Turtlebot 3 Autonomous Driving

Falezza Fabio VR409987

June 27, 2018

Verona University

Introduction

Autonomous Driving

"Autonomous driving" is a concept that group different tasks.
These could be:

- Traffic light detection
- Lane tracking
- Parking
- Level crossing
- Going through a tunnel

Autonomous Driving II

This presentation will be focused on *Lane tracking* task.
It will be showed how to setup a simulation environment in gazebo with different tracks to allow the turtlebot to perform the autonomous driving.

Installation

Prerequisites

In order to perform autonomous driving, it will be used the `turtlebot3_autorace_simulation` ROS package. It is supposed you have Ubuntu 16.04 with ROS Kinetic installed on your machine.

Install Dependency Packages

Proceed to install the Turtlebot dependency with the following command:

```
$ sudo apt-get install ros-kinetic-joy ros-kinetic-
    teleop-twist-joy ros-kinetic-teleop-twist-keyboard
    ros-kinetic-laser-proc ros-kinetic-rgbd-launch ros-
    kinetic-depthimage-to-laserscan ros-kinetic-rosserial
    -arduino ros-kinetic-rosserial-python ros-kinetic-
    rosserial-server ros-kinetic-rosserial-client ros-
    kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-
    map-server ros-kinetic-move-base ros-kinetic-urdf ros
    -kinetic-xacro ros-kinetic-compressed-image-transport
     ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-
    kinetic-navigation ros-kinetic-interactive-markers
```

Create Your Turtlebot ROS Workspace

Now we will create a ROS workspace to work in it. If you have already a ROS workspace you can use it and skip the commands below.

```
$ mkdir -p $HOME/turtlebot_ws/src
$ cd $HOME/turtlebot_ws
$ catkin_make
$ source ./devel/setup.bash
```

It is suggested to add the source to the file `setup.bash` to your `.bashrc` file.

Clone Turtlebot Packages

Now we will clone the Turtlebot packages:

```
$ git clone https://github.com/ROBOTIS-GIT/
    turtlebot3_msgs.git
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3.
    git
$ git clone https://github.com/ROBOTIS-GIT/
    turtlebot3_simulations.git
$ git clone https://github.com/ROBOTIS-GIT/
    turtlebot3_autorace.git
```

Clone Turtlebot Packages II

If you do not cloned the `turtlebot3_autorace_simulation` repository yet, do it with this command:

```
$ git clone https://github.com/falfab/
    turtlebot3_autorace_simulation.git
```

Then build all:

```
$ cd $HOME/turtlebot_ws && catkin_make
```

Folders Explanation

Move into the `turtlebot3_autorace_simulation` repository. You will notice different subfolders:

- `world/` contains the gazebo tracks models which will be used in the autonomous driving simulation
- `urdf/` contains the *Turtlebot3 Burger Pi* robot description
- `launch/` contains the ROS launchfile used to launch simulation environment and autonomous driving
- `config/` contains the `yaml` configuration file used to calibrate the camera compensation, the camera projection and lane detection

Install Environment Models

In order to run gazebo with the simulation environment, you need to put the *world description* inside the gazebo `model` folder. Execute the following steps to do it:

```
$ roscd turtlebot3_autorace_simulation
$ cp -r ./world/turtlebot3_autorace_track* $HOME/.gazebo
    /models
```

If you do not have the `.gazebo` folder is due you never opened gazebo. Open a terminal and run this command:

```
$ gazebo
```

It will setup his environment and create the folders you need.

Install Robot Model

In order to use the *Turtlebot3 Burger Pi*, you need to move the robot description inside the available robot models. Do the following steps:

```
$ cp ./urdf/turtlebot3_burger_pi* $HOME/turtlebot_ws/src
    /turtlebot3/turtlebot3_description/urdf
```

Remember to set it as the robot description you want to use:

```
$ export TURTLEBOT3_MODEL=burger_pi
```

It is suggested to add the last command to your .bashrc file.

How It Works

How It Detect Lane

The autonomous driving uses three ROS nodes to perform the task.

- `image_projection`: it project the camera input in order to remove perspective.
- `image_compensation`: it compensate the projected image. It is used but its effect is null because is not needed compensation in simulation environment.
- `detect_lane`: it detect the lane. It detect a yellow and a white lane in the projected and compensated image.
- `control_lane`: it receive an input from `detect_lane` and elaborate a `cmd_vel` to move the robot in the right direction.

