TERRAIN PREDICTION WITH A LOW-COST LIDAR SENSOR FOR MOBILE ROBOTS

R. Edlinger[1]*, A. Nüchter[2]

[1] University of Applied Sciences Upper Austria - raimund.edlinger@fh-wels.at
[2] Robotics and Telematics, Julius Maximilian University of Würzburg, Germany - andreas.nuechter@uni-wuerzburg.de

Commission II

KEY WORDS: 3d scanning, low-cost, terrain prediction, online terrain estimation, ToF sensors.

ABSTRACT:

Terrain modelling influences various aspects of mobile robot navigation. The ability to explore in rough terrain and to recognise ground conditions are essential to perform different activities efficiently, safely and satisfactorily. For this reason, intelligent vehicles and robotic systems need cognitive capabilities to understand the terrain and derive information from it. The information is mostly acquired and processed by very high resolution 3D-cameras and LiDAR sensors which provide full 360-degree environmental view to deliver accurate 3D data. The aim of this paper is to find out whether a low-cost sensor variant can measure sufficient and significant data from the terrain in order to modify the navigation behaviour and provide the correct control commands. In this paper we describe a low-cost sensor with Infrared Time-of-Flight (ToF) technology and 64 pixel depth image. Furthermore, different experiments on the detection of the sensor were conducted and with appropriate filters and signal processing algorithms the environmental perception could be significantly improved. In summary, our results provide both evidence and guidelines for the use of the selected sensor in environmental perception to improve local obstacle detection and terrain modelling, which we believe will lead to a very cost-effective improvement in competence and situational awareness.

1. INTRODUCTION

Optical metrology has become one of the standard measurement technologies for industrial automation and mobile robotics. The areas of application are very diverse and the trend and ingenuity of researchers shows that new areas of application are constantly being found. However, automated driving and free navigation also require that you not only scan the surroundings, but also explore the terrain and ground conditions. The research question is whether it is always necessary to use high resolution cameras and LiDAR sensors to measure reasonable terrain information or is it also possible to make predictions with much less information? The paper presents a sensor evaluation of a low-cost sensor for terrain prediction for mobile robots. As only the terrain directly in front and behind the robot is of interest, only a small amount of the high data point volume of a laser scanner would be used. Furthermore, scanning the previously mentioned area is rather difficult without tilting the scanner and rotating it or using a second one, which would either increase the complexity or the costs dramatically. Due to the low FOV of existing 3D LiDAR sensors (e.g. Velodyne, Ouster), the near areas around the robot cannot be measured. Another possibility would be to tilt these scanners and let them rotate infinitely (Daun et al., 2021). The disadvantage of multi-layer LiDAR sensors and high-resolution cameras is that the appropriate computing power must be available on the robot to process the large amounts of data right away. For this reason, especially in the selection and integration of sensors, the search was on for less expensive and smaller sensors with a UART interface, so that the sensor data can also be processed on a micro-controller and to communicate directly with the low-level-controller at the end. In this paper we introduce the low-cost (€ 124.00) TeraRanger Evo 64px[1] which is used and evaluated as depicted in Figure. 1.

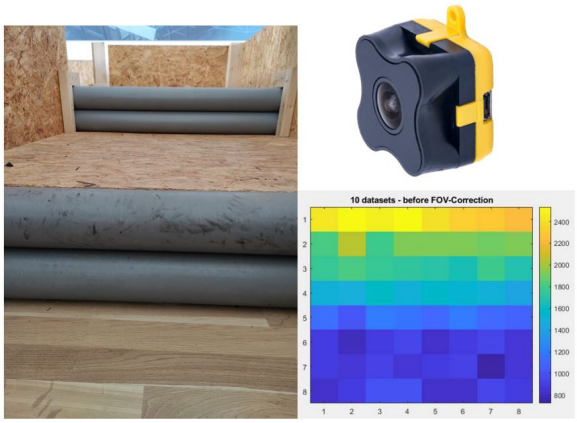

Figure 1. Measuring hurdles – one TeraRange sensor at a distance of 90cm to the obstacles

The contributions of this work are summarized as follows:

- Sensor evaluation, filtering and visualisation of a low-cost ToF sensor for terrain surveys under different ground conditions.
- Data acquisition and processing on a low-cost and low-power micro-controller board.
- Experimental results in indoor and outdoor environments.

