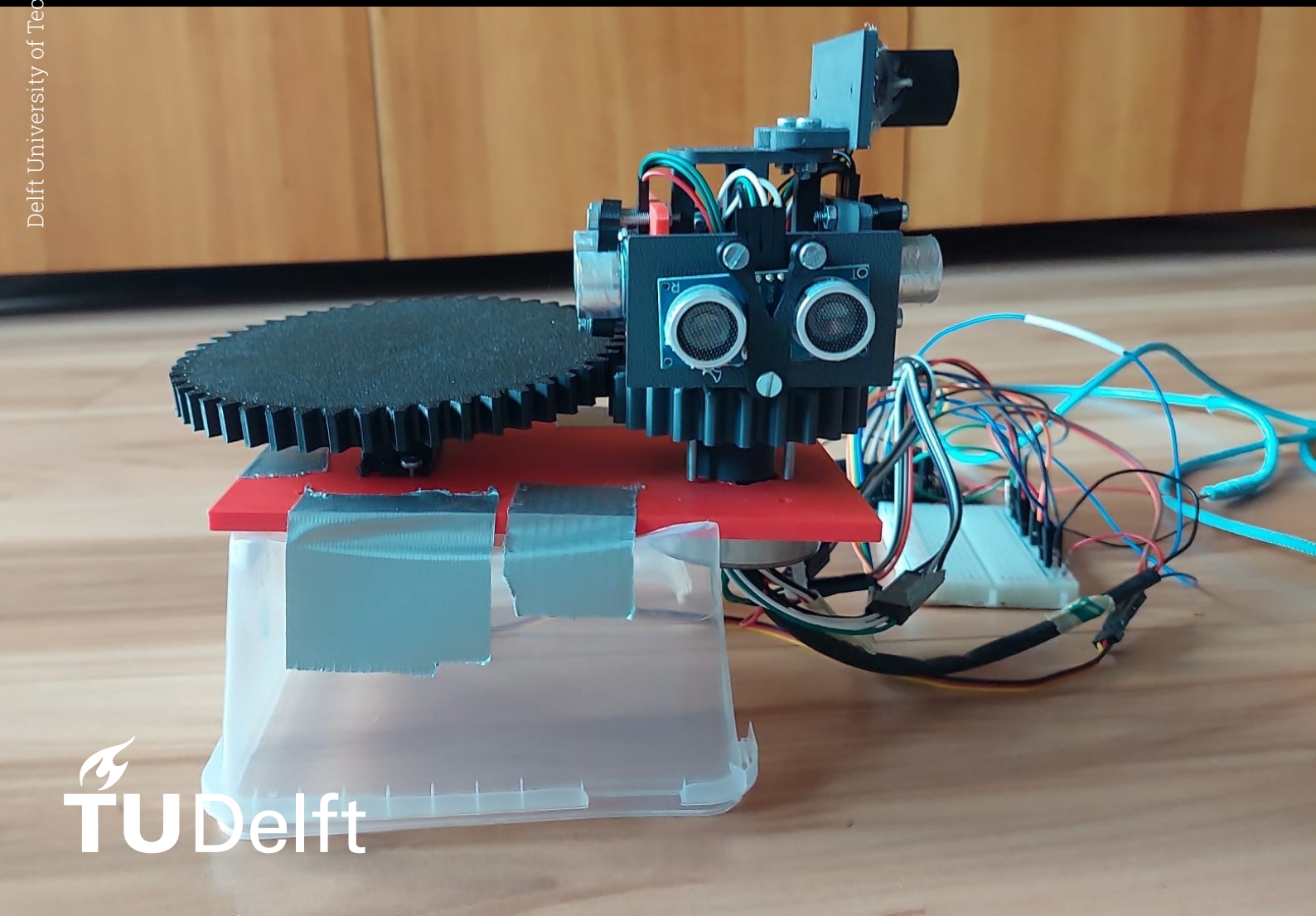
# Dynamic adaptive lidar sensing aided with an ultrasound sensor

EE3L11: Bachelor Graduation Project Electrical Engineering (2024/25 Q4)

Evert-Jan Beiboer & Jesse van der Kooij

Delft University of Technology



TU Delft

# Dynamic adaptive lidar sensing aided with an ultrasound sensor

by

Evert-Jan Beiboer & Jesse van der Kooij

to obtain the degree of Bachelor of Science
at the Delft University of Technology

Student numbers: 5352754 & 5608546
Project duration: April 23, 2025 – June 16, 2025
Thesis committee: Dr. G. Joseph, TU Delft, supervisor
Prof. Dr. Ir. L. Abelmann, TU Delft
dr. S.M. Alavi, TU Delft

Style: TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

# Preface

*This project is a culmination of about 9 weeks of effort in researching, designing and developing a system. Together with the two members of the other subgroup, we made a system capable of both sensor fusion and adaptive lidar scanning aided with ultrasound. We would like to thank our supervisors Dr. Geethu Jospeh, Dr Nitin Meyers and Ir. Peiyuan Zhai for guiding us through the process and providing valuable feedback. We would also like to thank the members of the other subgroup, Guus Dohmen and Thymen Vrijhoek, for thei contributions to the project. Finally, we would like to thank Stichting Neobots for loaning several of the components used in creating the prototype.*

*Evert-Jan Beiboer & Jesse van der Kooij*
*Delft, July 2025*

# Abstract

This thesis investigates a method to dynamically adapt the angular resolution of a 2D spinning lidar (Light Detection and Ranging) using an ultrasound sensor. The ultrasound sensor data is used to locate areas where a higher angular resolution is desired compared to other regions. It proposes two distinct methods to locate these areas based on real problems encountered in the autonomous automotive industry and assign numerical scores to areas of interest. These scores are then taken into account when deciding the angular resolution for that area. The adaptive algorithm is meant to increase the amount of data available on object of interest for use in object avoidance and trajectory mapping systems in autonomous vehicles. The thesis will cover the design, development and testing of the whole system.

