Environmental modelling for autonomous public transport

Martin Dimitrievski, David Van Hamme, Dimitri Van Cauwelaert, Gianni Allebosch, Ivana Shopovska, Maarten Slembrouck, Peter Veelaert and Wilfried Philips

Ghent University – Dept. of Telecommunications and Information Processing, TELIN-IPI St.-Pietersnieuwstraat 41, B-9000 Gent, Belgium

mdimitri@telin.UGent.be

Introduction

Autonomous vehicles can save lives and make the world a happier place for everyone but…

... they will not simply evolve from the driver's assistance systems of today
… the algorithms can not be taught for every possible traffic scenario
… they need to think by generalizing like humans do, only smarter
… they rely on very accurate perception of the environment
… in bad weather, current sensors simply don't work well enough
… the legislation is still not well defined

However, there is hope! A lot of key automotive companies are investing in research. Flanders is interested in implementing driverless buses for the public transport sector. *

It's a hot topic!

"Chinese bus manufacturers Yutong unveils a self-driving bus prototype!"
"First autonomous Toyota to be available in 2020"
"First fully autonomous Tesla by 2018, approved by 2021"
"Uber fleet to be driverless by 2030"
"Ford CEO expects fully autonomous cars by 2020"
"Next generation Audi A8 capable of fully autonomous driving in 2017"
"Jaguar and Land-Rover to provide fully autonomous cars by 2024"
"Nissan to provide fully autonomous vehicles by 2020"
"Intel CTO predicts that autonomous car will arrive by 2022"
"Sergey Brin plans to have Google driverless car in the market by 2018"

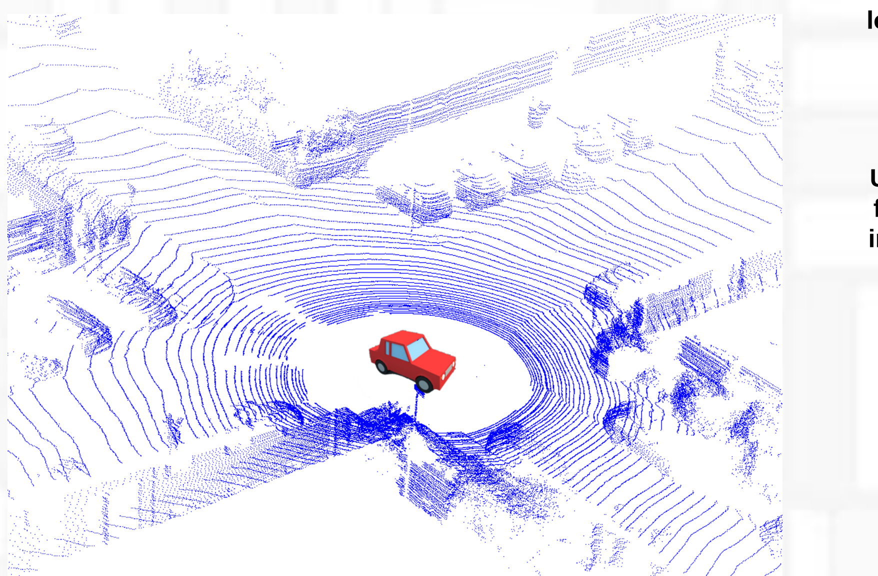
***source http://www.driverless-future.com/*

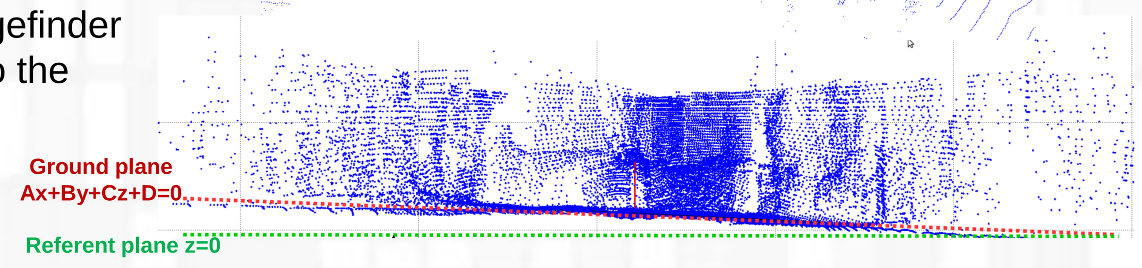
Tasks of an autonomous vehicle

Where am I now? EGO location

- Read the prior motion dynamics from the vehicle computer
- Find the ground plane in the current **laser scans** and project the visible objects on the ground plane
- Compute the **3DOF** pose change from the previous projection
- Find the exact position within the static map

