

## Detection, Ranging and Classification in LiDAR Systems

Modern cars as well as emerging autonomously driving cars are equipped with a wealth of sensors, including cameras, ultrasonic park sensors, radar sensors and recently LiDAR (Light Detection And Ranging) sensors. The latter are essential for assisted and autonomous driving, since LiDAR works also in harsh environments, where e.g. vision-based systems may fail.



Figure 1: LiDAR systems for autonomous driving

The LiDAR system consists of one or multiple Lasers, one or more mirrors to change the angle of the laser, a receiver that converts the light into digital signals, and a processing unit (FPGA or  $\mu$ C). The output of the LiDAR system is a 3D-point cloud per second where the value of each point corresponds to the reflection of the object. The AI system of the car will use this point cloud to identify and classify objects and track them over time.

The experience of the final system depends on the quality of the 3D point cloud from the sensor. Target of this thesis is to investigate and refine algorithm and optimize the quality of the detection (= when to generate a point in the cloud), the ranging (= distance of the point to the source/car), and classification (= value of the point) considering imperfections (noise, dynamic range, bandwidth, etc.) of system.

This is a highly important topic for reliable future sensors; hence Infineon offers a paid master/diploma thesis focusing on:

- Literature survey on state-of-the art detection, ranging and classification algorithm
- Algorithm implementation and evaluation in MATLAB simulation framework (based on measurement)
- (optional) Port the algorithm into the existing LiDAR hardware system (FPGA or real time C++ on PC)

Your profile

- Motivated and interested in signal processing
- Eager to understand more about vehicular sensors and be part of the research

If you are interested in participating, please contact us. We are looking forward to hear from you!

stefan.mendel@infineon.com
www.infineon.com