# Contents

# 1 Introduction

Automated environmental perception using sensors is becoming more and more commonplace in a wide range of applications. One example is in automotive vehicles, where a range of lidar, stereo or monocular cameras, radar, and ultrasonic sensors are being used to map out the environment for navigation and obstacle avoidance [4] [9]. Another example is with automated indoor robots, like warehouse logistic robots or cleaning robots, which generally use sensor in conjunction with SLAM algorithms to map out the environment [16]. Construction, land and resource surveying, and defense also use a range of sensors for environmental perception[11] [18] [3]. By automating the perception, it can reduce human error and increase the efficiency of processes. What exactly is being perceived and which sensors are being used ranges from application to application, like using lidar for depth perception in mapping out areas [16], using rgb cameras and lidar fusion for object recognition in autonomous vehicles [13], and ground penetrating radar for use in detection of landmines in war torn areas [3].

Two common sensors employed in environmental perception are the lidar (Light Detection and ranging) and ultrasound sensors. Below follows a short overview of both sensors.

## 1.1. Lidar

A lidar sensor, which stands for Light Detection And Ranging, is a sensor primarily used for measuring distance. The way a lidar measures distance can be divided into three types, a ToF (Time of Flight) lidar, an AMCW (Amplitude Modulation Continuous Wave) lidar, and an FMCW (Frequency Modulation Continuous Wave) lidar. Both the AMCW and FMCW lidars use a continuous laser emission with a known modulation, amplitude for AMCW and frequency for FMCW. The distance is then calculated by measuring how the modulation is reflected from the object. The most used lidar is the ToF lidar, which sends out a pulse moving at light speed and measures time it takes to reach to object and bounce back to the lidar. The distance is then calculated as $distance = time\ of\ flight\ * speed\ of\ light\ /\ 2$. As the lidar sends out a concentrated beam of light, it has an extremely narrow FoV, which makes it a very precise measurement tool. This makes it a very good sensor for applications were a high precision is needed, but also necessitates the use of some sort of system which can direct the lidar beam to different locations. The two main ways to do this are to either place the lidar on a mechanical spinning platform , usually called a spinning lidar, to get a full 360° FoV. Lidars like the Velodyne HDL-64E is one of those systems, with 64 horizontally stacked lidars, giving a horizontal FoV of 27°, making it a fully 3D spinning lidar. The other option is a solid state lidar, where the lidar itself is fixed. The most common approach to implementing a solid state lidar is a MEMS based lidar, where the lidar points to a movable mirror which reflects the lidar beam. These typically do not offer a full 360° view, but are more robust and less prone to failure than its mechanical counterpart. A downside of lidar in outdoor use is the performance loss in adverse weather conditions[22], so it is often used in conjunction with other sensors when used in these conditions.

## 1.2. Ultrasound

The ultrasound sensor is also a ToF sensor like the lidar, but it uses an ultrasound wave instead, thus the distance calculation uses the speed of sound instead of the speed of light. An ultrasound sensor for industrial use has a frequency which typically ranges from 30kHz to 300kHz. Higher frequency are used for short range applications, while low range frequencies typically perform better on the longer ranges. They have a minimum distance from which measurements can be made, as anything before that lies in its deadband and cannot be consistently measured. This deadband is smaller for higher frequencies. Another big difference is the FoV, as an ultrasound sensor FoV can range from 15° to 180°. This makes it a less precise distance detector than the ultrasound, as the precision decreases as distance and FoV increase, but it can measure a wider area with just one measurement, so it is better in applications were a quick measurement without much precision is needed, like for measuring volume in containers. Ultrasound also performs better than lidar in detecting transparent and poorly reflective surfaces.

## 1.3. Problem definition and state of the art

In most commercial applications of spinning lidar, the measurements are being distributed uniformly along the entire FoV. This makes it so the system has one singular angular resolution. This is contrast to something like the human perception, which uses the eyes highest angular resolution to scan the Regions of Interest (RoIs). This process is also called foveation and it has a counterpart in image processing called foveated imaging. This is where the angular resolution varies over the range of the image, with it being higher in the RoIs. This technique can save on precious resources like time and power in an application like autonomous driving vehicles, where the primary focus in object detection and trajectory mapping algorithms is on the objects in the foreground [19]. When driving a vehicle at high speeds, any contact to another vehicle can cause injury or even death to a large amount of people. It is therefore imperative to avoid any contact with other vehicles as much as possible. Avoiding any object, be it other cars or obstacles in the road, starts with detecting said objects, so appropriate avoiding action can be taken. [6] Finds that crashes occurring when changing lanes happens because not enough attention was being given to the surrounding area or because the other cars were in blind spots. Thus, focusing more on the objects in close proximity should give drivers a chance to avoid going on a collision course with other vehicles. Furthermore, reaction speed is also a big factor in collision avoidance, as a quicker reaction has a much higher chance to avoid a collision[12]. Then getting the information faster while not compromising the quality of information in paramount in collision avoidance. Previous work on adaptive lidar like in [17] used a topological grid map to avoid oversampling objects that were close by, thus saving resources that could then be used to detect objects further than the lidar would normally be able to. In [2], [7], [15] and [21], an RGB image was used to define the RoIs and adapt the sampling density over the image. [10] proposes the use of HD maps in cyberspace to define the RoIs. These methods all require either knowledge of the environment beforehand via the topological grid map or they require a high definition RGB image to be processed in a complicated algorithm, which takes a lot of computing power and time, possibly negating the savings that would occur by sampling less. What is needed is a method to define the RoIs in a fast and efficient manner. That brings us back to the ultrasound sensor, which can provide such a measurement in a fast and efficient manner while requiring just a fraction of the computing power to process compared to HD RGB.

