Praktische Arbeit
Intelligente Robotik

-

Dempster-Shafer-theory-based lidar and radar data fusion to develop an occupancy grid

Thomas Pfaller

January 16, 2024

Contents

1 Introduction

As part of the practical work of the intelligent robotics module, a program is to be written that creates an occupancy grid with the following properties. The occupancy grid shall be circular and the cells have to grow with an increasing radius. The occupation of a cell is given by a probability that is computed based on the dempster-shafer theory of at least two different kind of sensors.

Occupancy grids can be a component in the perception stack of autonomous vehicles. They provide a simple representation of the surrounding environment. Autonomous vehicles use sensors such as LIDAR and RADAR to gather data about their surroundings. One public dataset that provides such data in a suitable way for our task is the NuScene dataset from the company motional.

The code used in this project is available at
`https://github.com/tpfaller/OccupancyGrid`.

2 Dataset

The nuScenes dataset is a large-scale public dataset for autonomous driving. It contains 1000 driving scenes collected in Boston and Singapore, two cities with challenging driving situations. Each scene lasts for 20 seconds. Nuscenes collected about 1.4 million camera images, 390,000 LIDAR sweeps,and 1.4 million RADAR sweeps in 40,000 keyframes. The autonomous vehicle's sensor suite comprises six cameras, one LIDAR, five RADAR, GPS, and IMU, as shown in Figure 1. [1]

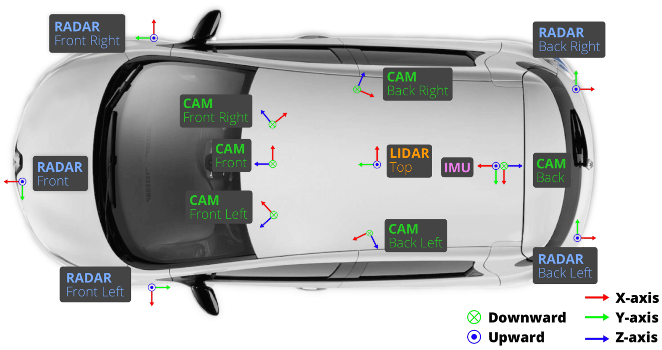

Figure 1: Sensor suite of an autonomous vehicle. [1]

3 Implementation

The flow of the program can be seen in the figure 8. At the beginning one scence from the dataset is selected. Afterwards there is a loop running over the samples that are provided by this scene. For every scene all sensor data are collected and projected into the ego pose of the car. Currently the program is only operating in birds eye view. Therefore is the next step to discard the z-dimension. Depending on the given arguments the pointclouds will be discretized with a polar or a square occupancy grid. A cell is occupied if there is at least one data point. There will be one grid for every data modality. This means at the current state we will receive one grid from the lidar sensor and one grid from all radar sensors. These two grids are then combined based on the dempster-shafer theory. Another step is the edge detection on the fused grid. This is currently a simple laplacian edge detection implemented in opencv. The results are than visualized till the last frame is processed. Then the program ends.

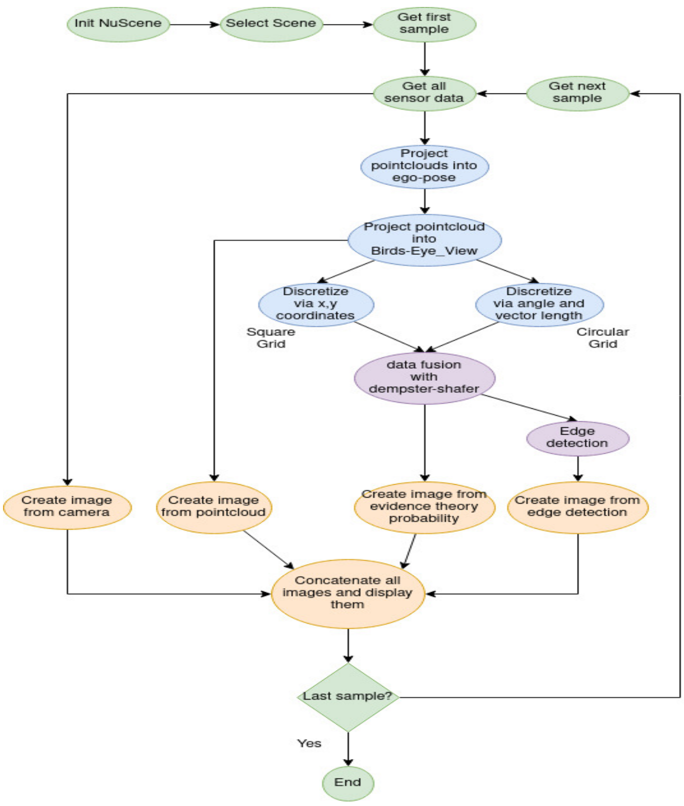

Figure 2: Flow diagram of the program.

