Detection of the Navigable Road Limits by Analysis of the Accumulated Point Cloud Density

Daniela Rato
Oriented by Prof. Vítor Santos
University of Aveiro

1.

INTRODUCTION AND RELATED WORK

LEVELS OF AUTONOMY [1]

- Level 0: No Autonomy
- Level 1: Driver Assistance
- Level 2: Partial Automation
- Level 3: Conditional Automation
- Level 4: High Automation
- Level 5: Full Automation

Road detection with LIDAR

[1] SAE International. LEVELS OF DRIVING AUTOMATION ARE DEFINED IN NEW SAE INTERNATIONAL STANDARD j3016, 2014.

ATLAS PROJECT AND ATLASCAR [2]

WHERE?

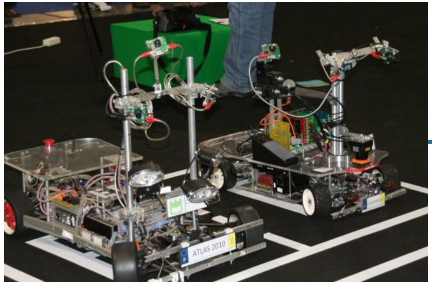
Laboratory of Automation and Robotics (LAR) in University of Aveiro (UA).

WHO?

Students and professors at the Department of Mechanical Engineering (DEM).

OBJECTIVE?

Development of advanced sensing and active systems designed for implementation in automobiles and similar platforms.



Atlas

Atlascar

Atlascar2

[2] http://atlas.web.ua.pt/

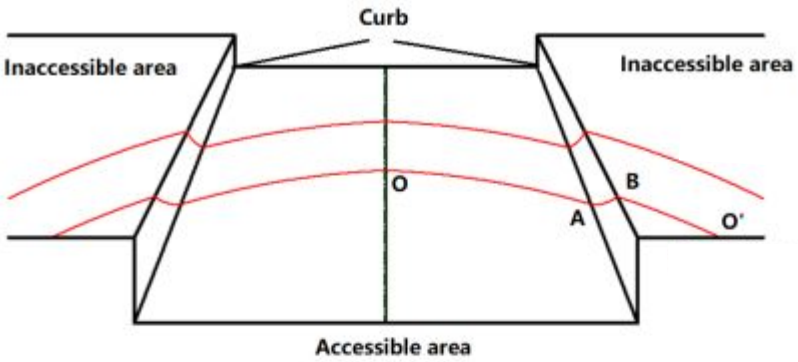
PROBLEM DESCRIPTION

PROBLEM

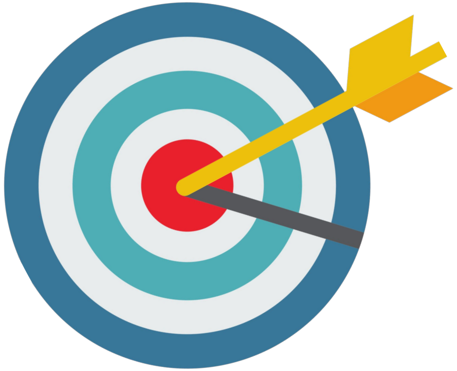
Identify **road limits** by analysing the **accumulated point cloud density.**

HOW?

Define a methodology to detect **physical/hard limits** by applying **edge detection techniques** to point clouds accumulated with the car movement.



OBJECTIVES



1. Develop a robust solution for **road detection;**
2. Test and integrate the solution onboard of the **AtlasCar2;**
3. Develop a methodology to perform **quantitative evaluation** of road limits.

RELATED WORK

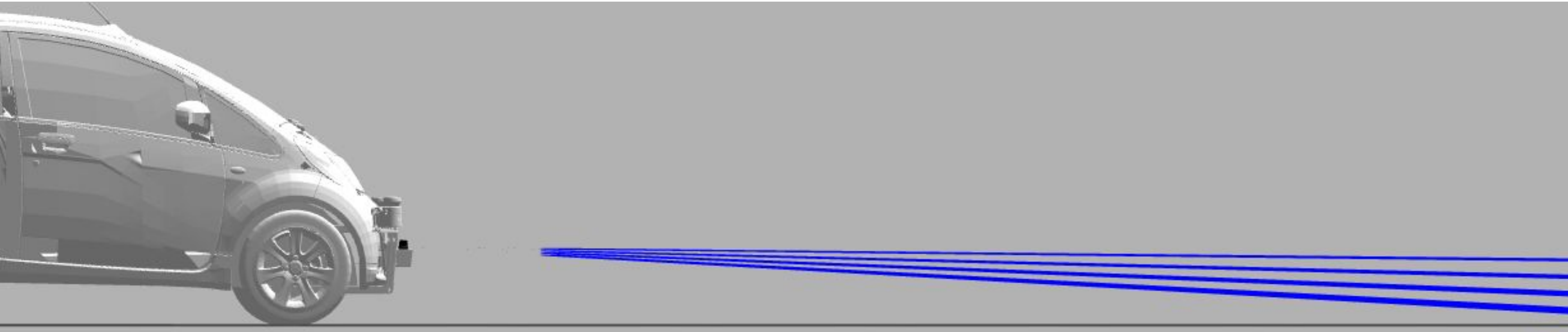
UNIVERSITY OF AVEIRO - TIAGO MARQUES[3]

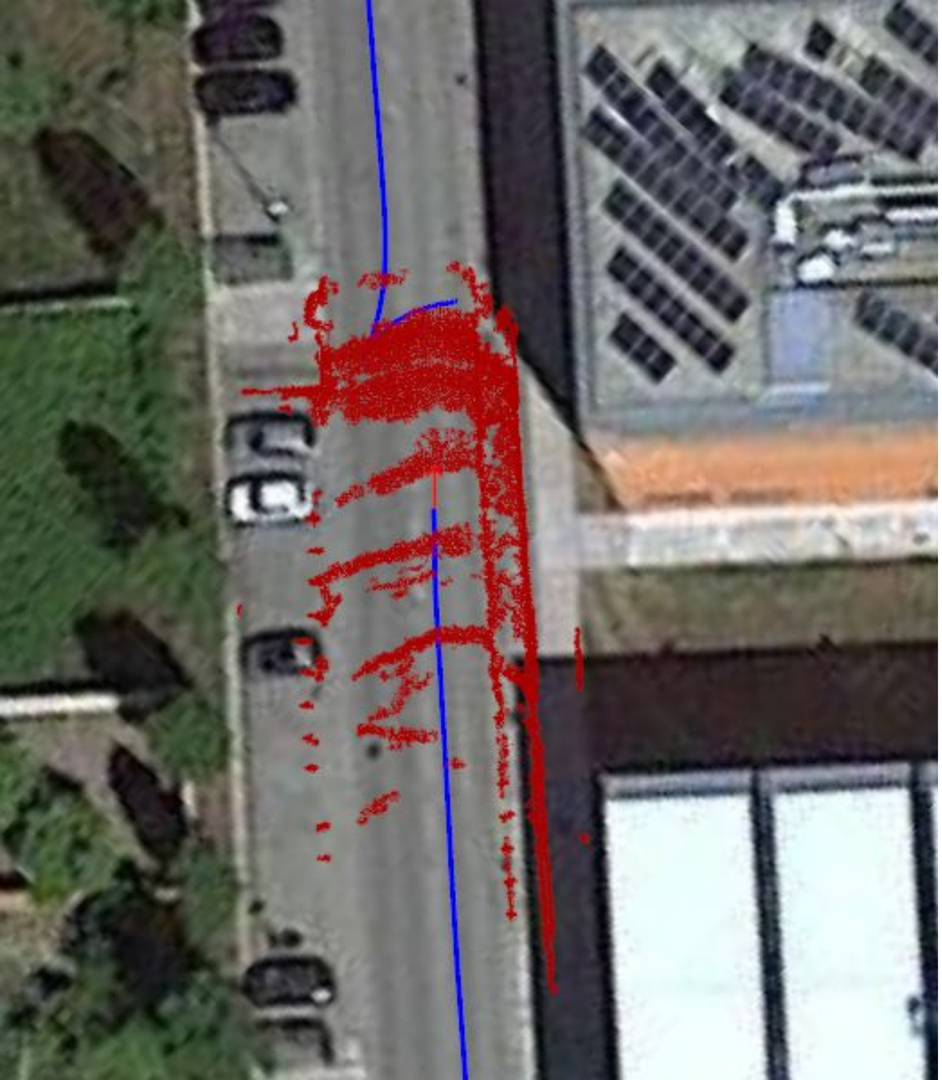
CONCEPT

Accumulate a point cloud with the car **movement**.

ALGORITHM

Eliminate points in voxels with **few neighbors** in a predefined radius with a **static** and **dynamic** parameterization.





RELATED WORK

UNIVERSITY OF AVEIRO - TIAGO MARQUES [3]