Rotating at 10Hz
64 laser beams
~1.3 Mpoints per second

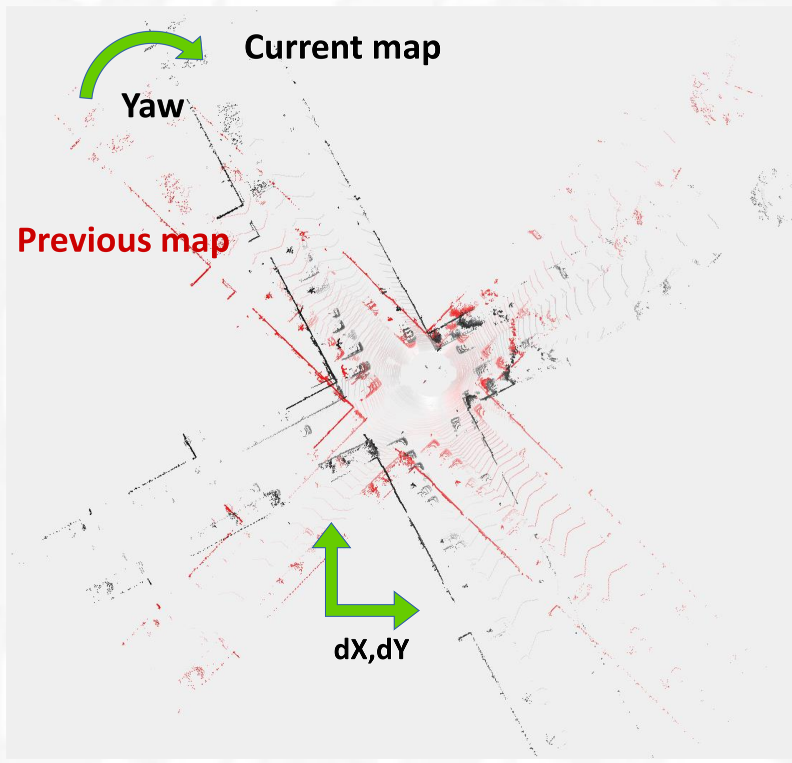




The main sensor is a rotating laser rangefinder (LIDAR) which measures distance to the surrounding environment

Project the points on local ground plane

Use the projection from the previous instant to estimate the pose change



We assume a locally flat world where the autonomous vehicle has 3 degrees of freedom [Yaw,dX,dY]

The yaw rate is equal to the rotation between the two projection images {**previous**, **current**}

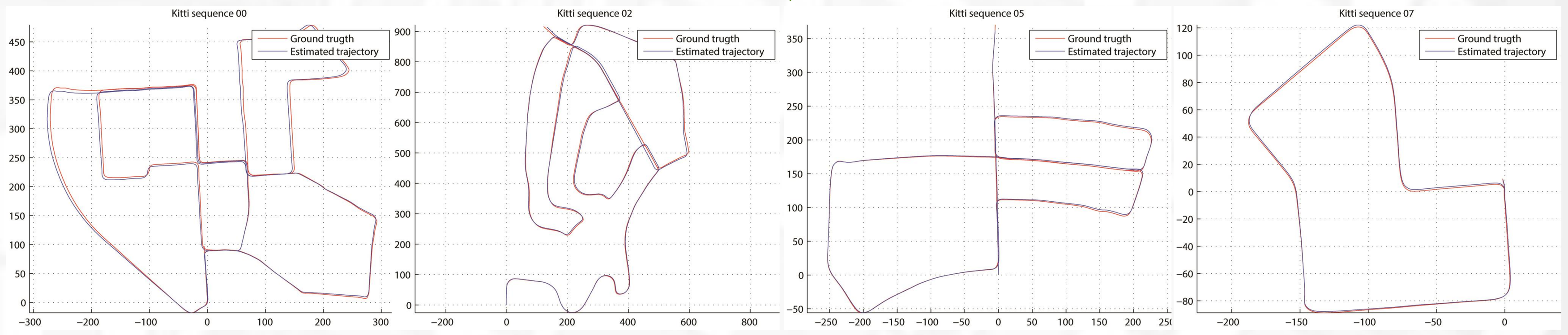
The translation is proportional to the number of row and column shifts needed to match the two projection images