Camera Input

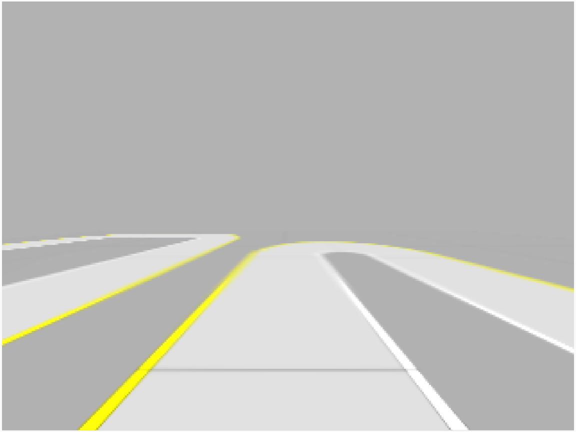


Figure 1: Image raw input

Camera Projection

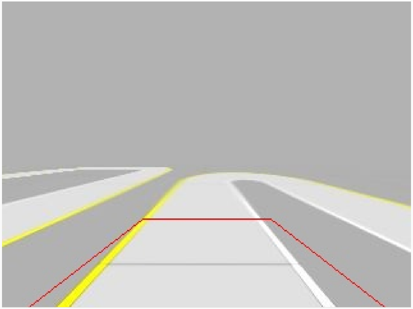


Figure 2: The red box show the area will be projected

Figure 3: Projected image

Lane Detection

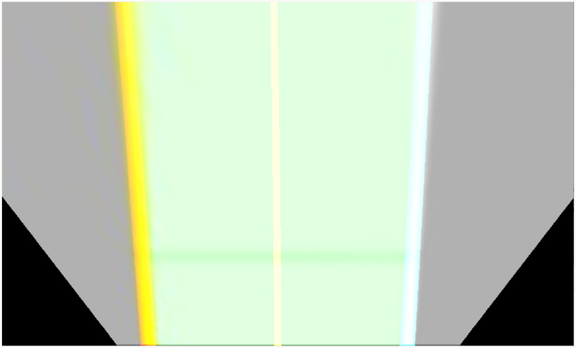
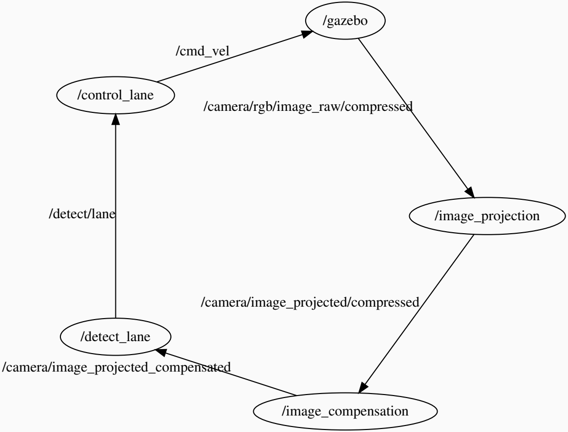


Figure 4: Lane Detection

Graph Representation

Below is shown how the nodes communicates:



Launch The Simulation

Launch Simulation Environment

In order to launch the simulation environment a ROS launch file is provided. Do the following command:

```
$ roslaunch turtlebot3_autorace_simulation gazebo.launch
```

This launch file could take five input arguments:

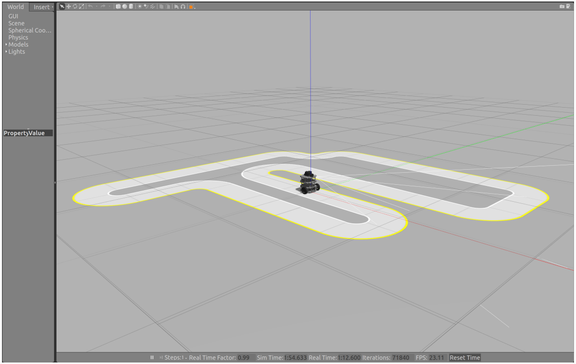
- x_pos: set the x start coordinate of the robot
- y_pos: set the y start coordinate of the robot
- z_pos: set the z start coordinate of the robot
- `track`: set the track to import in the environment
- `use_gui`: set if it has to load the user interface

By default it set $x = 0, y = 0, z = 0$, it set the track as `track1` and `use_gui`=True

Launch Simulation Environment II

Here is an example on how to launch the simulation with a different track:

```
$ roslaunch turtlebot3_autorace_simulation gazebo.launch
    track:=track2
```



Launch Autodrive Nodes

In order to launch the Autodrive nodes a ROS launch file is provided. Do the following command:

```
$ roslaunch turtlebot3_autorace_simulation autorace.
    launch
```

The calibration mode could be enabled by running the following command:

```
$ roslaunch turtlebot3_autorace_simulation autorace.
    launch calibration_mode:=calibration
```

Robot Visualization

In order to watch the image published on the ROS topics, a ROS launch file is provided. This launch file open *RViz* and show the robot odometry and the images published on `/camera/rgb/image_raw` and `/detect/lane`.

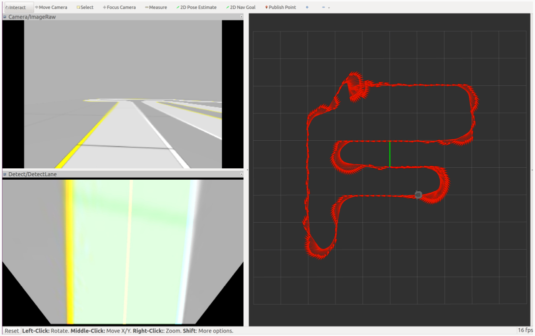
To run *RViz* do the following command:

```
$ roslaunch turtlebot3_autorace_simulation rviz.launch
```

Robot Visualization II

If your PC cannot execute Gazebo and *RViz* simultaneously you should try to launch Gazebo with the use_gui parameter setted to false:

```
$ roslaunch turtlebot3_autorace_simulation gazebo.launch
    use_gui:=false
```



Robot Circuit With Lap Counter And Timer

In order to makes the robot run in the circuit with spot logo recognition with Lap Counter and Timer do the following commands:

```
roslaunch turtlebot3_autorace_simulation circuit.launch
roslaunch turtlebot3_autorace_simulation autorace.launch
```

In order to visualize the Robot Vision you can use RViz also this time. Remember to use the following command which pass a different configuration file in order to view a most significant topic.

```
roslaunch turtlebot3_autorace_simulation config_file:=
    circuit.rviz
```

Robot Circuit With Lap Counter And Timer II

Here is a short video of the Robot visualization during the race:

- https://www.youtube.com/watch?v=FR27W19rrYA