Estimation of Structural Geometry Sensors and software to generate real-time maps of structural geometry in fish farms

ENGINEERING CYBERNETICS,
SPECIALIZATION PROJECT

By:
Gaute Vatne Nybø

June 24, 2019
Department of Engineering Cybernetics
Norwegian University of Science and Technology

Summary

This report is based on the artifex project. It discuss and poses an approach to an image processing subsystem . The subsystem would help the autonomous operation in and around the fish farm by sending out information about the structures in the water. Recognising the geometries in the water is a step towards fully automatising the operations on the locations. Getting this drone based vision system to work will be key to making it easier to oversee the production sights and would allow avoidance of coalition and tethering for every vehicle involved in such operations.

In the task description of the specialisation project it can be found that this report aims to reviewing literature and identify relevant algorithms for mapping the safe zones during operation.

The work has resulted in several different experimental modules used for detection and visualisation. The visualisation part of the project description was not originally included in the problem description as stated above, but has been added since, as its has been seen as a valuable addition.

	Revived	Tested	Implemented
Image pre processing	x	x	x
Height based cage detection	x	x	x
DNN based cage segmentation	x	x	x
Experimental SFM module	x	x	
Experimental mooring buoy detection	x	x	
Serialisation module	x		
WebGL GUI for output data	x	x	x
Sensor fusion IMU/GPS	x	x	

Preface

The report for the mandatory course TTK4551, taken during the fall 2018. The course counts for 7.5 credit points of the two-year MITK masters program at Norwegian University of Science and Technology. The project has been an enjoyable experience, thus offering a lot of challenges along the way. Working on a project that has an application in the real wold has been both inspiring and motivational .

I would like to thank my supervisor Associate Professor Annette Stahl and advisors Walter Caharija, Christian Schellewald and Eleni Kelasidi for their valuable advice, support and interesting discussions during this therm work

Gaute Vante Nybø
Trondheim, 2018-12-18

Table of Contents

Summary i

Preface ii

Table of Contents iv

List of Tables v

List of Figures vi

Abbreviations vii

1 Introduction 1
- 1.1 Background . . . 1
  - 1.1.1 Problem description . . . 2
  - 1.1.2 Limitations and scope . . . 2
- 1.2 Planning . . . 3
- 1.3 Data sets used throughout the report . . . 3
- 1.4 Equipment . . . 4

2 Theory 6
- 2.1 Image pre processing . . . 6
  - 2.1.1 Converting colour image . . . 6
- 2.2 Common linear kernels . . . 6
- 2.3 Sea filtering approach . . . 7
- 2.4 Segmentation and structure detection . . . 7
  - 2.4.1 Edge emphasising operators . . . 7
  - 2.4.2 Circular Hough Transform (CHT) . . . 8
  - 2.4.3 Harris Corner detection . . . 9
  - 2.4.4 Shi–Tomasi corner detection . . . 9
- 2.5 Transformation between inertial and drone . . . 10
  - 2.5.1 Obtaining NED the NED frame . . . 10
  - 2.5.2 The Euclidean transformation from body to NED . . . 11
- 2.6 Camera calibration and notation . . . 13
- 2.7 State estimation . . . 14
  - 2.7.1 Motivation for estimating states . . . 14

2.7.2 State estimation 15
2.7.3 Filtering techniques for navigation reviewed 16
2.8 CNN 17
2.8.1 Single Shot detectors 17
2.8.2 Object detection and the Yolo network 17
2.8.3 Training the network 18
2.9 Considerations in the artifex project 19
2.9.1 Distance of sight from vassal 19
2.9.2 Adjusting the lens 21
2.9.3 Ground Sample Distance - GSD 21
2.10 Compiling OpenCV to Node.js 22

3 Methodology 23
3.1 Structure geometry detection module 23
3.2 GUI 24
3.2.1 Visualisation of the operation 24
3.2.2 Front end 24
3.2.3 Building around existing libraries and frameworks 24

4 Results and discussion 25
4.1 Image processing 25
4.1.1 Prepossessing case study - Morphology method sea ltering . . . 25
4.1.2 Prepossessing case study - Lab space 26
4.2 Classical approach to segmentation 27
4.2.1 Experimental detection of mooring buoys 27
4.2.2 Hough cirlces for cage detection 27
4.2.3 Classical approach discussion 28
4.2.4 DNN based segmentation 28
4.3 Sensor ltering 30
4.3.1 Experimental test for STM 31
4.4 Illustration of concept 31