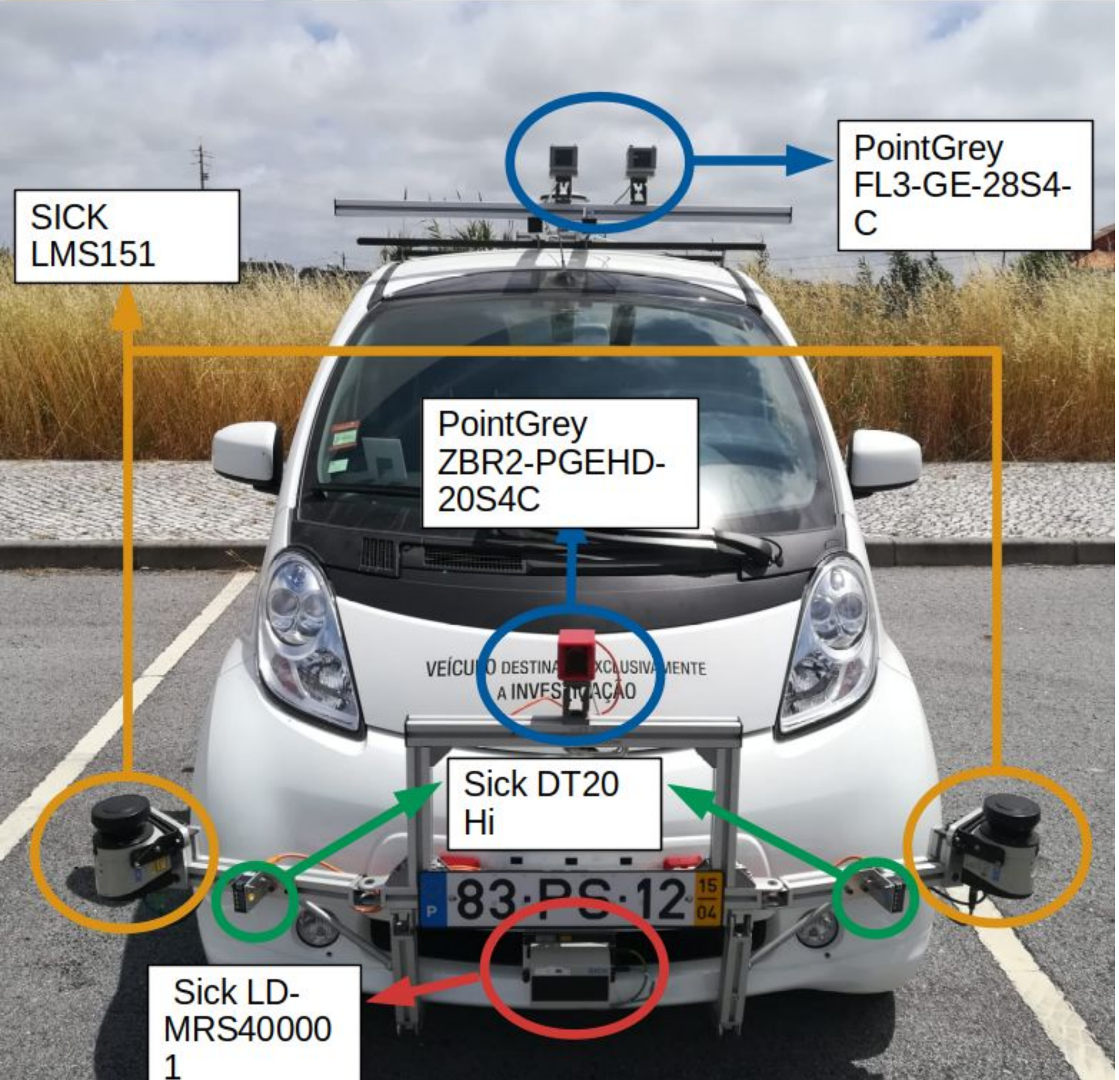
PROBLEMS AND LIMITATIONS

- No accumulation when the vehicle is **stopped**;
- No identification of **negative objects**;
- Change in the LIDAR **inclination** when **accelerating, decelerating and curving**;
- Poor results at **high speed**;

[3] Tiago Marques. Detection of road navigability for ATLASCAR2 using LIDAR and inclinometer data, 2017.

2.

INFRASTRUCTURE



ATLASCAR2

OBJECTIVE

Prototype for research in Advanced Driver's Assistance Systems

CAR

Mistubishi i-MiEV

EQUIPPED WITH

1 Sick LD-MRS400001

2 Sick LMS151

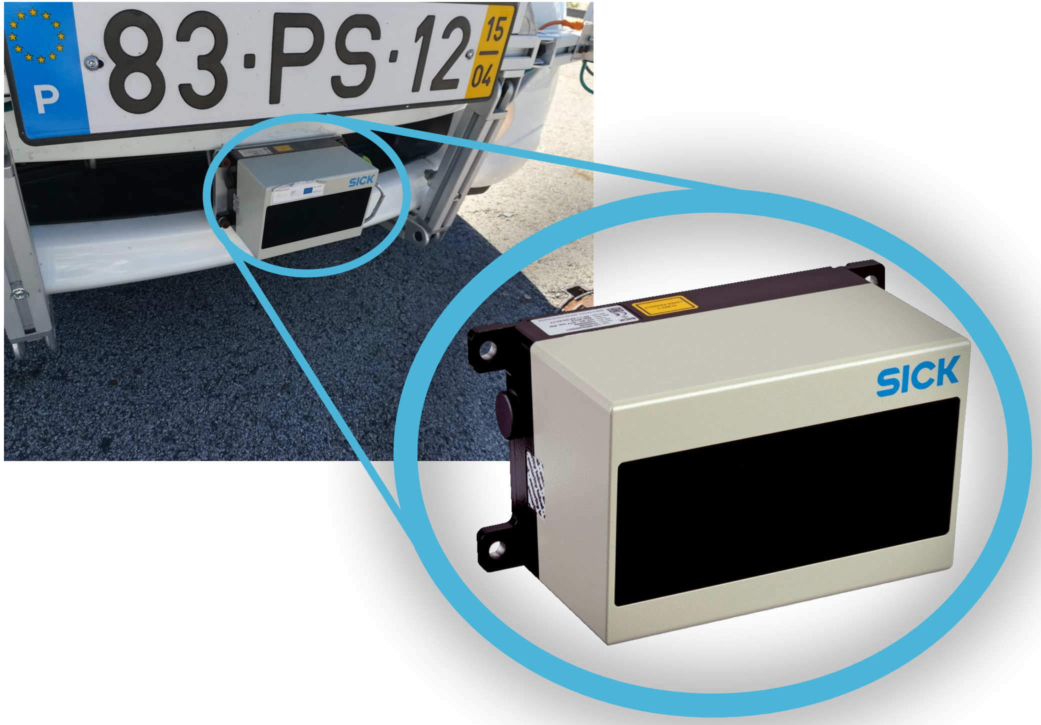
2 PointGrey FL3-GE-28S4

1 PointGrey ZBR2-PGEHD-20S4C

1 Novatel SPAN-IGM-A1+ Novatel GPS-702-GG

4 SICK DT20 Hi

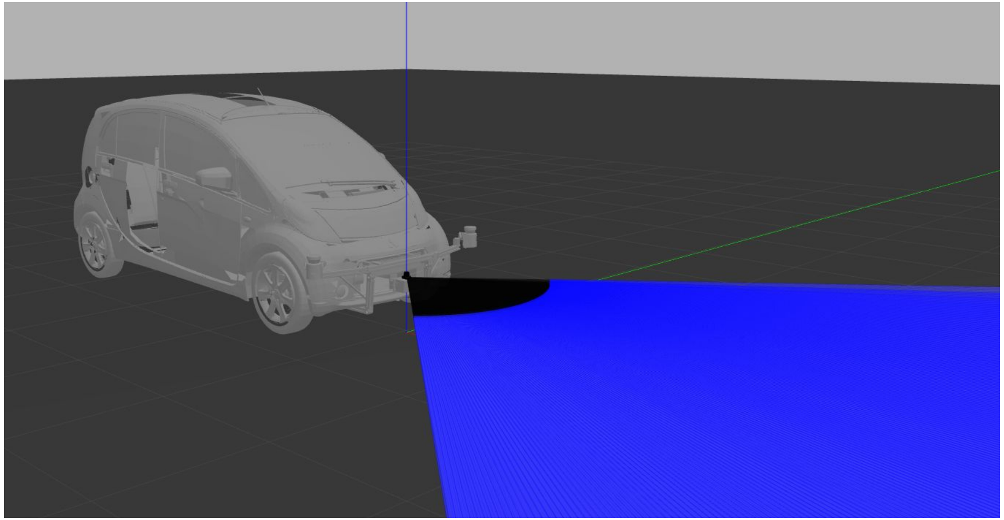
SICK LD-MRS400001



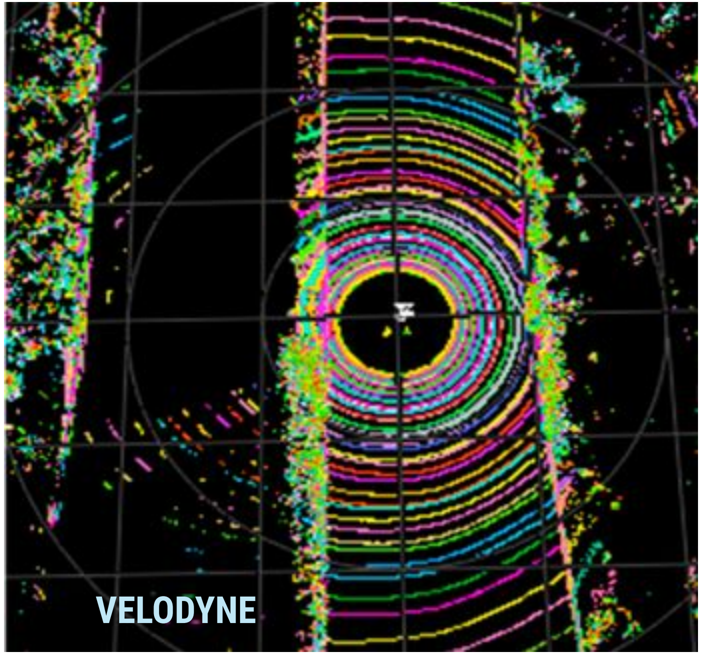
LIDAR FEATURES	
Application	Outdoor
Horizontal aperture	85º
Vertical aperture	3.2º
Scan frequency	50 Hz
Angular resolution	0.5º
Working range	0.5-300 m
Scanning range	50m

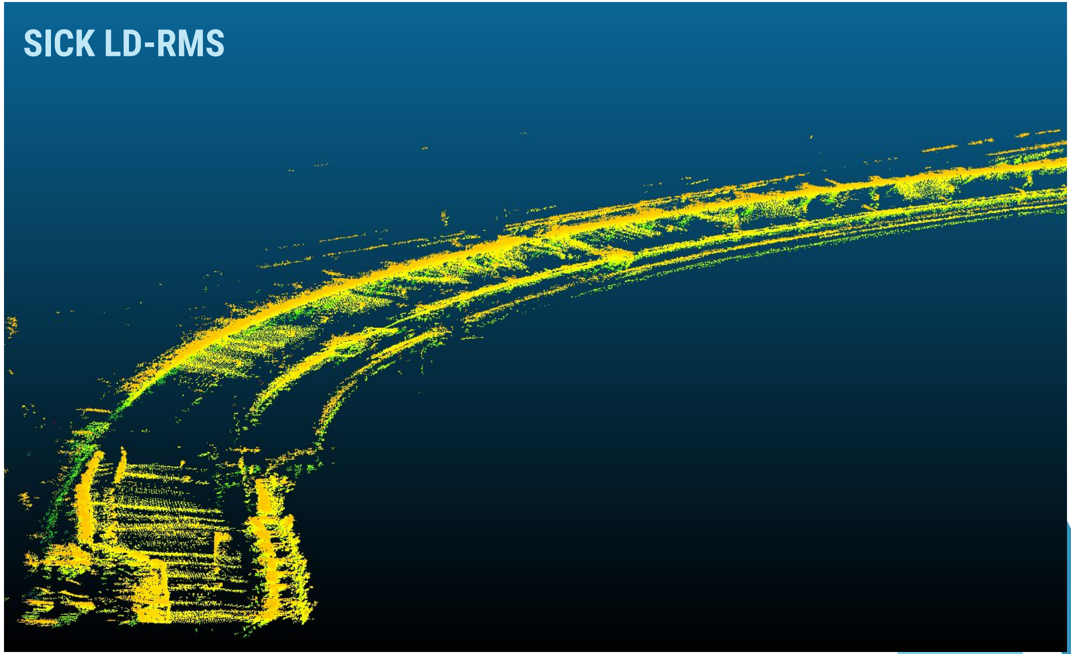
WHY PLACE THE LIDAR CLOSE TO THE GROUND?