These parameters can be robustly estimated in the Fourier domain using the peak in the 2D convolution of the amplitude spectra of the projection images

$$R(k) = \frac{M_T^{\rho,\theta}(k)\overline{M_{T-1}^{\rho,\theta}(k)}}{\left|M_T^{\rho,\theta}(k)\overline{M_{T-1}^{\rho,\theta}(k)}\right|},$$

The result is very accurate positioning relative to the beginning of the trajectory

2.94m position error and 3.80 degrees heading error per 1Km traveled
[tested on the KITTI odometry benchmark]



What is happening around me? Environmental perception

- Use the current laser scans to build a probabilistic **occupancy map**
- Classify which spaces are free of obstacles
- Don't ignore the occluded regions
- Reason about the **possible threat** at each location along the available area

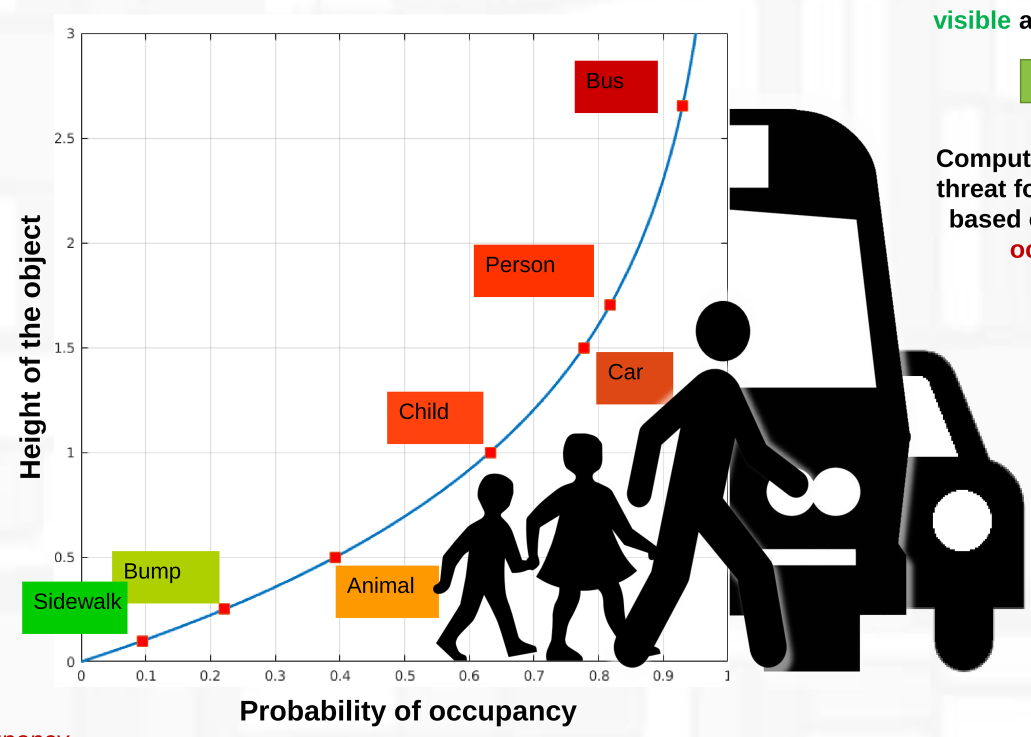
The occupancy map models the probability of a grid cell in space to be occupied based on the height of each measured object.

In order to update each map cell with the correct probability, the input LIDAR point cloud needs to be accurately registered to the global position.

For computational reasons we update the log odds $l_{i,j}$ for each grid cell i,j using the current measurement and the log odds from the past.

