marker-less AR

- **Projection AR** - A type of technology that projects artificial light onto real-world surfaces and thus allows for human interaction by detecting differentiating between an expected know flat projection and an altered projection. See fig.(6)

Figure 6: Illustration of projection AR

- **Superimposition based AR** - With this type of AR-technology, the augmented image can replace the original image, either partially or fully. Object recognition plays a vital role because the original view cannot be replaced by an augmented one if it does not know what object or the location of objects in the captured scene. See fig.(7)

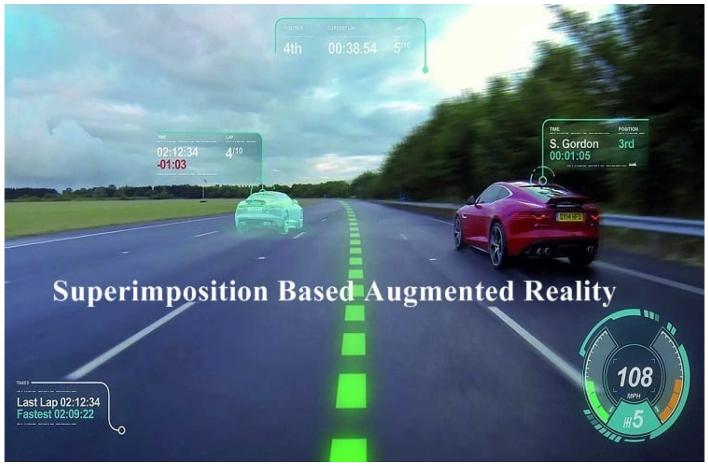

Figure 7: Illustration of superimposition based AR

Different types of AR hardware/devices

In order to visualize or present the augmented reality devices built up by complex hardware are necessary. These devices can very generally be segmented into two groups, one containing smartphones, tablets and computers with monitors while the other contains wearable devices like smartglasses and AR headsets. In the first segment one can expect to find 2D displays with augmented 2D information added to an underlying videostream. Examples of this could be the augmented face masks in Snapchat or the visualization of Pokémons in the "real" word in the Pokémon GO application fig.(8). The other segment contains wearable hardware like Microsoft's head-mounted HoloLens which uses light engines that shoots out images which when passes through a combiner manages to combine the projected image and the real world. Waveguides is used in the projection lenses to create series of diffraction gratings, this is to avoid projected holograms that "swims" in your vision. By doing this, these headsets or glasses manages to project 3D holograms into your 3D vision of the world, see fig.(9).

Figure 8: Marker-less AR on a smartphone through the Pokémon GO application

Figure 9: Illustration of how the 3D projected augmented reality could look like through Microsoft's AR Headset the HoloLens

2.2 Introduction to computer vision and image processing

To augment our reality, our computers must see the "reality" it is supposed to augment. This means we have to program our computers, so that they can obtain information from images and multi-dimensional data. Proper integration of all components, including camera, computer and software results in optimal computer vision system performance. It is also crucial that the camera is thoroughly calibrated. With an optimal performing system, the camera will act as a vision sensor that provides high-level information about it's surroundings, which is necessary to display accurate information through AR.

2.2.1 Digital images

Digital images are two dimensional arrays of numbers, where the dimension is determined by the amount of pixels in the image. Each pixel is represented by its own number in the array. This number contains information about the brightness of a given color in that specific pixel. We can utilize this representation to perform different operations on the image, operations better know as digital image processing. The goal of these operations is to either enhance the image or extract useful information from it. One could think of it as a type of signal processing where the input is an image and the output is some characteristics or features associated with that image [15]. There are different types of image formats, some of the most common in image processing is the following [16]:

- **Binary Image** - An image represented only by 0 and 1 values in the two-dimensional array, where 0 refers to black and 1 to white. Therefore, binary images are black and white images.

- **8 bit color format, grayscale image** - A image format constructed by 256 different shades of gray, where 0 represents black, 255 white and 127 gives gray.
- **24 bit color format (RGB and BGR)** - A colored image format divided into three deeper 8-bit formats which are red, green and blue. In this order, the image is of RGB format, a very common colored image format where blue is occupying the least significant area, green the second least and red the third least. Therefore, the name represents the stacking order of colors within the format. Another much-used format is BGR which is the standard image format in OpenCV (see section 2.3). In this format, colors are stored in color channels opposite to the order of RGB. Because of its 24 bit structure it yields $256 \cdot 256 \cdot 256 = 16\ 777\ 216$ possible color combinations [17, 18].
- **24 bit color format with HSV color model** - HSV is like RGB and BGR a format that describes the way various basic colors combine to create the rainbow of colors. HSV is through a bit different of a model compared to BGR and RGB which are defined by primary colors. HSV is more similar to how humans perceive color and are defined through hue, saturation and brightness value. This color representation describes colors (hue) in terms of their shade (saturation) and their brightness value. Hue represents a color wheel, a 0 to 360 representation of the color range from red to magenta. Saturation is the amount of gray in the color from 0 to 100 percent, while the value of brightness describes the intensity of the color from 0 to 100 percent, where 0 is completely black and 100 is the brightest and reveals the most color [19]

2.2.2 Image processing

With one of the introduced digital image formats as input, we can execute different image processing operations on this input. A description of the relevant operations for this project are described in the remainder of this subsection.

Kernel convolution
A kernel is a small matrix of ode dimension and dependent on the element values, a kernel can cause a wide range of effects on an input image. Among these effects are linear filtering operations such as blurring, sharpening, edge-detection and more. These effects are caused by convolving kernels and images. This is done through matrix convolution which is shown in eq.(1) where the kernel is a 3x3 matrix and the digital image a 5x5 matrix. The pixel that is being convoluted is defined by the pixel which the kernel is above at the current calculation, in eq.(1) that is the pixel of value 1 in the image. When the convolution is calculated, the pixel output value is to be changed by the normalized value of this operation. For this example this value is 58, see eq.(2). It has to be emphasized that the normalized convolution value has to be inserted into the current pixels place in the matrix in a new output image. If we are updating the value of the pixels in the actual image we are convolving in, we would mess up the whole process. The reason for normalizing the value before insertion in the output image is to make sure we are not making the image brighter or darker during the operation. Because of this, the whole kernel convolution process could be thought of as a big weighted average of the image, where the different weights determine what kind of effect we are applying to the image [20, 21].

$$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix} * \begin{bmatrix}10 & 20 & 30 & 40 & 50\\40 & 1 & 60 & 40 & 50\\70 & 80 & 90 & 40 & 50\\10 & 20 & 30 & 40 & 50\\20 & 30 & 30 & 40 & 70\end{bmatrix} =$$
$$(10 * 1) + (20 * 2) + (30 * 3) + (40 * 4) + (1 * 5) + (60 * 6) + (70 * 7) + (80 * 8) + (90 * 9) = 2605 \tag{1}$$

$$sum\ of\ kernel\ = (1+2+3+4+5+6+7+8+9) = 45, \ \Rightarrow normalized\ value = \frac{2605}{45} \approx 58 \tag{2}$$

Noise and Filtering

When taking a digital photo, the resulting raw digital image is often difficult to read for a computer vision system. This difficulty is caused by possible random variations in intensity, variations in illumination or poor contrast [22]. These variations in intensity represent noise in the image, and they can change rapidly from pixel to pixel. Some common types of noise are salt and pepper noise, impulse noise and Gaussian noise. Salt and pepper noise is caused by sharp and sudden disturbances in the image signal and appears as white and black pixels. While impulse noise comes from random occurrences of pixels with bright intensity values in the image. In addition, there is Gaussian noise which is noise that contains variations in intensity related to a Gaussian distribution, which is a good model for many kinds of sensor noise. A method used to reduce these noise disturbances in the image is linear filtering, these spatial operations can be done through kernel convolution, where the values in the kernel represent different types of linear filtering. The simplest being the mean blur where each element of the kernel is 1. Another very common blurring method is the Gaussian blur, it takes more care of the edges. It's a more controlled form than a mean blur. The difference comes from the Gaussian having different weighting, i.e. higher values in the center. So the further from the center you get, the less weight you have. This means that you are not going to blur too much, and when you are approaching an edge, you are not going to take too much away from it as the outlying pixels are weaker weighted [21]. A blurring filter is a type of low pass filter and therefore by suppressing the high-intensity noise, gradual changes in pixel intensity becomes easier to detect. There are also other types of filters such as sharpening spatial filters (high pass filter), i.e filters that highlight detail in the image, contrary to blurring, and non-linear filters, e.g median filters [23, 24].

For image processing, the two-dimensional zero-mean discrete Gaussian function eq.(3a) is used as a smoothing filter. For instance, this results in the 5x5 Gaussian blur kernel in eq.(3b). [25]

$$g[i,j] = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \tag{3a}$$

$$\frac{1}{273}\begin{bmatrix}1 & 4 & 7 & 4 & 1\\4 & 16 & 26 & 16 & 4\\7 & 26 & 41 & 26 & 7\\4 & 16 & 26 & 16 & 4\\1 & 4 & 7 & 4 & 1\end{bmatrix} \tag{3b}$$

Morphological Transformations

Morphological transformations are operations that improve the shape of segmented objects in an image. It is normally performed on binary images and is executed by kernel convolution. It's applied to binary images after segmentation and it works as a structuring element. Relevant for this paper is dilation and erosion, as well as the opening and closing operations. Opening being that you do erosion followed by dilation and for closing, you do them in the opposite order[26].

- **Dilation** - Does hole filling and connects blobs[1]. This is done by defining some binary shape, say a 3x3 matrix. Like kernel convolution, this is a sliding window operation which affects the pixel underneath the center of a sliding operator, which consists of 1's. The operation is done by sliding over the binary image and every time the number 1 in the kernel hits a 1 in the image, the pixel underneath the center of the kernel is given the value of 1 in the corresponding output image. Only one element in the kernel must correspond to a 1 in the image, that is because dilation work with the "or" operator. This way, things tend to get bigger in size and thus fill holes in the blobs.
- **Erosion** - Does hole enlarging and disconnecting of blobs. This operation is the opposite to dilation because it operates with the "and" operator. Therefore, every element underneath your sliding window operator has to be the number 1 in the binary image in order to result in a 1 (bright pixel) in the output image [27].

Thresholding and Segmentation

Thresholding is the simplest and most common method for image segmentation. The thresholding constants/thresholds are determined through trial and error. The reason for thresholding is to divide a digital image into multiple segments. One example is the process of going from a grayscale image to a binary. This is done by choosing some threshold value if a pixel in the image has a lower intensity than this threshold it is assigned a value of 0 (black) in the binary image. If the pixel value is above the threshold it's represented with 1 (white) [28]. An example of a threshold segmented binary image is shown in fig.(10).

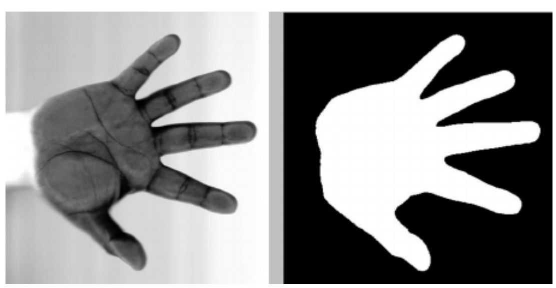

Figure 10: A grayscale image segmented by a threshold value, the result is presented in a binary image

Another way to utilize thresholding is by using masks. Masks can be seen as a layer laid on top of your original image causing your segmented parts to stand out on their own [29]. When using a mask on an image, the mask will often contain pixels with the same intensity value in the segmented parts, while the rest could hold the values of the original image, i.e. the background values. An example of a masked image is given in fig.(11). For thresholding by color on a colored image, an upper and lower bound threshold should be chosen. The pixels within this threshold is then filtered out from the background and grouped in its own segments[30].

[1]Binary large objects

Figure 11: A segmented digital image with a mask laid on top to highlight the segmented parts [31].

2.2.3 Digital camera sensor

In this section the basics around imaging models are introduced, this is the basis for digital imagining. Even though there are new modern sensors like RGB-D cameras and lidar sensors, we will in this section focus on the traditional digital camera model while the functional effect of RGB-D cameras will be represented by a stereo setup of the more traditional digital camera sensor.

The very basics elements of an imaging device is given by:

- **Aperture** - is an opening (or "pupil") in the camera system to limit the amount and angle of the incoming light rays. For a pinhole camera model, which is a much-used camera model because of its mathematical simplifications, the aperture is chosen so small that each ray from the scene passes undeflected through the image plane.
- **Optical system (usually a lens)** - is used to focus the light to a single point, i.e focuses the image onto the plane. This is done by deflecting parallel incoming rays so that they go through the focus point. Rays passing through the center are undeflected.
- **Imaging photosensitive surface** - is a two-dimensional array of sensor elements that accumulate charge during the exposure, once the exposure time is out, these charges are transferred out and digitized to further down the road be read by a computer or a digital monitor. The main design factors for this element is the number of sensor elements, chip size (plane size) and the resolution of the analog to digital converter. The plane consists of a 2D normalized image plane where the z-axis points out of the 2D plane, that is the z-axis of the camera frame (dotted line in fig.(12) and x-axis to the right, while y-axis points downwards). In addition, there is a digitized image buff which represents the pixels in the image. The center of this

digitized plane is where the z-axis crosses it, which is given by c_u and c_v. This represents the pixel location of the principal point, i.e the image center. The origin of this pixel plane is at the top left corner where u and v represents columns and rows, respectively. The difference between a representation of an image projection in 2D and the same image sampled and quantized in the image buffer is illustrated in fig.(13) [32].