The difference of perspective, with the LIDAR placed in the front of the car, close to the ground offers a *UNIQUE* point of view, allowing to focus on obstacles that delimitate the road instead of looking from the top of the car.



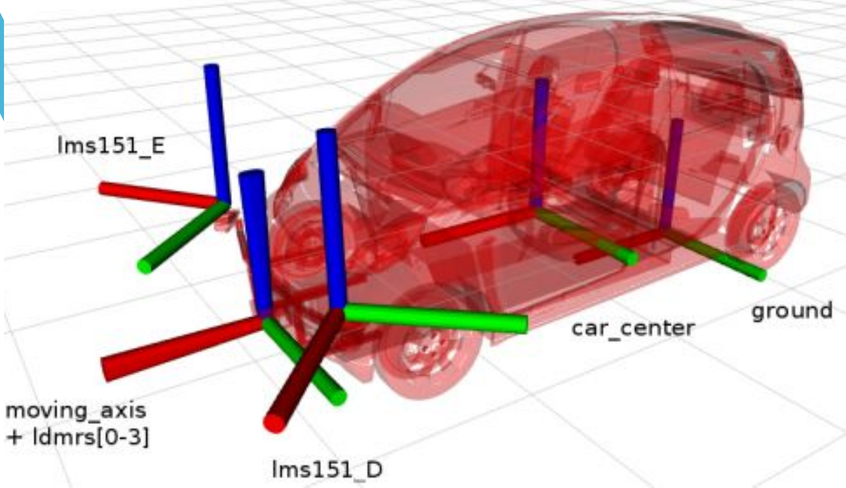
DIFFERENCE OF PERSPECTIVE BETWEEN OUR LIDAR AND A VELODYNE

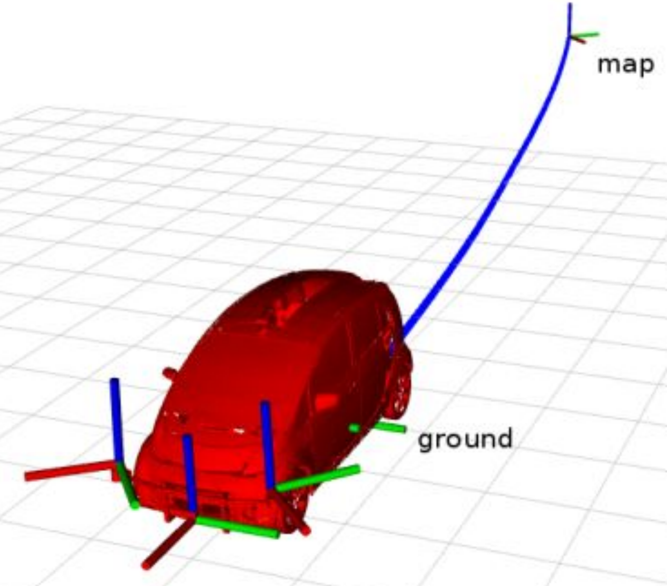


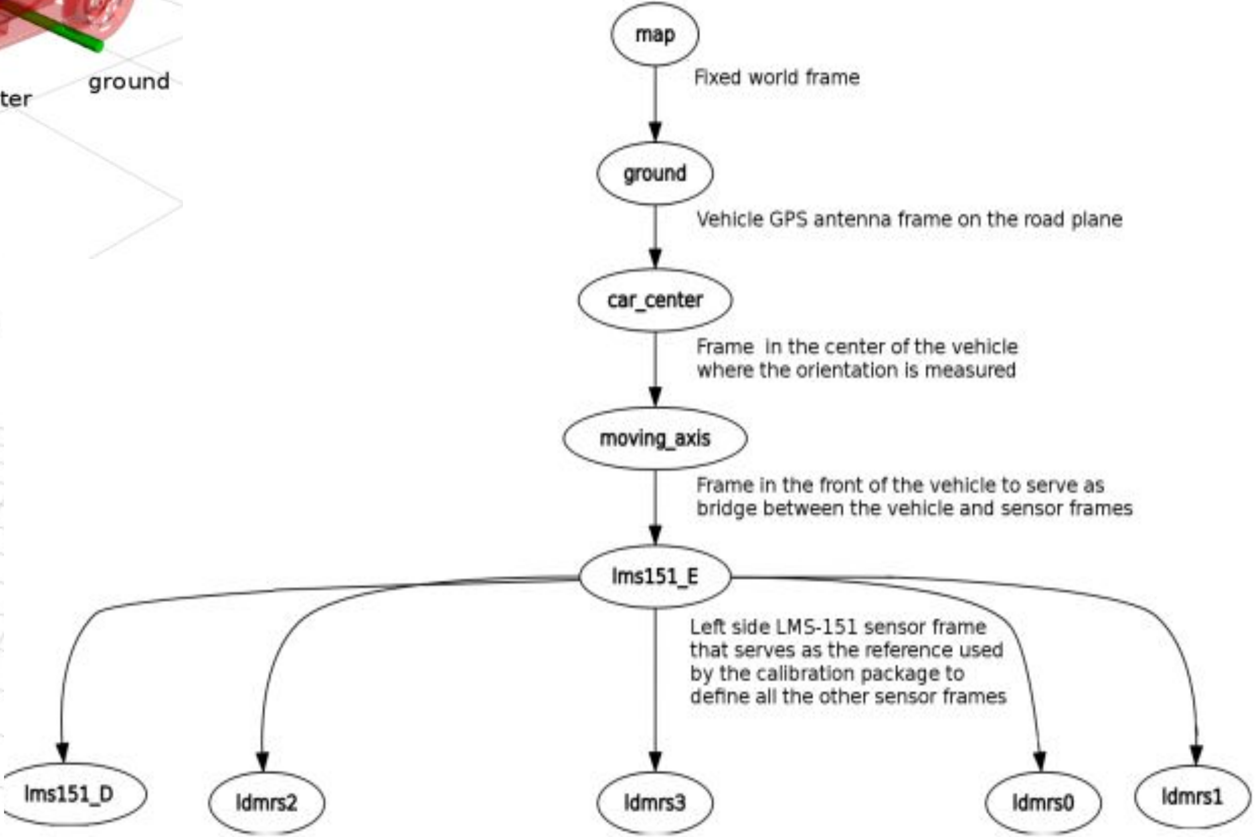


Novatel SPAN-IGM-A1 + Novatel GPS-702-GG

CAR FRAMES


lms151_E
ground
car_center
moving_axis
+ ldmrs[0-3]
lms151_D


map
ground


map
Fixed world frame
ground
Vehicle GPS antenna frame on the road plane
car_center
Frame in the center of the vehicle
where the orientation is measured
moving_axis
Frame in the front of the vehicle to serve as
bridge between the vehicle and sensor frames
lms151_E
Left side LMS-151 sensor frame
that serves as the reference used
by the calibration package to
define all the other sensor frames
lms151_D
ldmrs2
ldmrs3
ldmrs0
ldmrs1

3.

DEVELOPMENT OF A DENSITY GRID

HOW TO CONVERT POINT CLOUDS TO DENSITY?

Point clouds are **computationally heavy** to work with and the need to evaluate the point cloud density brings the questions…

- How to **divide the space** to calculate the density?
- Is it really necessary to evaluate **all the points**?
- What are the best **dimensions** for analysis?

The use of **occupancy grids** answer all that questions, allowing to fully parametrize a grid with the **desired dimensions and resolution** and place the grid in the **correct place**!

OCCUPANCY GRID

- Header
  - Timestamp
  - Frame ID
- Info
  - Resolution
  - Position
  - Height and Width
- Data
  - Row-major vector

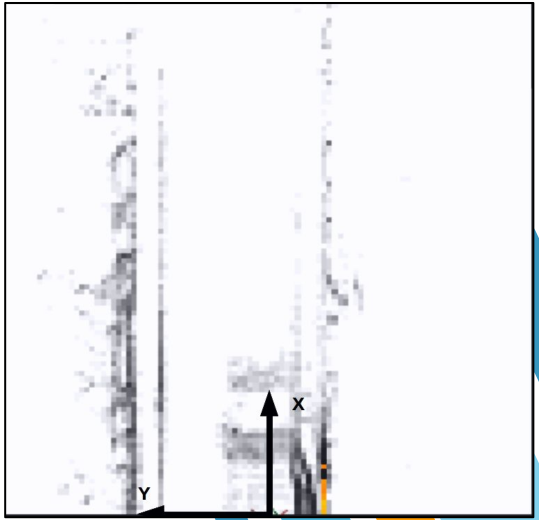
DEFINITION

2-D grid map in which each cell represents the probability of occupancy

DENSITY GRID

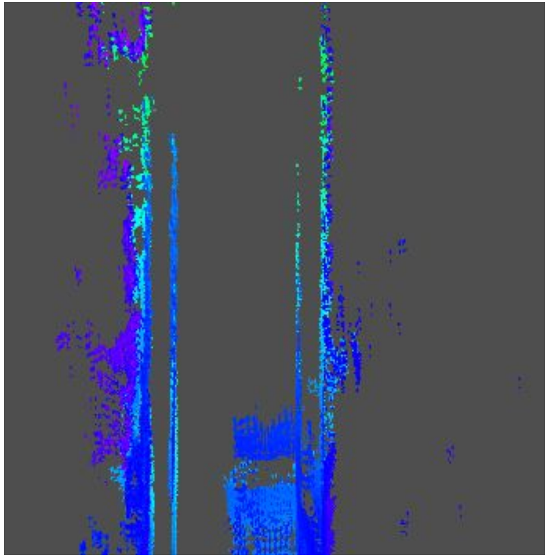
PRINCIPLES

1. The density in each cell equals the **number of points** within the coordinates of that cell;
2. **Normalize** the data vector from 0 to 100;
3. The altitude component is **discarded**;
4. The grid base frame is ***moving_axis***;
5. The grid was defined 40m ahead of the car and 20 m to each side of the car, making a total of **40 x 40 m**.