## 1.4. Goals

This section will provide the goals we are trying to achieve in this project. We aim to investigate methods of lidar foveation using RoIs based on an ultrasound sensor dynamically. Due to time restrictions we will only look at two different method applying to two different scenarios for autonomous driving. The first scenario is based on [5], where radar and ultrasound sensor measurement were fused for object detection and navigation. Here the system had trouble detecting the edges of objects. The methods goal is then to complement the wide FoV of the ultrasound with the lidar to better detect these edges. The second scenario is a general autonomous driving scenario, where objects in the foreground should be the RoIs. The goal of this method is to spend the available resources more effectively by sampling more points detecting these objects. After these methods have been developed, they will be tested

against a non-adaptive scan to see if they improve the results.

# 2

# Program of requirements

These requirements will be the guidelines for how the project should be developed.

## 2.1. Mandatory requirements

- The product must have a spinning sensor platform that can spin 360 degrees.
- The sensor platform must have at least one lidar sensor.
- The sensor platform must have at least one ultrasound sensor.
- The entire system must be controlled by an Arduino Uno R4 board [1].
- The maximum range of the lidar sensor must be at least 4 meters.
- The maximum range of the ultrasound sensor must be at least 4 meters.
- At least one method to adapt the spinning lidar point distribution based on the acquired ultrasound data must be developed.
- All calculations for the adaptive system must be done on the Arduino hardware.
- A metric needs to be defined to score the developed adaptive method against the pre-existing non-adaptive method.
- The developed adaptive method(s) must be scored against the non-adaptive method using the defined metric.
- The system must be able to receive commands from a separate server running on a computer.
- The system must be able to send data to a separate server running on a computer.

## 2.2. Trade-off requirements

- The product should preferably be compatible with the ELLAS platform[17]
- Different kinds of adaptive methods for different real life situations should preferably be developed.
- The developed adaptive method(s) should preferably score higher on the defined metric than the non-adaptive method.

# 3

# Design process

This chapter will lay out the design process that was undertaken during the project. It will shine a light on the the choices made during the project and provide justification for those choices.

## 3.1. Sensor platform

All designs were done by our sister subgroup consisting of Dohmen, G and Vrijhoek, T. We will provide the design choices and their justification in this section to give an overview of the complete sensor platform.

To be able to test out the adaptive system, a sensor platform must be constructed. Since this project was created as a continuation of the work done on adaptive sensing done with ELLAS[17], and because we have access to the complete platform that was used in that project, we decided to make a platform that could be swapped out with the platform of the ELLAS system, which can be seen in figure 3.2. The baseplate of the sensor array is shown in figure 3.1. This baseplate can be seamlessly swapped out with the baseplate of the ELLAS system.

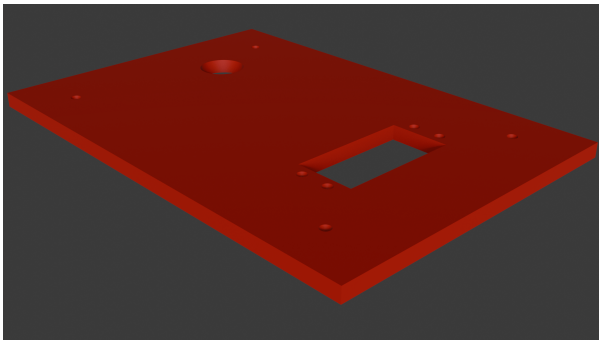


**Figure 3.1:** Baseplate for our sensor system.

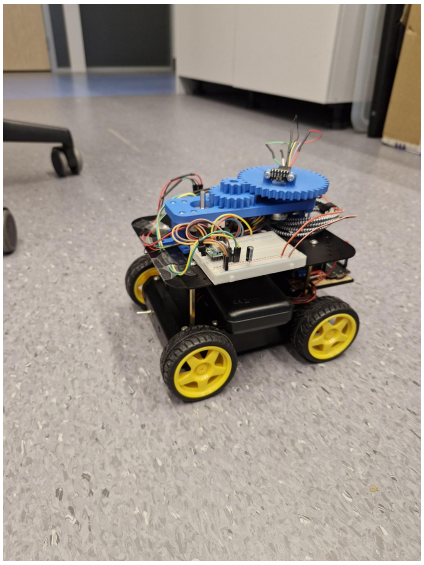


**Figure 3.2:** Picture of the ELLAS system. The black baseplate can be replaced with our baseplate and sensor array.

Once the baseplate was in order, the designing of the sensor array could begin. For the adaptive algorithm, at least one ultrasound sensor and one lidar sensor must be present on the platform. The ultrasound sensor itself was chosen to be the HCSR04 ultrasound sensor[8], as this is a cheap sensor that was readily available and the requirements for range and operating voltage were met. The lidar sensor was chosen to be the YDlidar SDM15[20], as this was already on hand at the time, as well as being within the range and operating voltage requirements. For the driving power behind the spinning system, the Parallax standard servo motor[14] was chosen, as it was on hand at the time, as well as having a high resolution.

