# Self Driving Car Simulation

Fetullah Atas

January 2019

## 1 Intro

This document explains Self Driving Car Simulation.

As a rule of thumb, in development process of Self Driving Car problem we need a simulator, to evaluate and validate our solution under different test cases. This idea in my mind, I decided to do a minimal Self Driving Car Simulation with a few perception tasks under Gazebo simulator. Pure contribution by me is the minimalist perception module. This module , detects Obstacles surrounding the car, creates a Local Costmap using LIDAR point clouds, Costmap can be used for some other tasks such as Path Planning, Obstacle avoidance etc. Last but not least, There exists fusion of point clouds and camera image, from this fusion we get RGB point cloud. So the Project Body is as following;

Self Driving Car
*simulator
*self driving car description
*utils
*perception

### 1.1 Simulator

Includes , necessery gazebo files such as simulated city, simulated models in city, Plugins for simulated Lidar and Car.

### 1.2 Self driving car description and Utils

Includes properties of simulated car, describes links joints, sensor links and their connections between each other. Utils includes message types being used.

### 1.3 Perception

Very staright forward, it creates a Costmap based on Lidar Point Clouds, and from this costmap we do a simple obstacle detection. Our Outcome from this, Local Costmap for obstacle avoidance, Objects detection, Camera image and LIDAR point cloud Fusion which gives RGB colored point cloud.

# 2 How to Run this Code?

This work bases on ROS kinetic and Gazebo-8, you will need a fairly modern laptop ROS and Gazebo to run the code. Here I provide bash commands to get there.

## 2.1 README.md

Copy and pasting bash commands from pdf file may lead some syntax errors, thus I created a Github Repo **ONLY** for setup instructions(README.md). Link to this file; README.md

Please carefully go through each command and make sure that all commands executed successfully, The instructions are for installaing ROS, Gazebo and Additional requirements. After you set up your envoirment In end of file there exist Executing Nodes section, Here a provide some screen shots of what you may see.

Running simulator, NOTE; make sure you source this file devel/setup.bash, in your workspace

do not copy commands from pdf file , refer to README.md

```
source devel/setup.bash
roslaunch simulator city.launch
```

After this you should be able to see figure.1

In a seperate terminal launch perception and RVIZ, again source devel/setup.bash do not copy commands from pdf file , refer to README.md

```
source devel/setup.bash
roslaunch perception perception.launch
```

Now You should see RVIZ open with Car Model, Detected Obstacles , LIDAR pointclouds ,Local Costmap and Simulated Camera. See , figure.2 , figure.3 , figure.4

**NOTE**; if you find steps , complicated, or long or for whatever reason you could not test it I also provide a nice descriptive video

## 2.2 Video

Link to Video is here.

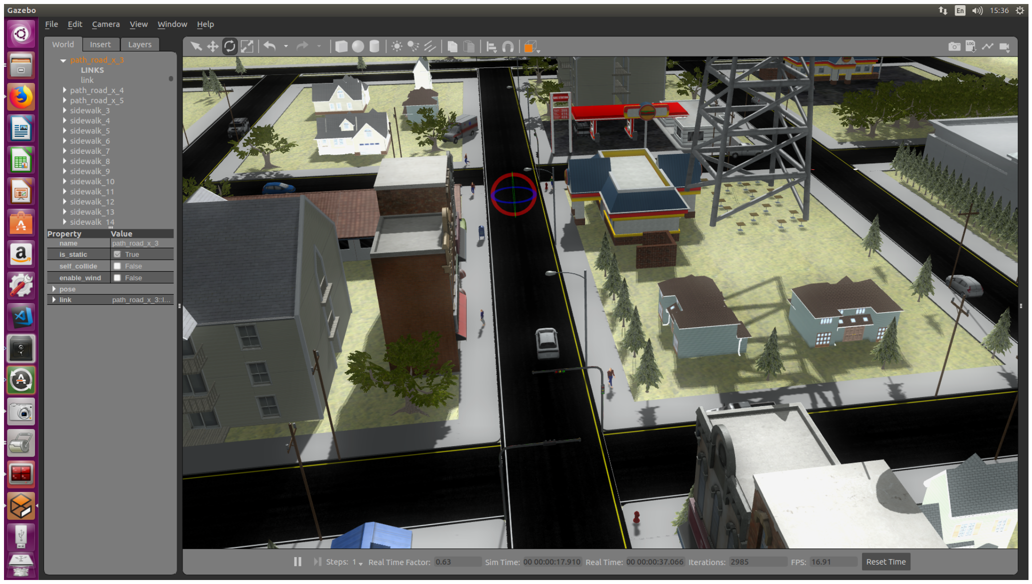


Figure 1: Simulator view

# 3 Conclusion

I appreciate you , for following this far, Thank you for your time, I would definetly like to hear your feedback, suggestions or criticism about my work. REACH ME OUT HERE; fetulahatas1@gmail.com