5 Conclusion and Future work 32
5.1 Concluding remarks 32
5.2 Future work 32

Bibliography 34

Appendix 36

List of Tables

1.1 Data sets 3
1.2 Drone data 4
1.3 Drone data 5

List of Figures

1.1 Potential for maritime wealth creation (Mrd. NOK / year) NTVA / DKNVS (2012) 1
1.2 Picture taken from Artifex 2
1.3 Flight pattern Korsneset 08/11/2018 4
1.4 Drone (Korsneset 2018) 4
1.5 Camera (Korsneset 2018) 5

2.1 Illustration Atherton and Kerbyson (1999) 8
2.2 Camera projected to ground plane 14
2.3 Graphical representation of estimating one state 15
2.4 Comparison sensor fusion by Cappello et al. (2015) based off GNSS, VBN and IMU 16
2.5 Pre de ned boxes exsample (taken from dos Santos (2018)) 17
2.6 Training of the network 18
2.7 Caption 19
2.8 Viewing distance projected 20
2.9 Left plot: D (;), Right plot: $\frac{d}{d}$ D (;) 21
2.10 Conceptual illustration of sparse image sensor projected on to the ground plane 22
2.11 Opencv implementation of frontend GUI Electron framework 22

3.1 Flowchart describing the image data ow 23
3.2 Communication ow during operation 24
3.3 Libraries and framework used making program front end 24

4.1 Performance review morphology based pre prosessing 25
4.2 Robustness test Lab space-based tlering 26
4.3 Experimental mooring buoy detection with Shi-Tomasi Corner Detection . 27
4.4 Dimensions of shcage at Korseset as of 2018 27
4.5 Robustness check hough circles 28
4.6 Results of ”You only look once realtime object detection” 29
4.7 Test of open source library Iphone 7 30
4.8 Test of open source library Iphone 7 30
4.10 Early stage of stitching together different modules 31

Abbreviations

RPAS - Remotely piloted aircraft systems
ROV - Remotely operated underwater vehicle
USV - Unmanned surface vehicle
CNN - Convolutional Neural Networks
GNSS - Global Navigation Satellite System
GPS - Global Positioning System
NED - Local north, east, down
ECEF - Earth-centered, arth- xed
VBN - Vision-Based Navigation (

Chapter 1

Introduction

1.1 Background

The sea accounts for 70 % of the earth surface area, but only 2 % of the world food production comes from the sea. Of this is Norway responsible of 2 % of this as of 2018.

In 2012 there was a opportunity study called ”Verdiskaping basert produktive hav i 2050” it was a part of ”perspektivanalysen” ordered by the government of Norway. where they predicted the growth until 2050. The work was conducted by NTVA, DKNVS og SINTEF. In the study we can nd that the sector is expected to go from a ve doubling as of 2010 to 2050, that means taking the productions of 1 million tonn of salmon to ve million in 2050.

Figure 1.1: Potential for maritime wealth creation (Mrd. NOK / year) NTVA / DKNVS (2012)

As of today the country is ahead of the curve when it comes to value creation but behind when it comes to production.
The predicted prices are higher but the volume is lower. This is because of the industry has a problem of health. The industry as a whole is strongly regulated growth regulated with respect to kilos per year produced, until the health problems are improved.

With this in mind this report aims to contribute towards a fully human less operation around and within one locality. Thus increasing the convenience and lowering the cost of inspection. More frequent and in dept inspection can make for better understanding and prevention of poor health conditions in and around sh farm.

1.1.1 Problem description

The problem can be stated as the following

Generate a static map of the sh

Mapping the safe zones where the USV, the RPAS and the ROV can operate

Figure 1.2: Picture taken from Artifex

The following is speci ed as input output in of the RAPS operation

Inputs
Images taken from above: they shall encompass the biggest possible area of the sh farm with a suf cient resolution

Output
Georeferenced coordinates of the key structures of the sh farm such as the buoys and the cages, as well as estimate the position of the mooring lines underwater

The problem of extracting and mapping structures in the sea will make the autonomous operation the sh farm easier for all autonomous vehicles involved. As all of the structural information outputed by the RAPS is broadcast when found.

1.1.2 Limitations and scope

The report is limited to literature study and testing of methods solving the problem at hand. There was also an functionality added throughout the semester, as the addition of GUI feature to the software package allowing for monitoring of the operation. Also it will be assumed at least a Nadia jetson or similar computing capabilities in the RAPS or USV for the algorithems to run on.

1.2 Planning

During the rst meeting it was decided to lay out a rough plan what the time was to be spent on in during the semester.

1. Literature study (week 38-14)
2. Schedule planning (week 37- 41)
   - Speci cations of the RPAS system of ARTIFEX
   - Pictures and methods from the SensorDrone project
   - Simple geometrical model of the farm and the anchoring system
3. Possessing of the 2015 pictures (week 37- 41)
4. Field drone ying (week 41- 44)
5. Possessing of the obtained data (week 44- 48)
6. Report writing (continuous)

1.3 Data sets used throughout the report

There are data to this project is speci ed in the task description, that is the pictures are to be taken at suf cient height and and resolution. There are also one addition who the task description is not taking into account that is bene cial to the problem solving. There where is always given a GNSS and barometer reading aside with the data set.