With all sensor choices now out of the way, the spinning platform could be designed. The servo has an operating range of 180°, so a gear ratio of 2:1 was taken to provide the main gear where all the sensors are located a full range of 360°. The ultrasound sensor has an FoV of 15°, so a natural subdivision of the complete 360° range would be 24 sectors of 15°. The maximum range of the ultrasound sensor is 4 meters and thus one measurement needs to take at most 25 milliseconds. For increased accuracy, the ultrasound measurement is done 3 times per sector, so that then becomes at most 75 milliseconds per measurement. Combined with processing overhead and servo rotational speed, the time it takes to complete one scan circle would be to slow. Since the ultrasound sensors are cheap and readily available, the decision was made to put them pointing in all four cardinal directions and align the lidar with one of those directions, as then the time to complete a complete scan circle becomes four times as faster. The complete sensor platform can be seen in figure 3.3.

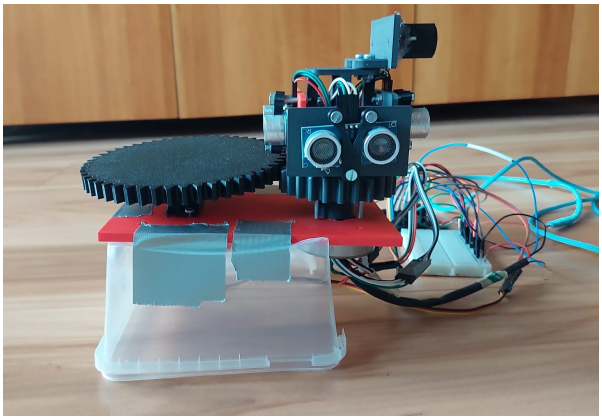


**Figure 3.3:** Complete model for the sensor array attached to the baseplate. The senor with the two silver circles pointing toward the camera is one of the ultrasound sensors. The sensor on top pointing to the right is the lidar sensor.

## 3.2. Adaptive algorithm

Normal spinning lidar operation has a fixed angular resolution. We will propose a method to change the angular resolution on the fly and apply it for two different end goals. To do this, we use the ultrasound sensor on the platform that is offset 90° in front of the lidar sensor, in the counter-clockwise direction. This ultrasound sensor will then take a measurement every 15°, so one every sector, and progressively built up information on the sectors in front of the lidar. A schematic of the sectors and sensor configuration can be seen in figure 3.4. From the ultrasound measurement values in the sectors, some measure of interest will be used to assign an angular resolution to that sector, with RoIs being given a higher angular resolution compared to other sectors. As we cannot directly control the angular resolution, we will assign the amount of lidar measurement points, $M_N$ each sector must contain, thereby indirectly controlling angular resolution per sector. The algorithm itself will be run on the Arduino hardware to show a low amount of computing power is needed for the algorithm.

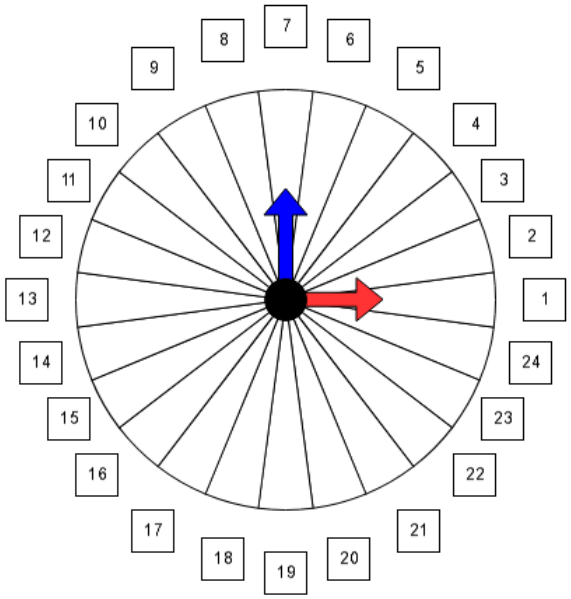



**Figure 3.4:** A schematic of the sector divisions and sensors. The red arrow is the lidar sensor and the blue arrow is the ultrasound sensor at t = 0.

### 3.2.1. Constructing the RoIs

To translate what is interesting into mathematical terms, we have developed something we will call an interest index. This is an array of interest values, which we will call the scores, for every sector. A higher score means that the sector is of higher interest compared to sectors of a lower score, thus making the high scoring sectors the RoIs. We will apply some restrictions in creating the index. First, the scores must be in the range [0, 1]. By restricting the range to [0, 1], we can easily compare across multiple regions. Second, the total sum of all scores in the index must be equal to one. By having all points sum to one, the index behavior stays consistent when more points are added. The information gathered by the ultrasound sensor will be used to calculate the scores. The calculation itself is situation dependent, so we will explain it for both scenarios below.

Index 1: Ultrasound Edge Complement
As was found in [5], an ultrasound sensor, in combination with a radar sensor, had trouble detecting the edges of objects. Thus to complement the ultrasound sensing, we will assign the RoIs to be sectors that have a sharp difference in distance to the next sector. We will call the array with ultrasound data $U[N]$, where $U[6]$ returns the ultrasound measurement for sector 6. Then a new array called $D[N]$, where $D[N] = |U[N] - U[N+1]|$. We now have an array with the differences of neighboring ultrasound measurements, in the positive direction. Through empirical measurements it was found this algorithm would overcompensate and assign too many points to sectors with sharp edges, so the decision was made to add an extra step to it. This step takes $D[N]$ and applies a kernel of $[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$, creating the new array $W[N]$, where $W[N] = \frac{1}{3} \cdot (D[N-1] + D[N] + D[N+1])$, to smooth over the edges and simultaneously compensate for the one directional difference. Finally, $W[N]$ is normalized to abide by our two restrictions and put into array $I[N]$, the index array containing the scores for each sector.