3.1 Pointcloud preprocessing

For the preprocessing of the pointcloud are five parameters provided. The first two are a minimum and maximum height. In our particular environment the most interesting objects are other road users. The street as well as object that are very high, example given bridges are not useful and can therefore be filtered to save computation and get better interpretable results. The next two are a min distance along the x and the y axis. This is because the sensors can also detect the car itself. This is also producing noise. Finally the last filterparameter is the max distance. This is filtered by default at 150 because most of the sensor data is within this range. The lidar sensor is also just able to measure things within 100m.

3.2 Calibration

The nuscene dataset is also providing already calibrated data. The rotation and translation matrix for every sensor can therefore easily extracted from the dataset.

3.3 Polar grid

The sensor data is divided into a polar grid with the following steps. At the beginning we initialize two onedimensional arrays. One for the distance to the middle point and one for the angle. For the distance array a parameter for the step size and one parameter to increase step size are provided. The angle resolution is also variable. The second step is to transform the sensor data from carthesian coordinates to polar coordinates. This is done with a norm and arctan function from numpy.

3.4 Dempster-Shafer theory

As mentioned in the introduction. Both sensor modalities are fused based on the dempster-shafer theory. The mass function is currently only using values given by the user. The user has to provide probabilities for the occupation, non-occupation and unsureness for every modality.

4 Experiments

In this section we display the result of our programm for some configurations. Every example uses the eleventh frame of the second scene from the nuscenes mini dataset. The figures are following this particular structure. The outer left image is from the front camera. The second image from left is the raw pointcloud. The blue pixels are lidar points. The red ones are radar points. The green point in the middle shall represent the car. The third image is than the resulting grid after the dempster-shafer combination as displayed in the flowdiagram. The outer right image is than the result of the edge detection.

4.1 Only LiDAR

The figures in this chapter display the results if the dempster-shafer combination accepts only but every occupied cell from the lidar sensor. This scenario is created by assuming an exact sensor for lidar and a completly unsure sensor for radar. With this setting we can compare the results coming from the different grid settings.

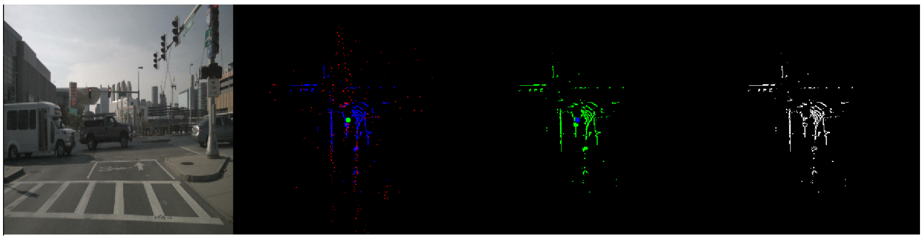

Figure 3: Square occupancy grid using only lidar data.

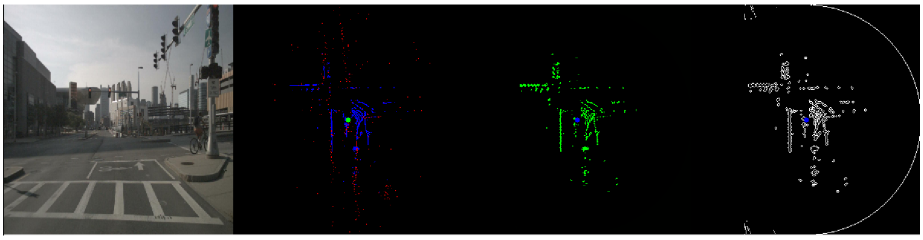

Figure 4: Polar occupancy grid using only lidar data.