2. RELATED WORK

Terrain surveying in mobile robotics is a broad research area and includes recognition, learning and corresponding control. Here we give a brief overview of related work and refer interested readers to articles (Borges et al., 2022),(Sevastopoulos

* Corresponding author
[1] https://www.terabee.com/shop/3d-tof-cameras/teraranger-evo-64px/

and Konstantopoulos, 2022) and (Wettergreen et al., 2008) for more comprehensive information. A new field of research emerges when terrain analysis is supported with artificial intelligence (Lee and Chung, 2021), (Guastella and Muscato, 2020), (Arena et al., 2021). The most commonly used technologies for distance detection and high-resolution distance measurement in autonomous navigation systems are 3D cameras and LiDAR sensors. In recent work (Roriz et al., 2021), Roriz et al. present a comprehensive overview of current market-ready LiDAR sensors for the automotive sector. There is a broad number of commercial LiDAR solutions available in the market. The applications for 3D sensors are very diverse and are used for terrain analysis and prediction in addition to environment modelling. A comprehensive review of terrain traversability can be found in (Sock et al., 2016), (Shaukat et al., 2016) and (Papadakis, 2013). The robot-centric elevation mapping method in (Fankhauser et al., 2014) and (Fankhauser et al., 2018) estimates the terrain profile including confidence bound.

But all the works mentioned so far use a high-resolution and expensive LiDAR or camera sensor. Therefore, we were looking for an appropriate sensor technology that is on the one hand cheap, small in size, low weight and on the other hand easy to integrate. The low-cost sensors available on the market, such as Microsoft Kinect 1, V2 or Azure, Asus Xtion 2 or Intel Realsense, have a much higher resolution, but always have USB 2.0 or 3.0 as their data interface. This has the disadvantage that you have no possibility of operating the sensors with a low-power micro-controller.

3. DATA ACQUISITION AND METHOD

The low-cost TeraRanger Evo is developed for indoor use and delivers a 64px outputs (matrix of 8x8) high-speed distance readings over a 15 degree FOV, with a maximum range of up to 5 meters. The technical specifications of the sensor are shown in the following Table 1:

Principle	Infrared ToF	
Resolution	8x8 matrix (64 pixel)	
Mode	closed-range	fast
Range [m]	0.1 - 5	0.5 - 5
Update rate [fps]	80	130
Accuracy	+/- 5cm 0.1m to 1m +/- 10cm 1m to 3m +/- 15cm >3m+	+/- 10cm <3m +/- 15cm >3m
Resolution [cm]	0.5	
FOV	15°	
Supply voltage	5VDC	
Supply current	80mA - 250mA	
Interfaces	USB 2.0 Micro-B, UART (3.3V level)	
Weight	15 g	
Dimensions	29x29x22mm	

Table 1. TeraRanger Evo 64px specifications.

3.1 Data acquisition

The data acquisition of the TOF-Camera on the OpenCR board[2] is realized via UART interface. Furthermore, the camera is set to close-range-mode as only near obstacles are of interest and distance-ambient-print is used as tests proved this mode to be more accurate. Even though this is the default setup of the camera it is still specified within the setup just to be safe. Because of this setup the camera sends the sensor output as 64 distance values and 64 ambient values. First, a header of 0x11 is sent which indicates the start the frame. Afterwards, there are 128 data-bytes representing the distance values. The data of one pixel is sent as high byte and low byte with each having a 1 at the most significant bit (MSB). Then, the sensor sends 0x13, which is the header of the ambient dataset, followed by another 128 data-bytes for the ambient values. Next, there are padding bytes in the form of 0x80 followed by 8 CRC-bytes. Finally, the sensor sends 0x0A which indicates the end of the frame. This protocol results in a data transmission of 269 bytes. As the last character of the transmission is the end-of-line-character, every byte is read separately and stored into a buffer. Following the setup in Figure 2 of the OpenCR-Board, the program waits until it receives the header 0x11. Then, all received bytes are saved in a buffer. After receiving 0x0A, the length of the buffer is checked as it should be 269 bytes long. If this is not the case the data within the buffer is discarded and the program waits for the next data set. If the buffer is 269 bytes long, the CRC is calculated to check if the transmission was not faulty. According to the user manual of the TOF-sensor the CRC-32-mpeg has to be calculated out of the sent data excluding the last 9 bytes which include the end of frame byte and the 8 CRC-bytes sent by the sensor. The result has to be compared to the value sent by the sensor. This value is constructed by only using the lower nibble of the 8 CRC-bytes. Last, the distances can be extracted from the sent data.