Table 1.1: Data sets

Location	Date	Camera	Type	Project
ACE Korsneset	04/08/2015	DJI FC350z	Pictures	Artifex
	08/11/2018	DJI FC350z	Pictures and video	Artifex
Rataren	2015	DJI FC350z	Video	Artifex

Remark- The last data set obtained has the most relevance to the project because of the relevance with regards to ful lling the speci cation listed in the section above.

Parsing and visualising the driven distance

As mentioned there has been conducted several experiments over the year. To better evaluate the sets and video given there was made two script with python and MATLAB to visualise. It rst parses the text le given into a csv, then matlab can be used to visualise the data, the script also supports visualising loss of GPS connection.

Figure 1.3: Flight pattern Korsneset 08/11/2018

1.4 Equipment

The equipment used throughout the testing phase of the artifex project was provided by Maratime Robotics.

P-GPS Accuracy	[0.5 m, 1.5 m]
Max Pitch Angle	25
Max Wind Resistance	8 m/s
Package Dimensions	525 mm
	480 mm 640 mm
Weight	9.5 kg
Hovering Time	16 min

Figure 1.4: Drone (Korsneset 2018)

The camera is xed to a gimbal and it is regulating the camera to an attitude that is aligned parallel to the ground plane. The camera and gimbal package is listed below.

Sensor	Sony EXMOR 1/2.3
Resolution	4096x2160 (25fps)
FOV	94 deg
Lens	20mm
Vocal length	3.6mm

Figure 1.5: Camera (Korsneset 2018)

Chapter 2

Theory

2.1 Image pre processing

2.1.1 Converting colour image

Grayscale

OpenCV has the following convection of the coloured image to grayscale.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad 0{:}299 \quad 0{:}587 \quad 0{:}114 \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.1}$$

Otsu's Method

This method was rst proposed by Otsu in 1979 in the paper Otsu (1979). The technique relies on a stochastic approach to background ltering. The basic idea is to iterate trough all the threshold values and calculating the variance spread at in each iteration.

$$\sigma_w^2(t) = q_1(t)\ \sigma_1^2(t) + q_2(t)\ \sigma_2^2(t) \tag{2.2}$$

2.2 Common linear kernels

Common kernels used trough out the project

Gaussian

$$H_{(i;j)} = \frac{1}{2\ \ 2} e^{-\frac{(i)^2+(j)^2}{2\ \ 2}} \tag{1}$$

Sharpening

$$Y_{i;j}(k) = I(i + \alpha \frac{g_x(i;j)}{g_x^2 + g_y^2}\ k; j + \alpha \frac{g_y(i;j)}{g_x^2 + g_y^2}\ k): \tag{2.3}$$

2.3 Sea ltering approach

A colour space is a mapping of the pixels contained in an image to some other representation. There are several conversions of how one could represent the information encoded in the image. The goal is often to make make for easier thresholding or adjustment of an abstract value associated with the pixel, for example thresholding or adjusting a saturation rather then a red intensity value. The usage of colour spaces can be one method of separating the sea form the structures in the sea shown in the paper M. Klimkowska and Lee (2017). The paper discusses ltering sea and uses the Lab space posed in 1976 the CIE recomended the colorspace CIE LAB, CIE publication 15.2.

The following equation is used for the Lab mapping.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.4}$$

2.4 Segmentation and structure detection

Many higher more abstract image processing algorithms has its ground basis in the edge emphasising algorithms

2.4.1 Edge emphasising operators

Sobel

Sobel is trying to solve the problem of approximating the derivative of the the intensity along a binary image. Taking the derivative along one axis

$$G_x = \begin{bmatrix} +1 & 0 & 1 \\ +2 & 0 & 2 \\ +1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.5}$$

Used to obtain the kernel

$$G = \sqrt{G_x^2 + G_y^2} \qquad \text{or} \qquad G = |G_x| + |G_y| \tag{2.6}$$

There exists more derivations of this operator, notably the scarr, who replaces the kernel with bigger coef cients when the image kernel is small in size. But the base idea of sobel still stands, and that is to take the approximate the derivative along x and y axis of the image.

Canny

There are still no concern of the absolute localisation of where the edge is found in the sobel. Problems associated with Sobel could be

Gives out no answer for de nite max for the edges in the image

The noise can be present and ampli ed

Solving the problem of not knowing exactly where the edges are, canny calculates the gradients along a speci ed edge and the size of these, the largest one will be the direction of decent.

$$= \text{atan2}(G_y; G_x) \tag{2.7}$$

Knowing the direction of descent there the algorithm can obtain the max pixel value of the edge and scrap the other values along the axis of decent.