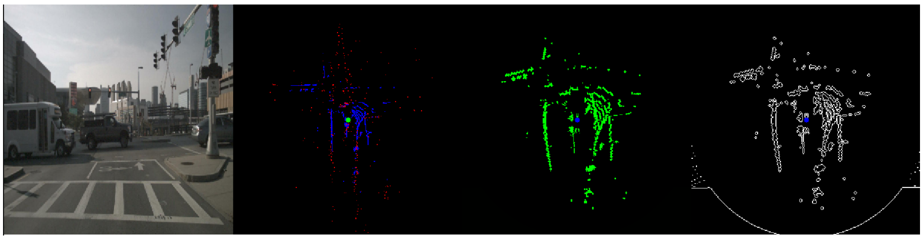

Figure 5: Polar occupancy grid using only lidar data. This grid uses a higher resolution along the angles and the cells are growing with increasing radius.

4.2 Exact Sensors

This section displays the occupancy grids if we assume that every sensor is exact. This assumptions is met, when the probability that the cell is occupied if a point from a sensor is within a cell is one. This configuration leads to very few occupied cells. Because through the combination rule now only cells where both modalities produce points are valid.

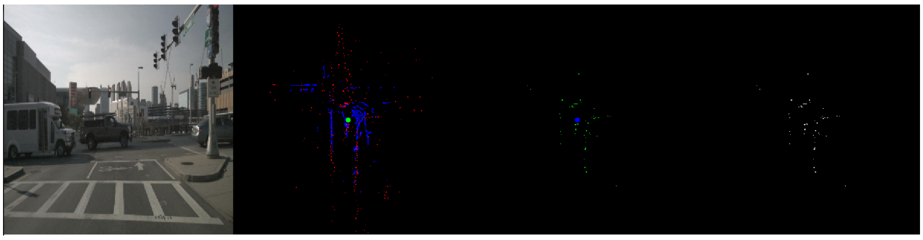

Figure 6: Square occupancy grid with exact sensors assumption

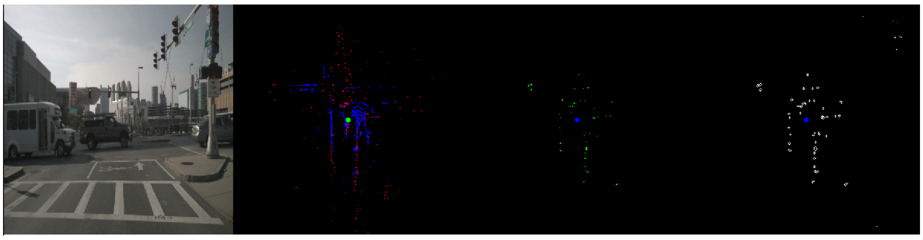

Figure 7: Polar occupancy grid with exact sensors assumption

4.3 Mixed Sensors

Finally we also want to show what is possible if we use reasonable values for the mass function of both modalities.

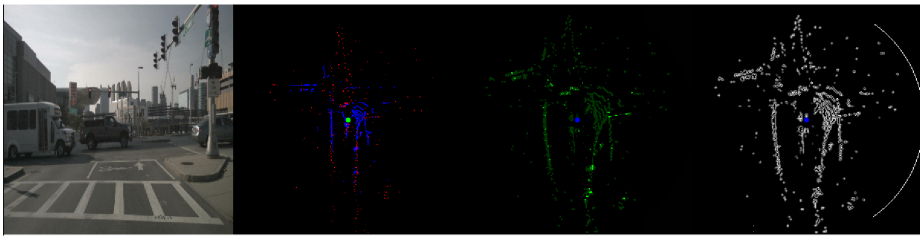

Figure 8: Polar occupancy grid with balanced values for both sensor modalities.

5 Discussion

5.1 Challenges

The first thing that made the fine adjustement a challenging task is the fact that the different sensors provide a very different resolution. Most of the time all of the five RADAR sensor provide together less than 500 points while the LiDAR sensor usually outputs more than 30000 points. This leads to the fact that a fine granular occupancy grid will be extremly sparse for the RADAR data while a coarse occupancy grid will decrease the fine resolution of the LiDAR data.

Another challenge is that the sensors do not collect the data simultaneously. Therefore there is a translation in the pointcloud from moving objects between the different sensors. This can be a problem if the grid is using a high resolution.

5.2 Opportunities for improvement

This section is written to show some possible paths to improve the current status of the program without going into the time-consuming implementation phase.

A thinkable solution to compensate the different spacial resolution of the sensors could be the use of occupancy grids with different granularity for each sensor and a fusion of those.

At the moment the occupancy grid is build based on sensor data to a given time span. The resulting grid could be optimized by combining the sensor data over time.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.