2016 14th International Conference on Control, Automation, Robotics & Vision
Phuket, Thailand, 13-15th November 2016 (ICARCV 2016)

Sensor Fusion of Depth Camera and Ultrasound Data for Obstacle Detection and Robot Navigation

Dariush Forouher
Institute of Computer Engineering
Universität zu Lübeck
Lübeck, Germany

Marvin Große Besselmann
Institute of Computer Engineering
Universität zu Lübeck
Lübeck, Germany

Erik Maehle
Institute of Computer Engineering
Universität zu Lübeck
Lübeck, Germany

***Abstract*—Depth cameras have gained much popularity in robotics in recent years. The Microsoft Kinect camera enables a mobile robot to do essential tasks like localization and navigation. Unfortunately, such structured light cameras also suffer from limitations. Exposing them to direct sunlight renders them blind, and transparent objects like glass windows can not be detected. This is a problem for the task of obstacle detection, where false negative measurements must be avoided.**

At the same time, ultrasound sensors have been studied by the robotic research community for decades. While they have lost attention with the advent of laser scanners and cameras, they remain successful for special applications due to their robustness and simplicity.

In this paper we argue that depth cameras and ultrasound sensors extend each other very well. Ultrasound sensors are able to correct the problems inherent to camera-based sensors. We present a sensor fusion algorithm that merges depth camera data and ultrasound measurements using an occupancy grid approach. We validated the algorithm using obstacles in multiple scenarios.

I. INTRODUCTION

Robots using depth cameras have been hugely successful in recent years, especially in research. They provide very detailed information about the environment while being inexpensive, making it possible to use multiple units on a single robot.

In the past ultrasound sensors had been very actively used in robotic research. However after the advent of affordable laser scanners they have largely fallen out of use.

The drawbacks of structured light depth cameras are twofold. First, they require much more processing power compared to laser scanners. Additionally, and this is the focus of this paper, they have optical limitations. The Microsoft Kinect camera can be blinded by direct sunlight. It cannot detect transparent glass walls. And it cannot handle mirrors very well.

In research this is not necessarily a problem, as one can often choose the surroundings to avoid these issues. When developing mobile robots for wide commercial use however, one might have to provide safety guaranties even in environments that may contain glass walls or overexposure with direct sunlight. Service robots have to handle obstacle detection robustly even under these circumstances.

In this paper, we argue that ultrasound sensors are an excellent extension to depth cameras and able to mitigate the shortcomings of structured light depth cameras. We will present a sensor fusion approach to use depth cameras as well as ultrasound sensors to create an occupancy grid map of the robot's surroundings. Our approach is to aggregate the ultrasound sensor data using the Occupancy Grid Mapping algorithm [1] to create a probabilistic map. That map is then layered with the data obtained with the depth cameras to create an occupancy grid map finally used by the navigation software.

The rest of this paper is structured as follows: Section II outlines our sensor fusion algorithm. Section III shortly reiterates the properties of the sensors used and details our test setup and implementation. In Section IV we present our experimental results. Finally in Sections V and VI we do a final analysis and conclude this paper.

A. Existing work

Before laser scanners became affordable, robot navigation was built mostly on ultrasound sensors. Works like [1], [2], [3], [4] and [5] describe how to localize and navigate a robot using only ultrasound sensors. While these systems worked, they were limited by the inherently imprecise measurements of ultrasound sensors.

In recent years robot navigation has focused on camera sensors like the Microsoft Kinect or similar devices. The results are mixed, but in environments well suited to these cameras navigation (like office floors) results have generally been very good [6], [7], [8].

Combining ultrasound sensors with (two-dimensional) cameras to improve obstacle detection was done (among others) by [9] and [10] with promising results. This is the work most similar to what we present in this paper. However, they used two-dimensional cameras and extracted features from those images. In [9] the edge detection filter used resulted in a map with sparse objects (lines representing walls). In contrast to that we used depth cameras providing three dimensional point clouds of the environment and used these clouds to create a detailed occupancy map of the surroundings.