[2] https://emanual.robotis.com/docs/en/parts/controller/opencr10/

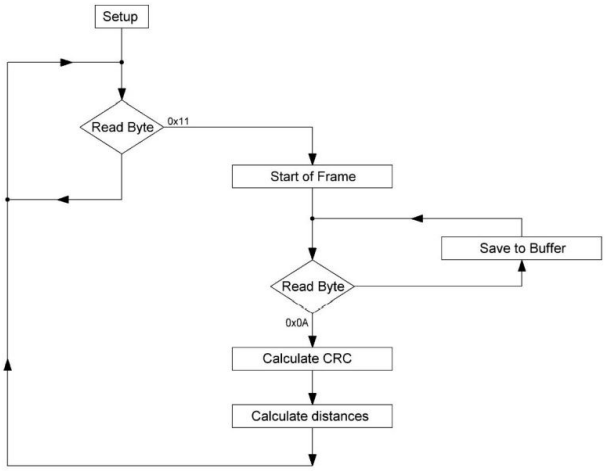

Figure 2. Structure of the data acquisition on OpenCR-Board.

3.2 Method

With the angle φ in Figure. 3 it is simple to calculate the normal distance l by using the tangent function. Using the similarity of triangles, the angle φ can be calculated for every single pixel. For this we assume that the centre of each pixel at the plane is 50mm from its neighbour in X- and Y-direction. Knowing, that the FOV of the sensor is 15° we can now calculate l for this configuration with

$$l = \frac{175\,mm}{\tan(7.5°)} = 1329.52\,mm \tag{1}$$

By calculating w using the X- and Y-coordinates of a pixel measured from the centre of the plane, the angle φ for every

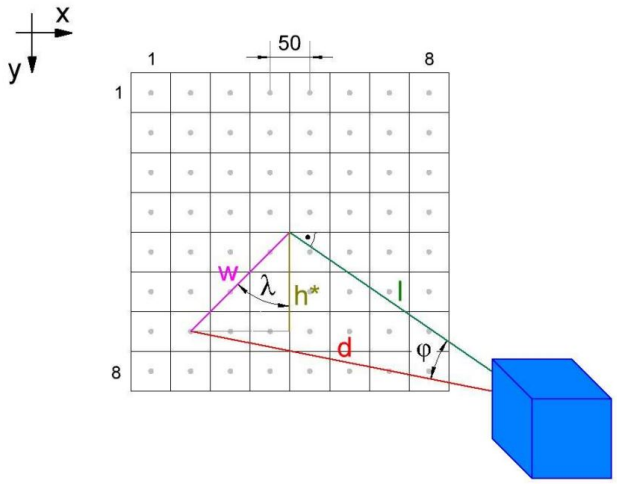

Figure 3. FOV-Correction: geometric references and correction matrix ϕ.

pixel is calculated with

$$\varphi = \arctan\left(\frac{w}{l}\right) \tag{2}$$

This results in the following 8x8-matrix ϕ containing the angles φ to calculate the normal distances l of each pixel out of the measured distances d. With the matrix ϕ in Figure 3 an 8x8-matrix with the corrected distance values can be easily calculated by multiplying the pixel's value with the cosine of its corresponding angle φ. The relevant data to plan an optimal path