Index 2: Automotive Foreground Detection
Here the RoIs will be the places where the ultrasound detects an object in the foreground. Once again, the array with the ultrasound data is called U[N]. A new array is then created called $L[N]$, where $L[N] = \frac{1}{U[N]}$. Array $L$ is then normalized to abide by our two restrictions and put into array $I[N]$, the index array containing the scores for each sector.

### 3.2.2. Final processing

With these two indexes defined, the amount of points per sector can be allocated, which stand in direct relation to the angular resolution with (3.1), where $\alpha$ is the degrees per sector, $M$ is the amount of points in the sector, and $\beta$ is the angular resolution in that sector.

$$\beta_N = \frac{\alpha}{M_N} \tag{3.1}$$

Then to get the amount of points for a sector N, $M_N$ from (3.1), (3.2) is used, where $I[N]$ is the interest value for that sector, $A_p$ is the average points per sector the system should aim for as set by the user, and $K$ is the amount of sectors that currently have an interest value assigned. To further explain, picture a hypothetical situation where every score in the index is the same value. Then it follows that all sectors are equally interesting and should all get the same amount of measurements. But the size of $I$ is increasing as more ultrasound measurements are made and through the normalization process, each score is decreasing as the sum of all elements increases. Thus, the normalization biases the sectors in the beginning. To fix this, $I[N]$ is multiplied by $K$, which ensures an equal amount of measurements per sector, should the sectors all have the same relative score, namely one measurement per sector. For an array $I$ where the scores are not equal, the average measurements per sector will be one. By multiplying with $A_p$, this average can be increased, but the value of $A_p$ depends on the application. For our tests, it will be set at a value of 3.

$$M_N = round(I[N] \cdot A_p \cdot K) \tag{3.2}$$

## 3.3. Lidar Angular Resolution Adaptation Algorithm

The entire process can be seen in pseudo code in algorithm 1. In line 5, the function $Calculate\_Index()$ uses one of the two methods previously defined to calculate the interest index. At the end of a scanning loop it will send the ultrasound data array and the lidar data array together with the measurement angles to the server, which can then process and display it.

**Algorithm 1** Lidar Angular Resolution Adaptation Algorithm

```
1:  A_p = 3
2:  while True do
3:      for N = 1, 2, 3...24 do
4:          Ultrasound_measurement(N)
5:          Calculate_Index()
6:          M_N = round(I[N] · A_p · K)
7:          for i = 0 to i = M_N do
8:              Lidar_measurement()
9:          end for
10:     end for
11:     Send_arrays()
12: end while
```

## 3.4. Limitations

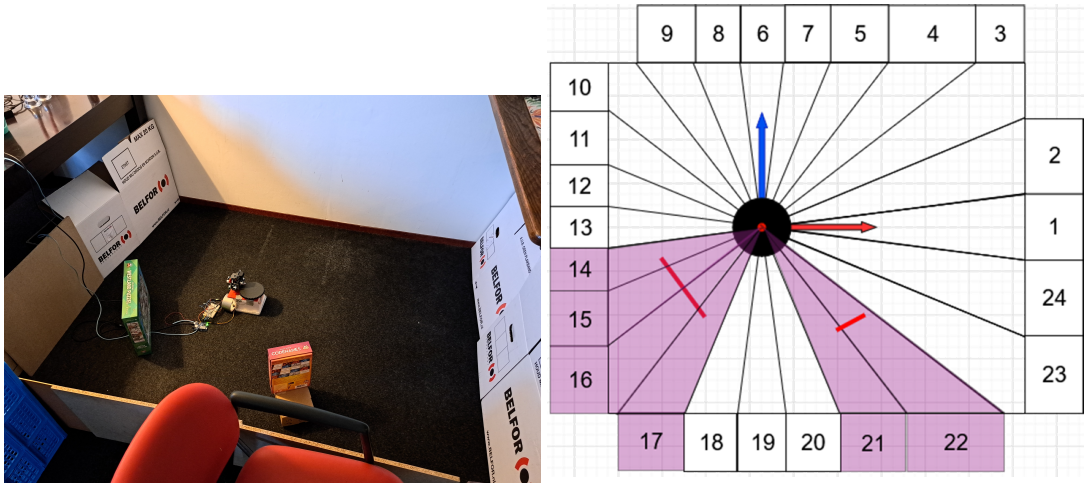
We will discuss some limitations that this design has for our use of it. As all the sensors are on the same rotating platform, they cannot be be controlled separately from each other and must wait for each other to finish scanning. This can lead to delays in the scanning an entire circle. Also, because the rotating assembly is driven by a servo with a 180° operating range attached to a gear operating at a 2:1 gear ratio, the sensor setup will only be able to complete a 360° rotation in one go before having to rotate back in the other direction. This also means that the lidar will only reach the sectors with interest values after a 90° rotation, which are 6 sectors with the current setup. These sectors will therefore have the average amount of points per sector $A_p$ as describe in (3.2). As our system is dynamic and any ultrasound data is only used when the lidar arrives at the sector with that data, using multiple ultrasound sensors has no benefit to out adaptive algorithm, and we will only use the one with a 90° offset to the lidar in the counter clockwise direction. The total amount of current the sensors and servo need is more than the Arduino board can provide on its own, so those are now powered by an external power bank. The sensor control and data processing still happens in the Arduino hardware.