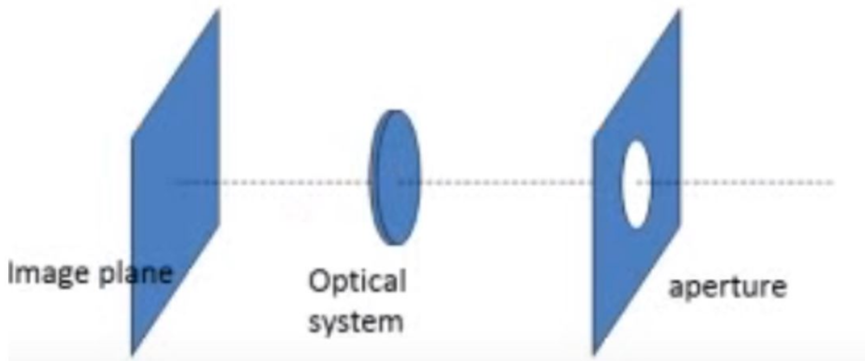

Figure 12: The basic optical system of a camera

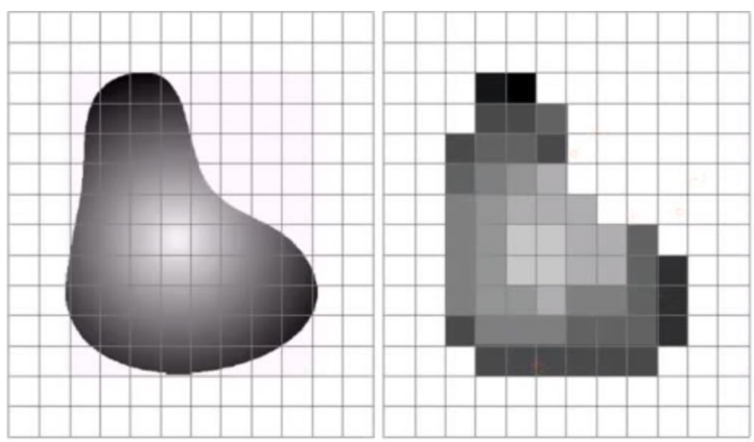

Figure 13: To the left: A continuous image projected onto the 2D image

Camera characteristics

The parameters determining the internal characteristics of the camera is given by a matrix called "the intrinsic matrix" this matrix is found by calibrating the camera. Camera calibration can be done by taking several images of a black and white checkerboard with a measured size of each square. By running this through a calibration algorithm that detects the corners between each square you get the calculated intrinsic matrix for your camera. This matrix contains the focal length f, optical center, a skew parameter and pixel scaling factors. In addition, from camera calibration you get the lens distortion coefficients which is needed to make sure that straight lines in the real world appears as straight and not curved lines in the image. The camera matrix is built up by the intrinsic and the extrinsic matrix, the latter being a translation matrix containing information about the position of the camera in the world. When these two matrices are known one can calculate the transformation from pixel coordinates to 3D world coordinates. This relationship is given by eq.(4) and the reference coordinate systems is illustrated in fig.(14) [33].

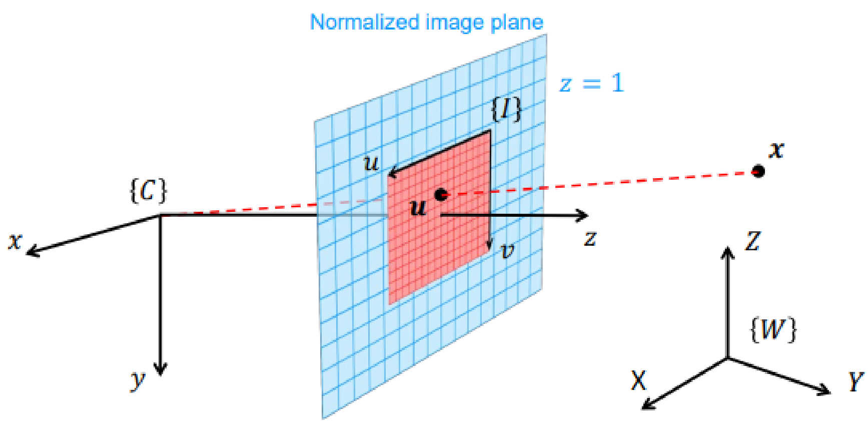

Figure 14: Camera and World coordinate frame C and W with normalized image plane (blue) and image buffer (red)

$$\tilde{u} = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{cw} & t^w_{cw} \\ 0 & 1 \end{bmatrix} \tilde{X}^w \tag{4}$$

Camera setup

Because a monocular camera setup only consists of one camera it is cheaper, more reliable and takes less space compared to a stereo setup (2 cameras). However, the drawback of not having stereo vision is the scale. With only one camera, the scale parameter needs to be specified in order to get some scale information. When stationary it's difficult to extract 3D information from a standard 2D camera sensor without any additional information, resources such as known 3D points/models in the scene or structured light. However, if the camera is moving, and the relative position and orientation (pose) between each image frame is known, we can extract depth information from epipolar geometry and perform the coordinate transformation given in eq.(4).

Stereo camera is another type of setup that requires two cameras. These two are calibrated together with a known relative pose between them. Whereas the monocular setup is depending on already detected models or points in the scene, a stereo setup can determine the position of 3D points in an arbitrary scene. The depth information is now given by disparity, i.e how different parts of the two images align with a constant horizontal shift between them, the baseline. So, by knowing the intrinsic parameters from a stereo calibration and the baseline between them, a measured point in the left image could be found along the epipolar line in the right image. This is a great constraint, limiting our search for the corresponding point from the whole image to only a line of pixels. The reasoning behind this constraint is the fact that a ray from the left image will hit and go through a 3D point in the world. To find and triangulate this exact point in the right image, one can send out different rays crossing that ray from the left camera. The rays that cross the ray from the left image lay along the epipolar line in the right image. At the point where these rays have the same or very similar intensity characteristics yields a 3D point in the world. The problem of finding the corresponding point between the two cameras in a stereo setup is called "the correspondence problem" and is seen as the most difficult part of stereo vision. An illustration of this concept is shown in fig.(15).

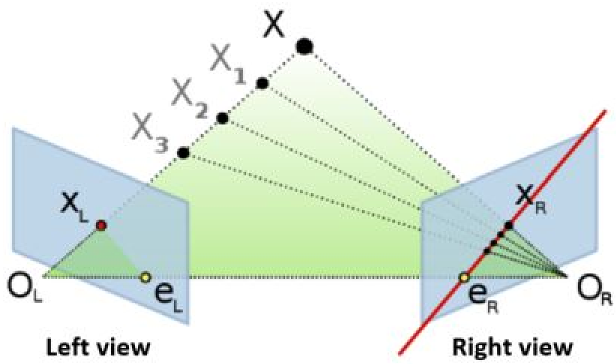

Figure 15: Illustration of possible stereo camera setup, with dept of field capabilities

2.2.4 Illumination and contrast

Lighting and therefore also contrast is key to accurate, robust and reproducible computer vision results. The computer creates images by analyzing the reflected light from an object, not by analyzing the object itself. Therefore, lighting is an extremely important part of the vision system. The total light seen by a camera is all the reflected, absorbed, transmitted and emitted light, see fig.(16). Some filtering methods are therefore recommended to control the exposure. In order to achieve the best lighting for the given situation, different types of lighting techniques can be used. For the remainder of this section, we are going to look at two different illumination techniques and contrast.

Backlighting
This is a technique where the camera and lighting source are acting in opposite directions. Therefore the lighting will reveal the outlines of the objects in the scene. This method is helpful in situations where shape and edge detection are of interest. As a result, backlighting is widely used for dimensional measurements.

Structured light
This technique projects accurate patterns of light onto a target. Normally the patterns could be plane, grid or more complex shapes. The lighting source is mounted at a known distance and angle from the camera giving the relative pose between them. Through this information and displacement or distortion measurements, the shape, size, and position of the object can be calculated. In addition, a narrow pass filter should be used on the camera to filter the reflected light from the ambient light. This method is useful for acquiring dimensional information and calculation of volumes

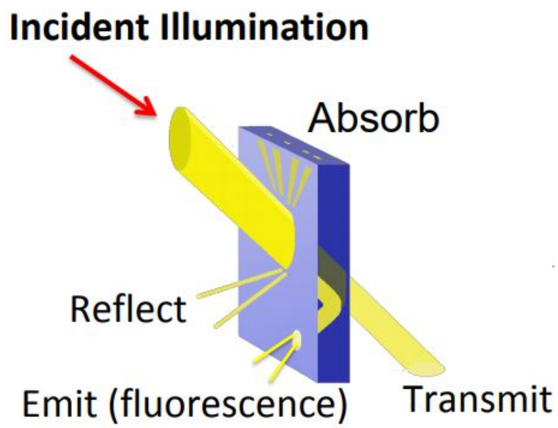

Figure 16: Spreading of light waves in Illumination

Contrast

Contrast is the difference in luminance or color that makes an object or its representation in an image distinguishable. Therefore, to best detect features of interest, we must create as much contrast as possible. An example of the power of contrast is presented in fig.(17 and 18). The one on the left is lit with UV and strong red 660nm ambient light. No filter used. The one on the right is lit with the same UV and red 660nm lighting, but with the application of a 510nm short pass filter. One can see that the right picture better highlights the contrasts of the object because of the increased contrast. This means that because of greater contrast, we can easier distinguish certain elements of the object from each other, and thereby detect features such as edges and corners. A method to increase the level of contrast is to change the lighting direction, which can be done through reconfiguration of the setup geometry. That is the geometric relationship between object, light, and camera. Another method is the one shown in the two mentioned figures, where filtering of light affects the wavelength or character of the lighting that is collected by the camera.

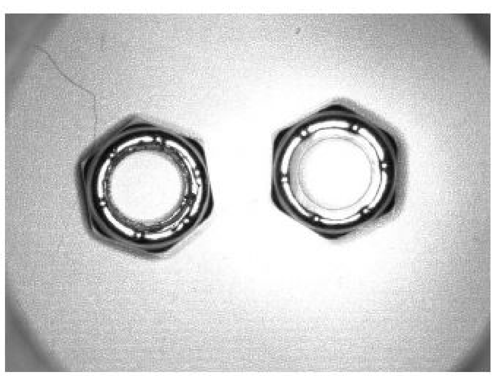

Figure 17: UV and strong 660nm ambient lighting

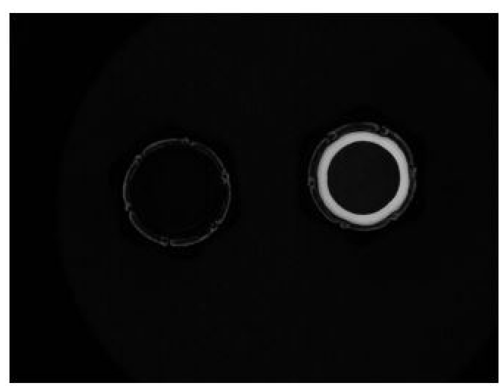

Figure 18: UV and strong 660nm ambient lighting, with 510 nm Short Pass filter applied

2.2.5 SLAM and VO

SLAM is Simultaneous Localization And Mapping, this means that you want to localize yourself in a map that does not exist yet, so you have to create it at the same time as you localize yourself in it. This map is created by using some kind of visual sensor, e.g. a camera, and then detect different landmarks in the incoming images. These landmarks are identified by features in indirect SLAM methods such as ORB SLAM[34]. This SLAM method uses the FAST feature detector to find features (corners) and a modified BRIEF feature descriptor is used to find alike features in new incoming images. These methods are indirect because they do not look at all information available in the image, they just extract characteristical information from it. There is also direct methods like LSD-SLAM[35] and SVO[36] that searches through all the pixel information in the image to find correspondence between incoming images. The reason you want this correspondence is to triangulate points in a map so that you can make an estimate of your relative position to the point in your map and thus localize yourself within your map. Your map will then often consist of keypoints, which are the most distinctive points detected by a single image frame. A map built of keypoints will result in a sparse map, that is a map with less information and therefore occupies less memory in the computer compared to a dense method that tries to make a map by reconstructing the 3D scene captured by the camera.

By tracking yourself in a map of recognized characteristics you introduce the possibility to perform loop closures. That is when you recognize the correspondence between your current image frame and a previous image. It is then possible to correct for accumulated drift from estimation uncertainties in your position because you now get additional information about your whereabouts. The principal of loop closures is therefore in many ways similar to the feedback loop in a closed loop system. This is also where Visual Odometry (VO) differs from SLAM, VO will only save a fixed amount of previous image frames and thus not have the ability to perform loop closures. VO is therefore heavily exposed to drift. In fig.(19 and 20) one can see how SLAM keeps image frames from its whole trajectory, it can, therefore, recognize places it has been before and thus performs loop closures to update the map and fix the accumulated drift. These loop closures are detected through bundle adjustment. Compared to VO in fig.(21) one can see that there is only a fixed number of saved poses in the map which means it cannot recognize places it has earlier captured [37, 38].

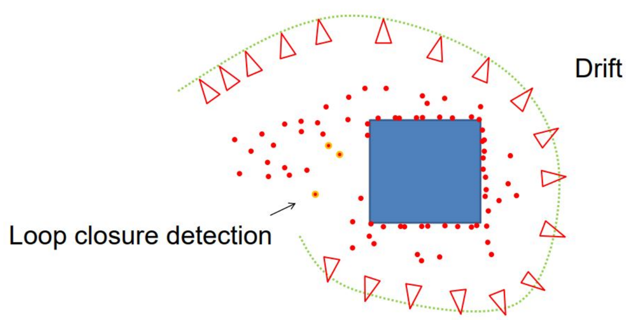

Figure 19: Illustration of a SLAM pose graph with the mapped keypoints, one can see that the camera frames is exposed to drift because they do not follow the green line perfectly.

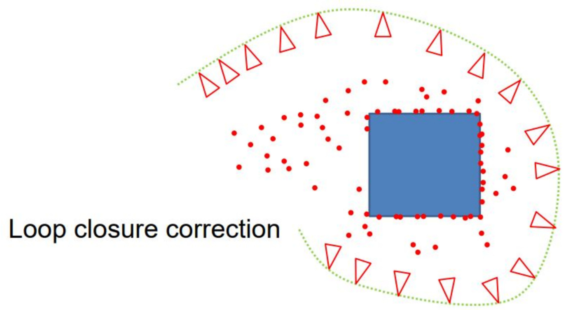

Figure 20: The same pose graph as above but now with loop closure performed. Drift is no longer a concern

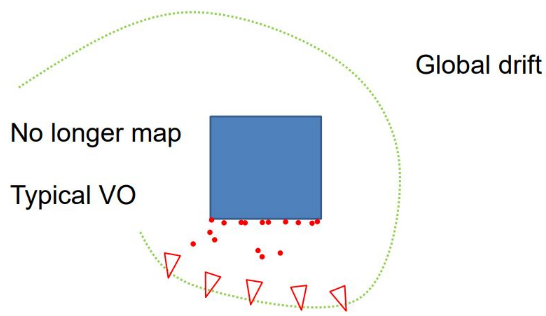

Figure 21: Illustration of a VO pose graph with sliding window mapping (a fixed amount of image frames in the map

2.3 Tools for computer vision and augmented reality implementation

To investigate the thesis statement, different programming languages, software, developer kits, and libraries are used. A brief introduction and description of the most important is given here to build a theoretical background. Section (**??**) presents information about how these tools are used to obtain results in this project.