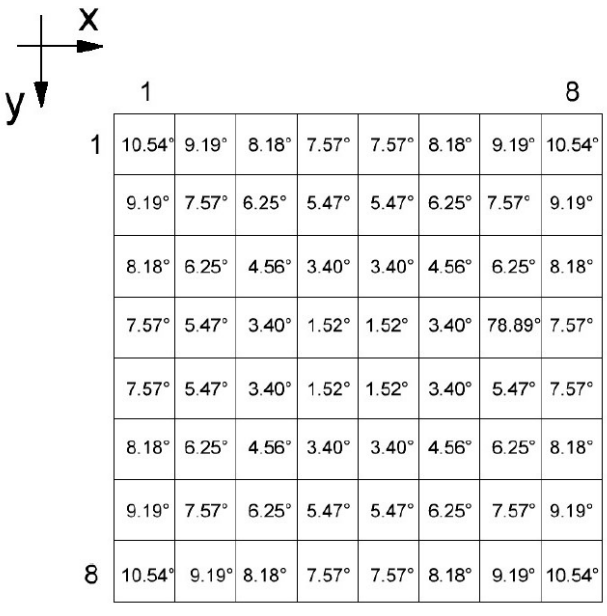

Figure 4. Correction matrix $\wedge$

to cross the obstacle are the height h of the obstacle and the distance l from the robot to the obstacle, see Figure 5. Furthermore, it is useful to calculate β in case of a slope. Comparable to calculating the normal distances of the pixels, the height h^* can be calculated by using the angle λ which can be determined beforehand by using similar triangles.

$$\lambda = \arctan\left(\frac{x}{y}\right) \tag{3}$$

For the calculation we only need the X- and Y-coordinates of the pixels measured from the centre of the plane, see Figure. 3. Using this matrix $\wedge$ in Figure. 4 the height of an object

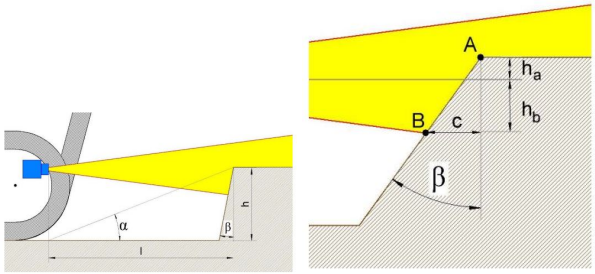

Figure 5. FOV for camera (left) and detailed view of slope calculation (right).

at pixel (x,y) can be determined by calculating $h*$ and adding or subtracting it to the height h_c at which the camera is placed depending on if the pixel is under or above the middle line. The height function we used here is

$$h^* = w \cdot \cos(\lambda) = d \cdot \sin(\varphi) \cdot \cos(\lambda) \tag{4}$$

$$h = h_c \pm h^* \begin{cases} + \text{ for } y \leq 4 \\ - \text{ for } y > 4 \end{cases} \tag{5}$$

Terrain prediction: To calculate the slope β of the obstacle we need the data of the pixel A which is the highest point of the nearest object and the data of pixel B which is the lowest pixel in the same column of pixel A. As shown in Figure. 5 h_a and h_b correspond to h^* of the pixel A and B respectively and c is the difference between the distances l of pixel A and B. Furthermore, we need to divide the 8x8-matrix into a top half (row 0-3) and a bottom half (row 4-7) to calculate the side adjacent to β properly:

$$\beta = \arctan\left(\frac{c}{x}\right) \begin{cases} x = h_a + h_b \text{ if A} \cap \text{B not in same half} \\ x = h_a - h_b \text{ if A} \cap \text{B in top half} \\ x = h_b + h_a \text{ if A} \cap \text{B in bottom half} \end{cases} \tag{6}$$

The flipper angle α can be calculated by using the following equation

$$\alpha = \arctan\left(\frac{h}{l}\right) \tag{7}$$

The 8x8-matrix can be interpreted as images where the pixel values range from 0 to the maximum distance measured, instead of the usual 0 to 255. The structure of the evaluation of the sensor data, obstacle detection and data pre-processing is as follows. This type of sensor is usually less accurate, provides less data and is limited in its FOV in comparison to laser scanners (Figure. 6).