# 4

# Results

## 4.1. Setup

To test the effectiveness of the adaptive algorithm, a test area of 1.9 meter by 1.6 meter was walled off and 2 objects were placed inside the area, as can be seen in figure 4.1. Originally, the test area was much larger, as the program of requirements states a minimum range of 4 meter is required, but the test results were not satisfactory at these ranges, hence the area was downsized.




**(a)** Picture of the test setup.

**(b)** Schematic of the test setup with the sectors shown. The purple sectors contain the objects.

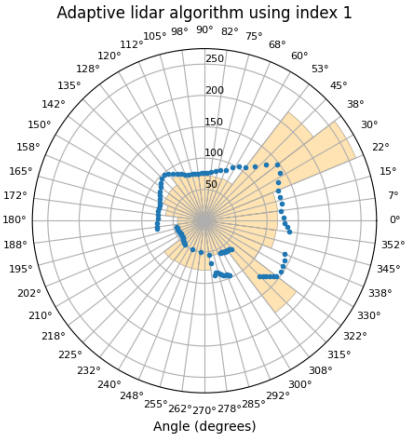
**Figure 4.1**

## 4.2. Procedure

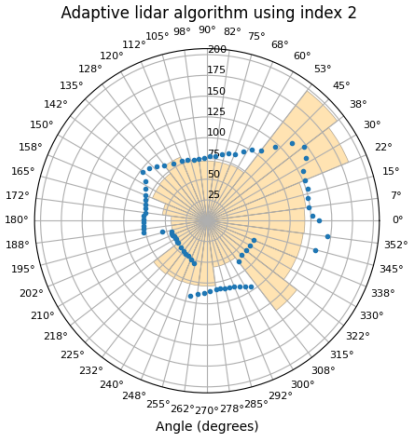
The scene we are testing is a static scene and the algorithm will be tested with both indexes separately. First a test with a non adaptive lidar is done using an angular resolution of 5°. This will be the basis for comparison between adaptive algorithm and the status quo. Each index is tested, where the variable $A_p$, the average measurements per sector, is set to three. This makes the angular resolution on average 5° , which makes it comparable with the non adaptive scan.

## 4.3. Data

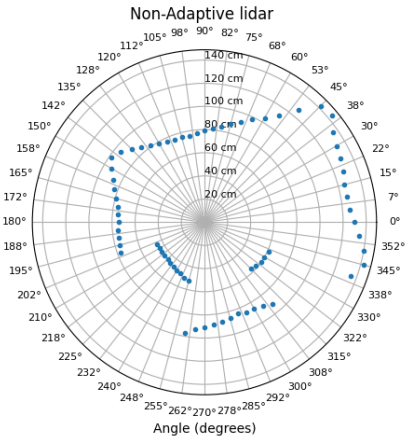
The raw data acquired in the measurement setup with the previously defined procedure is shown in figures 4.2a, 4.2b and 4.2. Something immediately noticeable is the discrepancy between the ultrasound data and the lidar. This problem was even more pronounced in the larger test setup, and as it is the basis for the adaptive algorithm it was the primary reason for downsizing to this setup.




**(a)** Adaptive lidar algorithm with index 1




**(b)** Adaptive lidar algorithm with index 2




**(c)** Non adaptive lidar

**Figure 4.2:** Results using the different lidar algorithmes. Where the lidar data is shown in blue and ultrasound data is shown in orange.

For each sector, the angular resolution can be calculated using (3.1). To compare between sectors and to see the effectiveness of the algorithm figure 4.3 is made, which shows the calculated angular resolution for each sector and each scan.