II. SENSOR FUSION

The main purpose of this sensor fusion approach is to compensate the shortcomings of a structured light depth camera with the aid of ultrasound sensors. Given the fact that depth cameras are much more accurate and detailed than ultrasound sensors, they will be used as the primary sensing device. The ultrasound range data is only used if the data from the depth camera is determined to be not sufficient (see below).

978-1-5090-3549-6/16/$31.00

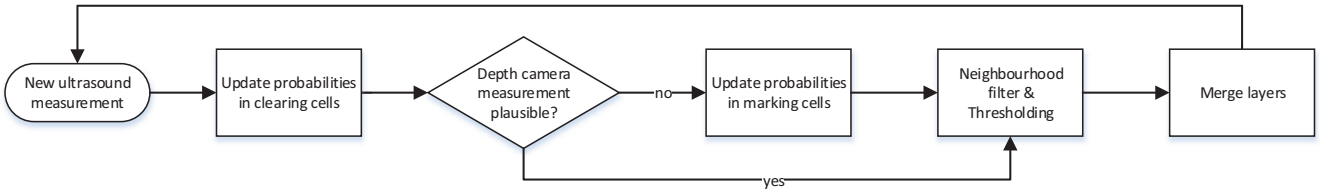


Fig. 1. Shows the update sequence of the ultrasound costmap layer. This process is done for every new measurement, generated by each of the ultrasound sensors. At the beginning, every new measurement leads to a new separation of grid elements into marking and clearing cells, which then get a probability from the sensor model assigned. Afterwards we update the probabilities of the clearing cells in the *Ultrasound Layer* with the new range measurement. Subsequently we check whether the measurement of the depth camera is plausible, as described in Section II. If the data from the depth camera is not plausible we use the ultrasound information to update the probabilities in the marking cells. Otherwise this step is skipped. After that, the probability values contained in every grid element are reduced to a binary free/occupied status. In the final step we merge the *Ultrasound* and *Camera Layer*.

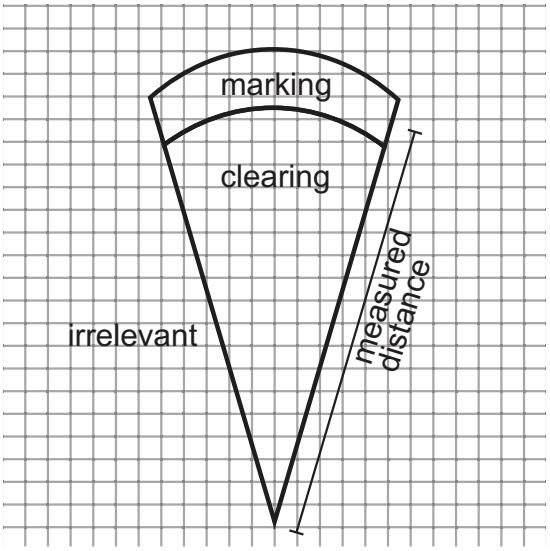


Fig. 2. Shows the grid element distribution of the cells of our map. It consists of three parts. The part outside the ultrasound cone is not covered by the sensor and therefore marked irrelevant. The marking area represents a small slope around the area in which the measured obstacle is presumably located. For all cells within the marking area the occupation probability is increased inside the grid map. The part between the marking area and the sensor head is denoted as the clearing area. Cells in this area lower the occupation probability.

The mapping process is divided into the steps depicted in Figure 1. Since an occupancy grid mapping approach [1] is used, the single grid cells of the map are initially categorized as marking, clearing or irrelevant cells as shown in Figure 2.

Each grid element inside a small slope around the marking area of the ultrasound cone becomes a marking cell. This means that the measurement is likely caused by an object in some of these cells. All elements between the sensor head and the marking area become clearing cells, because the ultrasound wave passed through all those elements without obstruction. It is therefore unlikely that there are any obstacles present in any of these cells. All other elements are irrelevant, because they are located outside of the sensor's field of view.