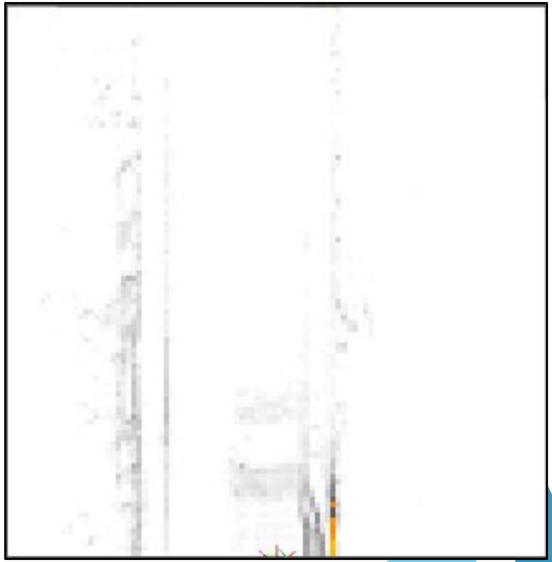


DENSITY GRID

Camera view



Correspondent point cloud

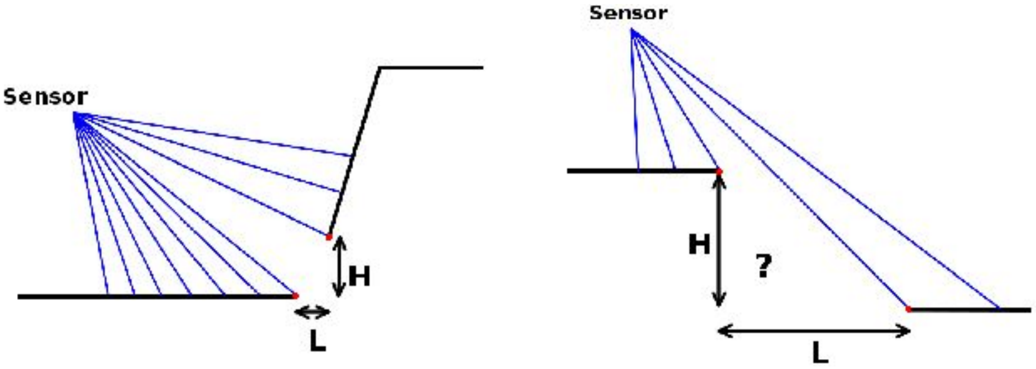


Correspondent Occupancy Grid

4.

GRADIENT AND OTHER EDGE DETECTION ALGORITHMS

HOW TO IDENTIFY NEGATIVE OBSTACLES?



(a) Positive Obstacle

(b) Negative Obstacle

DENSITY VARIATIONS

Positive obstacles → high density zones

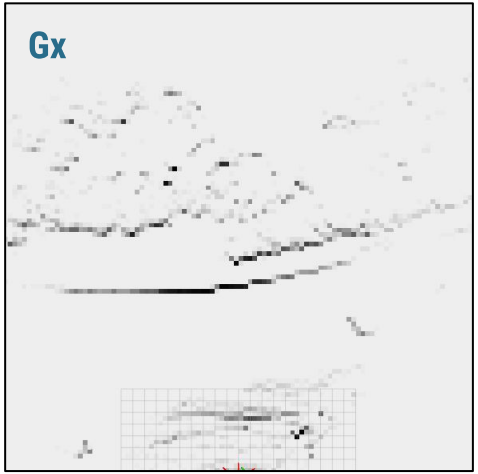
Negative obstacles → shadow zones = zero density

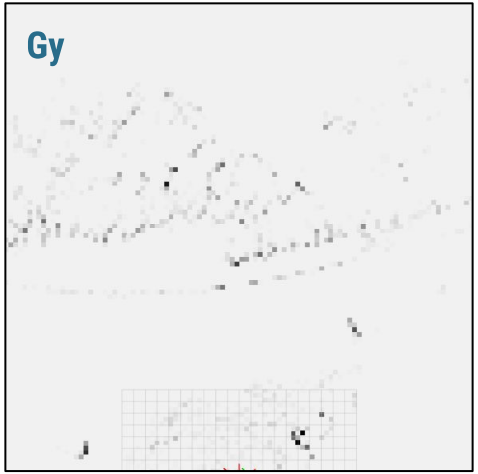
Simple Gradient and other more complex edge detection filters are able to detect both **positive and negative** density changes!

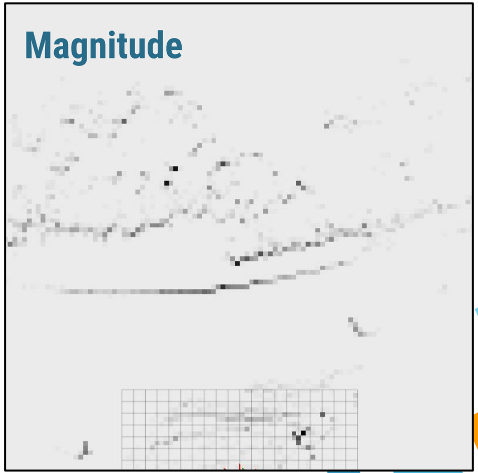
DENSITY GRADIENT

$$F_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad F_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \left\|\vec{G}\right\| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$$

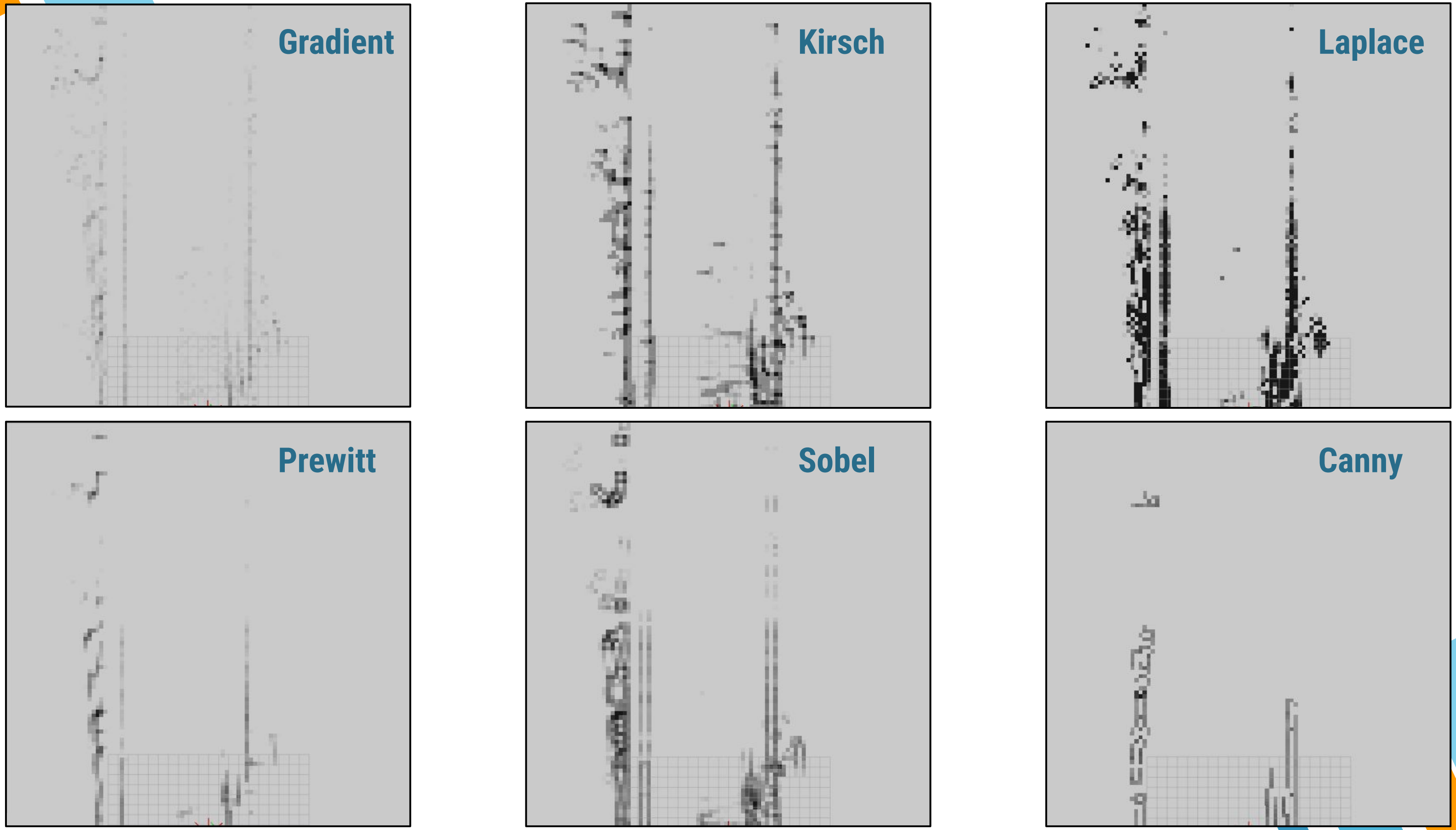
Grid → OpenCV → Image → 2D edge detection filters → Threshold → Grid