There still might be some noise present. The classical way to do this is to a hysteresis threshold where the allowed values in the pictures lie inside a region region d

$$d = \max(d_1; d_2)$$

Final step is to suppress all the values that are not connected to the edge of intensity d_1 with intensity larger or lower than d_1 or d_2 are suppressed. (Fang et al. (2009))

2.4.2 Circular Hough Transform (CHT)

In 1990 there was an survey summing up different methods of doing the circular hough transform (Yuen et al. (1990)) papers centeral to the creation of the methods where revived, like the central paper in 1975 Kimme et al. (1975)). There has been some contributions to the method since then, noticeably in 1999 there was a method proposed how has formed the scene (Atherton and Kerbyson (1999)).
Although there are many papers written on the subject since. Most om them include the following steps of nding circles.

1. Accumulator Array Computation.
   Here the circles drawn around the main circle casts votes that for it being a main circle in the image. As every circle iterated upon shares the one main point at the centre

Figure 2.1: Illustration Atherton and Kerbyson (1999)

2. Center Estimation
   Used for counting and processing the votes made from the edges iterated upon.

3. Radius Estimation
   Not all the circles then to have the same radius, so an estimate needs to be done.

The equation for iterating trough the circles can be expressed as:

$$y = \frac{\cos}{\sin} x + \frac{r}{\sin} \tag{2.8}$$

Hough elipse

First preposed algoritem in 2002 Xie and Ji (2002)

$$a(x \quad p)^2 + 2b(x \quad p)(y \quad q) + c(y \quad q)^2 = 1 \tag{2.9}$$

$$ac \quad b^2 > 0 \tag{2.10}$$

For now this algorithm was found to be suf cient, there needs to be a consideration and performance comparison with algorithms nding ellipses as described in Fitzgibbon and Fisher (1995)

2.4.3 Harris Corner detection

The most common edge detection algorithm is Harris corner detection. The theory relies heavily on the edges described in previous section about edges. In 1988 the Chris Harris and Mike Stephens proposed a method for nding corners heavily based on previous edge ltering algorithms Harris and Stephens (1988). Mathematically its borrowin from the sobel kernal (eks of sobel kernel 2.5)

$$M = \sum_{(x;y)2W} \begin{matrix} G_x^2 & G_x G_y \\ G_x G_y & G_y^2 \end{matrix} \tag{2.11}$$

WhenG is computed there is an threshold set and if

$$R = \det(M) \quad k(\mathrm{trace}(M))^2 = \quad {}_1 \; {}_2 \quad k(\; {}_1 + \; {}_2)^2 \tag{2.12}$$

Why corners

In the paper original proposed, the problem is described in relation to the growing topic of robotic vision. An corner allows systems to note down a position of a region of pixels. Contrary to an edge, who is less strictly de ned. Many of the more sophisticated tacking algoritems today relies hugely on edges .

2.4.4 Shi–Tomasi corner detection

Sometimes refereed to as the Kanade–Tomasi algorithm Shi and Tomasi (1994). Are is an extension of the Harris corner detection. The basic idea is that the algorithm acts more selective for the points conserved and this might increase the robustness in the tracking. In this project the selectness can be used to extract mooring buoys. Assuming these are

the most dominant edges found in the picture.
As the harris edge detecetion calcualtes high eigenvalues $_1$ and $_2$ conclues there exists an edge. The Shi–Tomasi concludes by using the R as

$$R = \min(\ _1;\ _2) \tag{2.13}$$

This is a way of threshold lamda values, and can be highly ef cient for corner detection applications.

2.5 Transformation between inertial and drone

The notation used in describing the qaudcopter is inpired by the book Beard and McLain (2012)

$$v^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \text{longitudinal velocity} \\ \text{lateral velocity} \\ \text{normal velocity} \end{bmatrix} \qquad ! = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \text{roll rate} \\ \text{pitch rate} \\ \text{yaw rate} \end{bmatrix} \tag{2.14}$$

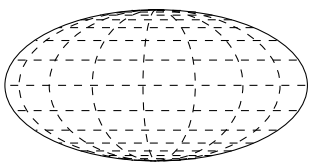
$$= \begin{bmatrix} \ \\ \ \\ \ \end{bmatrix} = \begin{bmatrix} \text{bank angle} \\ \text{pitch attitude} \\ \text{heading} \end{bmatrix} \qquad p^n = \begin{bmatrix} p_n \\ p_e \\ h \end{bmatrix} = \begin{bmatrix} \text{north position} \\ \text{east position} \\ \text{height} \end{bmatrix} \tag{2.15}$$