Each one of the marking and clearing cells gets a probability according to the sensor model depicted in Figure 3. Given a specific grid element and the associated sensor data (i.e., the range measurement and the orientation of the sensor when the measurement was taken), the sensor model returns the probability of how likely it is that this range measurement was caused by an object inside this cell.

After this first step, the information stored for the marking and clearing cells has to be integrated into the occupancy grid map. We use two different layers, that is one grid map for each sensor type. The *Camera Layer* is created by using the depth information from the depth camera. This layer is not probabilistic. For the *Ultrasound Layer* a Bayes map is used to store the ultrasound data individually.

Both layers will be constantly updated throughout the mapping process with new sensor data. After every update step the two maps are merged into a single map. That map is then used for obstacle detection and path planning.

To easily create a one-to-one correspondence between cells the properties of the *Ultrasound Layer* are chosen to match the size and resolution of the *Camera Layer*.

The update process of the *Ultrasound Layer* can be divided in two parts. First, the probability update step of the clearing cells. Second, the probability update step of the marking cells, for which additional cases have to be examined. For the obstacle clearing process on the other hand, no cases have been considered.

Prior to the obstacle marking process, all marking and clearing cells as well as their corresponding elements in the *Camera Layer* have to be analysed. More precisely, the three cases shown in Figure 4 have to be considered. Figure 4 shows the ultrasound cone with the underlying grid. Filled cells represent objects inside the *Camera Layer*.

Figure 4(a) shows case A, in which the *Camera Layer* marks one or multiple cells inside the detection area of the ultrasound cone. Therefore we can assume that both sensors detected the same object and since the depth camera is more accurate, the probability of the marking cells will not be updated.

Case B (Figure 4(b)) occurs if there is an object inside the *Camera Layer*, which is located in front of the ultrasound detection area. In this case we assume that the ultrasound measurement is wrong, since it should have been reflected by the object sensed by the depth camera. Again the probability

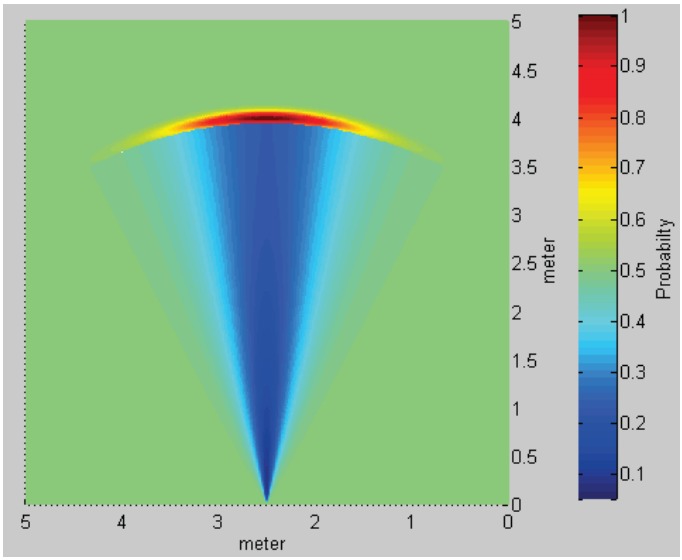


Fig. 3. Shows the used sensor model for the Devantech SRF08 ultrasound sensor. Here a five by five meter area, in which the sensor cone lies, is depicted. The ultrasound emitter is placed at the bottom at 2.5 meter on the x-axis. This model was created by using two multidimensional Gaussian distributions, one for the marking and one for the clearing area.