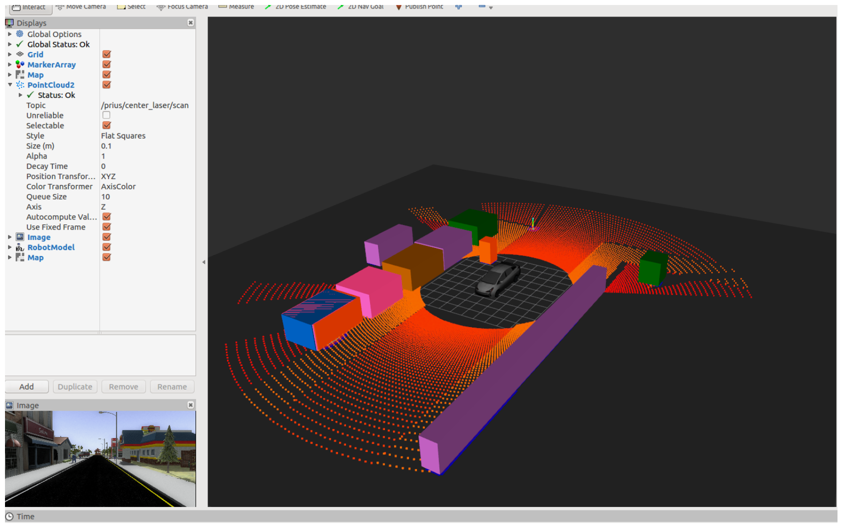


Figure 2: Obstacle Detection, Local Costmap , Simulated Lidar and Simulated camera view

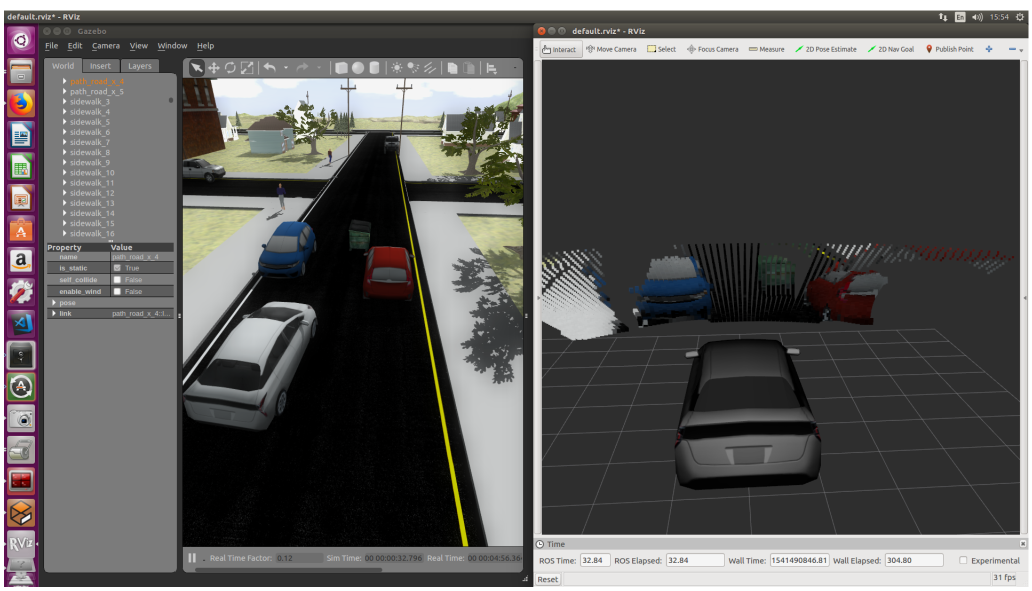


Figure 3: Camera Image and LIDAR Point Cloud Fusion

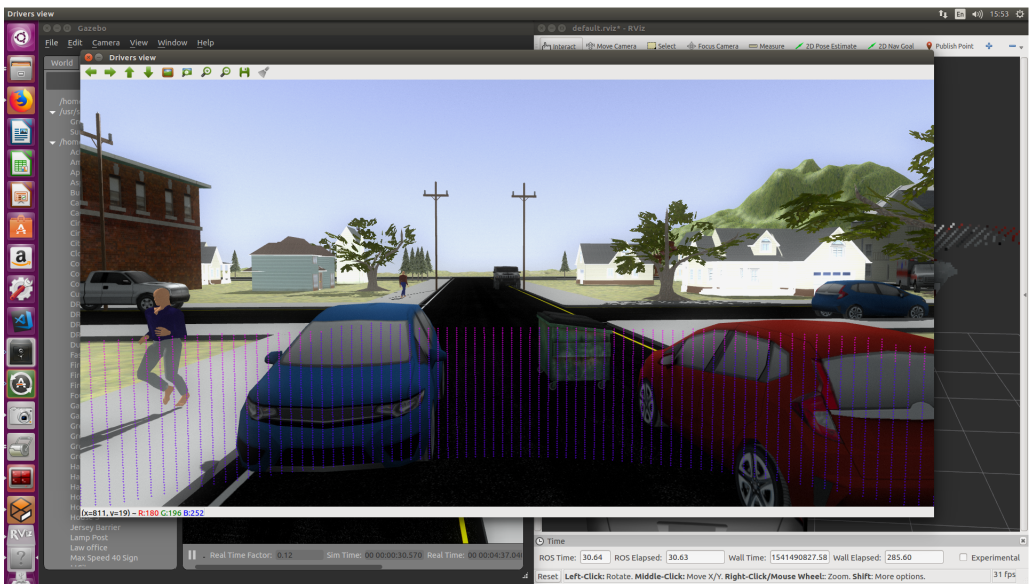


Figure 4: LIDAR Point Cloud Projection onto Image