Python
Python is an interpreted, object-oriented, high-level and general-purpose programming language [39]. Compared to other programming languages, pythons high-level and relatively easy to understand syntax makes it one of the simpler languages to learn. It supports modules and packages as well as standard and third-party programming libraries. In programming, a library is a collection of pre-made routines that a programmable program can use. Libraries are therefore useful for storing frequently used routines [40]. Python was created by Guido van Rossum and first released in 1991 and have in the later years experienced an extreme growth in its user base [41].

C#
C# is a general object-oriented programming language that enables the developer to build a variety of secure and robust applications that run on the .NET framework[42]. The .NET framework is a software framework that executes coded programs in a software environment in contrast to the usual execution in a hardware environment. This makes the execution more secure, with better memory

management, and improved handling of errors, interruptions, etc[43]. Like the .NET framework, C# is developed by Microsoft and is in many ways an updated improvement of many C and C++ features. Therefore, the syntax of C# should be instantly recognizable to anyone familiar with C, C++ or Java. With the help of Visual Studios also developed by Microsoft, Visual C# provides an advanced code editor, convenient user interface, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET framework[44].

OpenCV

OpenCV or Open Source Computer Vision is a computer programming library of functions used for real-time computer vision. The library was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products [45]. The software package was originally developed by Intel but was later supported by Itseez which is now acquired by Intel. Under the BSD license[46], this library acts as a cross-platform free to use open source library of programming functions. The package itself is written in C++ and has therefore primarily an interface for C++ coding. However, there are bindings to different languages, making it possible to utilize the library interface in Python, Java, and MATLAB. Today the library has a wide range of algorithms, more than 2500 in fact [45], which includes both computer vision and machine learning algorithms. Some of the features possible with these algorithms are face recognition and detection, object identification, camera calibration, tracking of camera movement and establishment of markers to overlay the displayed video with augmented reality. OpenCV is widely used with its estimated number of downloads exceeding 14 million.

Unity

Unity was released in 2005 as an OS X-exclusive game engine. In 2018 however, the engine was extended to support 27 platforms, with Windows and Linux being among them. It was developed by Unity technologies and can be used to support the development of both 2D and 3D features and has functionality among different needs and area of applications, for instance, game development. It also includes AI pathfinding tools which are a navigation system that allows intelligent movement of nonintelligent objects through automatically generated meshes based on the scene geometry, as well as a versatile AR/VR interface. In addition, there is an assets store which contains thousands of resources, tools, and extensions to help project development [47]. The engine is closed-source, however, it is available for free use when the user's revenue or funding does not exceed 100 000$ per year. In order to make the scene dynamic, C# scripts are attached to the scenes and various models within the scene. Resulting in a fully programmable dynamic scene [48].

2.4 Introduction to Subsea oil & gas fields

In order to investigate the project statement some basic information about the subsea industry is desired. However, following the scope of this project, this will be a very brief section to get familiar with the basics within subsea fields and operations.

2.4.1 Subsea field

Ever since the first subsea well was brought into production in 1962, the production technology and subsea infrastructure have made immense improvements. Potential economic savings drove these improvements together with the opportunity to increase hydrocarbon recovery rate and the possibility to produce from wells in harsher environments and at deeper depths. Subsea fields are built up of several closed placed wells on a template layout or more spread out wells in a satellite field. On each wellhead, a huge valve-tree module (Xmas-tree) is mounted. This module contains the Subsea Control Module, which makes it possible to remotely control every valve on the Xmas-tree and thereby control the whole flow from the well. Depending on the complexity of the subsea field, additional modules and structures can be placed on the seabed [49, 50].

Today a large number of topside platforms have several subsea wells and fields connected to them. From this, operators can drill new subsea wells and connect these to already implemented topside production facilities at different fields, giving greater flexibility in field layout as well as economic savings. An example of this is the Gullfaks field, where smaller fields of subsea wells such as Vigdis, Gullfaks south, Gullveig and Rimfaks (see fig.(22)) where drilled and installed after the installation of the main topside Gullfaks platforms.

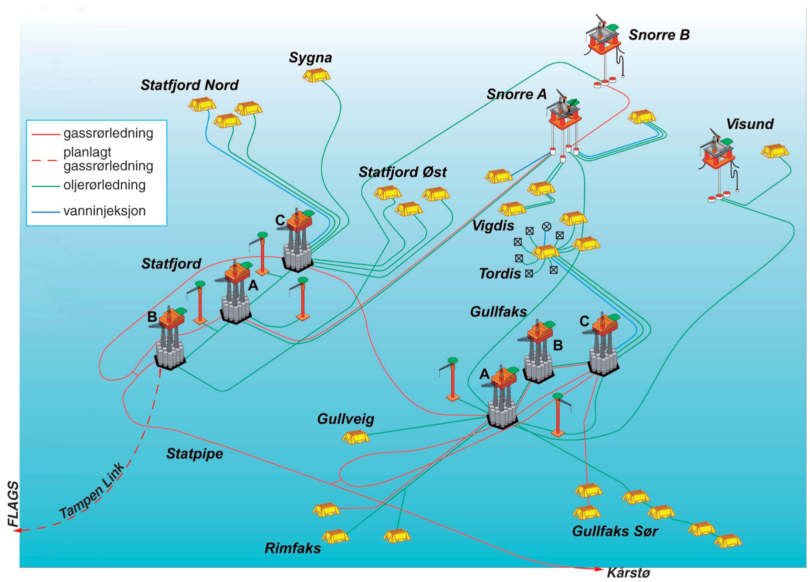

Figure 22: Field layout around topside platforms Gullfaks, Stafjord, Snorre and Visun

Subsea fields are not limited to connect to already existing infrastructure. Other common methods used at the Norwegian Continental Shelf (NCS) is pipelines to an onshore facility (Ormen Lange) Fig.(24) and subsea field tied back to an FPSO (Knarr) Fig.(23). An FPSO is a more flexible topside structure because it can disconnect itself from the field and float away if required or when production is finalized. The principle of a pipeline tie-in to shore is highly topical in regards to the industry's ambition to make subsea production factories. A method that would get rid of any topside or onshore processing facilities, resulting in distribution of oil & gas directly from the subsea factory. [51]

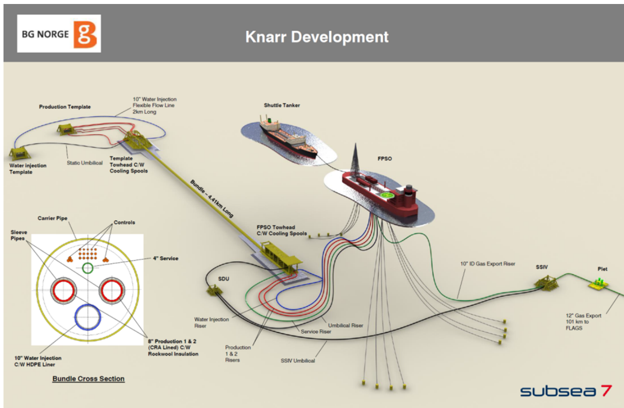

Figure 23: Subsea to FPSO (Knarr field)

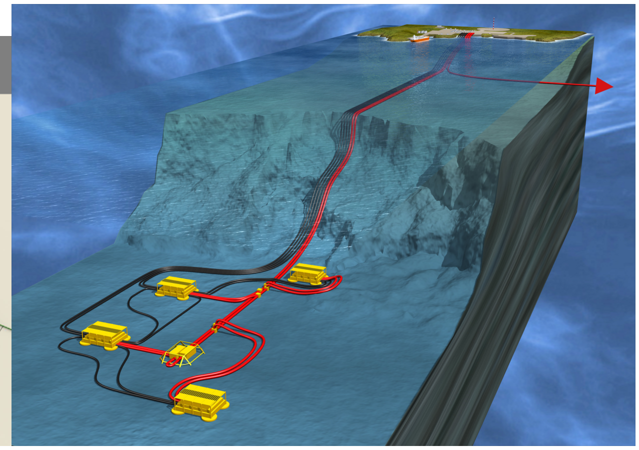

Figure 24: Subsea to onshore facility (Ormen Lange field)

2.4.2 Different types of Subsea structures

Several different types of subsea structures can be found on the seabed. Among them are some key structures that UIDs (see section 2.4.4) frequently interact with. In general, orange colored components on these structures is equipment designed for UID interaction. For the remaining of this subsection, two types of structures that the UID frequently interact with will be described. [52, 53]

Xmas-trees

The Xmas-trees lives on top of the wellhead. It contains an arrangement of valves, pipes, fittings, and connections. They can be arranged in a cluster on a fixed template or spread out as a single satellite well. There are two different types of trees, a horizontal with its master valve in a horizontal bore and a conventional tree with its master valve in the vertical bore. The valves are controlled through electrical or hydraulic signals and can also be manually controlled by a UID. As mention, at this structure, one can also find the Subsea Control Module, SCM. This module act as the wells "brain" because it allows for remote control of the valves via hydraulic and electrical signals. The main tasks for UIDs on these trees are manual operations of valve positions and SCM replacement intervention.

Subsea manifolds

To reduce the use of subsea pipelines and risers and to optimize fluid flow, manifolds are installed on the seabed. Its an arrangement of piping and valves used to gather produced fluids or to distribute injected fluids. This structure acts as the main interface between the production pipeline, flowline, and well. The manifold gathers flow from several satellite Xmas-trees and distributes the combined flow further into the pipeline system. When UID interacts with manifolds it is mainly through valve operations and replacement of jumpers, cables, and hoses. Like the Xmas-trees, manifolds also have a dedicated UID panel like the one in fig.(25).

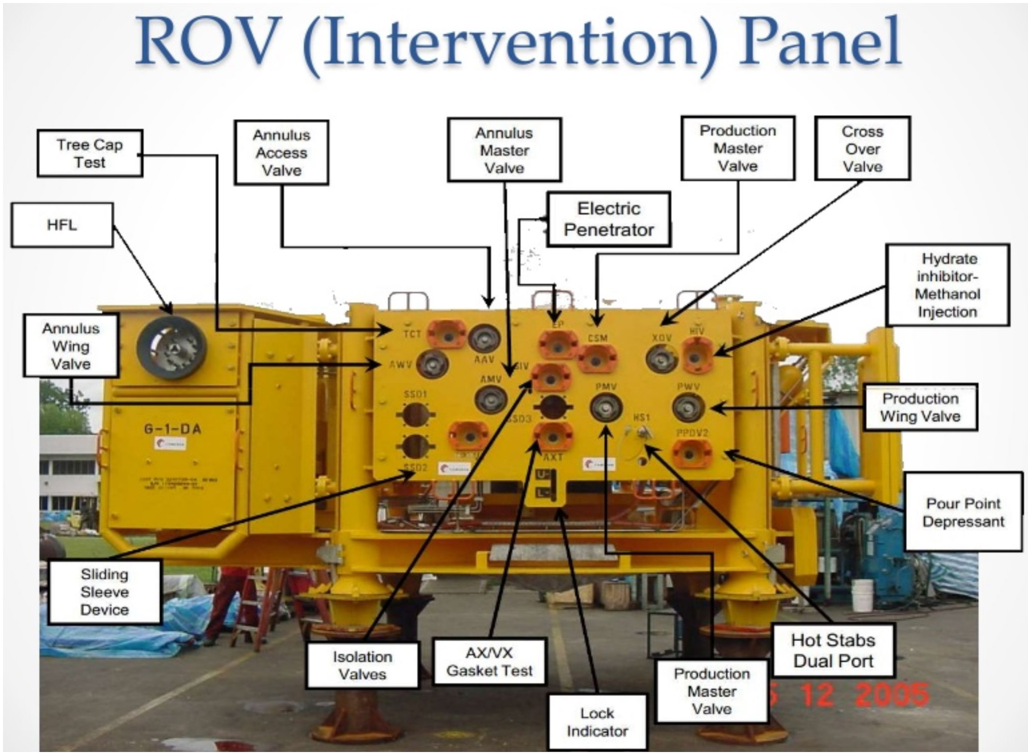

Figure 25: An example of a ROV/UID interface panel used for different IMR operations

2.4.3 Subsea inspection, maintenance and repair (IMR) operations

The main goal of IMR operations is to maintain a sustainable flow of oil and gas from subsea wells. Structures at the seabed is built highly modular, so that when a component fails it can be replaced or repaired. The IMR operation consist of three main different types of operations, these are:

- **Inspection** - This activity is mainly focusing on documenting the condition of a seabed installation or to visually plan and map where new equipment or pipelines should be installed.
- **Maintenance interventions** - These are performed on the well itself or on other subsea structures like the Xmas-tree and manifold. Generally, these operations could be everything from cleaning of subsea equipment to routine replacement of parts that are exposed to wear and tear.
- **Repair interventions** - If damaged equipment are detected through inspections or sensors, repair interventions are executed if possible to salvage the life of the component.

These operations are often of a smaller scale than other subsea operations and are also executed frequently. Because these installations are placed on the seabed, the operations are performed with remotely controlled UIDs from an IMR vessel topside. These vessels are versatile, highly technical, and are designed to continue operations in harsh weather conditions. They also have a moon pool where they deploy the UID from [54, 55].

2.4.4 Underwater intervention drones (UID)

UID is a wide definition of different types of remotely operated vehicles used in different subsea operations. For IMR operations, especially maintenance and repair operations, advanced ROVs with different tools and equipment are used, see fig.(26). These can be controlled from a vessel at the seafloor or from locations on shore. Usually, these drones have to be connected to a vessel topside through an umbilical that houses electrical communication and hydraulic cables. These drones contain at least one visual device such as a camera to let the operator navigate the drone at the seabed. The kind of ROVs from fig.(26) is also equipped with two manipulators. One rather basic that is used to grip onto handlebars on subsea structures to keep the ROV fixed during a specific task on the structure. The right one is more advanced and has several degrees of freedom to help carry out the task or operation. It is also controlled by an advanced tailor-made joystick showed in fig.(27) [56, 57]. In the continuation of the report, both UID and ROV will be used to describe the kind device introduced in fig.(26).