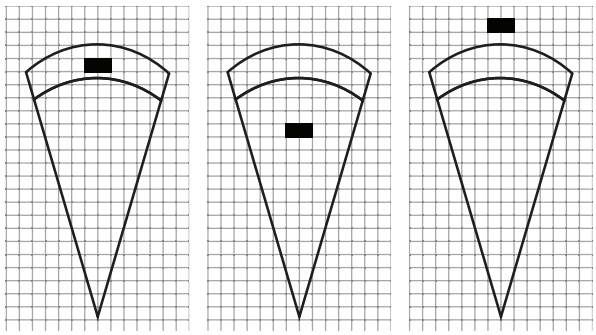


Fig. 4. The three considered cases for the ultrasound-depth correspondence. The underlying grid represents the current map created solely by the depth camera. Filled cells represent objects and white cells free spaces. In the middle of each Figure the cone of a single ultrasound sensor is depicted, which is divided in a lower and upper part. The lower part is just free space, whereas the upper part represents the detection area of the ultrasound, formed by a small slope around the measured distance. 4(a) Shows case A, in which the distance measurement of the ultrasound and the inscribed object of the depth camera match. In 4(b) the object recognized by the depth camera lies inside a space which is considered unoccupied by the ultrasound sensor. Case C is depicted in Figure 4(c) where the ultrasound sensor measures an obstacle, although the depth camera has not inscribed any object in this distance or in front of it.

of the marking cells will not be updated.

Case C (Figure 4(c)) is the most interesting case. Here the ultrasound waves were reflected by an object, which has not been detected by the camera. As the obstacle has not been detected by the camera, the probabilities of the marking cells are updated in this case.

In conclusion, if case C occurs but case A and B do not, the information inside the marking cells will be used to update the probabilities of the *Ultrasound Layer*. Otherwise those ultrasound measurements will be discarded.

For the update steps we used the Occupancy Grid Mapping approach [1]. Specifically we took the equations and assumptions from [11].

The next step is to apply a neighbourhood filter to the *Ultrasound Layer* to reduce the number of false positive grid elements. Here, every cell will be checked whether it and at least one of its 4-connected neighbourhood cells occupation probability exceeds a certain threshold. This step is used to exclude solitary grid elements and to reduce the probabilistic estimations of each cell to a binary free/occupied status.

Finally the *Ultrasound Layer* is merged with the *Camera Layer*.

III. IMPLEMENTATION

All tests were performed on a custom robot platform equipped with multiple *ASUS Xtion Pro Live* depth cameras (which are very similar to the Microsoft Kinect) as well as an array of Devantech SRF08 ultrasound sensors. The depth camera has VGA resolution and about 55 degree field of view. The positioning of the ultrasound sensors is depicted in Figure 5. To aid the sensor fusion, the field of view of the ultrasound sensors is included in the field of view of the depth cameras. Ultrasound sensors work by sending out a high-pitched sound pulse and measuring the time delay of the response, as it is reflected back by an object. As the speed of sound is known, one can approximate the distance to the observed obstacle. Only objects in the field of view of the ultrasound sensor reflect back a response. In theory multi-echo information could be used to pinpoint the distance to multiple obstacles. However for simplicity reasons we did not attempt to model this and only use the first echo response returned to us by the ultrasound sensor. The sensors used have a relatively narrow opening angle of 30 to 40 degrees. This helps to pinpoint the origin of the echo response.

Unlike the depth cameras, our ultrasound setup was custom-designed for this robot. Only the ultrasound sensors themselves were standard components. Ultrasound measurements in the array are done sequentially, with only one sensor active at any time. This results in an sample rate of about 8 Hz per ultrasound sensor.

The Robot Operating System (ROS) [12] served as our software platform. More specifically we used *gmapping* and *amcl* and the depth cameras for localizing the robot in a known map. Navigation was done using the *navigation* stack, specifically *move_base* and *costmap_2d*.

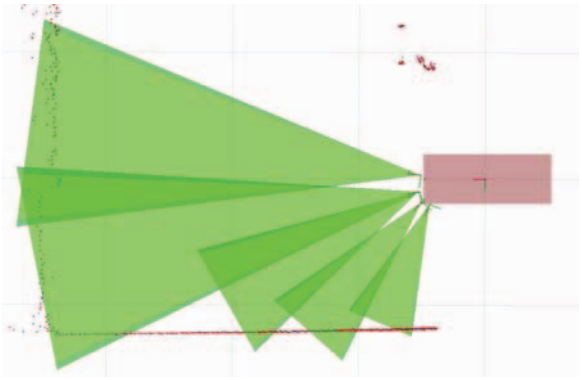


Fig. 5. Field of view of the ultrasound sensors. On the right the robot's dimensions are roughly indicated. The field of view of the depth cameras covers the field of view of the ultrasound sensors.