2.5.1 Obtaining NED the NED frame

The GNSS reading the RPAS gives out WGS (World Geodetic System) 84 readings, these are given in[; l]

The GNSS coordinates gives out WGS readings, witch models the earth as an ellipsoid, more presise, oblate spheroid. This geometical model needs a cople of parameters including equatorial radius a, attening factor f,

$$f = 1 = 298.257223563$$
$$a = 6378137.0\ m$$



With these parameters speci ed by the WGS 84 will allow to compute the identities involving attening. Polar radius and eccentricity respectively.

$$b = a(1 \quad f)$$
$$e = \sqrt{f(2 \quad f)} = 0.0818191908$$

Geodetic to ECEF

With the parameters speci ed the EFEC cordinates can be computed (by Heiskanen and Moritz, 1967).

$$x_{ec} = (h + N)\cos(\ )\cos(l) \tag{2.16}$$

$$y_{ec} = (h + N)\cos(\ )\sin(l) \tag{2.17}$$

$$z_{ec} = (h + (1 \quad e^2)N)\sin(\ ) \tag{2.18}$$

with

$$N(\ ) = p\frac{a}{1 \quad e^2(\sin(\ ))^2} \tag{2.19}$$

$$P_{ec} = \begin{bmatrix} x_{ec} \\ y_{ec} \\ z_{ec} \end{bmatrix} \tag{2.20}$$

ECEF to NED

$$R_{n=e} = \begin{bmatrix} \sin(l_0) & \cos(l_0) & 0 \\ \cos(\ _0)\sin(\ _0) & \sin(\ _0)\sin(l_0) & \cos(\ _0) \\ \cos(\ _0)\cos(l_0) & \cos(\ _0)\sin(l_0) & \sin(\ _0) \end{bmatrix} \tag{2.21}$$

$$P_n = R_{n=e}(P_e \quad P_{e;ref}) \tag{2.22}$$

2.5.2 The Euclidean transformation from body to NED

Assuming the structures observed is ridged in NED $= [\ ;\ ;\ ]$ where yaw

$$R_x(\ ) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & s & c \end{bmatrix}; \quad R_y(\ ) = \begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ s & 0 & c \end{bmatrix}; \quad R_z(\ ) = \begin{bmatrix} c & s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

This can be integrated up to the angles needed to transform observed objects. The rotation of the objects observed by the camera system is described by

$$R_b^n(\ _{nb}) = R_z(\ )R_y(\ )R_x(\ )$$

$$= \begin{bmatrix} \cos\ \cos & \cos\ \sin\ + \sin\ \sin\ \cos & \sin\ \sin\ + \cos\ \sin\ \cos \\ \cos\ \sin & \cos\ \cos\ + \sin\ \sin\ \sin & \sin\ \cos\ + \cos\ \sin\ \sin \\ \sin & \sin\ \cos & \cos\ \cos \end{bmatrix}$$

There is a using this angles we get a singularity when the pitch 90 is But This can be assumed to not be a problem, considering the normal behaviour of an drone. The translation is described by the vector $= \ x;\ y;\ h\ ^{>}$.

The full translation then becomes

$$p = \underbrace{\begin{matrix} R & t \\ 0 & 1 \end{matrix}}_{\text{GPS and IMU}} \underbrace{\begin{matrix} p_s \\ 1 \end{matrix}}_{\text{Observation in body}} \tag{2.23}$$

The p_s needs to come from type of slam system that as obtained some depth information about the point of interest

Assuming perfect Gimbal

Assuming the gimbal is perfect, we get $= 0$ and $= 0$

$$R_b^n(\ ) = R_z(\ )I^2 \tag{2.24}$$

Adding the assumption of perfect height tracking becomes

$$\xi = \ p_n;\ \ p_e;\ \ h_0\ ^{>} \tag{2.25}$$

Then the translation becomes

$$p = \begin{matrix} R & t \\ 0 & 1 \end{matrix} \begin{matrix} p_s \\ 1 \end{matrix} \tag{2.26}$$

$$= \underbrace{\begin{matrix} \cos & \sin & 0 & p_n \\ \sin & \cos & 0 & p_e \\ 0 & 0 & 1 & h_0 \\ 0 & 0 & 0 & 1 \end{matrix}}_{T} \begin{matrix} p_s \\ 1 \end{matrix}$$

2.6 Camera calibration and notation

In order to translate the points in the image obtained into the right measurement units there is needed to do a calibration. The model of the camera includes

Pinhole camera model Zhang (2000)

Lens distortion Heikkila and Silven (1997)

The reason for that the pinhole algorithm does not contain the radial and tangential distortion is that it the model does not contain a lens, therefore this is accounted for in a later step. Models here is obtained in Bradski and Kaehler (2013)