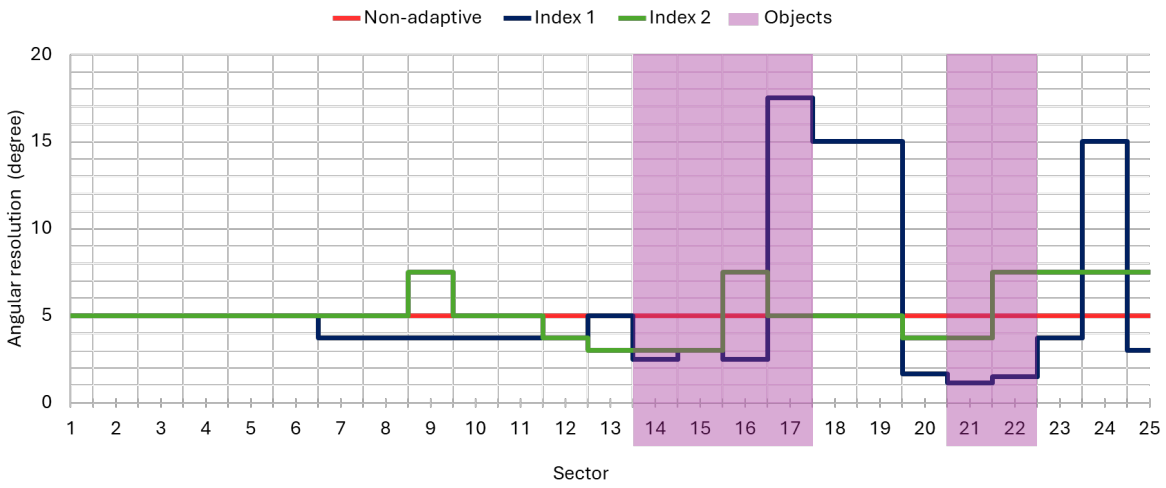



**Figure 4.3:** Graph showing the calculated angular resolution in each sector. The purple sectors contain the objects.

Looking at the angular resolution for the non-adaptive scan, it stays at a constant 5° as expected, and this will be used as the baseline to which we compare the adaptive algorithm. Using index 1, the ultrasound edge complement index, the angular resolution should be improved around the edges of the objects. Sectors 14 and 17 are the edges for the first object, where we can see an improvement of angular resolution in sector 14, but a big decline in sector 17. For the second object, there is an improvement in both sectors. For index 2, the automotive foreground detection, a general improvement of angular resolution in the sectors containing the objects is expected. For the first object, an improvement is seen in sector 14 and 15, a decline in sector 16, and a return to baseline in sector 17. For object 2, only sector 21 shows an improvement, while 22 is a decline. The behavior of the algorithm looks to be wrong at first glance, as it fails to improve the angular resolution in some of the sectors, but reviewing the data more closely, it looks like the ultrasound data has an angular offset of 1 sector, meaning that the ultrasound measures 1 sector ahead of time. To compensate for this offset all sectors should be shifted by 15°. With this shifted sectors the algorithm behaves as expected.

# 5

# Discussion

To evaluate the performance of the algorithm we will compare the angular resolution in the regions of interest to the non-adaptive lidar scan. The non-adaptive angular resolution is a constant 5° in every sector, as expected. For the first index, the ultrasound edge complement index, the angular resolution in the sectors containing the object is around 2.5° at the edges of the objects, accounting for the offset. This is a big improvement over the non-adaptive scan and shows the potential improve object recognition using both lidar and ultrasound data, as was the goal for this index. For the second index, the automotive foreground detection, the angular resolution in the regions of interest was around 3°, accounting for the offset. Still an improvement over the non-adaptive scan, but the results were not as consistent. This was attributable to high variance of the ultrasound data. This index would require further testing in different setups to see if it is viable for real world traffic situations. Overall, the algorithm works on the Arduino hardware pretty well and it is highly modular, with a change of index being able to focus on different areas.

Relating back to the program of requirements, it is clear that some goals were not or not fully met. A spinning platform was made with the correct sensors. The sensors also where selected while keeping the minimum range requirements in mind, nevertheless the system itself only worked in test setups up to a meter. The system is also entirely controlled by the Arduino board, with a separate server running on a computer sending the which algorithme should be done and receiving the data. The system also has two methods to adapt the lidar angular resolution based on the ultrasound scan, which were compared to the non-adaptive lidar, albeit without the new metrics, as those were scrapped after the final revision of the project. The system is also compatible with the ELLAS platform, so that optional goal was met.

The performance of the algorithm itself is satisfactory to an extend. The theoretical framework behind it is sound and for simulated data it works exactly as it should, but testing it in a real world scenario highlighted its sensitivity to erroneous data. Anytime the ultrasound sensor would miss an object or get an erroneous reading, the results from the lidar would suffer from it and the ultrasound only gave correct data about half of the time, a problem that was even more apparent at longer ranges. Examples of this are the offset mentioned in the previous chapter as well as the ultrasound not measuring the border correctly in the first few sectors. Furthermore, the test was a static environment, so nothing can be said about the performance in dynamic environments. It is still unclear if these problems would be solved by building a better prototype platform using a better ultrasound sensor, or that they are inherent to the algorithm itself. The algorithm is lightweight enough to run on the Arduino hardware in real time, so it can be adapted to a whole range of sensor platforms to evaluate its performance and see if any improvements are made.

# 6

# Conclusion

The algorithm dynamically adapting the lidar angular resolution using an ultrasound sensor can work with one of the two interest indexes that were developed during the project. The algorithm produces a higher angular resolution for regions of interest compared to a non-adaptive scan at ranges up to a meter in a static setup. This shows the potential of the algorithm to be adapted to more advanced setups for use in real world dynamic traffic situations.

## 6.1. Future work

Recommendations for future work based on this thesis are the use of a better ultrasound sensor for the preliminary scan. This would probably eliminate the high variance in ultrasound data used by the indexes and the algorithm, which in turn should produce better result. Another potential solution to this problem is to separate the ultrasound sensor and the rotating base plate, making the ultrasound sensor a static sensor only measuring certain sectors. More sensors can then be added to cover more sectors. This is more akin to the sensor arrays located on autonomous vehicles using ultrasound and should produce more consistent measurements. A different approach to solve that problem could also be to change the sensor type, like using radar instead of ultrasound to gather sector data, or a combination of both. Another area of interest is to expand the lidar scanner to a three dimensional scanner, as these are typically used on autonomous vehicles, and the algorithm might need some tinkering to account for the extra spatial dimension.

# References


[1] *Arduino Uno R4*. URL: https://docs.arduino.cc/resources/datasheets/ABX00087-datasheet.pdf.

[2] Alexander Bergman, David Lindell, and Gordon Wetzstein. “Deep Adaptive LiDAR: End-to-end Optimization of Sampling and Depth Completion at Low Sampling Rates”. In: Apr. 2020, pp. 1–11. DOI: 10.1109/ICCP48838.2020.9105252.

[3] Claudio Bruschini et al. “Ground penetrating radar and imaging metal detector for antipersonnel mine detection”. In: *Journal of Applied Geophysics* 40.1 (1998), pp. 59–71. ISSN: 0926-9851. DOI: https://doi.org/10.1016/S0926-9851(97)00038-4. URL: https://www.sciencedirect.com/science/article/pii/S0926985197000384.