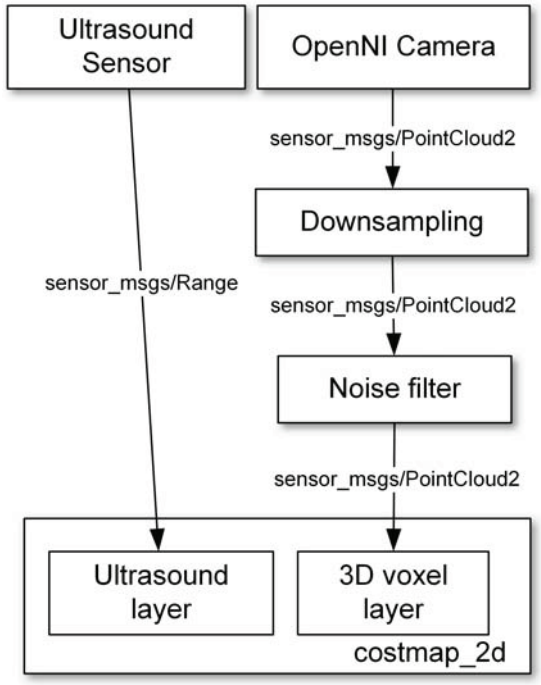


Fig. 6. Data flow of sensor data to the costmap. Boxes indicate software modules. The text aside the arrows hints the ROS datatype of the message.

For our tests we used a *rolling window* costmap that moved with the robot. It has a size of 6.0 x 6.0 m and a resolution of 0.03 m.

The Bayes probability map is implemented as an additional layer to *costmap_2d*. That layer also performs the sensor fusion as outlined in the previous Section and uses the API of *costmap_2d*. Specifically, the layered costmap architecture presented in [13] was used as the foundation for our implementation.

Obstacles observed by the depth cameras are stored using the *voxelgrid* layer, one of the costmap layers available in ROS by default. We performed preprocessing on the the depth pointcloud before handing it over to the costmap, as detailed in Figure 6. First we do temporal and spatial downsampling of the camera point cloud to reduce the resource requirements. Then we apply a statistical outlier filter to reduce the number of false positives. The resulting point cloud is split into parts and sent to *costmap_2d*, which then incorporates the data into a 3D voxel grid.

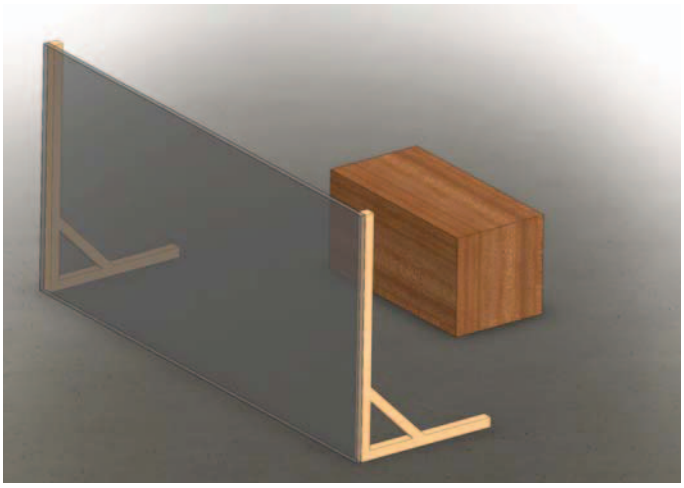


Fig. 7. Illustration of the test setup. Corresponds to the obstacle maps shown in Figure 8.

IV. TESTS AND ANALYSIS

In the previous Sections, various limitations of the depth camera were stated. For this paper we did tests in three scenarios in which the depth camera data alone fails to reliably construct an accurate occupancy grid map. In all these tests we manually manoeuvred our robot through the environment.