Pinhole camera model

$$p = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} k_u f & 0 & u_0 \\ 0 & k_v f & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsic matrix}} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

There are several notations of focal lengths expressed in pixel units:

$$k_u f = f_u = {}_u$$
$$k_v f = \underbrace{f_v}_{\text{OpenCV nation}} = {}_v$$

Radial distortion correction

The radial distortion in Opencv and Matlab Camera Models

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.27}$$
$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2.28}$$

Tangential distortion correction

$$x_{corrected} = x + [2 p_1 xy + p_2(r^2 + 2x^2)] \tag{2.29}$$
$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2 p_2 xy] \tag{2.30}$$

Summery

The coef cients obtained by calibrating this kind of calibration is

$$\text{Intrinsic}_{\text{coefficients}} = (f_u \quad f_v) \tag{2.31}$$
$$\text{Distortion}_{\text{coefficients}} = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) \tag{2.32}$$

2.7 State estimation

The internal sensors are measuring the relative position of the drone and gives out fast updates but are prone to drift the GPS receiver provides out the absolute location but it gets updated less frequently and it may be noisy in this scenario a where its desirable to estimate stats, one of the techniques used can be kalman ltering, to fuse measurements and nd the optimal estimate of the exact position of the AUV. This section discusses different approaches of ltering including Kalman.
The following state vector x can written as a combination of states and their related biases.

$$x = [p^n; \quad ^n; \underline{p}^b; b_a^b; b_!^b; b_z^n]^T \tag{2.33}$$

The multi-Sensor system model for the the AUV can be written as

$$\begin{aligned} u_t &= [a^b; !^b]^T \\ v_t &= [v_a; V_!; v_{b_a}; v_{b_!}; v_{b_z}]^T \\ x_{t+1} &= f(x_t; u_t; v_t) \end{aligned} \tag{2.34}$$

Modelling with the sensor model based on Fossen (1994).

$$a_{imu}^b = R_b^n(\quad)(\underline{v}_{m=n}^b + g^n) + b_a^b + w_a^b \tag{2.35}$$

$$!_{imu}^b = !_{m=n}^b + b_w^b + w_w^b \tag{2.36}$$

$$m_{imu}^b = R_b^n(\quad)m^n + b_z^b + w_z^b \tag{2.37}$$

2.7.1 Motivation for estimating states

Importance of yaw

Adding to what is mentioned in the above section it can be said that at high altitudes the error in the reading from for the manometer can result in a very high error taking the point found in the object detection module to NED. There therefore the it can be useful to do estimation.

Figure 2.2: Camera projected to ground plane

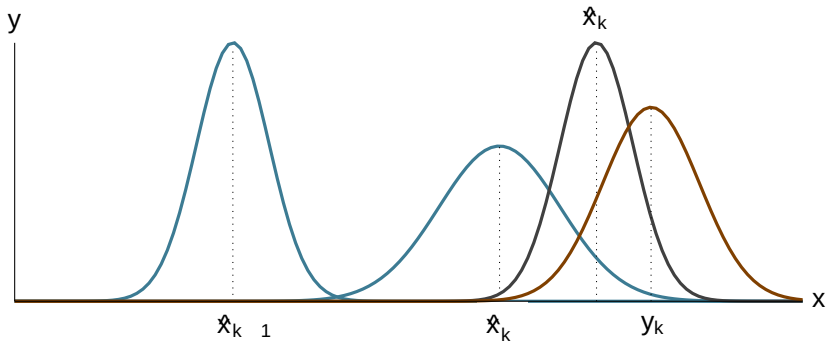


Figure 2.3: Graphical representation of estimating one state

Dreadlocking

In a GPS fallout dreadlocking it gets harder to estimate the position through GPS since the receivers line-of-sight to satellites these blocks and GPS signal is weak in this case it may be desirable to trust the IMU readings which will yield acceleration. However for the position its needed to take the double integral of the acceleration, unfortunately this operation is prone to drift due to small errors accumulating over time to get better position estimates you can use IMU measurements along with other readings, but the other measurements may be affected by biases and noise.

2.7.2 State estimation

KF- most basic theory

One of the problem associated with Kalman lters is that it assumes a zero mean Gaussian white noise. And if the state transition function is linear the noise remains its property and distribution. The same can be said for the measurement function. However if the state state transition function is not linear the noise may not have contained the Gaussian distribution and therefore the lter may not converge.

EKF

However,f $(x_t; u_t; v_t)$ is nonlinear, and the resulting state distribution may not be Gaussian. And therefore, the Kalman lter algorithm may not converge.