[4] Qiping Chen et al. “Sensing system of environmental perception technologies for driverless vehicle: A review of state of the art and challenges”. In: *Sensors and Actuators A: Physical* 319 (2021), p. 112566. ISSN: 0924-4247. DOI: https://doi.org/10.1016/j.sna.2021.112566. URL: https://www.sciencedirect.com/science/article/pii/S0924424721000273.

[5] Niels Damsgaard, Simon S. Gaasdal, and Søren Bonnerup. “Multi-sensor fusion with radar and ultrasound for obstacle avoidance on Capra Hircus 1.0”. In: *Mechman symposium* (2023). URL: https://www.aaudxp-cms.aau.dk/media/5ydaypsq/paper-multi-sensor-fusion-with-radar-and-ultrasound-for-obstacle-avoidance-on-capra-hircus-1-0.pdf.

[6] G. M. Fitch et al. *Analysis of Lane-Change Crashes and Near-Crashes*. 2009. URL: https://www.nhtsa.gov/sites/nhtsa.gov/files/811147.pdf.

[7] Eyal Gofer, Shachar Praisler, and Guy Gilboa. “Adaptive LiDAR Sampling and Depth Completion Using Ensemble Variance”. In: *IEEE Transactions on Image Processing* 30 (Jan. 2021), pp. 8900–8912. DOI: 10.1109/TIP.2021.3120042.

[8] *HCSR04 Ultrasound sensor*. URL: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf.

[9] Jacob Lambert et al. “Performance Analysis of 10 Models of 3D LiDARs for Automated Driving”. In: *IEEE Access* 8 (2020), pp. 131699–131722. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3009680.

[10] Yuhang Liu et al. “Software-Defined Active LiDARs for Autonomous Driving: A Parallel Intelligence-Based Adaptive Model”. In: *IEEE Transactions on Intelligent Vehicles* 8.8 (2023), pp. 4047–4056. DOI: 10.1109/TIV.2023.3289540.

[11] Amin Malekmohammadi et al. “Sensing Systems in Construction and the Built Environment: Review, Prospective, and Challenges”. In: *Sensors* 23.24 (2023). ISSN: 1424-8220. DOI: 10.3390/s23249632. URL: https://www.mdpi.com/1424-8220/23/24/9632.

[12] Daniel McGehee, Elizabeth Mazzae, and G Baldwin. “Driver Reaction Time in Crash Avoidance Research: Validation of a Driving Simulator Study on a Test Track”. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 44 (July 2000). DOI: 10.1177/154193120004402026.

[13] Eliraz Orfaig, Inna Stainvas, and Igal Bilik. *Enhanced Automotive Object Detection via RGB-D Fusion in a DiffusionDet Framework*. 2024. arXiv: 2406.03129 [cs.CV]. URL: https://arxiv.org/abs/2406.03129.

[14] *Parallax standard servo*. URL: https://nl.mouser.com/datasheet/2/321/900-00005-Standard-Servo-Product-Documentation-v2.-462659.pdf.

[15] Francesco Pittaluga et al. “A MEMS-based Foveating LIDAR to enable Real-time Adaptive Depth Sensing”. In: *CoRR* abs/2003.09545 (2020). arXiv: 2003.09545. URL: https://arxiv.org/abs/2003.09545.

[16] Yingying Ran et al. “A Review of 2D Lidar SLAM Research”. In: *Remote Sensing* 17.7 (2025). ISSN: 2072-4292. DOI: 10.3390/rs17071214. URL: https://www.mdpi.com/2072-4292/17/7/1214.

[17] Thymon Rhemrev et al. “ELLAS: Enhancing LiDAR Perception With Location-Aware Scanning Profile Adaptation”. In: *IEEE Sensors Journal* 25.5 (2025), pp. 8766–8775. DOI: 10.1109/JSEN.2025.3526733.

[18] Charles Toth and Grzegorz Jóźków. “Remote sensing platforms and sensors: A survey”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 115 (2016). Theme issue ’State-of-the-art in photogrammetry, remote sensing and spatial information science’, pp. 22–36. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2015.10.004. URL: https://www.sciencedirect.com/science/article/pii/S0924271615002270.

[19] Zhenming Xie, Woongsun Jeon, and Rajesh Rajamani. “Low-Density Lidar Based Estimation System for Bicycle Protection”. In: *IEEE Transactions on Intelligent Vehicles* 6.1 (2021), pp. 67–77. DOI: 10.1109/TIV.2020.3010728.

[20] *YDLIDAR SDM15 DATA SHEET ALPHA*. URL: https://www.ydlidar.com/Public/upload/files/2024-08-05/YDLIDAR%20SDM15%20Data%20Sheet_V0.1.5%20(240801).pdf.

[21] Hengxu You et al. “Adaptive LiDAR scanning based on RGB information”. In: *Automation in Construction* 160 (2024), p. 105337. ISSN: 0926-5805. DOI: https://doi.org/10.1016/j.autcon.2024.105337. URL: https://www.sciencedirect.com/science/article/pii/S0926580524000736.

[22] Yuxiao Zhang et al. “Perception and sensing for autonomous vehicles under adverse weather conditions: A survey”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 196 (2023), pp. 146–177. ISSN: 0924-2716. DOI: https://doi.org/10.1016/j.isprsjprs.2022.12.021. URL: https://www.sciencedirect.com/science/article/pii/S0924271622003367.