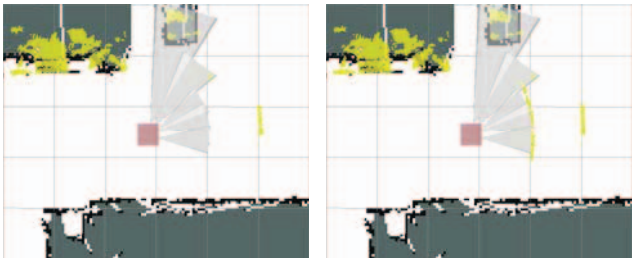
These tests and the resulting maps depicted in Figures 8-10 were created by the mapping algorithms of *costmap_2d*, once with (Figures 8(b) and 9(b)) and once without (Figures 8(a) and 9(a)) our sensor fusion approach. To ensure comparability, we used the same pre-recorded sensor data for both test cases.

A. Scenario 1: Large glass wall

The first and hardest scenario for depth camera is a glass surface. As described in Section III, the ASUS camera used emits an structured light pattern and constructs a depth image from the distortion of that mentioned light pattern. But since the infra-red light just passes through the glass and is not reflected by it, no distortion of the light pattern occurs. Therefore a glass surface is completely invisible for the depth camera.

To highlight this particular problem and its resulting complete blindness of an entire sensor type, we used two different test set-ups.

For the first set-up a 2.0 x 1.0 meter transparent acrylic glass pane (Figure 7) with a stand on each side was used. More specifically the pane was placed orthogonally in the robots trajectory.



(a) Camera only (b) Camera and ultrasound

Fig. 8. Scenario 1. Figure 8(a) shows the map generated only from depth camera data, whereas Fig. 8(b) shows the resulting map using our sensor fusion approach. The two shown maps are created from test set-up one and visualized in the ROS rviz package. In this scenario we placed a two by one meter transparent pane in the trajectory of the robot and additionally placed a box behind the pane. Here the gray and black areas represent the previously recorded ground truth map of our test environment. Also the robot footprint and its ultrasound cones are visible. As a visual aid an underlying grid with one by one meter cells is shown in the background. Lastly the yellow pixel represent the obstacles inscribed by the depth camera and ultrasound sensors. Both maps are similiar with the following important exception: In Fig. 8(b) the transparent pane is clearly visible, whereas in Fig. 8(a) it is completely absent.

In Figure 8(a) the resulting map using only the depth camera is shown. Here, only one of the two stands at the edges of the glass pane is visible, but the pane itself is not. The box behind the pane on the other hand is clearly distinguishable, which shows that the infrared light pattern goes through the pane and is not reflected by it.

Figure 8(b) shows the occupancy map utilizing our sensor fusion approach. Here, all obstacles which were recognized in Figure 8(a) remain incorporated in the map. Due to the use of ultrasound sensor data now additionally the glass pane is clearly visible. This case shows that our sensor fusion approach noticeably improves the detection of transparent objects in our tested scenario.

B. Scenario 2: Small glass wall

For the second test set-up a smaller 1.0 x 0.5 meter pane was placed orthogonally to the wall (Figure 9(a)). There was no gap between the wall and the glass pane.

Figure 9(a) shows the resulting map of this test set-up, again generated only from data of the depth cameras. Here too the transparent pane is not visible at all, which would result in problems for an obstacle avoidance algorithm.

Figure 9(b) on the other hand shows the resulting map, with incorporated ultrasound data. Here the acrylic glass pane becomes clearly recognizable and can safely be bypassed.

An issue in this scenario arises from the large opening angles of our ultrasound cones. The dimensions of the manually inscribed ground truth and the measured diameter of the transparent pane in the map, generated by our algorithm, do not match exactly. Due to the large opening angles of the sensors the diameter of the inscribed obstacles will be often overestimated. This problem could be possibly fixed in future work, by improving the used sensor model or by using a more evenly distributed sensor placement.