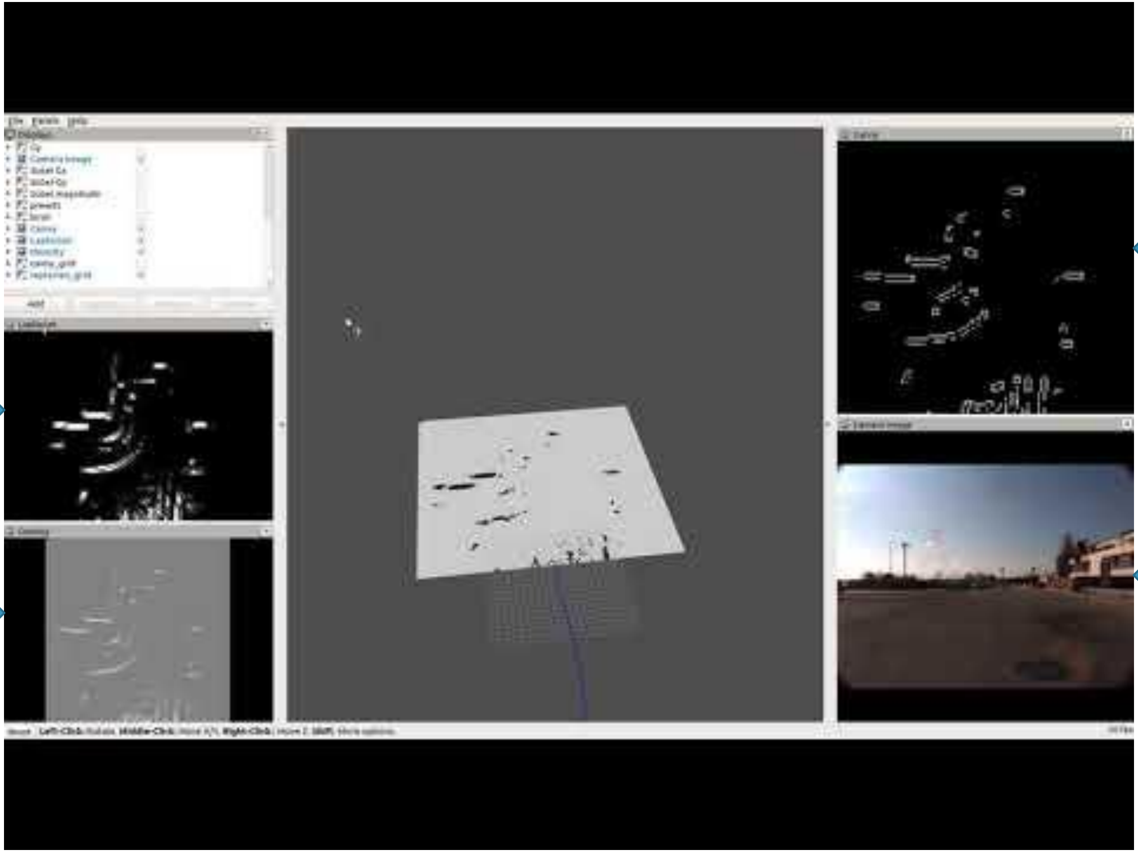
EDGE DETECTION TECHNIQUES



EDGE DETECTION TECHNIQUES

Laplace

Density



Canny

Camera

5.

GROUND TRUTH APPLICATION

HOW TO EVALUATE THE QUALITY OF THE DETECTED LIMITS?

1. Create a KML file with the car path;
2. View the path on Google Earth and draw road limits for a section of that path;
3. Read those limits in the program;
4. Draw a grid with the real road limits;
5. Draw a grid with the detected road limits;
6. Mathematically compare the limits.

CREATING THE CAR PATH

1. Subscribe to the *\gps* topic to gather the car coordinates information;
2. Create a KML file with the correct headings;
3. In each frame the values of latitude and longitude are added to the file;
4. Close the file handler in the end of the program;
5. Visualize the data on Google Earth.

MARKING OF A GROUND TRUTH

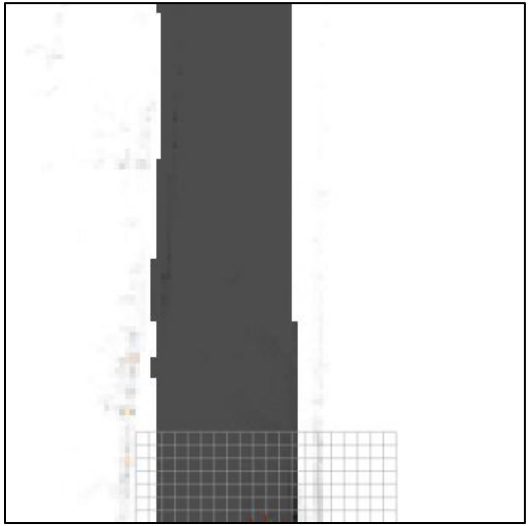
COORDINATES CONVERSION

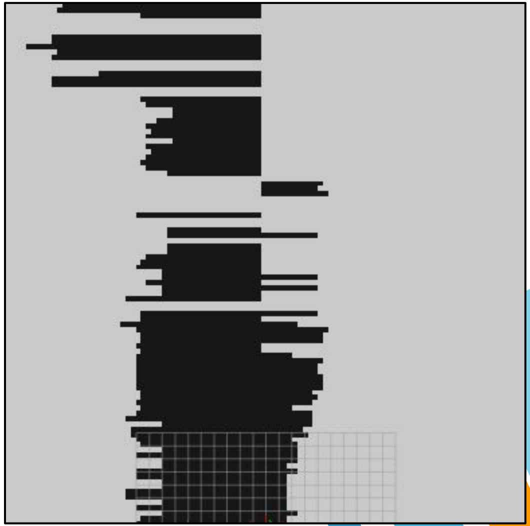
1. Convert the car **latitude and longitude** to the Universal Transverse Mercator (**UTM**) frame;
2. Convert every **road limit** point to the UTM frame;
3. Calculate the **difference** in meters between **each point coordinates and the car coordinates**;
4. **Rotate** the obtained coordinates to the ***moving_axis* orientation** (z rotation of the car azimuth);
5. **Add 2.925 m** to the x coordinate of each point (translation between the *ground* frame and the *moving_axis* frame);
6. Create a continuous line between points with an **interpolation** function.

$$\begin{Bmatrix} x_{correct} \\ y_{correct} \\ 1 \end{Bmatrix} = \begin{bmatrix} \cos(yaw) & \sin(yaw) & 0 \\ -\sin(yaw) & \cos(yaw) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} x_{utm\_point} - x_{utm\_car} \\ y_{utm\_point} - y_{utm\_car} \\ 1 \end{Bmatrix}$$

NAVIGABLE SPACE

Space within road limits, where the car can, allegedly, navigate with safety.

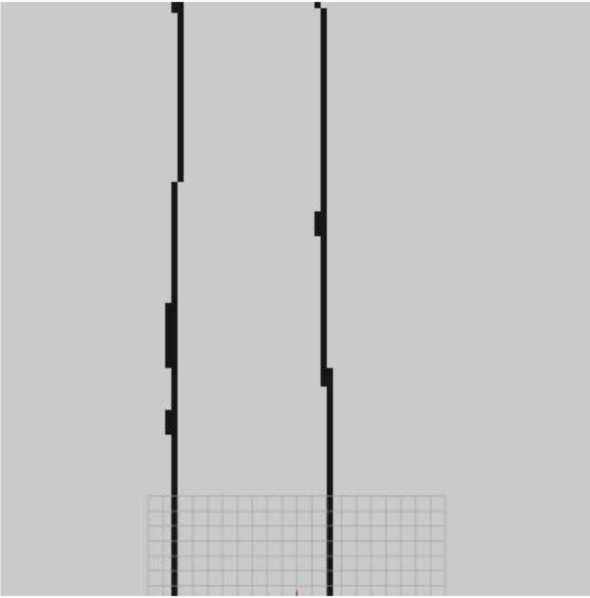




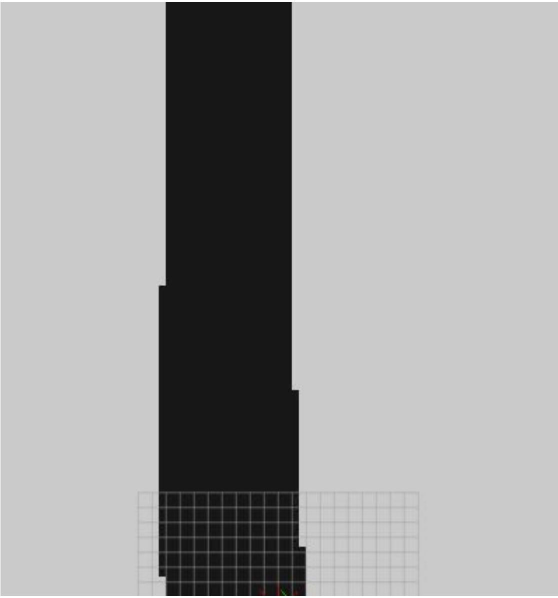
HOW TO CREATE A GRID WITH THE NAVIGABLE SPACE DETECTED WITH AN ALGORITHM?

1. Choose the algorithm to evaluate;
2. Remove road noise due to excessive accumulation (depending on the filter);
3. Apply an algorithm to only keep the closest limits to the car on either side of the car;
4. Fill those limits to create the navigable space.

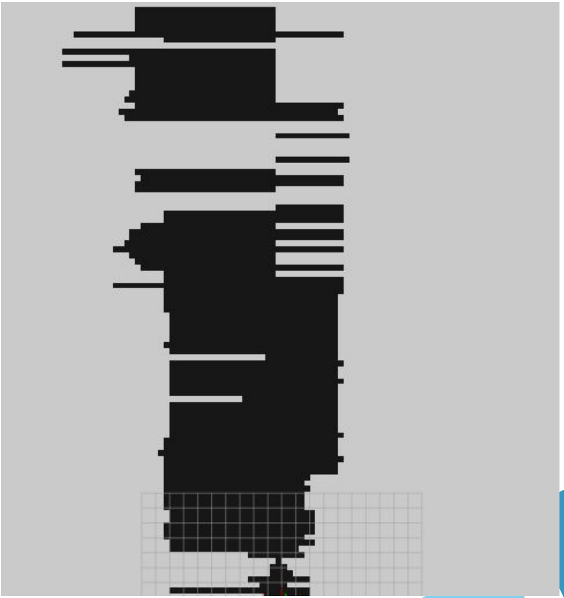
GROUND TRUTH VISUALIZATION



Ground truth lines



Ground truth of navigable space



Example of navigable space for Laplace operator

QUANTITATIVE EVALUATION

CONCEPT

CONCEPT

Binary evaluation based on positives and negatives.