Analytically obtaining the derivative may be dif cult

Computational cost of computing numerically may be high

The system function may be non continuous at one point, making the differential unde ned

Linearisation in some cases may be a bad approximation for high non linear system

UKF

Instead of linearizing the funciton at hand, the unscented Kalman lter aims to approximate the probability distribution. It works be selecting a minimal set of sample points such that their mean and covariance is the same as the distribution, these are referred to as sigma points. Each of the points are propagated tough the true nonlinear system and the new mean and covariance are computed. Witch is then used to get the new estimate.

2.7.3 Filtering techniques for navigation reviewed

The need for state estimation can be relevant. As in the data collection at Korsneset 2018 the GPS fell out, then there would be desirable to estimate position based off of the sensores contained inside the different IMU packages. It could also be RPAS for the simple reason of getting better estimate of states in a certain time step considering the GNSS system is slowing than the IMU with respect to update frequency. In the paper Tailani et al. (2014) it is proposed that an EKF approach can be used to get the desirable estimated states. However in the paper Cappello et al. (2015) and Ren and Ke (2010) the authors tries to compare the ways of ltering giving a certain benchmark of each method.

Figure 2.4: Comparison sensor fusion by Cappello et al. (2015) based off GNSS, VBN and IMU

However implementing and testing them self for the speci c localisation task at hand is not in the scope. To hold complexity down it is considered to take one open source library that is based off Roetenberg et al. (2005), where there is proposed usage of an complementary lter.

2.8 CNN

CNN stands for convolutional neural networks. The basic idea is to take in image data though the layers that lters (convolves). These lters has kernals and are adjusted within the layers so that these lters are adjusted automatically to extract the most useful infraction to the task at hand. What separates the convolutional neural network from the traditional neural network is that the layers are not fully connected, they are rather what we call locally connected.

2.8.1 Single Shot detectors

The previous networks neural networks has divided up the it in to sub tasks

Providing region proposals

Quality classi er classify these proposals

This makes for high accuracy but not fast processing of images.

However in the Single shot detectors they do this by instead have a set of pre-de ned boxes to look for objects. Instead of having one part of the program dedicated to providing region proposals.

Figure 2.5: Pre de ned boxes exsample (taken from dos Santos (2018))

2.8.2 Object detection and the Yolo network

Yolo is one of the single shot detectors, hence the name ”you only look once”.
The network also tacles the porblem formulation of object detection: given a image, output bounded boxes where the objects of interest while also labelling. Detecting objects have historical been a big task for computers. There have been a lot of advances in recent years of making it more accurate, however as mentioned before the computational load has been considerable associated with these problems. Until the yolo network proposed a solution to this Redmon et al. (2015). Since then there has been some notable improvements to the rst version of the network proposed.

Loss function

The loss function can be written like the following

$$
\begin{aligned}
&{}_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
&+ {}_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - \hat{C}_i)^2 + {}_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} (C_i - \hat{C}_i)^2
\end{aligned}
$$

Note that form the original posed loss function the last therm are not present because of in the projects application there was no multiple classes.There is no reason for the simpli ed cost funciton, there is only one class used in the project so far.

2.8.3 Training the network

After labelling the pictures with ground trough and generating the corresponding XML there the network could be trained.

The network was trained for little over 150 epochs

Figure 2.6: Training of the network

2.9 Considerations in the artifex project

One consideration to take into account when specifying the hardware and software of the artifex project. This is the constraint on the height of the drone. There needs to be a video stream link from the USV and the drone, this is decided to be cabled so far in the project as of now.

Figure 2.7: Caption

There are is a maximal distance depending on the length of the cable attached to the vassal. This cable is attached to the vassal by design of the artifex project.

2.9.1 Distance of sight from vassal

The distance can be written as a function of the angle from the vassal to the RPAS.

$$D( ) = d_1( ) + d_2( ) + d_{safe} \tag{2.38}$$

Simplifying the problem down to the two dimensional space we get the gure 2.8.

Figure 2.8: Viewing distance projected

The gure contains two separate triangles, who can be used to obtain the geometric relations with descriptor as the angle.

$$\cos(\ ) = \frac{d_{safe} + d_1}{h_1 + h_2} \qquad \tan\ \frac{}{2} = \frac{d_2}{d_r \sin(\ )\quad h_1} \tag{2.39}$$

Substituting into 2.38

$$\begin{aligned} D(\ ) &= d_{safe} + d_1(\ ) + d_2(\ ) \\ &= d_r \cos(\ ) + (d_r \sin(\ )\quad h_1)\tan(\ \ ) \end{aligned}$$

Differentiating with respect to angle yields the following equation

$$D^0(\ ) = \quad d_r \sin(\ ) + d_r \cos(\ )\tan\ \frac{}{2} \tag{2.40}$$

This can be set to t zero to check for the optimal angle

$$D^0(\ ) = 0$$

$$d_r \sin(\ ) = d_r \cos(\ )\tan\ \frac{}{2}$$

From this the relation for optimal angle between the RPAS and the vassal can be obtained based on the speci ed lens used in the project

$$= \tan\ \frac{}{2} \tag{2.41}$$

The calculation shows that there needs to be an concern regarding the height of the RPAS.