In summary, the presented sensor fusion approach significantly improves the map and further enables the detection of objects not visible to the depth camera.

C. Scenario 3: Without depth camera

In the last scenario the depth camera is not available. We rely solely on ultrasound data for obstacle detection. Here it will be examined how the implemented *costmap_2d* plugin behaves without any data from other sensors.

In Figure 10 the resulting map without depth data can be seen. The test set-up was the same as for Scenario 2. Here the wall at the bottom and the acrylic pane on the left side of the Figure are again clearly visible. Yet the map is rather sparse. This arises from the fact that most of our ultrasound sensors were aligned to the front left. Because of this, only the map in front of the robot was updated. All other areas, which were not in the field of view of any sensor remained uncertain.

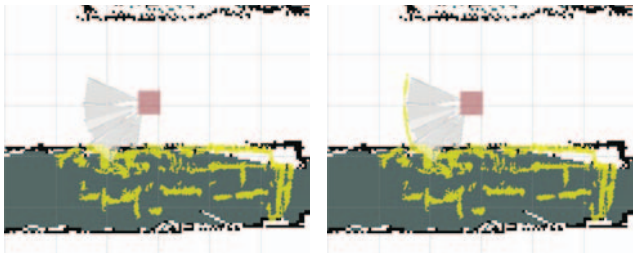
But nevertheless, in case of a breakdown of the depth camera during operation, objects on the trajectory of our robot will continue to be recognized and incorporated into our map. Thus we are able to continue safely navigating through the environment, even if our main sensors (the depth cameras) have broken down. Therefore our sensor fusion approach may also be suitable as a fail-safe for malfunctioning sensors.

The before mentioned sparsity arises from the arrangement of our ultrasound sensors, which mostly pointed to the front-left. The resulting map will very likely be improved by adding more ultrasound sensors to cover the whole front of the robot. Hereby the obstacles would be observed from several directions, which might improve the probability estimation of single grid elements. This might significantly reduce the number of false positive estimations.

V. Remaining Issues and Future Work

While the algorithm worked well in the tested situations we observed a few problems during our tests.

In one case we altered the first scenario by rotating the acrylic glass pane. As soon as the rotation exceeded an angle of 50 degrees, we received incorrect measurements due to



(a) Camera only (b) Camera and ultrasound

Fig. 9. Shows the map created from Scenario 2, again visualized using rviz. In this scenario we placed a transparent pane of one meter diameter orthogonally to a wall. Figure 9(a) depicts the map generated only from depth camera data, whereas Fig. 9(b) shows the resulting map using our sensor fusion approach. As before both maps are nearly identical with the exception of the pane, which was again correctly inscribed in Fig. 9(b) but not in Fig. 9(a). The only issue is that the dimensions of the pane are overestimated, due to wide ultrasound cone openings.

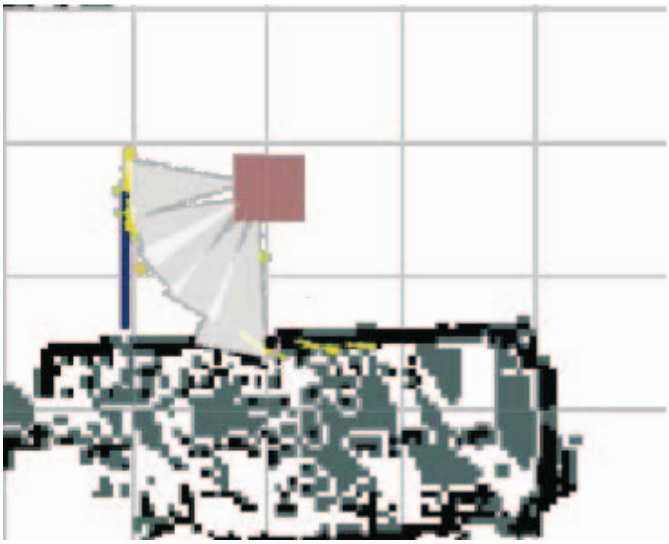


Fig. 10. Scenario 3. Shows the resulting map created without any camera data. Here we blinded the depth camera to simulate a failed sensor. Nevertheless it shows that, even without the data from the main sensor, a rudimentary map is created which allows us to continue navigating through the environment without colliding with obstacles in the trajectory of our robot.

specular reflections, which currently are not handled by our sensor model. By using a more complex sensor model the appearance of those measurements might be reduced.