DEFINITIONS

- **True Positive (TP)**: a cell that is **correctly** identified as being from the inside of the **navigable space**
- **False Positive (FP)**: a cell that is **falsely** identified as being from inside of the **navigable space**
- **True Negative (TN)**: a cell that is **correctly** identified as being **outside the road limits**
- **False Negative (FN)**: a cell that is **falsely** identified as being **outside the road limits**

QUANTITATIVE EVALUATION

INDICATORS

$$\text{Precision}/\text{PPV} = \frac{TP}{TP + FP}$$

$$NPV = \frac{TN}{TN + FN}$$

$$Specificity/TNR = \frac{TN}{TN + FP}$$

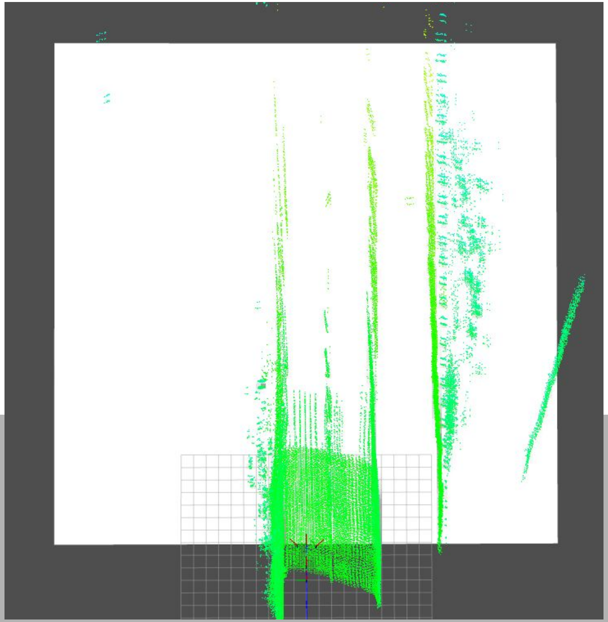
$$\boxed{\text{F-measure} = (1 + \beta^2)\frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}}$$

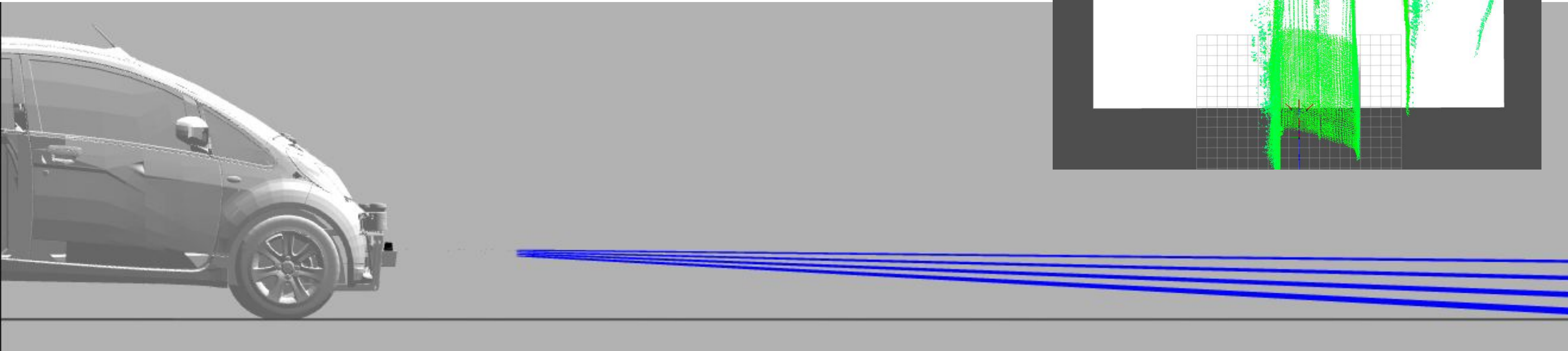
$$\text{Recall}/\text{Sensitivity}/\text{TPR} = \frac{TP}{TP + FN}$$

$$\boxed{\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}}$$

LIMITATIONS

1. There is an excessive accumulation of points in the first 10 m in front of the car due to the LIDAR inclination
2. The ground truth application is not prepared to contemplate curve situations

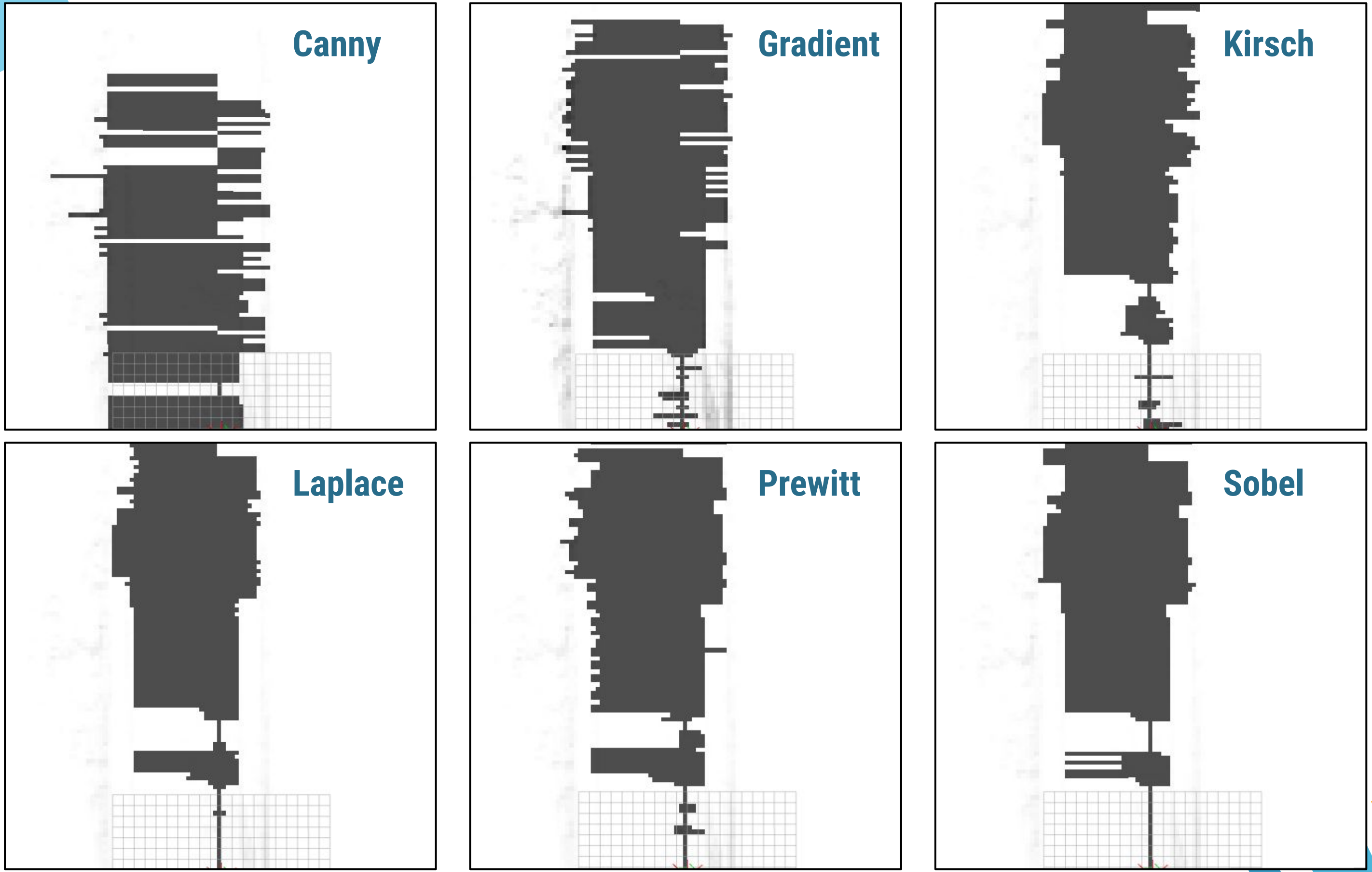




6.

TESTS AND RESULTS

QUALITATIVE EVALUATION



QUALITATIVE EVALUATION

CONCLUSIONS

- Although being the only one that results relatively well **close to the car**, **Canny** produces **poor results** when more distant from the car, with lots of **gaps** in the detection.
- All the edge detectors, apart from Canny produce **poor results** in the **10 m ahead of the car**.
- The **simple Gradient** filter is the one with **fewer gaps** in the detection and more clear road, apart from the initial meters.
- **Laplace and Prewitt** produce **similar results** with some **gaps** in the middle of the road.
- **Sobel and Kirsch** produce a **defined road** but further away from the car than the rest of the algorithms.

QUANTITATIVE EVALUATION

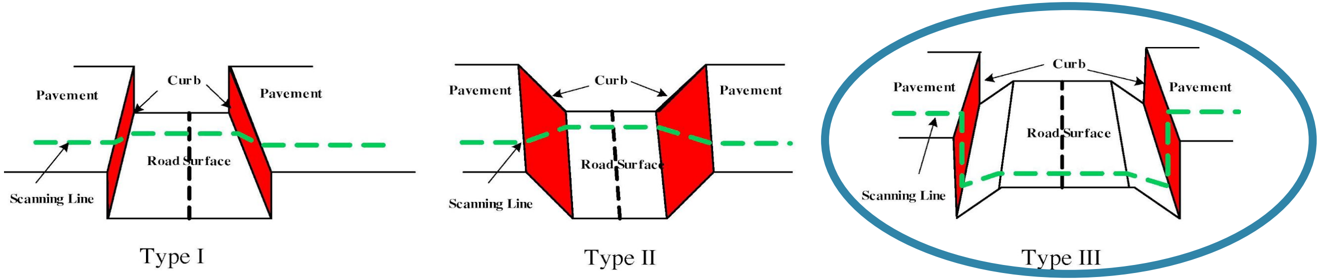
ALGORITHM'S PERFORMANCE

Filter	Precision	Specificity	NPV	Sensitivity	F-measure	Accuracy
Laplace	84.1	90.5	70.7	60.5	70.0	75.9
Gradient	83.8	86.6	84.5	83.3	83.0	84.6
Sobel	83.9	89.8	71.8	63.1	71.7	76.7
Prewitt	81.6	87.5	78.9	72.6	76.0	80.2
Kirsh	86.9	89.9	72.3	66.7	75.4	78.1
Canny	86.0	91.5	66.6	52.3	62.5	71.7

Table 2: Statistical indicators in each algorithm's performance with 0.4m/cell and no threshold applied (10 m to 30 m).