2.9.2 Adjusting the lens

For this section can be considered to be variable, then the function of distance becomes.

$$D(\ ;\ ) = d_{safe} + d_1(\ ) + d_2(\ ;\ ) \tag{2.42}$$

Differentiating with respect to angle of view

$$\frac{d}{d}D(\ ;\ ) = d_r \cos(\ )(1 + \tan(\ )^2) \tag{2.43}$$

The other differentiation with respect to can be found in the equation 2.40. Putting these tougher one gradient can be found.

Numerical plotting

Setting the rope distance to 45 m (d_r (= 45) and the camera offset to d_1 = 0:5 meters we get the following relation between the angle of view, angle from he vassal to the drone and the height.

Figure 2.9: Left plot: $D(\ ;\ )$, Right plot: $\frac{d}{d}D(\ ;\ )$

Last we can consider

$$= 2\arctan\frac{d}{2f} \tag{2.44}$$

Allowing for change in vocal length what the speci cation of the lens f and d.

2.9.3 Ground Sample Distance - GSD

From the previous section we can know that keeping the angle somewhat high is bene- cial to the maximising the distance of wieving. This is the sensor projected down to the ground. The relationship can be described with the following equation

$$GSD = \frac{p}{fW}R \text{ [m=px]} \tag{2.45}$$

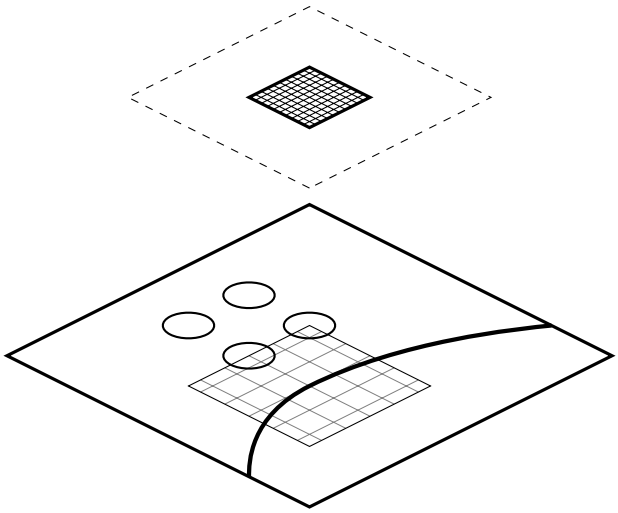


Figure 2.10: Conceptual illustration of sparse image sensor projected on to the ground plane

Where R is the distance from the optical centre, w is the length of the pixel. This has to be taking into account while considering the height of operation.

This could also be taken into account. The geometric relation is derived in equation 2.45. This relation can be extended to get an constraint on how high the drone can go, with respect the projected pixels at the ground.

2.10 Compiling OpenCV to Node.js

Node.js is the runtime environment of the executing the JavaScript. The javaScript runtime is built on Chrome's V8 JavaScript and WebAssembly engine. The node-gyp build tool can be used to compile c++ projects into node addons.. It is based on gyp, a similar tool to Cmake. To automatically compile addons written in c++ v8 to what is called node addons.

Figure 2.11: Opencv implementation of frontend GUI Electron framework

Chapter 3

Methodology

3.1 Structure geometry detection module

The proposed image segmentation system is consisting of the following three modules. STM, deep neural network and hough lines. The the statistical model is

Figure 3.1: Flowchart describing the image data ow

The euclidean transform would take in parameters from the sensor module, but is not drawn in to the ow chart.

3.2 GUI

3.2.1 Visualisation of the operation

The data generated at the USV can be sent to a base station for the operators. To make the operators gain insightful information about the operation and avoid threats in the testing phase of the artifex project. The USV contains the data generated from the drone. And should also contain the information about the speci c location of the ROV and of course the UAV.

Figure 3.2: Communication ow during operation

3.2.2 Front end

The idea of the proposed GUI implementation is for the program modules to be bundeled together as one program, running off JavaScript. The proposed way of doing it may

The idea is that when the user is launching the frontend javascript/ Electron application, the c++ binary is called by the program and is working in the background.

3.2.3 Building around existing libraries and frameworks

The importance of JavaScript as an front end language is growing rapidly. This could be seen in the relation to how it is the most popular used for web front end development. The

(a) Electron (b) Nodejs (c) Threejs

Figure 3.3: Libraries and framework used making program front end