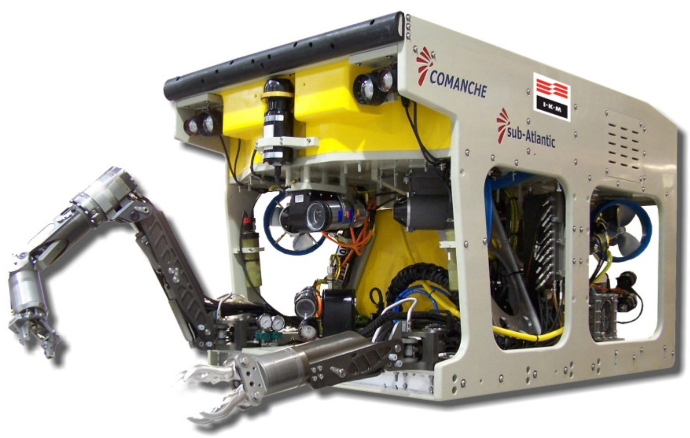

Figure 26: Illustration of a work class IKM subsea ROV (UID) with the basic left grip manipulator and advanced right handed manipulator

Figure 27: Controller for the right hand manipulator showed in fig.(26)

3 Method

In this section the methods used to investigate the project thesis statement is presented. This includes both theoretical and practical approaches to the problem. These will be described in this section together with the used software setup to produce the implemented results. A detailed description of the practical implementation is also given in section 4.

3.1 Approach to the problem

Based on the task description and scope of this project it is desired to use this project to get familiarized with the introduced topics and technology. The reason for this is the desired to continue the work with this topic in a master thesis. To get a feeling for the potential of both augmented reality and computer vision, some kind of practical implementation would be beneficial. In order to achieve this and to answer the task description a thorough literature search is used. In addition, a visit to Oceaneering's headquarters functions as a great source of information to this project.

The implementation is achieved by separating the computer vision back-end from the more font-end based augmented reality implementation. This makes for an understanding of how these technologies functions on their own. In further work, combining these two together in a single seamless system implementation is fundamental. The goal is that the implementation presented in section (4) will give valuable experience to prepare for a more complex implementation in the continuation. To make sure this experience is collected, one might look at the implementations in this project as basic, but it is desired to fully understand the mechanics of the basic implementation rather than being confused by some complex and over complicated implementation.

3.2 Literature search

An initial wide literature search is used to initiate the investigation of the project statement and to discover different approaches to the problem. Because of the wide spectrum of possible approaches, a more specific approach is chosen (section (4)) and thus narrows the literature search. When searching for literature, several references is used to validate the credibility of the once used. These searches is done mainly with the help of NTNU's library search engine, ScienceDirect and Google searches. The superior topics search for are listed bellow:

- Augmented reality
- Computer Vision
- Image processing
- Programming tutorials within Python and C#
- Subsea fields and structures

3.3 Participation on SEAVENTION project meeting

The subject of this thesis is part of a bigger project named SEAVENTON. This is a cooperation between Innovation Norway, scientists at NTNU and SITFEN, and industry partners TechnipFMC, Equinor, Oceaneering and IKM. In mid-September, there was a project meeting at Oceaneering's headquarters in Stavanger. At this meeting different members of the project held presentations on various topics. The team also got a tour of Oceaneering's headquarters and UID-workshop. The tour included a visit to their onshore "Mission control center" where the team was presented for live UID operations. This meeting ended up being a great source of information for the continuation of this project. The opportunity to be briefed on the state of UID and ROV operations from the industry leaders within this field was helpful in understanding which areas could benefit the most from new technological innovations.

3.4 Method used for object segmentation

To obtain practical results in this report, several software programs, programming languages and libraries are utilized. Based on previous knowledge in python, the OpenCV library is a natural place to start to get some initial results. With the help of anaconda, both OpenCV and the Spyder compiler are installed. Through these tools image processing is done to do object segmenting in images. In order to learn and get familiar with OpenCV, several online guides and instructions is used as the source for useful knowledge and experience. The results generated from these tools are given in section (5.2) and a detailed description of the implementation is found in section (4.2)

3.5 Method used for AR implementation

For the AR implementation the Vuforia software developer kit is installed together with the Unity engine. Unity's primary programming language is C#, however, there are methods and extensions that gives the opportunity to use python API. After some research, the impression is that these methods affect the frame rate of the end product and because it is not the primary language it might not be as robust solution as C#. Another possible direction is to utilize OpenGl's cross-platform and bind it together with Python. With the help of OpenCV, AR implementation is possible with this python setup. Since Unity is using OpenGL to render the scenes the understanding is that Unity would provide more tools to faster end up with an end result compared to the lower level method using OpenGL. This together with the scope where the desire is to get introduced to one of the C programming languages are the deciding factors when choosing to go down the path of Unity and C#. The great community and forums around Unity, a crash course in Unity held by Hackerspace-NTNU, and online tutorials in both Unity and C# creates the foundation for the knowledge used in this implementation. In section (4.3) the implementation is described in detail.

4 Practical implementations

To aid operators during UID operations, with the approach of this problem in mind, some developed software system is needed. However, such an complex system is not within the scope of this project. To get a deeper understanding of augmented reality and computer vision, this technology is instead used in two different less complex implementations. These two implementations work as a bare minimum to test the realization potential of the proposed augmented operator interface given in fig.(32). With this in mind, this section aims to describe the process of producing the results presented in section (5).

4.1 Design of augmented operator interface

From information and talks with UID operators at the SEAVENTION project meeting and initial theoretical research, an augmented operator on screen interface is designed. The resulting design is provided in fig.(32) and further described in section 5.1. The interface is designed using tools in Microsoft Power Point and is therefore only an illustration of a proposed front-end to how an augmented layer could look like on a videostream from a UID. Since this is a illustration of a desired end product, it is used as a very ambitious goal within this project and for further work. Therefore the following work will be aimed towards realizing and validating the possibility and potential of such an augmented operator interface.

4.2 Detection of UID interface components on subsea structure

Through the gained knowledge from the computer vision literature search and tutorials and guides for python OpenCV, a script for object segmentation is developed. This script is to identify orange colored objects on subsea structures. This is done through a color filter which detects and mark colors with properties within the set filter thresholds. The thresholds is set by a lower and upper band, represented by two arrays, min orange and max orange in the script (found in the enclosure). Each row of the arrays represent values for hue, saturation and value respectively. Due to different brightness levels in the reference image and real world operations, the filter value is tuned to obtain the results given in section 5.2. In addition to the HSV-filter, some image transformations is used to obtain satisfying results.

The first function in the script is a pure scaling function for fast and easy presentation of the resulting processed image in the console, in addition to easier chose kernel and file size during tuning. Secondly a function to convert our segmented layer (mask) from gray to RGB is made. This is done because the soon to be applied transformations are done in gray scale. The mask is chosen to be RGB because the result is desired to be presented as a layer on-top of our colored image. Weights is also added to tune the brightness of the mask and original image.

For the actual segmentation function, the image is first converted from BGR to RBG as the default format of OpenCV is BGR. Next up is the smoothing of the image, which is done with a Gaussian blur filter. This filter is easily implemented through the OpenCV library. It asks for our initial image, a kernel size, and a standard deviation value in x and y direction. The smoothing of the image highlights the gradual change in colors and because we are going to filter based on HSV values, the filtered image is converted from RGB to HSV. Following up is the implementation of the upper and lower bands of the color filter. These are added with variables for convenience during tuning. Finally the mask containing the color filtered elements is defined. This is done through an in range function who adds all the elements within the color filter threshold of the smoothed image to a new layer. Lastly our segmented layer is ran through a pair of Morphological Transformations. First a closed method where dilation followed by erosion is executed to close potential small black holes in the segmented object. Then an open method is used to remove noise in the layer through the opposite operation, erosion followed by dilation. At the end of the function the combined segmented image is made through overlaying the mask on top of the original image, resulting in a new output image.

```
#Tuning parameters
green = (0,255,0)
kernsize_x = 15
kernsize_y = 15
low_red = 5
low_green = 120
low_blue = 100
high_red = 20
high_green = 256
high_blue = 256
s_kernx = 1
s_kerny = 1
scale_val = 1000
```

Figure 28: Variables to be tuned during the tuning process

Tuning process

The tuning of the object segmentation filter is mainly based on trail and error and different tuning parameters is tuned. These are all listed in fig.(28), and the original image the tuning is performed on is shown in fig.(34), while the segmented image is found in fig.(33). First out is scale_val who control the dimension of the output image. This value is tuned to control the produced file size and run time. Since the width of the image is determined through scale_val, a method is tried where the kernel size is determined by the chosen scale. A symmetrical kernel of size $\frac{scale\_val}{100}$ is tested (given that it is ode). To check if this is a good kernel size fitting method, the results are compared to smaller kernels of size 3x3 and 5x5. It was fast seen that a kernel size of 3x3 performed the best for different scales of the image. Thus the tried concept of choosing kernel size based on image scale did not yield any satisfying results. This could also be related to the theory in section 2.2.2 as a small size kernel would mean several more sliding operations giving a more thoroughly calculated result. Therefore the kernel size was ideally found to be 3x3 as this dimension did not yield a big compromise in runtime.

While tuning the sigma values its noticeable how an increased sigma value leads to larger segmented blobs in the mask, making the segmented area exceed the actual orange object area. This observation resulted in a tuned value of 1 for both sigmas. Next up is the tuning of the HSV threshold values. Because it is not trivial to relate these values directly to a given color, a HSV plot is made from the colors in the original image. This plot i shown in fig.(29). It is a bit difficult to separate the orange hue values from the yellow with the full spectrum presented. Therefore a highlighted plot with x-axis limitations is made, see fig.(30). With this plotted information, initial values for max and min HSV arrays is set to $[0, 45, 50]^T$ and $[10, 255, 200]^T$ respectively. It should be noted that the scales introduced for HSV values in section (2.2.1) is represented by values from 0 to 255 in OpenCV. With this filter threshold, almost all the orange objects are segmented, however, many of them are only partly let through the filter. The threshold is therefore increased by 10 for max hue value. With this increase the whole orange objects are detected and let through the filter, but there is some unmarked dots on some of the segmented parts together with filtered parts of the black and white pipes. With further tuning it is found that increasing the min values for saturation and brightness value fixes these problems. Which makes sense as 0 in brightness value gives back and low saturation gives very bright colors (white). From the concluded tuning process it is found that lower orange limit of $[0, 100, 80]^T$ and upper limit of $[20, 255, 255]^T$ yields the most satisfying results.