QUANTITATIVE EVALUATION

EFFECT OF TYPE III CURBS - NEGATIVE OBSTACLES

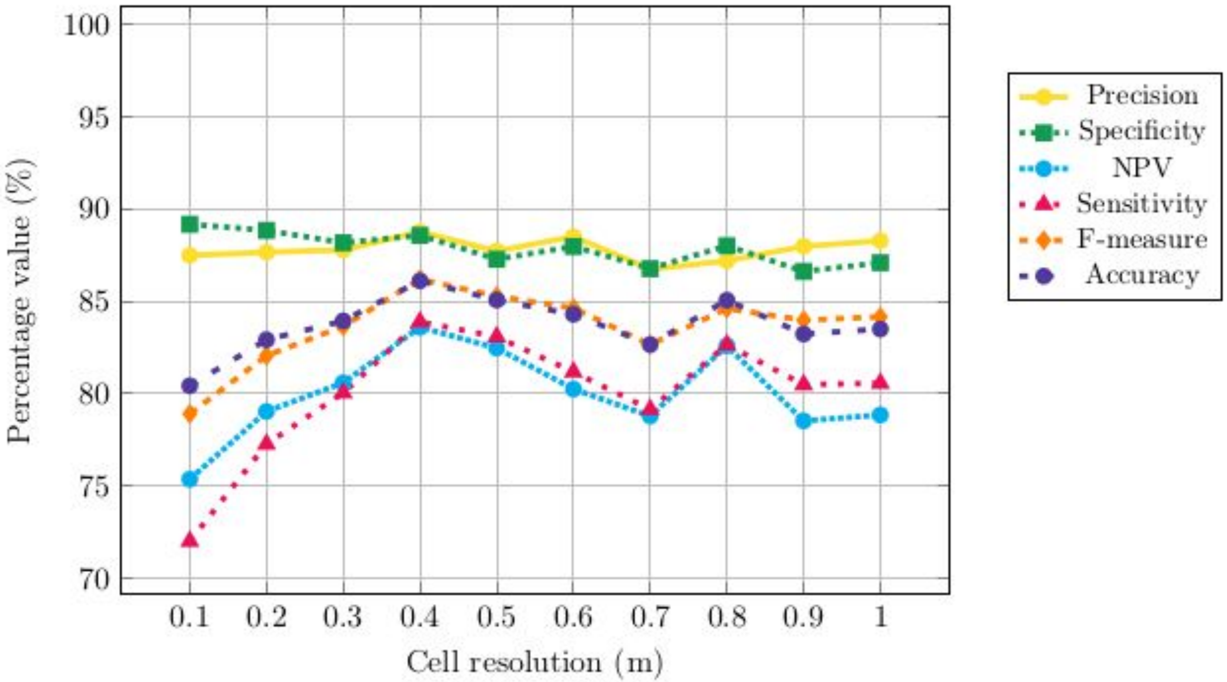


Filter	Precision	Specificity	NPV	Sensitivity	F-measure	Accuracy
Laplace	94.6	97.9	76.0	52.8	67.6	80.1
Gradient	85.0	91.8	83.6	72.0	77.9	84.0
Sobel	91.0	96.9	76.2	51.4	65.5	79.4
Prewitt	86.9	93.8	79.5	63.0	73.0	81.6
Kirsh	94.6	97.8	78.9	59.6	73.0	82.7
Canny	78.9	95.2	67.5	27.3	39.6	68.8

Table 3: Performance of algorithms in the presence of Type III curbs with 0.4m/cell and no threshold applied (10 m to 30 m).

QUANTITATIVE EVALUATION

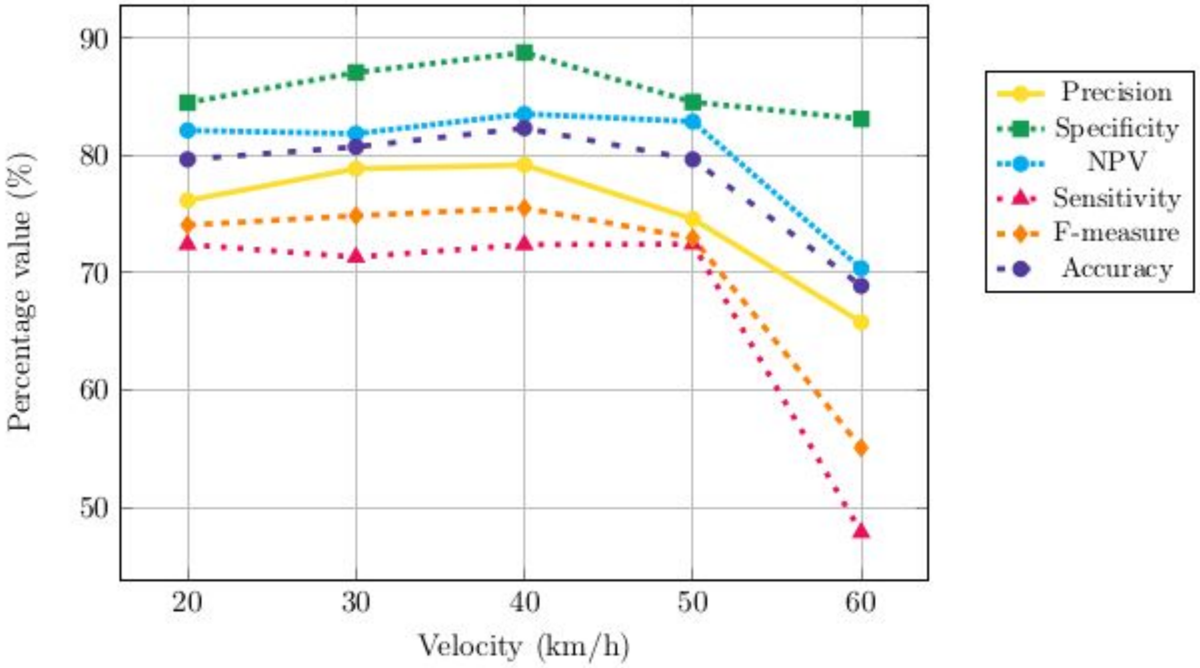
EFFECT OF OCCUPANCY GRID RESOLUTION



Graphic 1: Effect of occupancy grid resolution for the Simple Gradient algorithm with no threshold..

QUANTITATIVE EVALUATION

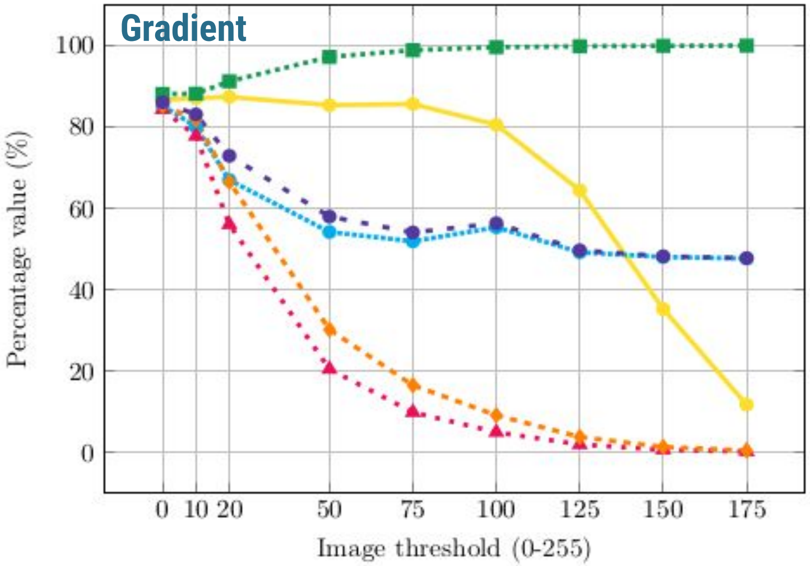
EFFECT OF CAR VELOCITY

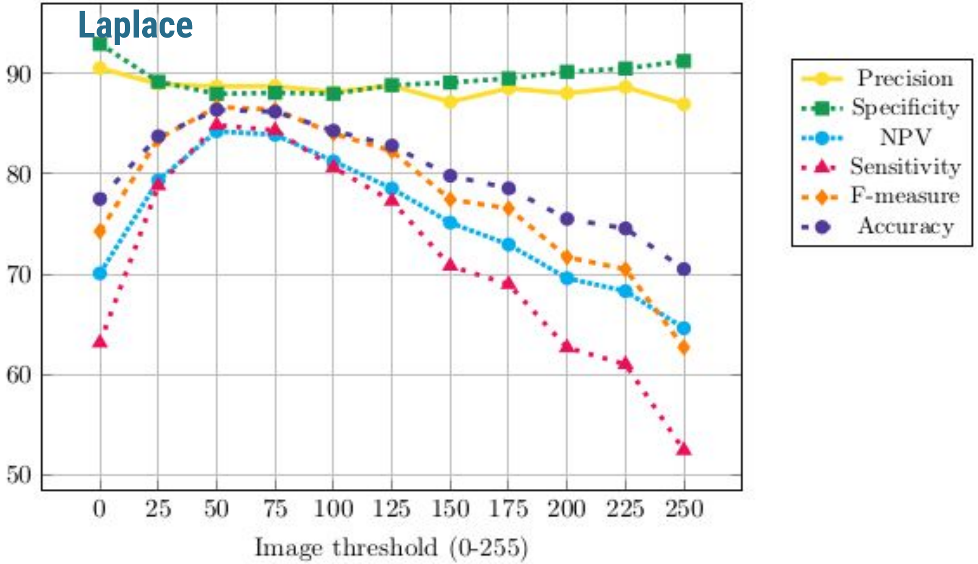


Graphic 2: Effect of car velocity for the Simple Gradient algorithm with no threshold and 0.4m/cell of resolution.