4. EXPERIMENTAL RESULTS

For the evaluation and measurements, a 3D-printed part of the housing was mounted on a tripod to ensure that the sensor was

at the correct height. In the spirit of rapid prototyping, the concept was first tested using MatLab to evaluate the sensor data, as programming on an ARM micro-controller – or as in this concept on the OpenCR-Board – does not have sufficient visualization capabilities to test the prototype properly.

Furthermore, saving data that was evaluated on a computer to be able to present it elsewhere is also complicated. Therefore, the python program that is provided by the manufacturer of the ToF-sensor was used to get the sensor data and modified so that the data is saved in a CSV-file. First, only the data of the ToF-sensor that is facing forward is evaluated in MatLab. Hence, distance values within an 8x8-matrix are used to detect obstacles. The 8x8-matrix can be interpreted as pictures where the pixel values range from 0 to the maximum distance measured instead of the usual 0 to 255. Thus, image processing tools can be used to alter the matrix, but it is advised to only use simple tools as complicated functions might prove difficult to implement on an ARM-micro-controller. The structure of the evaluation of the sensor data and obstacle detection is shown in Figure 6

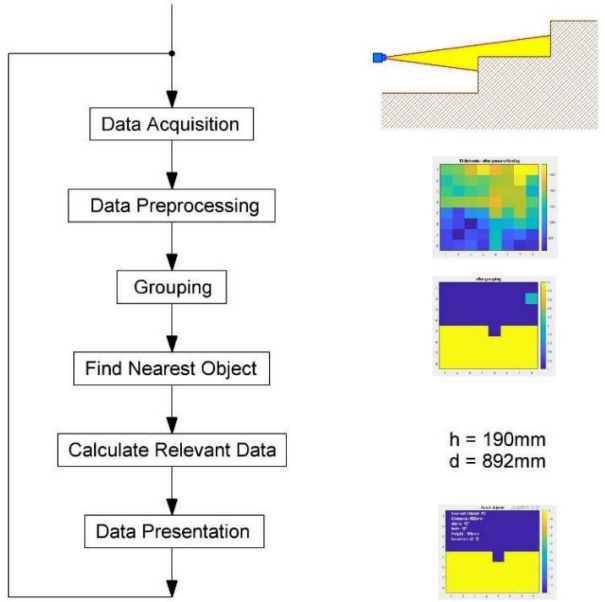

Figure 6. The structure of the evaluation of the sensor data and obstacle detection from a TeraRanger Evo 64px.

The simplest obstacle to encounter is a wall, see Figure. 7. In theory this system should detect one object across the entire 8x8-matrix. The sensor was evaluated against various conditions and objects and analysed in detail using the developed software.

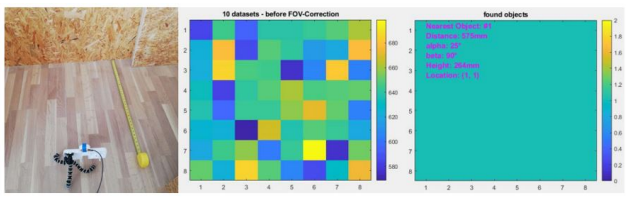

Figure 7. Experimental result wall – one sensor at 30 cm distance.

4.1 Bars

Another measurement was a bar, a 10x10 cm obstacle on the ground that can obstruct the robot's path. The following data was measured with a bar at 30,60,75 and 90 cm with a wall behind it at 175 cm distance. As was to be expected due to the limited FOV, a 10 cm long bar can only be seen from a minimum distance of:

$$l = \frac{h_c - h}{\sin 7.5°} = 68.95\,cm \tag{8}$$

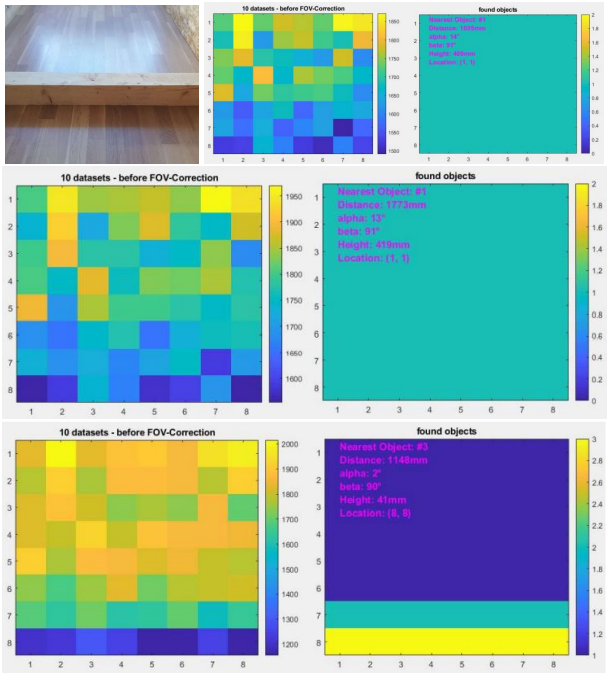

Figure 8. Experimental results: bar – one sensor at 30 cm, 60 cm and 90 cm.

4.2 Hurdles

The hurdles are obstacles that are specified within the MOB1 of the Rescue League rulebook[3]. This is a 20 cm high rolling pipe obstacle to climb up and down, as shown in the following figure. However, due to the diameter of the pipes used, the actual height of the hurdles is 24 cm. As these pipes do not have hard edges like stairs, they might be difficult to detect.

At the distance of 60 cm the data was not really use-able as the sensor sent a lot of errors for the pixels. This is probably caused by the scattering of the light beams caused by the round pipes.

Curiously, measurements at close range and when farther away (30 cm and 90 cm) where reasonably accurate. At 30cm the height of 24cm was nearly at the end of the FOV and the pipes were simply detected as one object. At 90 cm you can clearly see the curvature of the pipe demonstrated by the 4 different objects from row 1-4. Measurements in between seem to be rather difficult and it is more luck-based if the beams are refracted a lot or if you obtain useable data. If this creates problems while actually controlling the flippers or if the measurements at close range and greater distances are enough, needs to be investigated further.

[3] https://rrl.robocup.org/2022-rules/

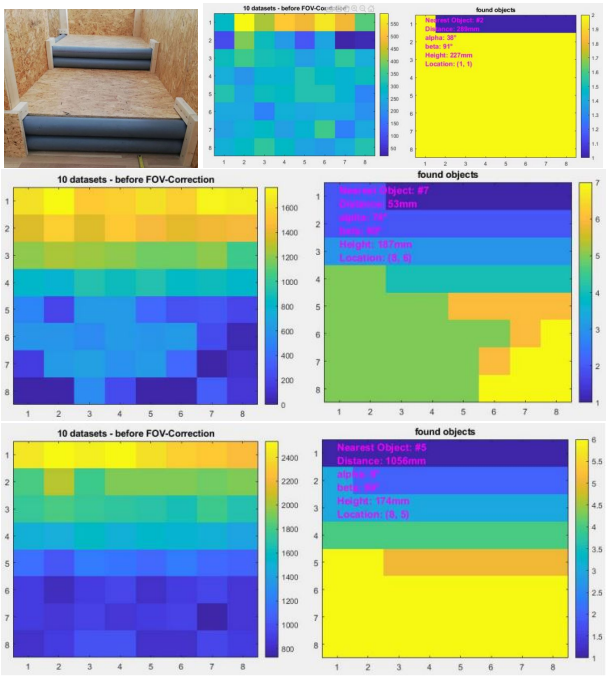

Figure 9. Experimental results: hurdles – one sensor at 30 cm, 60 cm and 90 cm.