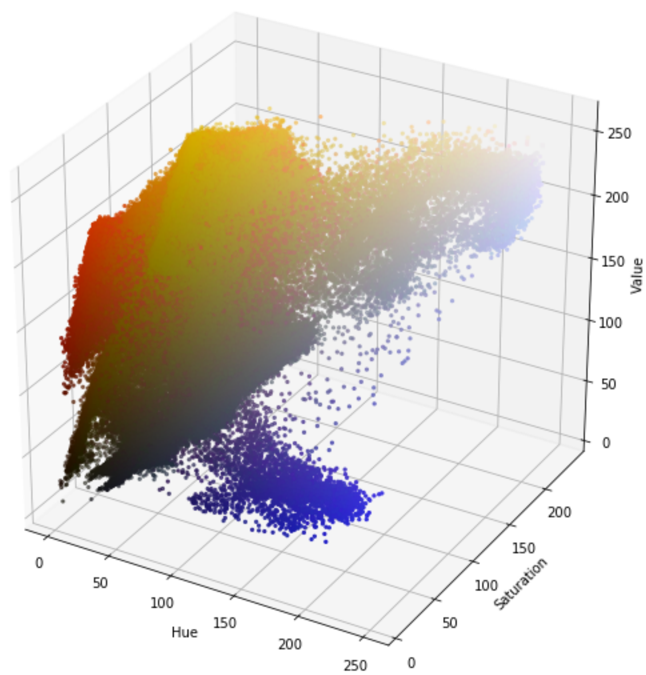

Figure 29: HSV color plot of object segmented image

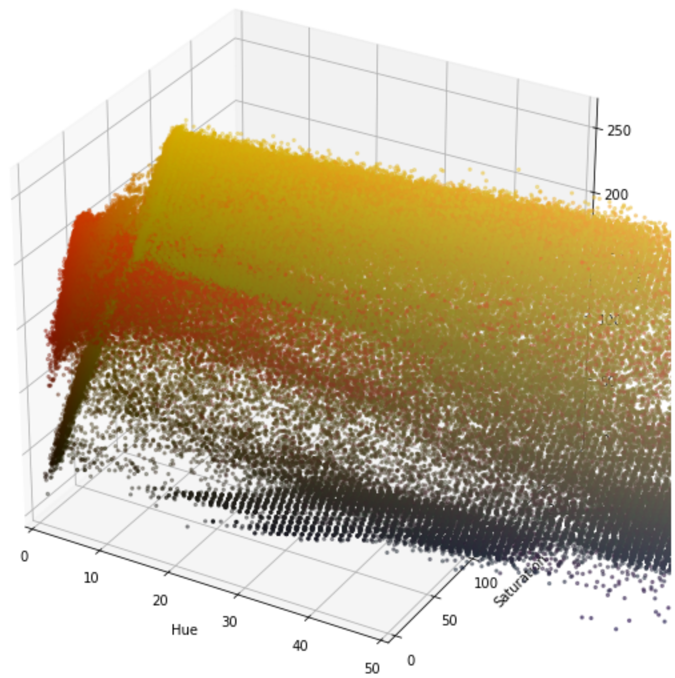

Figure 30: A zoomed HSV color plot of object segmented image

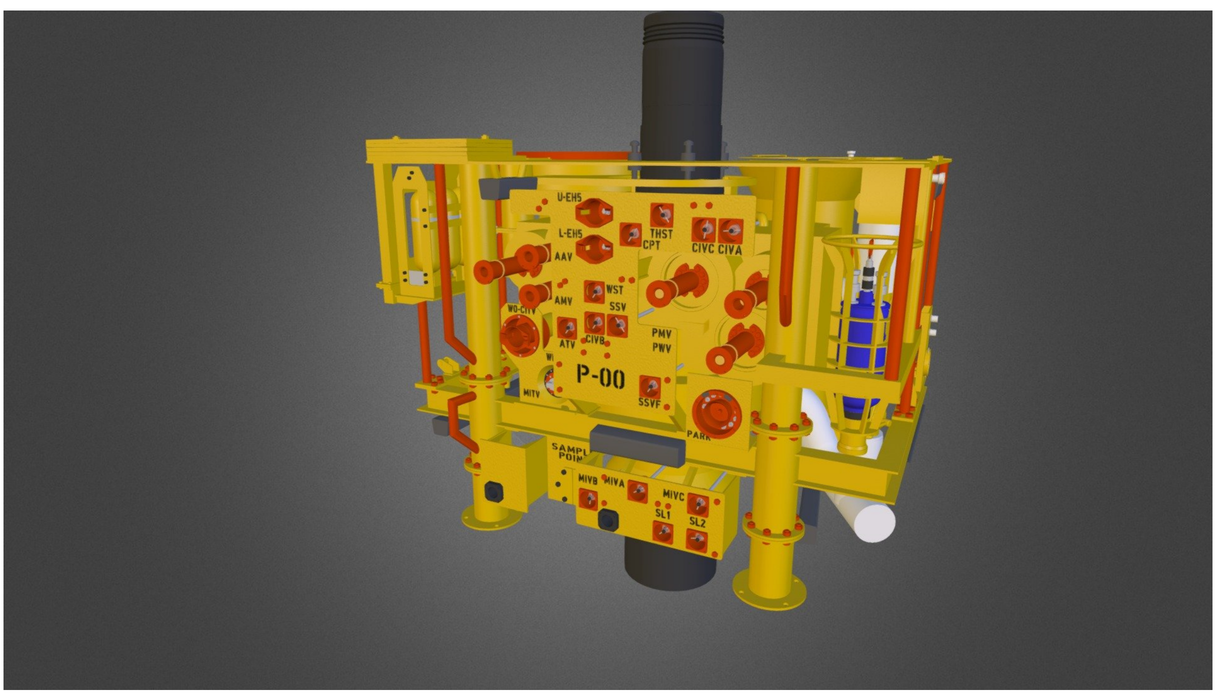

Figure 31: Original x-mass tree model

4.3 Implementation of proposed augmented operator interface design

With the necessary tools (Unity, Vuforia and MS Visual studios) installed and knowledge gained from several introductions and tutorials in C#, it is desired to realize the augmented operator interface concept design. By building scenes in Unity it is possible to display an augmented layer on top of a videostream, just like the interface design in section (5.1). To initiate the developed scene, the localization and tracking capabilities of Vuforia is utilized. By feeding in a reference image to this developer kit in Unity the AR scene is set when the AR camera, also brought to Unity by Vuforia which in this case is the camera on a Microsoft Surface computer, detects this reference image. This is the image on the table in the demonstration given in the enclosure. So, Vuforia, with some inputs, takes care of the tracking and localization of the real world scene as seen from the AR camera. This can be related to the same task as the segmentation of orange components performed in the implementation above. With this now taken care of by the developer kit, the further focus can be directed towards augmenting the interface of the camera scene with a real time updated information layer.

First up, the information layer was equipped with a red 3D Cartesian coordinate system and a joystick. The coordinate system is there to display information about distances from the operated vessel to the desired target in x, y and z direction. To illustrate the working principles of the presented concept, an operable character is added to the scene. This is done by downloading a free 3D model of the internet and implement it in Unity. To make the character move, the joystick is to be used. This is possible with the help of two C# scripts found in the enclosure. The first one relates touch interactions on the screen to movement of the joystick, and outputs this in world coordinates through screen to world coordinate transformation. The second one couples the x and z movement directions of the character to the horizontal and vertical joystick pointers, respectively. With the help of the Unity engine, animations during walking and stand still is also implemented.

So, when the character is movable, the tracking is next up. It is desired to track the character's position and give information abut its whereabouts relative to a reference point. This will simulate the same effect as the target lock concept introduced in section (5.1). To present this information in the augmented layer, a reference cube is added to the scene together with another C# script (found in the enclosure) that calculates the difference in position between the reference point and the character. These values are then stored in strings and are read out to augmented scene in real time. These values are then presented in the scene next to their corresponding arrow in the coordinate system.

Last up, the lines that illustrates the path from the UID to the target in fig.(32) is implemented in this augmented layer. These lines are added with a final C# script (found in the enclosure) which adds lines based on the values presented next to the coordinate system. Because these values are updated in real time, the lines is also updated as the character is moving.

5 Results

In this section the results from the described implementations in section (4) is presented. The augmented operator interface in sect.(5.1) represents a system that utilizes both AR and computer vision to improve the quality of subsea IMR operations. The following two implementations in sect.(5.2) and (5.3) showcases the potential of similar technology, but in a simpler form than what an actual augmented operator interface would require. Following the presentation in this section the results are to be discussed in the next section (6).

5.1 Augmented operator interface

This proposal will add a layer of navigational information through augmented reality on top of the received videostream from the UID-camera system. With this added information, live updated three dimensional distances will appear on the operator's screen and illustrate how far the operated vessel is from the desired target. An Cartesian coordinate system is also added to the interface to give the operator information about the reference directions of the displayed distances. This interface does also allow for a live updated path towards the target and should with complex computer vision systems manage to plan the best collision free path towards this target.

The reference point of the directional information found on the augmented interface is produced by a target lock function. This function would allow the operator to click at a desired destination on the subsea structure. When clicked the augmented distance information should appear and the scene like the one proposed in fig.(5.1) is initiated. This kind of directional information would give operators full information about the depth of field towards their target, even though the video is only displayed on a regular 2D monitor. This operator interface would help out in every IMR operation where depth information and path planing is of beneficial use.

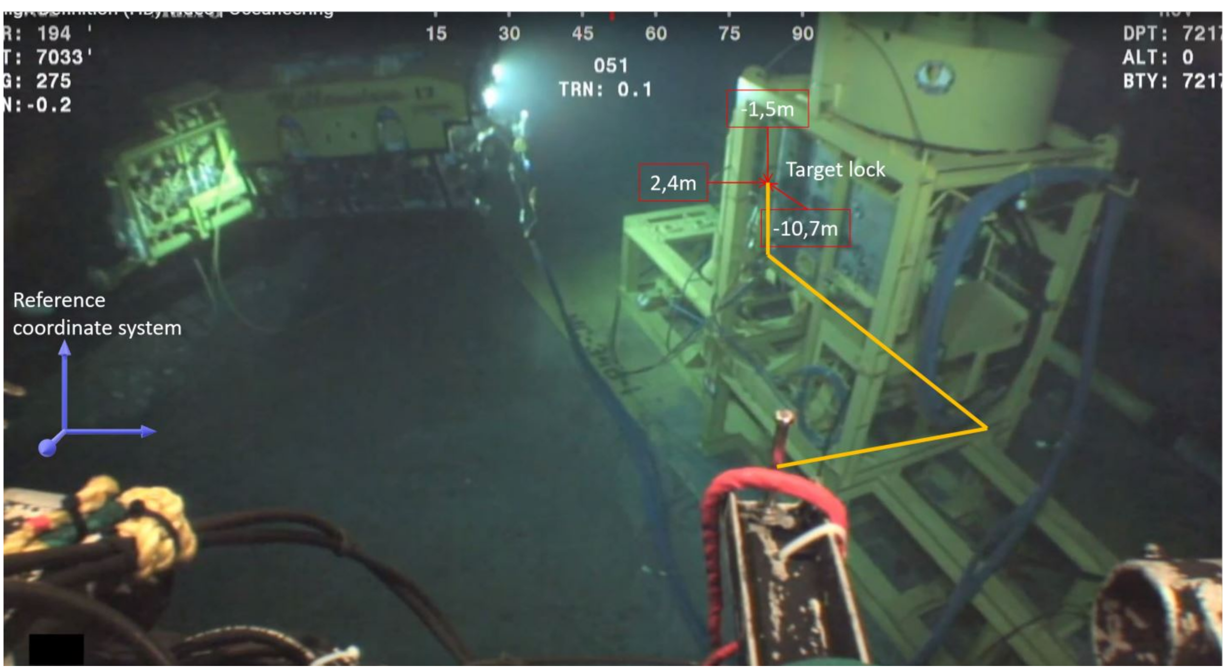

Figure 32: Design proposal for UID operator

5.2 Detection of UID interface components on subsea structures

Through segmentation by thresholding the orange components, thus the components of interest during UID operations, of a subsea X-mass tree is filtered out as individual objects by the computer. These segmented objects are highlighted in a bright mask placed on top of the original input image(fig.(34)) resulting in fig.(33). The resulting figure shows the results from the tuning process presented in section (4.2), and represents a satisfactional compromise between quality of segmentation and resolution on one side, and file size and run time on the other. By comparison one can see that pretty much every orange colored element is highlighted by the mask, but also that some yellow parts inside some of the large orange components is also marked. An example is the yellow area inside the three hose connections on the right side of the center of the X-mass three.

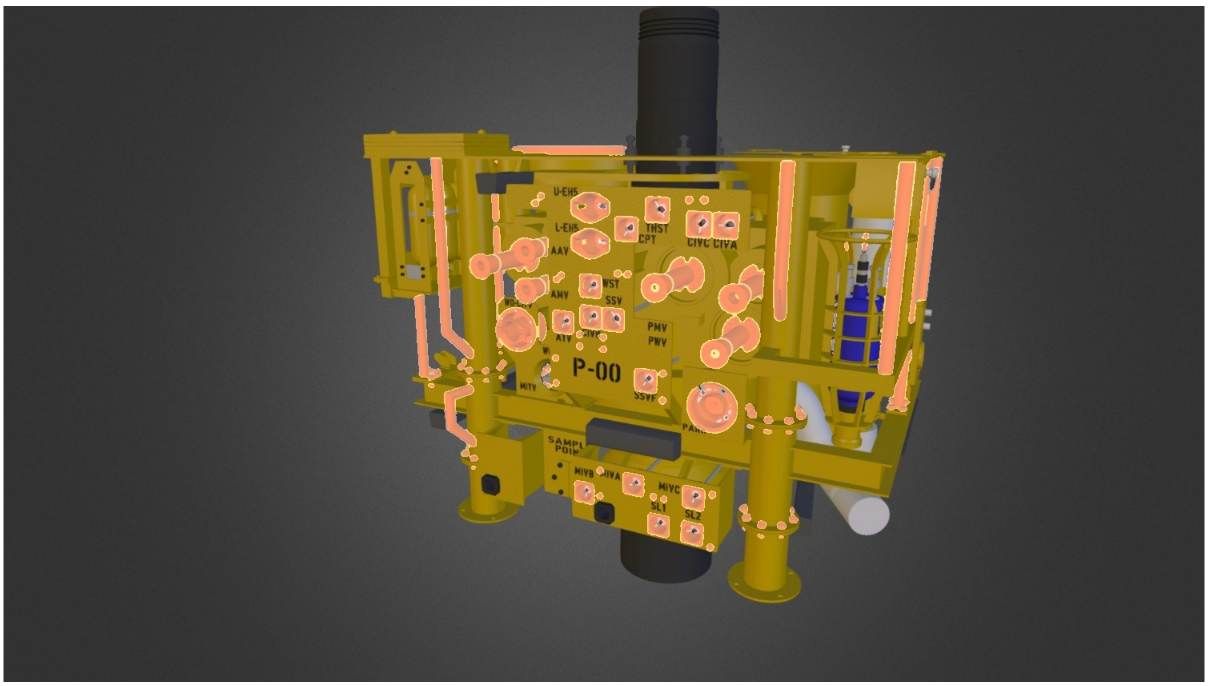

Figure 33: Edge detection based on color separation with scale = 1500, sigma = 1 and kernel size 3x3

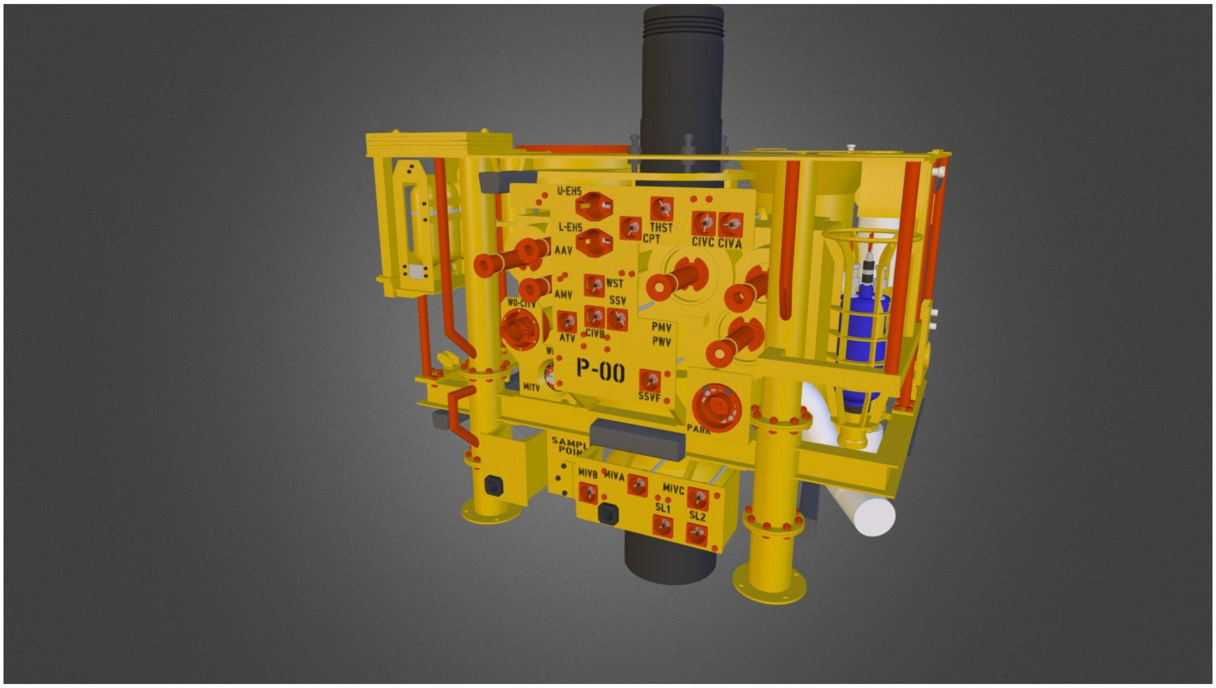

Figure 34: Original x-mass tree model

5.3 Implementation of proposed augmented operator interface design

A video demonstration showing the results from the implemented AR application in Unity is given in the enclosure. In fig.(35) a screenshot from the augmented scene is shown. Compared to the conceptual augmented operator interface in fig.(32) one can see that the navigational information is moved down to the right corner and utilizes the reference coordinate system as directional guidelines. In addition, the implementation have direction lines that shows a potential path from the character to the reference. The target is here the blue cube in the scene. In the video demonstration the y value is constant, because the character is only programmed to move in the 2D x and z plane.