Another problem, caused by the wide opening angle of the ultrasound sensors, arises by driving too close to the wall. Here protruding objects of a wall alongside the robot are sometimes considered as an obstacle in front of it. This problem is caused by sensing an obstacle in the outer regions of the ultrasound cone and the lack of sensors on this side of the robot to clear these false positive measurements away. But, as mentioned in the previous Section, this is caused by a lack of dissimilar data and can also be avoided by adding more ultrasound sensors with varying orientations.

Finally, positioning of ultrasound sensors is very critical and a robot specific task. We tried ultrasound sensors from different vendors and different positioning setups until we found one giving us acceptable results. Finding such a setup is very time consuming. A systematic approach or guide on how to integrate ultrasound sensors into a robot setup would be very helpful.

VI. CONCLUSIONS

In this paper we argued that depth cameras and ultrasound sensors by themselves each have shortcomings in regards to obstacle detection. But if we combine the data of both sensor types they complement each other very well.

We presented an approach for sensor fusion between these two sensor types that reduces the number of false positives by doing plausibility checks on the ultrasound measurements.

We tested the approach in three different scenarios involving glass panes, which are very hard to detect using structured light cameras. Including ultrasound sensors gave the ability to detect these objects, without increasing the rate of false positives significantly.

REFERENCES

[1] H. Moravec and A. Elfes, "High Resolution Maps from Wide Angle Sonar," *International Conference on Robotics and Automation*, vol. 2, 1985.

[2] A. Elfes, "Sonar-based real-world mapping and navigation," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 3, pp. 249–265, 1987.

[3] J. J. Leonard and H. F. Durrant-Whyte, *Directed sonar sensing for mobile robot navigation*. Kluwer Academic Publishers Dordrecht, 1992, vol. 448.

[4] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[5] J. L. Crowley, "World modeling and position estimation for a mobile robot using ultrasonic ranging," in *Proceedings of 1989 IEEE International Conference on Robotics and Automation*. IEEE, 1989, pp. 674–680.

[6] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *Proceedings of 1989 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1697–1702.

[7] J. Hartmann, J. H. Klüssendorff, and E. Maehle, "A unified visual graph-based approach to navigation for wheeled mobile robots," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, 2013, pp. 1915–1922.

[8] D. S. O. Correa, D. F. Sciotti, M. G. Prado, D. O. Sales, D. F. Wolf, and F. S. Osorio, "Mobile Robots Navigation in Indoor Environments Using Kinect Sensor," *2012 Second Brazilian Conference on Critical Embedded Systems*, pp. 36–41, May 2012.

[9] A. Ohya, A. Kosaka, and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *Robotics and Automation, IEEE Transactions on*, vol. 14, no. 6, pp. 969–978, 1998.

[10] P. G. Kim, C. G. Park, Y. H. Jong, J. ho Yun, E. J. Mo, C. S. Kim, M. S. Jie, S. C. Hwang, and K. W. Lee, "Obstacle Avoidance of a Mobile Robot Using Vision System and Ultrasonic Sensor," ser. Lecture Notes in Computer Science, D.-S. Huang, L. Heutte, and M. Loog, Eds. Springer Berlin Heidelberg, 2007, vol. 4681, pp. 545–553.

[11] S. Thrun, "Learning Occupancy Grid Maps With Forward Sensor Models," *Autonomous Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[12] M. Quigley and K. Conley, "ROS: an open-source robot operating system," *ICRA Workshop on Open Source Software*, vol. 32, pp. 151–170, 2009.

[13] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pp. 709–715.