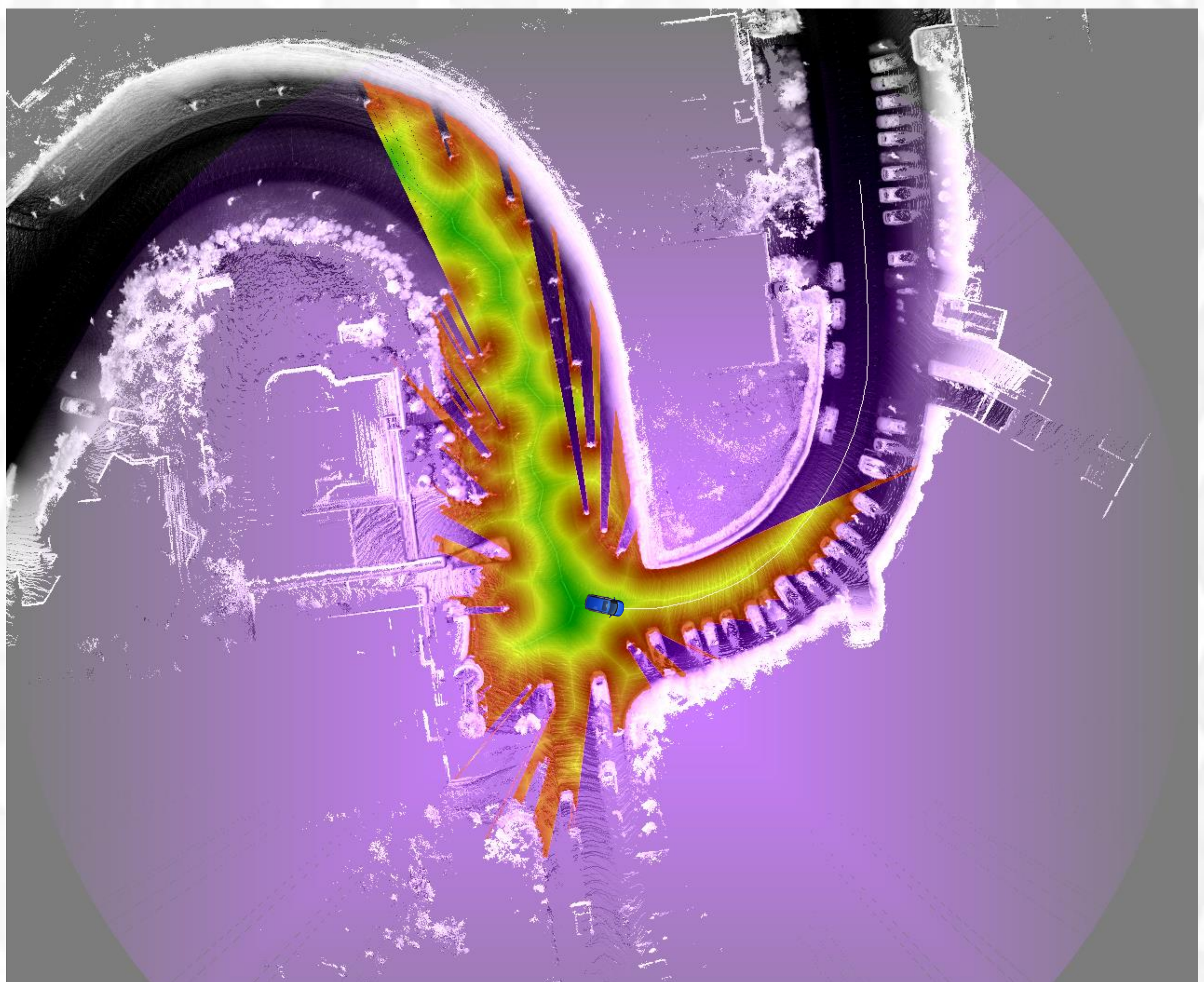
$$l_{i,j} = \log\frac{p(m_{i,j}|z_T)}{1-p(m_{i,j}|z_T)} + \log\frac{1-p(m_{i,j})}{p(m_{i,j})} + l_{i,j}^{past},$$

Measured probability of occupancy
Prior log odds for occupancy



Use the static map and the current position to find visible and occluded space

Compute the static level of threat for the visible space based on the distance to occupied cells



Desired trajectory
World map with static occupancy probabilities
Range of perception for sensors onboard
Level of threat in the currently visible space

What will happen next? Tracking, prediction

- Detect and classify the **moving objects** into (pedestrians, cyclists, vehicles…)
- Apply prior knowledge about the category to estimate the position and motion
- Track the object's behavior from the past samples
- Predict the threat for collision with the EGO vehicle

What I want to do? Path planning

- Consider the following:
  - Probability of occupancy from static map
  - Threat level for the free space ahead
  - Probability of collision with moving objects
  - Obey traffic regulations (lane, speed, traffic signs…)
  - Obey vehicle dynamics and reasonable accelerations
- Compute the optimal trajectory and speed profile

What will I do? Action

- Send steering, throttle and breaking information to the onboard computer
- Use the vehicle model as **feedback loop** to predict the next EGO position and give prior to step 1

Ongoing research

Real world tests at Zwijnaarde Technology Park with a prototype electric bus

Collaboration with project partners

* The work was financially supported by IWT through the Flanders Make ICON project 140647 "Environmental Modelling for automated Driving and Active Safety (EMDAS)"