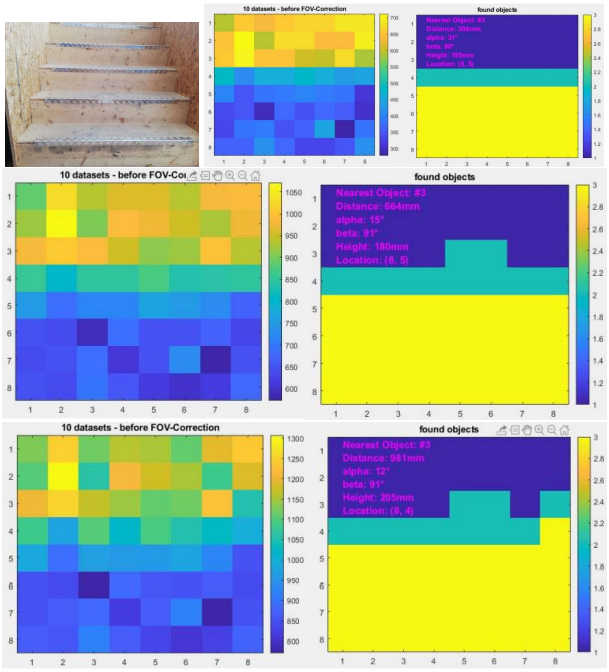

Figure 10. Experimental results: stairs – one sensor at 30 cm, 60 cm and 90 cm.

4.3 Stairs

Stairs are obviously a very common obstacle. Ideally the 8x8-matrix should contain 2-3 objects that are horizontally aligned depending on the distance from the robot to the stairs. One step has a height of 20 cm and a depth of 24 cm. Here the expected result is that the algorithm groups the pixels into two objects. One at the bottom half for the first step and one at the top half for the second step. As one can see above, even though the result includes three objects, they are grouped in a logical manner. The measurements at row 4 are probably slightly off due to the re-fractions at the edge of the step. Yet, the algorithm determines the closest object to be the step at the bottom. Furthermore, the accuracy of the height and distance measurements have satisfying accuracy.

4.4 Extended FOV with two ToF sensors

The measurement at 70 cm detects the bar properly but the distance measurement with one ToF sensor is very inaccurate, see Figure 11. The accuracy seems to increase with the distance between the camera and the bar. Nevertheless this is problematic. For this obstacle, this system does not work and needs to be adjusted. As the measurement of smaller obstacles does not seem to be possible with one sensor at the height of 19 cm due to its limited FOV, the system was expanded by a second camera at a height of 10 cm. This second sensor should in theory detect the smaller objects while the first one is responsible for larger obstacles. Now it needs to be determined if the algorithm still works or if the errors of one camera hinder the result of the other one. Furthermore, it needs to be evaluated whether the data is now simply an 8x16-matrix or if the pixels where the FOV of the two sensors overlap need to be tallied up.

4.5 Different objects

Different materials were also tested to evaluate the sensors behaviour. Some of the different objects measured can be seen below.

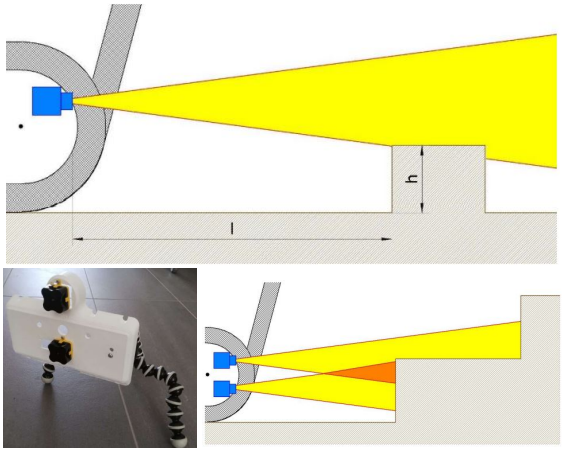

Figure 11. Camera setups and comparison of different measurement configurations - single ToF LiDAR FOV and two sensors with overlapping FOV.

After measuring different materials and objects the following conclusions could be drawn:

- Reflective surfaces, unless they are highly polished, such as the aluminium lid, the TV-screen or other metal surfaces, can be easily detected.
- Mirrors that are perpendicular to the line of sight can still be measured properly.
- Slightly tilted mirrors can be detected within distances of $<$ 80cm but then disappear, due to the reflection of the light.
- Glass or windows can be detected using this sensor.

4.6 Outdoor

Trying to measure similar objects as in section 4.5 outside delivered the following results:

- Surfaces such as wood or cardboard boxes can be detected within distances < 1m;
- Reflective surfaces such as metals are hardly detectable and mostly result in errors;
- The accuracy is worse with about $\pm$10cm instead of $\pm$5cm.