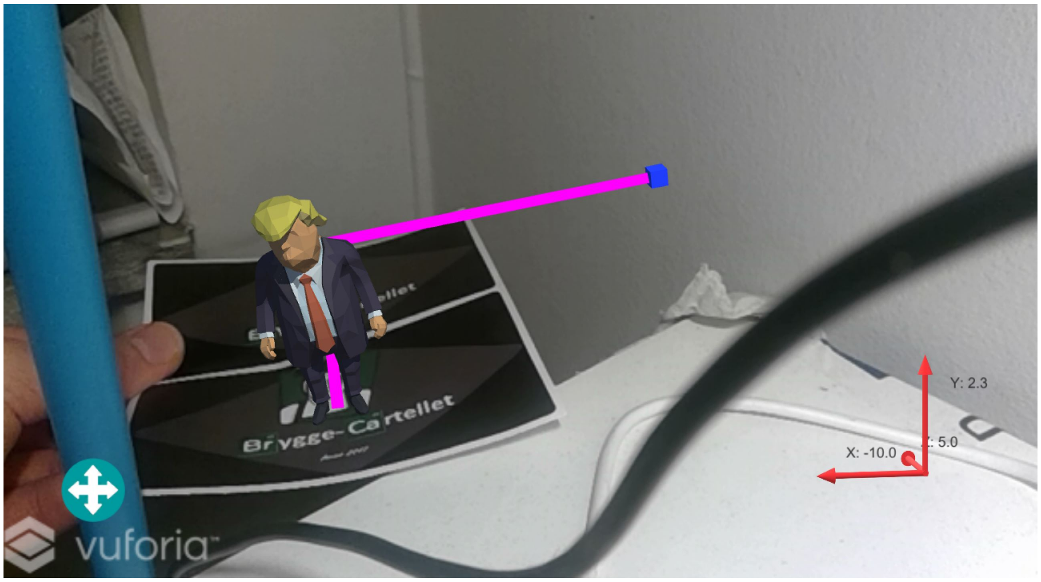

Figure 35: Illustration of the implemented augmented scene based on the augmented operator interface in section (5.1)

6 Discussion

In this section the approach to the problem as well ass the questions rased in the project task section is to be discussed.

6.1 Challenges of bringing computer vision subsea

In recent years visual navigation algorithms like SLAM and VO have been implemented in different environments such as in the air with unmanned areal vehicles, on the roads through autonomous cars and even in security cameras inside buildings. With the desire of increasing the amount of autonomous operations at sea and subsea, a proper navigation algorithm for these harsh environments would go a long way in realizing this desire. There are however, multiple challenges of bringing a vision based position measuring system underwater. A handful of these challenges are to be discussed, with subsea operations in mind, throughout this section.

6.1.1 Ambient light

Underwater things are less visible because of lower levels of natural illumination cased by rapid attenuation of light [58]. This means, an external lighting source is necessary underwater, however, lighting in water exposes fare more reflection than in air. This creates a lot of noise in the image which acts negatively on the contrast. When the contrast is reduced, edge detection will be more difficult. It is therefore desired to reduce and control this negative effect. This same effect is also caused by ambient light, so there is also a desire to control and negate this type of lighting.

There are different approaches to solving these lighting problems, where to most effective, but unfortunately least likely is to turn off the ambient light contribution. However, the attenuation effect of being underwater could actually help us out here in reducing the amounts of ambient light. Another solution more likely, but more expensive is to build a shroud. It's a very effective, but time-consuming method that introduces certain constraints to the system. One could also overwhelm the ambient light contribution with a source of high-power lighting. This is an effective solution but requires more cost and complexity, but seeing as there is a low level of natural illumination subsea, high-power lighting would be a reasonable way to illuminate the subsea scene. The last method up for discussion is to control the ambient light with pass filters. A highly effective method, that only requires a narrow-band source light. With pass filters light can be excluded or preferred based on wavelength. For instance, one can achieve up to 4x reduction of sunlight and 35x reduction of fluorescent light with pass filters [59]. Today's big work-class UIDs are usually equipped with both high-power lighting and reflectors to direct the lighting to a focus point. This would effectively help reduce the un-attenuated ambient light during operations. One should think however, that there is no ambient light during operations on subsea fields. But there are several subsea fields at shallow ground, especially at the Norwegian continental shelf, where the shallowest is found at 65 meters depth in the Tyrim field [60]. So this, together with the fact light might be detected as far as 1000 meters down in the ocean [61], supports the case for awareness of ambient light reduction during subsea operations [62, 63].

6.1.2 External light source, loss of colours and turbidity

Even though lighting is often needed when navigating and executing operations with remotely operated UID's subsea, it might not be as simple as just a flip of a switch when lighting is needed. One of the challenges is the fact that water absorbs light and therefore attenuates it as mentioned. This leads to selective absorption of light in water, where red light, which has the longest wave length and therefore contains the least amount of energy, is absorbed first. That means, colored light at the opposite side of the spectrum, compared to red, can travel longer and therefore increase the lighting before it is absorbed [58]. This means, when choosing type of light for underwater operations the color of it has to be taken into account. With this in mind, divers and ROV operators have found that lighting in temperature color area ranging from about 4 to 7000 kelvin is ideal [64, 65, 66, 67]. When looking at these temperatures in a color temperature scale as the one in fig.(36), one can see that they are found in the area between yellow, green and light blue, which all will have a longer travel range than the red light further down on the scale. On the other hand, this loss of colours can also lead to an environment that appears monotonous. Color gradients can be a valuable information source in computer vision and might therefore limit the possible image processing approaches if the environment turns monotonous. A similar difficulty is faced in locations where the sea floor consist of mainly sand and does therefore not offer much recognizable texture to extract landmarks from. This limits position tracking capabilities for computer vision systems. These mention difficulties are some of the main concerns regarding pushing complex computer vision systems like SLAM to the underwater world. However, a benefit with the subsea field environment is the recognizable structures. These structures could help break the monotony environment that a plain sand seabed could be. Each structure has its owns characteristic features that should be recognizable for computer vision systems. In addition, we must remember that we are operating a maneuverable ROV/UID, so when we are closing up on different structures the close distance should reduce the amount of colour loss in the scene, especially with the help of an external light source.

Today, the light source of choice for ROV operators is LEDs. These are efficient units that of-

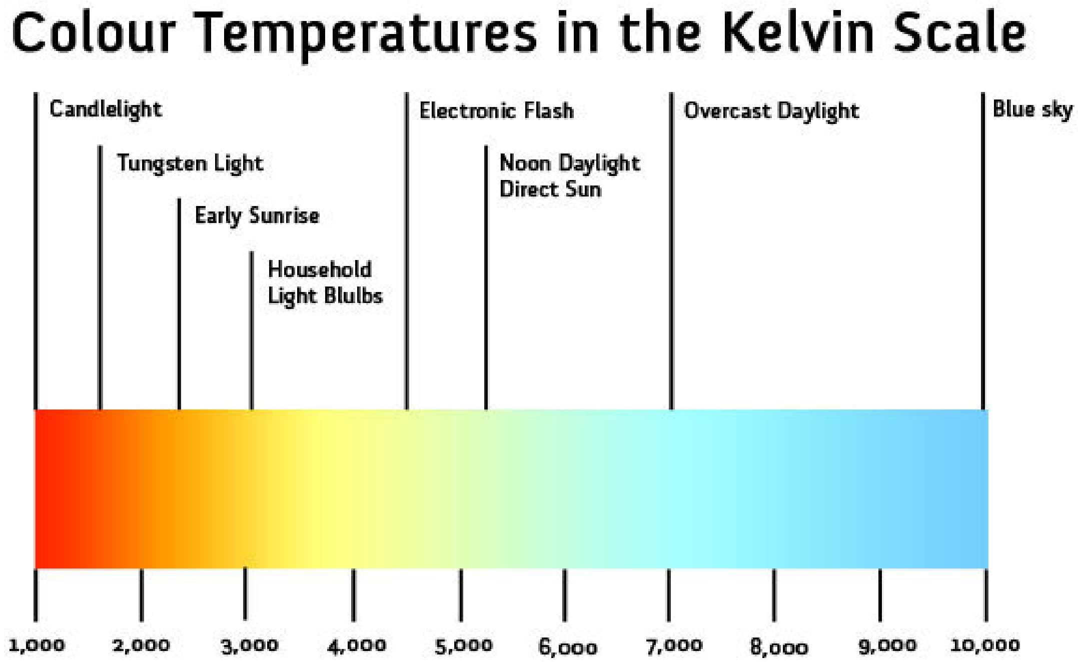

Figure 36: Spreading of light waves in Illumination

fers dimming capabilities and almost instant toggling, features that are used often during deep-sea observations and operations.[63] The main reason for this frequent dimming is the problem encountered by the enormous amount of fish in the ocean. These fish are often attracted by light in the sea and can therefore often hinder and make ROV operations more difficult due to high accumulations of fish over time [68, 69]. A simple work around to reduce the intensity at which fish are attracted is to dim the light based on how much is needed during different operations and tasks. This is though an unfortunate situation for oil operators as any hinder towards efficiency during operations are linked to big additional costs when operation time is increased.

An additional challenge with external lighting subsea is the scattering and absorption of light due to the turbidity of seawater. This often results in a bloom effect in the image, which makes it more difficult for the camera to focus on the scene [70]. A method used to avoid this is range-gating. This is a method where one can open and close the aperture in the camera based on the relative distance between the camera and the scene you actually want to observe with the camera. To avoid the immediate reflections and bloom effect from particles close to the camera, one could send out a pulse of light, close the aperture for some nano-seconds during which the initial reflections are hitting a closed lens and would not be projected in the image. Then the aperture is opened again to receive the light reflections from objects in the depth-scene that actually are of interest [71]. This method is however most accurate in long range imaging and arises greater uncertainties around the aperture closing time when the camera approaches the object of interest. A consequence of these discussed difficulties of subsea imaging have resulted in development of ultra low light subsea cameras. These cameras are developed based on decades of offshore operation experience, and can perform in light conditions as low as 0.0003 lux [72]. With such high low light capabilities, the need for high-power external lighting is drastically reduced, meaning that light intensity can be reduced, something that reduces the accumulation of fish and blooming effects in the received image and therefore also the problems associated with lighting underwater.

As discussed there are several characteristics of seawater that affects the vision of a camera, some more crucial than others. This amplifies the need for a very robust computer vision system for successful underwater implementation. On the bright side, tools such as known dimensions of subsea structures, advanced position systems, sensor data, and computing power on the ROV can help out in making such a robust system. Therefore in future work an interesting aspect is to investigate how utilization of these on board features of the ROV could help reduce the affects or make way for workarounds around the challenges presented in this section. Another interesting aspect would be to investigate the possibilities to train deep learning systems to detect and filter out disturbances such as the ones introduced by the turbidity of seawater. Maybe a way to do this could be through utilization of the presented tool in [73] for turbidity measurement.

6.2 Discussion of implementation

In this subsection a brief discussion of how the implementations where executed is found. The discussion is mainly focusing on reflecting over how the developed systems performed and touches on different approaches for solving the same problem.

6.2.1 Detection of UID interface components on subsea structure

First of all, the greatest limitation of this implementation is the controlled environment in the image with even illumination and no disturbances from turbid seawater. Secondly it is not a videofeed, its simply a high definition still image. This makes for a great thresholding "playground", however, it would not hold up in the real world on a live videofeed in real time. The image processing script is simply not efficient enough to handle the fast changing environment.

So, the script represents a somewhat textbook thresholding technique. You start by blurring your image to dampen the affects of noise in the image and increase the gradual change in pixel intensity values. This makes for easier thresholding conditions compared to a raw image. Then you place your thresholding values (which were represented in HSV values as this seemed like the better choice from section 2.2.1) to segment some desired part of the image and highlight the segmented part with a mask. Finally you want to refine your segmented blobs with morphology transforms. This pretty much sums up the whole segmentation operation, and you are left with a system you can tune to perform according to your desire. It is worth nothing that when the script was ran on different images than the one tuned for in section (4.2), it performed surprisingly good. Some of these images can be found in fig.(37) and (38). As seen from these images, the script does a great job in the artificial one. This is probably because the chosen artificial orange color chosen in the image lays within the threshold. There is though some intense artificial lighting in the image which it seems to handle in a good way. For the latter one, which is a raw image taken from a ROV videofeed found on YouTube [74], the script does an OK job despite the harsh environments. These examples shows that even this somewhat basic script is not way off in an actual operational setting. Which strengths the potential, as there is available technology and processing power to develop tailored systems that should do a better segmentation and detection job. Another interesting observation from fig.(38) is the fact that it is not ROV interface equipment exclusively that are colored with red/orange color. This is worth noting and bring into further work and development as this could lead to misunderstandings and noise for the tracking system if not handled accordingly.

To summarize this subsection, there are far more robust and comprehensive segmentation and object recognition methods than the color filtering method utilized in this project. However, it does give an intended and much needed introduction to image processing through the usage of essential image operations. With the endless tuning opportunities, it is also possible to get a feeling for how the different parameters affects the result and the input image in it self. This thresholding process can be summed up by saying it gives much needed basic knowledge and experience in using image processing for computer vision, which builds a foundation to further build more advanced, versatile and robust systems.

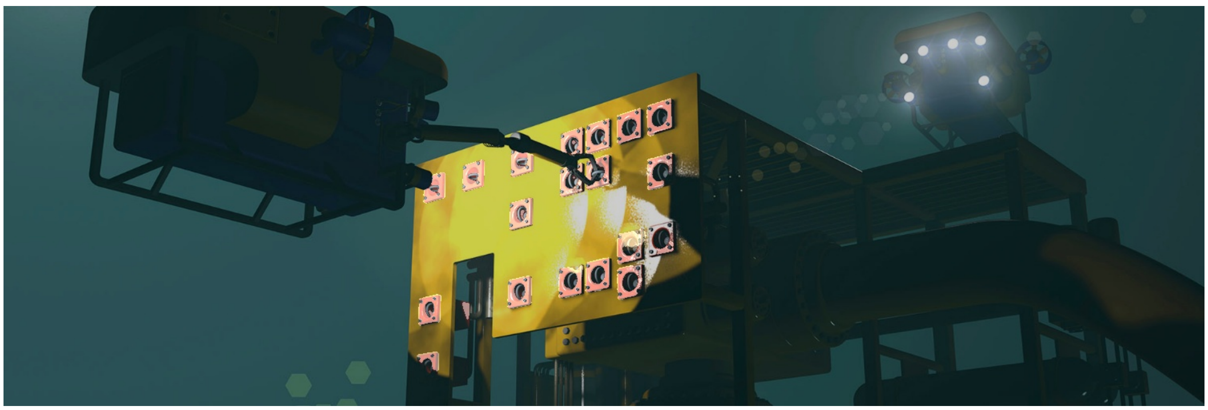

Figure 37: An artificial subsea image segmented with the implemented segmentation by thesholding script

Figure 38: An image take from a videofeed during real subsea operation, segmented with the implemented segmentation by thesholding script

6.2.2 Implementation of proposed augmented operator interface design

In order to get a feeling for augmented reality based programming, a system testing the concept of the the proposed operator interface in section (5.1) is developed. The inclusion of a random 3D model found on the internet and a printed image for scene recognition as well as a joystick to control and navigate the character is placeholders for the actual UID and its videostream. These placeholders was chosen arbitrary based on convenience of implementation.