QUANTITATIVE EVALUATION

EFFECT OF GRADIENT THRESHOLD





Filter	Laplace	Gradient	Sobel	Prewitt	Kirsch	Canny
Threshold (0-255)	50	0	25	25	50	0

Table 4: List of optimized gradient threshld for each algorithm.

QUANTITATIVE EVALUATION

ALGORITHM'S PERFORMANCE WITH OPTIMIZE PARAMETERS

Filter	Precision	Specificity	NPV	Sensitivity	F-measure	Accuracy
Laplace	88.9	88.1	84.8	85.6	87.1	86.8
Gradient	89.4	89.0	83.4	83.9	86.5	86.3
Sobel	87.1	87.6	80.4	79.6	83.0	83.5
Prewitt	87.6	88.5	77.2	74.4	80.0	81.4
Kirsh	86.6	86.0	84.8	85.3	85.9	85.7
Canny	87.3	91.1	67.5	56.0	66.3	73.2

Table 5: Result of the performance of the algorithms with the improved parameters.

7.

CONCLUSION AND FUTURE WORK

CONCLUSIONS

PERFORMANCE

- The **Simple Gradient** produced the **best results** detecting the navigable space in all situations tested.
- The **Kirsch and Laplace** edge detectors also proved to produce **good detection results**.
- The algorithm **threshold** that optimizes detection is **different from filter to filter** due to the characteristics of the same and noise sensitivity.
- The algorithms have a **stable performance up to 50 km/h** and from that value the performance, although acceptable, begins to decrease.
- The cell resolution that optimizes the detection of the navigable space is **0.4 m/cell.**

CONCLUSIONS

CONTRIBUTIONS

- The use **gradient** as a tool to **detect hard limits** of the road in a **moving car;**
- Development of a method able to **detect all types of curbs**;
- Development of a tool to **evaluate algorithms performance**;
- Test and prove the **efficacy** of the method in **real time**;
- **Reduce computational effort** of point cloud accumulation.
- An **article** submitted in the *Fourth Iberian Robotics Conference* named "Detection of Road Limits using Gradients of the Accumulated Point Cloud Density".

FUTURE WORK

DEFINITION

» Combine the work developed in **lane detection** using **cameras** and create a **multi-sensorial** algorithm with the possibly to create an occupancy grid with **different levels of probability** according to the detected features;

» Fuse the results of **several edge detection algorithms** may also be interesting to obtain more complete and robust information;

» Find a solution to the behavior of the accumulated point cloud in **roundabouts**;

» Add **one or more LIDARs** to cover a bigger range of road and setting the sensors to asynchronous times for more reliability at higher velocities;

» Improve the quantitative evaluation program to contemplate **more situations**.

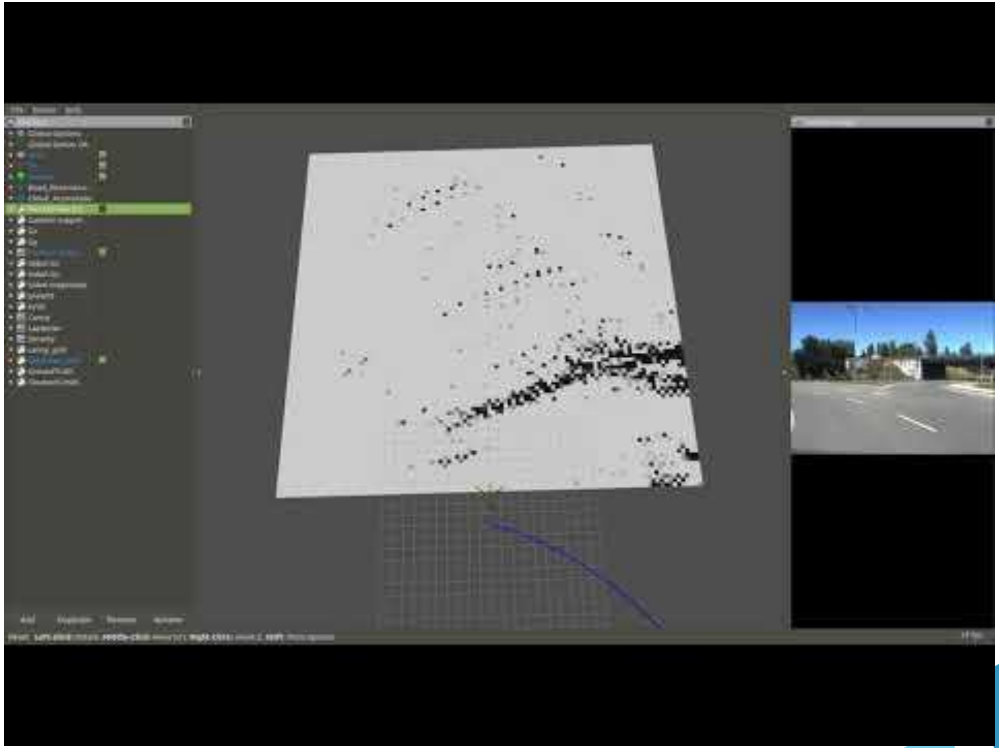
THANKS!

Any questions?

You can find me at

» danielarato@ua.pt

» https://www.linkedin.com/in/daniela-rato/



RELATED WORK

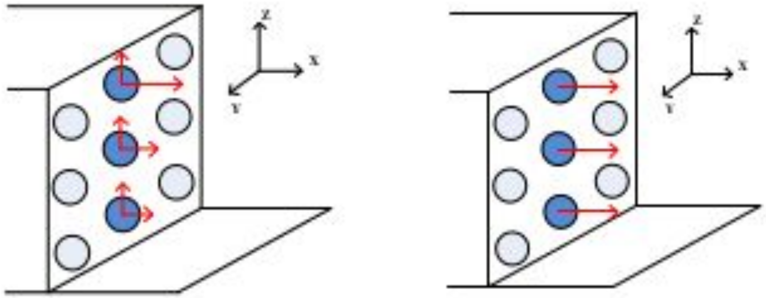
OTHER WORK - XU ET AL.[4]

APPROACH

Calculating the **difference of density** in adjacent voxels in 2D and then adding the 3rd **dimension** as the **difference of elevation between voxels**.

METHODOLOGY FOR ROAD CLASSIFICATION

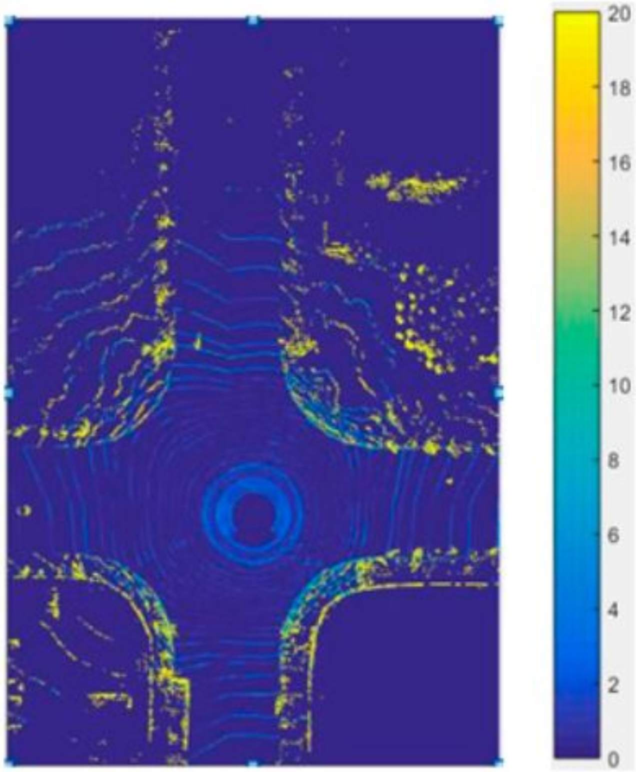
- One large gradient → voxel within one surface;
- Two large gradients → voxel in the intersection of two surfaces;
- Three large gradients → voxel in the intersection of three mutually non-parallel surface;



[4] Sheng Xu, Ruisheng Wang, and Han Zheng. Road curb extraction from mobile LiDAR point clouds. 55(2):996–1009, 2017.

RELATED WORK

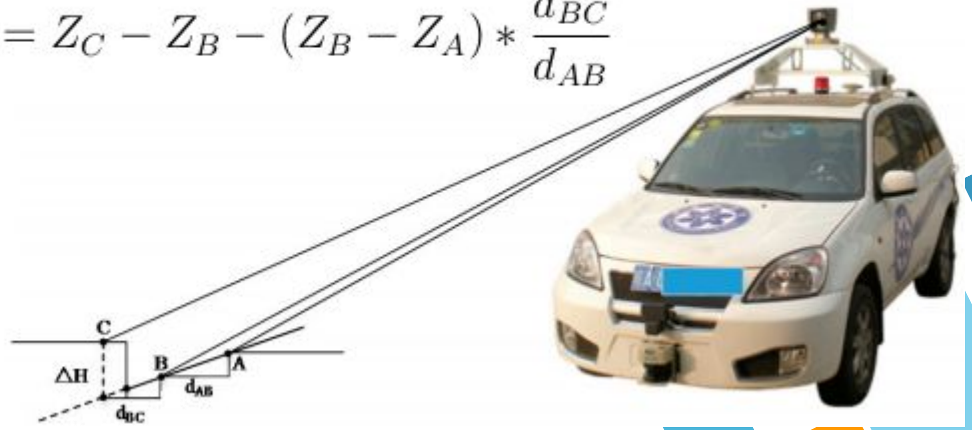
OTHER WORK - HUANG ET AL.[5]



APPROACH

A prediction method is used to find the height difference between two points and create an elevation map with the predicted measures.

$$\Delta H = Z_C - Z_B - (Z_B - Z_A) * \frac{d_{BC}}{d_{AB}}$$



[5] Rulin Huang et al. "A Practical Point Cloud Based Road Curb Detection Method for Autonomous Vehicle". In: Information 8.3 (July 30, 2017), p. 93. ISSN : 2078-2489. DOI : 10.3390/info8030093.

CONCLUSIONS

EVOLUTION

	CHALLENGE	SOLUTION
01	No identification of negative obstacle	Use of gradient as a tool to identify both positive and negative obstacles
02	Problems with variations in RPY values	Normalization of density values at every frame
03	Problems with variations of speed	Normalization of density values at every frame + new accumulator with low computational effort