5. CONCLUSION

The experiments have shown that with the TeraRanger Evo it is possible to analyse objects and terrain with only 64 pixels. This restriction also makes it possible to operate two sensors simultaneously and to further analyse the data on a micro controller. The low-level integration then also has the advantage, that the overall system is quickly ready to go once powered up and easy to use. The experiments also showed that the data from several measurements had to be averaged to reduce the noise and then the distortions are corrected by the FOV. After calibration, the pixels are grouped into objects using connectivity-8 if their distances are within a certain range. Finally, the closest object is searched for and its dimensions to the obstacle are calculated.

Initial tests have shown that such a minimalist system with a sensor that provides only 64 data points per measurement is probably sufficient to reliably detect objects and perform rough measurements. If a mobile robot approaches an obstacle from a certain angle, the object would probably be split horizontally into several different objects. This would not be a problem for the flipper control because the angle of the two flippers on one side cannot be adjusted separately. However, to achieve correct and logical results, the algorithm should be adjusted to recognise such an obstacle as one. Furthermore, this problem also applies when hitting slopes. These will most likely be recognised as horizontally aligned objects and not as an object with a slope. Therefore, the algorithm should definitely be adapted to deal with this problem.

With the experiments, all the manufacturer's specifications could be confirmed. The accuracy of the camera is most likely sufficient and most surfaces can be detected without any problems. Since the sensor emits its own infrared light signal, it is not dependent on the lighting of the room. However, since sunlight contains infrared light in its spectrum, outdoor performance is limited. Although this is somewhat problematic, the sensor is still considered suitable for this application because many mobile robot applications are indoors.

REFERENCES

Arena, P., Patanè, L., Taffara, S., 2021. Learning risk-mediated traversability maps in unstructured terrains navigation through robot-oriented models. *Information Sciences*, 576, 1–23.

Borges, P., Peynot, T., Liang, S., Arain, B., Wildie, M., Minareci, M., Lichman, S., Samvedi, G., Sa, I., Hudson, N. et al., 2022. A Survey on Terrain Traversability Analysis for Autonomous Ground Vehicles: Methods, Sensors, and Challenges. *Field Robotics*, 2(1), 1567–1627.

Daun, K., Schnaubelt, M., Kohlbrecher, S., von Stryk, O., 2021. Hectorgrapher: Continuous-time lidar slam with multi-resolution signed distance function registration for challenging terrain. *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, 152–159.

Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M., Siegwart, R., 2014. Robot-centric elevation mapping with uncertainty estimates. *International Conference on Climbing and Walking Robots (CLAWAR)*.

Fankhauser, P., Bloesch, M., Hutter, M., 2018. Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization. *IEEE Robotics and Automation Letters (RA-L)*, 3(4), 3019–3026.

Guastella, D. C., Muscato, G., 2020. Learning-based methods of perception and navigation for ground vehicles in unstructured environments: A review. *Sensors*, 21(1), 73.

Lee, H., Chung, W., 2021. A self-training approach-based traversability analysis for mobile robots in urban environments. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 3389–3394.

Papadakis, P., 2013. Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence*, 26(4), 1373–1385.

Roriz, R., Cabral, J., Gomes, T., 2021. Automotive LiDAR technology: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

Sevastopoulos, C., Konstantopoulos, S., 2022. A survey of traversability estimation for mobile robots. *IEEE Access*.

Shaukat, A., Blacker, P. C., Spiteri, C., Gao, Y., 2016. Towards camera-LIDAR fusion-based terrain modelling for planetary surfaces: Review and analysis. *Sensors*, 16(11), 1952.

Sock, J., Kim, J., Min, J., Kwak, K., 2016. Probabilistic traversability map generation using 3d-lidar and camera. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 5631–5637.

Wettergreen, D., Wagner, M., Jonak, D., Baskaran, V., Deans, M., Heys, S., Pane, D., Smith, T., Teza, J., Thompson, D. R. et al., 2008. Long-distance autonomous survey and mapping in the robotic investigation of life in the atacama desert.