The choice of using the Unity game engine was based on the desire to use C# as the programming language, in combination with the convenience the Vuforia developer kit introduced with its computer vision object tracking algorithm. Programming such an practical and visual implementation while learning a new programming language help a lot in understanding how the script had to be written in order to achieve the desired functionality. Here Unity is also helping out with its user friendly interface, its big user community and also the great functionality documentation. These are all factors in favor of choosing Unity over other alternatives like OpenGL. Another method discovered later on in the project was using ROS, where this tool would function like the connection between the AR layer and computer vision tracking. Using ROS would maybe yield a better implementation considering that this connection is needed in further work, but on the other hand this kind of implementation might have been more complex and thus not ending in a working result within the deadline.

6.3 Discussion of results

In the following subsection the results from section (5) will be discussed.

6.3.1 Augmented operator interface

From the experience and conversations at Oceaneering's live UID operation at their mission control center in Stavanger, it became easier to picture the struggles and difficulties faced by the UID operators. Only the dark conditions and a subsea environment very different from the above sea-level surroundings can introduce complications on its own. However, adding the fact that operators often use 2D monitors to navigate in this harsh 3D environment makes the process more time consuming and potentially frustrating. Because one of the UID manipulators is solely used to grip onto structures during intervention and maintenance, griping tasks can be a sizable part of operations. When operators don't have any depth of field information, such basic operations can be more difficult than imagined. My feeling after leaving the mission control center was a mix of frustration and restlessness. This feeling came when watching the operator use several attempts to grab onto the handlebar placed on the ROV-interface panel of the subsea structure and what seemed to be a bit chaotic process of organizing and finding the right hoses at the seabed. In addition, I got the opportunity to ask the operators some questions after the operation. They gave me the impression that some kind of depth of field information would help them immensely. Today they have the opportunity to overlay a 3D model onto the 2D videofeed from the UID-camera, this helps to some degree, but it can also interfere with the operator's field of view.

Based on this experience and demo, a proposal for improved operator interface for better awareness during operations is developed. The resulted proposal is illustrated in fig.(32). This interface is dependent on a working method for target/destination lock to set a reference. For this, a robust tracking and localization computer vision system is needed. This type of system is achieved with implementation of working a SLAM or VO systems or the simple segmentation implementation in section (5.2). However, the immediate challenges of implementing such systems is discussed in section (6.1).

After conversations with the UID operators them self, there is no doubt that an improved operator display has the potential to reduce the time of operations. As the approach of this project is through AR, the more realistic and technologically available implementation of AR seems to be through an AR layer on a videofeed displayed for the operator on a 2D screen. Implementation through AR headsets such as Microsoft HoloLens has also been investigated. The potential for operators to actually view the UID footage in 3D with multiple layers of augmented 3D information can introduce great benefits for the operators. However, today these headsets have a shallow field of view which reduces the operator's side view and can, therefore, results in sudden collisions between UIDs and subsea structures. Also, the fact that these operations can take several hours makes drifting in the image a potential problem. Drifting in the image could make the pilots "sea-sick" which by no means are ideal during operations. As this approach to the problem focuses on the operator's general view and awareness feeling during operations with information presented reliably and organized, the on display augmented reality turns out to be the simplest with best benefits within the scope of this project.

6.3.2 Detection of UID interface components on subsea structures

For the proposed system design to work during real operations, there have to be a system to detect and segment different objects. Because of the limitations of this paper, it was decided to demonstrate the possibilities for segmentation through color thresholding, rather other more demanding solutions such as tracking through systems like visual odometry and SLAM. With the method and results presented in section (4.2) and (5.2) respectively, it was shown how, through some image processing, the computer could segment certain geometry from the rest. This segmentation is fundamental if a target lock function is to be implemented in a complete system. Fig.(33) shows the result from the tuning process of the filters, and represents a compromise between file size and run-time on one side and accuracy on the other side. Clearly, a significant part of the orange structures is highlighted, but due to restrictions in accuracy some of the outlining geometry is also marked. However, the outlaying marking seems to be restricted to only some pixels, the segmentation of this image should therefore be sufficient in a target lock situation (under these very limited conditions). Also, if needed one could also decrease the sigma (variance) in the Gauss blur to increase accuracy at the expense of longer run time. The difficulty is nonetheless implementing this type of segmentation in a videostream, where the segmentation continuously have to update and recalculate as the camera host is moving. Adding to this is the difficulties discussed in section (6.1). These challenges represents uncertainties that could drastically change the conditions from the ones represented by the pictures this segmentation filter where tuned on. This raises the question about the robustness of the segmentation method for use in demanding subsea environments. Fore a more robust and comprehensive solution one could look towards tracking solutions involving visual odometry or SLAM, however, even these systems will have hard times performing in harsh subsea environments. For such an implementation to work, the algorithms and general software has to be tweaked to increase performance under water. On the hardware side, interesting developments utilizing time of flight technology to increase the visual range in turbid waters, could help solve some of the problems related to image quality robustness [75]. Nevertheless, there are ROV operations executed every day on different subsea oil and gas fields. Operations performed by ROV operators who only have the videofeed as reference when maneuvering the ROV. If this information is sufficient for the operators to make decisions, chances are, the same videofeed (maybe in stereo) could be sufficient for a computer to understand the operating environments. An interesting though here is to implement some kind of deep learning network that over time could become specialized in understanding and adapting to the video information from subsea ROV's in different water visibility conditions.

To summarize, there are still many challenges to be faced before we can get underwater imaging results as good as the results above sea-level. However, with clever solutions and workarounds, computer vision solutions should definitely have the potential to track and identify subsea objects in the foreseeable future.

6.3.3 Implementation of proposed augmented operator interface design

To realize the concept-idea of information based augmented layers built upon a 2D videostream, the tools mentioned in section (4.3) was used. Since the detection method given in section (5.2) is limited to only still images and not a videofeed, the Vuforia extension to Unity was used. This SDK is not open source, so its tracking and localization algorithm is not available, however, since it takes a input images and detect it in the real world, it must do some kind of feature detection of the image and then uses this obtained information to compare these features with the information coming from the camera. Based on this feature comparison and edge detection, the input picture could be detected. A similar result could be obtained by implementing an algorithm like SIFT in OpenCV. This was not done due to the convenience of the Vuforia SDK in Unity.

So, with the tracking and localization processes handled by Vuforia and understood through the theory in section (2.2.5) and method in section (5.2) the focus could be directed towards making the augmented layer. In Unity this is done though scenes which is placed on top of the displayed videostream. The scene was designed based on the results presented in section (5.1), with some modifications. Instead of having the dimensions presented at the target, this information was moved to the left corner, to not block the line of sight towards the target. It was initially desired to have the information as close to the target as possible, because this is usually where the operators vision is focused. However, this could in many situations compromise the sight around the target destination, so the information was moved down into the corner. This introduce a small compromise for the operator as this persons vision focus area are expanded to the corners. To work around this and the potential problem of this information in the corner, as well as any other information in the augmented layers blocking areas of interest in the image, a solution could be to implement easy drag and drop functions to move the information about in the layers. Also a transparency bar to adjust transparency of each individual information box could be very use full, in combination with a function to toggle on and off different information boxes. Such an function could especially be useful on the path lines, as this information has the greatest potential to block features in the videostream. It is also worth mention the limitation of the coordinate system in the implementation presented in section (6.3.3) compared to what is needed during operations, which is the facts that it is fixed in the scene. Therefore, when moving moving the camera, or UID in the operation case, the axis is not moved. A much needed improvement is therefore to make a dynamic coordinate system that updates the axis as the camera perspective is changed.

In this implementation there is only on layer, but added functionality when for example scrolling the mouse-wheel to access different layer should be investigated in further work. A method to implement a self designed localization and tracking program as a back-end to the front-end implementation in this section should also be further investigated.

6.4 Camera setup

In the case of stationary imaging, a monocular camera setup could not be utilized by a subsea ROV for depth information during operations. However, the ROV is moving and by estimating the relative pose between each incoming camera frame from the videofeed, we could utilize the same epipolar geometry as stereo vision does (section (2.2)). In this case, the baseline would be the relative pose between each frame. By setting this up as a visual odometry (VO) or monocular SLAM system, tracking of the target lock objects could be possible, but there is a lot of uncertainties such as the discussed underwater disturbances and loss of tracking. This is however, not part of the scope of this paper and will be a interesting field to further investigate in further work.

For the stereo case, we can initiate with a depth of field view, without the need of relative motion. Our baseline is know and the cameras are calibrated together to get the camera matrices. For the sole purpose of tracking 3D geometry of objects to enable the mentioned target lock function, stereo vision seems to introduce the necessary tools, as it manages to get 3D metric scale information from the scene by triangulation. Therefore the complexity of the computer vision system is reduced, as we are not reliant of more advanced tracking and mappings systems such as VO and SLAM. In addition, the oil and gas industry is characterized by its high costs, so the additional cost of an additional camera sensor for this kind of setup should be reasonable. Seeing as the ROV's are big constructions, there should also be enough room to fit two sensors. Within this reasoning, the stereo setup stands out as the more robust and actually also simplest from a software perspective. However, if we think a bit further than the scope of the project, some simultaneous tracking and mapping system is more future-orientated. Especially when oil and gas operators such as Equinor is aiming towards autonomous ROV operations [76]. Therefore, as mention, these system introduce a very intriguing field of study because they open up far more possibilities than the scope of this paper.

6.5 Possibilities regarding implementation of AR and computer vision in IMR operations

The value from adding augmented reality to subsea IMR operations is not limited to convenience and better working conditions for the operators, it also has the potential to reduce the operation time, improve collision control and increased customization of easy accessible information during different types of operations. Operation costs in the oil & gas industry is very high due to the immense potential earnings lost during production stop, with the accumulated expenses of ship and UID rent and several other smaller factors like salary. Therefore would a reduction in operation time potentially result in great earnings for offshore operators. Still, it would be important that the added additional equipment (if necessary) to the UID would not lead to exaggerated expenses.

If a type of SLAM is to be used as the computer vision part, collision control could be implemented as an additional feature with its own AR layer of information. When first getting a robust computer vision system that works seamlessly and manages to understand its surroundings, could lead the way for a whole lot of possible new features, the greatest being completely autonomous operations.

This is something the industry is pushing towards as autonomous IMR operations could be executed whenever needed at short notice and following pre-programmed schedules at all hours 365 days of the year [76]. So implementation of improved displayed information on a 2D monitor for the UID operators, with its necessary back-end software, could potentially lead the way for other exciting milestones towards an autonomous subsea production facility. However, even though these drones one day could operate by them self, there will most likely be use for operators or humans that oversees the operation though a monitor with the videostream form the UID. Therefore, layers of augmented information presented on these displays would be just as useful as during operations executed remotely by operators. And the information displayed could also be customized to the relevant format for such a type of active monitoring.

7 Conclusion and further work

It is found that UID operators yearn some kind of depth of field information during their operations. With the augmented interface presented in this paper, this information could become conveniently available for the operators. Information that is not only limited to one specific IMR operation but rather a wide spectrum of different IMR and also other UID operations where the depth of field information is desired. To equip operators with this kind of information means the potential to reduce the time of operations through nullifying time wasted due to the inconvenience of navigating a UID in a 3D world with only 2D information available. The technology behind the suggested concept was separated into two different limited implementations to document the potential to build the suggested system. It is also found that an IMR operation contains many griping tasks, thus a target lock method was developed to select the desired destination and get 3D navigational information towards that target. However, this augmented information does not come without its sets of challenges. It's problematic to bring computer vision systems underwater because of the drastically changed conditions compared to air. But with technological improvements, such as more advanced sensors, and new software contributions like novel SLAM methods, the toolkit to overcome and work around these challenges are growing.

These technological tools should be further investigated and modified in order to tailor fit them to subsea operations. Methods to utilize information like 3D models of excising subsea infrastructure at the seabed to aid computer vision systems should look into. The concept of training some kind of deep learning algorithm to help the computer vision system cope with the harsh environments is also very interesting. And of course, a technique for object tracking is needed to make the target lock function work. Here it should be decided if the computer vision system should only be limited to tracking, or if it should also contain localization and mapping. With the latter functionality expansion possibilities becomes greater and drift awareness through loop closure would also be possible, but the system complexity is also increased. It would also be beneficial to go deeper into the subsea aspect, looking at what sensor information is currently available at the UIDs for potential sensor fusion purposes. Of course, a method to merge the computer vision system with the AR graphics so they work seamlessly together as a front- and back-end is another thing that is essential for the proposed system to work, and thus should be looked into accordingly.

References

[1] Steve Krar. *The IMPORTANCE of MAINTENANCE*. 2016. URL: https://www.automationmag.com/images/stories/LWTech-files/94%5C%20Intelligent%5C%20Systems.pdf (visited on 12/12/2018).

[2] Andrew Medal. *How Technology Is Saving the Oil and Gas Industry*. 2017. URL: https://www.inc.com/andrew-medal/how-technology-is-saving-oil-gas-industry.html (visited on 12/12/2018).

[3] AkerBp. *Aker BP vil digitalisere og innovere norsk sokkel*. 2018. URL: https://www.akerbp.com/aker-bp-vil-digitalisere-og-innovere-norsk-sokkel/ (visited on 12/12/2018).

[4] oxford dictionary. *Augment*. 2018. URL: https://en.oxforddictionaries.com/definition/augment (visited on 08/30/2018).

