VROOM / auth

Virtual Route Optimization Mobility
Aristotle University of Thessaloniki

Bosch Future Mobility Challenge

ARISTOTLE UNIVERSITY OF THESSALONIKI

Technical Report #1

Authors

Theodoros Papafotiou [*Team Leader*]

Efthymia Amarantidou [*Team Member*]

George Koutroumpis [*Team Member*]

Nikolaos Tsachouridis [*Team Member*]

Responsible Mentor

Emmanouil Tsardoulias

Date

February 12, 2021

1 Introduction

The current report summarises the technical progress of the team V.RO.O.M. (Virtual Route Optimization Mobility) in the Bosch Future Mobility Challenge 2021 until December, 20th, 2020. Particularly, the activities planned during the time-period between the 23th of November and the 20th of December and their current status, as well as the upcoming activities are described shortly.

2 Planned activities

The planned activities during the aforementioned period, additionally to the team members in charge for each activity are shown below:

- Perception
  - **Lane detection**
    @George Koutroumpis & Nikolaos Tsachouridis
  - **Traffic Sign detection**
    @Theodoros Papafotiou & Efthymia Amarantidou
  - **Traffic Light detection**
    @Theodoros Papafotiou & Efthymia Amarantidou
  - **Intersection detection [Early implementation]**
    @George Koutroumpis
  - **Obstacle detection [Early implementation]**
    @Theodoros Papafotiou
- Control
  - **Lane Following [Early implementation]**
    @Nikolaos Tsachouridis
- Working tools
  - **Early track setup using the provided 3D printed parts of the original track and traffic signs**
    @Efthymia Amarantidou

3 Status of Planned Activities

3.1 Lane Detection [Done]

3.1.1 Goals

The first milestone we set for our algorithm was to detect straight lane lines. After achieving this, our next target was also to detect curved lines.

3.1.2 Implementation

For our Lane Detection algorithm we employed the Python API OpenCV. For test videos, since our stage is not set-up yet, we used videos from Bosch Future Mobility Challenge's website.

1. **Initial implementation**
   - The image is pre-processed by turning it to gray scale
   - The Canny Edge Detection is used
   - The processed image is fed into a Hough Line Transformation for line detection.
2. **Improvement**
   - A trapezoid mask is applied to the image, so our car can only detect lines in close proximity to it
   - The image is monochromatically transformed, using binary operators
   - The aforementioned process is followed with some fine tuning of the Hough Transform parameters

3.1.3 Issues

In the camera footage provided in the Documentation the lighting was very harsh resulting in resulting in glare (burned out) issues in several frames of the footage in some areas of footage. In these areas there where too many pixels with intensity values of 255. After adjusting brightness and through trial and error experimentation we could not recover information from those areas. The problem that this causes is that during Lane Detection, our algorithms recognize these areas as lines no matter what the threshold or thresholding method is. Since this problem is quite common in real life and we want to be ready for heterogeneous conditions, we are investigating further to resolve it.

3.2 Traffic Sign and Light Detection [Done]

3.2.1 Goals

Our goal was to detect and recognize both the traffic signs and the traffic lights, in order to be able later to incorporate them in the decision process. To achieve this we followed two different approaches, which will be analyzed in the next section.

3.2.2 Implementation

1. **Initial Implementation**
   - A dataset with pictures from different traffic signs and traffic lights was created.
   - The model is trained using the YOLO ("You Only Look Once") algorithm.
2. **Improvement**
   - For redundancy, hierachical recognition of the traffic signs was additionally implemented.
   - The shape and the color of the signs is initially detected and then the text is recognized - when needed.
   - The tesseract Python library is used for the text recognition.

3.2.3 Issues thus far

The main issue with our initial approach is that a lot of resources are needed, since a satisfying train of the model with confidence level of detection $> 90\%$ demands at least 3 full days in standard PC. In addition, the disadvantage of the second method, is that it is not as effective as necessary and there are still some issues with the accuracy of the text recognition. We concluded on using the first method, training the model in our university's server cluster while checking the effectiveness of our algorithm again.

3.3 Activities under [Development]

Activity	Goals	Implementation
Obstacle Detection	Recognizing of objects all over the track apart from traffic signs and traffic lights. Used for detecting the pedestrian, the bumpy part of the track, as well as the other RC cars on the track.	CSRT Tracker [slower, very accurate object tracking] MOSSE Tracker [very fast, not so accurate object tracking]
Intersection Detection	Recognizing intersections on the track.	Utilize the results of the Lane Detection algorithm, to detect the road intersection patterns.
Lane Following	Keep the vehicle within the bounds of the lanes.	Utilize the detected lanes, and by calculating relative distances from the car to the lanes, steer and accelerate the vehicle accordingly.
Track Settlement	Settlement of the original track where the challenges will take place	Traffic signs and traffic lights 3D printed Parts of the road printed on blackboard base

4 General status of the project

In general, a early approach of the car's perception was implemented and all scripts were tested in simulation situations. Due to the difficulty of initiating the Gazebo simulation on the team members' computers, the team used unit and functional tests for verifying the effectiveness of each implementation.

5 Upcoming activities

According to the suggested timeline of the team, during the period between the 20th of December 2020 and the 24th of January 2021, the upcoming activities are:

- Perception Integration
  - Lane detection [Simulation and Finalization]
  - Traffic Sign detection [Simulation and Finalization]
  - Traffic Light detection [Simulation and Finalization]
  - Intersection detection [Simulation and Finalization]
  - Obstacle detection [Simulation and Finalization]
- Control
  - Lane Following [Simulation and Finalization]
  - First implementation of Intersection Navigation
  - First implementation of Roundabout Navigation
- System Integration
  - Initial code integration from simulation to real-life car
- Working Tools
  - Obstacles 3D printing
  - Final track settlement in laboratory facilities