[5] Margaret Rouse. *Augmented reality (AR)*. 2015. URL: https://whatis.techtarget.com/definition/augmented-reality-AR (visited on 08/30/2018).

[6] Jolene Fiarchuk. *What is AR?* 2016. URL: http://mobileedar.weebly.com/what-is-ar.html (visited on 08/30/2018).

[7] Wikipedia. *Augmented reality*. 2018. URL: https://en.wikipedia.org/wiki/Augmented_reality (visited on 08/30/2018).

[8] Austin Berner. *Augmented Reality – The Past, The Present and The Future*. 2018. URL: https://www.interaction-design.org/literature/article/augmented-reality-the-past-the-present-and-the-future (visited on 11/30/2018).

[9] Coursera. *The history of augmented reality*. 2015. URL: https://www.coursera.org/lecture/ar/the-history-of-augmented-reality-OnMis (visited on 11/30/2018).

[10] Michael Isberto. *The history of augmented reality*. 2018. URL: https://www.colocationamerica.com/blog/history-of-augmented-reality (visited on 11/30/2018).

[11] Neil S. *Current State of Augmented Reality*. 2018. URL: https://medium.com/@bizg33k_88988/current-state-of-augmented-reality-e727c268620a (visited on 11/30/2018).

[12] Scottie Gardonio. *The Current State of Augmented and Virtual Reality*. 2017. URL: https://www.iotforall.com/current-state-augmented-virtual-reality/ (visited on 11/30/2018).

[13] Hemendra Singh. *What is Augmented Reality? – Types of AR and Future of Augmented Reality*. 2018. URL: https://dev.to/theninehertz/what-is-augmented-reality--types-of-ar-and-future-of-augmented-reality--1en0 (visited on 08/25/2018).

[14] Reality. *The Ultimate Guide to Understanding Augmented Reality (AR) Technology*. 2016. URL: https://www.realitytechnologies.com/augmented-reality/ (visited on 08/25/2018).

[15] Dr. Gholamreza Anbarjafarl. *Digital Image processing*. 2014. URL: https://sisu.ut.ee/imageprocessing/book/1 (visited on 11/24/2018).

[16] Nishant Kumar. *Digitial image processing basics*. 2015. URL: https://www.geeksforgeeks.org/digital-image-processing-basics/ (visited on 11/24/2018).

[17] Wayne Fulton. *A few scanning tips*. 1997. URL: https://www.scantips.com/basics1d.html (visited on 11/24/2018).

[18] kaurdavinder. *Difference between RGB and BGR*. 2015. URL: https://lifearoundkaur.wordpress.com/2015/08/04/difference-between-rgb-and-bgr/ (visited on 11/24/2018).

[19] Jacci Howard Bear. *The HSV Color Model in Graphic Design*. 2018. URL: https://www.lifewire.com/what-is-hsv-in-design-1078068 (visited on 11/24/2018).

[20] University of Paris. *Kernel (image processing)*. 2018. URL: http://www.ai.univ-paris8.fr/~amsi/PGSI1819S1/docs/wiki_matrices_convolution.pdf (visited on 11/24/2018).

[21] Dr. Mike Pound. *How Blurs & Filters Work - Computerphile*. 2015. URL: https://www.youtube.com/watch?v=C_zFhWdM4ic (visited on 11/24/2018).

[22] Brian G. Schunck Ramesh Jain Rangachar Kasturi. *Machine Vision*. McGraw-Hill, 1995.

[23] Wilfrid Laurier University. *Lecture 6 Sharpening Filters*. 2016. URL: https://bohr.wlu.ca/hfan/cp467/12/notes/cp467_12_lecture6_sharpening.pdf (visited on 10/30/2018).

[24] Diffraction limited. *Low-Pass Filtering (Blurring)*. 2016. URL: https://diffractionlimited.com/help/maximdl/Low-Pass_Filtering.htm (visited on 10/30/2018).

[25] OpenCV-Srf. *Low-Pass Filtering (Blurring)*. 2018. URL: https://www.opencv-srf.com/2018/03/gaussian-blur.html (visited on 10/30/2018).

[26] OpenCV. *Morphological Transformations*. 2015. URL: https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html (visited on 09/20/2018).

[27] Udacity. *Structuring Element*. 2015. URL: https://www.youtube.com/watch?v=9lqH5XLI-V4 (visited on 09/20/2018).

[28] Wikipedia. *Thresholding (image processing)*. 2018. URL: https://en.wikipedia.org/wiki/Thresholding_(image_processing) (visited on 11/30/2018).

[29] Franklin Veaux. *What is image masking?* 2015. URL: https://www.quora.com/What-is-image-masking-1 (visited on 11/30/2018).

[30] xinapse. *Masking Images*. 2008. URL: http://www.xinapse.com/Manual/masking.html (visited on 11/30/2018).

[31] Jonathan Hui. *Image segmentation with Mask R-CNN*. 2018. URL: https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272 (visited on 11/30/2018).

[32] William Hoff. *EENG 512 - Computer Vision*. 2012. URL: https://www.youtube.com/watch?v=A-MzzCasnuM&list=PL4B3F8D4A5CAD8DA3&index=2 (visited on 11/30/2018).

[33] Trym Vegard Haavardsholm. *Lecture 5.2 Pose from known 3D points*. 2018. URL: https://www.uio.no/studier/emner/matnat/its/UNIK4690/v16/forelesninger/lecture_5_2_pose_from_known_3d_points.pdf (visited on 11/30/2018).

[34] J. M. M. Montiel Raul Mur-Artal and Juan D. Tardos. *ORB-SLAM: a Versatile and Accurate Monocular SLAM System*. 2015. URL: https://zaguan.unizar.es/record/32799/files/texto_completo.pdf (visited on 09/25/2018).

[35] Jakob Engel, Thomas Schops, and Daniel Cremers. *LSD-SLAM: Large-Scale Direct Monocular SLAM*. 2014. URL: https://webcache.googleusercontent.com/search?q=cache:52wCoQ2uouoJ:https://vision.in.tum.de/_media/spezial/bib/engel14eccv.pdf+&cd=1&hl=en&ct=clnk&gl=no (visited on 10/25/2018).

[36] Davide Scaramuzza Christian Forster Matia Pizzoli. *SVO: Fast Semi-Direct Monocular Visual Odometry*. 2014. URL: https://www.ifi.uzh.ch/dam/jcr:e9b12a61-5dc8-48d2-a5f6-bd8ab49d1986/ICRA14_Forster.pdf (visited on 09/25/2018).

[37] Trym Vegard Haavardsholm. *TTK21 – Introduction to Visual SLAM*. 2018. URL: https://piazza-resources.s3.amazonaws.com/jmgn7bpu252ij/jn1z3jm0ool1z0/TTK21_Lecture1_IntroductiontoVSLAM.pdf (visited on 10/19/2018).

[38] Michael Kaess. *Factor Graphs for Robot Perception*. 2018. URL: https://www.youtube.com/watch?v=Q313pTMAdcM&t=1118s (visited on 10/19/2018).

[39] Python team. *What is Python? Executive Summary*. 2018. URL: https://www.python.org/doc/essays/blurb/ (visited on 10/30/2018).

[40] Vangie Beal. *library*. 2013. URL: https://www.webopedia.com/TERM/L/library.html (visited on 10/30/2018).

[41] Wikipedia. *Python (programming language)*. 2018. URL: https://en.wikipedia.org/wiki/Python_(programming_language) (visited on 10/30/2018).

[42] Techopedia. *C# (C Sharp)*. 2018. URL: https://www.techopedia.com/definition/26272/c-sharp (visited on 10/30/2018).

[43] Wikipedia. *.NET Framework*. 2018. URL: https://en.wikipedia.org/wiki/.NET_Framework (visited on 10/30/2018).

[44] Microsoft team. *Introduction to the C# Language and the .NET Framework*. 2015. URL: https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework (visited on 10/30/2018).

[45] OpenCV team. *About - OpenCV library*. 2018. URL: https://opencv.org/about.html (visited on 10/30/2018).

[46] Wikipedia. *BSD licenses*. 2018. URL: https://en.wikipedia.org/wiki/BSD_licenses (visited on 10/30/2018).

[47] Unity team. *The worlds leading real-time engine*. 2018. URL: https://unity3d.com/unity (visited on 10/30/2018).

[48] Unity team. *Unity (game engine)*. 2018. URL: https://en.wikipedia.org/wiki/Unity_(game_engine) (visited on 10/30/2018).

[49] UTF. *Subsea Technology*. 2014. URL: https://www.utc.no/utf/om_subsea/subsea_technology/ (visited on 09/25/2018).

[50] UTF. *Subsea Technology*. 2014. URL: https://www.utc.no/utf/om_subsea/subsea_history (visited on 09/25/2018).

[51] Offshore-magazine. *Evolution of subsea well system technology*. 2006. URL: https://www.offshore-mag.com/articles/print/volume-66/issue-7/subsea/evolution-of-subsea-well-system-technology.html (visited on 09/25/2018).

[52] Oil Field wiki. *Subsea Structures and Equipment*. 2010. URL: http://www.oilfieldwiki.com/wiki/Subsea_Structures_and_Equipment (visited on 09/25/2018).

[53] AkerSolution. *Subsea Structures*. 2018. URL: https://akersolutions.com/what-we-do/products-and-services/subsea-structures/ (visited on 09/25/2018).

[54] Jan Roald Johannessen Idar Alfred; Jonassen. *A Subsea Operation in Action: A Brief Overview of How IMR Subsea Operations are Organized and Executed*. 2018. URL: https://brage.bibsys.no/xmlui/handle/11250/2502668 (visited on 09/25/2018).

[55] Tor B.Gjersvik Aksel A.Transeth Ingrid B.Utne Ingrid Schjølberg. *Next Generation Subsea Inspection, Maintenance and Repair Operations*. 2016. URL: https://www.sciencedirect.com/science/article/pii/S2405896316320316 (visited on 09/25/2018).

[56] Rigzone. *How Do ROVs Work?* 2008. URL: https://www.rigzone.com/training/insight.asp?insight_id=343&c_id= (visited on 09/25/2018).

[57] Equinor. *How we think UIDs will be used by Statoil in the future*. 2018. URL: https://32zn56499nov99m251h4e9t8-wpengine.netdna-ssl.com/wp-content/uploads/2017/07/ID2017_Att09_Statoil_TomGlancy.pdf (visited on 11/25/2018).

[58] Wikipedia. *Underwater Vision*. 2018. URL: https://en.wikipedia.org/wiki/Underwater_vision (visited on 11/24/2018).

[59] Daryl Martin. *Basic lighting techniques for machine vision*. 2016. URL: https://www.visiononline.org/userAssets/aiaUploads/file/CVP_Beginning-Lighting-for-Machine-Vision_Daryl-Martin.pdf (visited on 09/20/2018).

[60] Drilling ell solutions. *Subsea wells in the Norwegian Continential Shelf*. 2015. URL: https://drillingwellsolutions.wordpress.com/2015/04/05/subsea-wells-in-the-norwegian-continental-shelf/ (visited on 11/24/2018).

[61] U.S. Nation ocean service. *How far does light travel in the ocean*. 2015. URL: https://oceanservice.noaa.gov/facts/light_travel.html (visited on 11/24/2018).

[62] BlueRoamer. *Reflector (photography)*. 2018. URL: https://en.wikipedia.org/wiki/Reflector_(photography) (visited on 11/24/2018).

[63] Larson Electronics. *Lighting Designs, Equipment and Requirements for Underwater ROVs*. 2017. URL: https://www.larsonelectronics.com/a-424-lighting-designs-equipment-and-requirements-for-underwater-rovs.aspx (visited on 11/24/2018).

[64] Seatronics. *Imenco SeaLED 25 Subsea Light*. 2018. URL: https://seatronics-group.com/equipment-sales/video/subsea-lighting/imenco-sealed-25-subsea-light (visited on 11/24/2018).

[65] Remote Ocean Systems. *L300 FULL RANGE DIMMING LED*. 2018. URL: http://www.rosys.com/shop/l300-dc-full-range-dimming-led/ (visited on 11/24/2018).

[66] BlueRobotics. *Lumen Subsea Light for ROV/AUV*. 2018. URL: https://www.bluerobotics.com/store/retired/lumen-light-r1/ (visited on 11/24/2018).

[67] BlueRoamer. *What is best light color*. 2015. URL: https://forum.openrov.com/t/what-is-the-best-light-color/3453 (visited on 11/24/2018).

[68] Anne Jieli Louise Solem. *Analysis of current ROV Operations in the Norwegian Aquaculture*. 2017. URL: https://brage.bibsys.no/xmlui/bitstream/handle/11250/2452264/17869_FULLTEXT.pdf?sequence=1#page=12&zoom=100,0,153 (visited on 11/24/2018).

[69] Lifefrom led. *LED boat lights: Are they good for fishing?* 2014. URL: https://lifeformled.com/blog/led-boat-lights-are-they-good-for-fishing/ (visited on 11/24/2018).

[70] Wikipedia. *Bloom (shader effect)*. 2018. URL: https://en.wikipedia.org/wiki/Bloom_(shader_effect) (visited on 11/24/2018).

[71] Sensors unlimited. *Laser Range-Gated Imaging for Imaging at Long Ranges and Through Obscurants (Fog, Smoke, Mist, Haze, Rain)*. 2014. URL: http://www.sensorsinc.com/applications/military/laser-range-gating (visited on 11/24/2018).

[72] Claxton. *ULTRA LOW LIGHT SUBSEA CAMERAS*. 2018. URL: https://claxtonengineering.com/products-services/camera-systems/ultra-low-light-subsea-cameras/ (visited on 11/24/2018).

[73] Mullins. *A novel image processing-based system for turbidity measurement in domestic and industrial wastewater*. 2018. URL: https://www.ncbi.nlm.nih.gov/pubmed/29528334 (visited on 11/24/2018).

[74] Oceaneering. *ROV High Definition (HD) Video Oceaneering*. 2011. URL: https://www.youtube.com/watch?v=CoOwTOX5dpo (visited on 08/30/2018).

[75] Jens T Thielemann. *EENG 512 - Computer Vision*. 2016. URL: https://www.sintef.no/en/latest-news/eagle-eyed-subsea-camera/ (visited on 10/02/2018).

[76] Marianne Bryhni Asla. *Technology road map for Underwater Intervention Drones (UID^{TM}*. 2018. URL: http://www.subops.no/content/uploads/2018/05/Marianne-Bryhni-Asla-pa-nett.pdf (visited on 10/30/2018).