LineDetection任务

github地址

背景：ubuntu18, melodic

LineDetection Simulation

项目基于github地址实现。

实验背景：Ubuntu18.04, Melodic

Dependency

turtlebot3相关配置安装与编译

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
git clone https://github.com/ROBOTIS-GIT/turtlebot3.git
git clone https://github.com/ROBOTIS-GIT/turtlebot3_simulations
git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs
cd ~/catkin_ws
catkin_make
```

OpenCV安装

下载opencv3.4.13，解压

```
cd opencv-3.4.13
mkdir build
cd build
# configure, generate即可
cmake-gui ..
make
sudo make install
```

Build

编译LineDetection

```
cd ~/catkin_ws/src
git clone https://github.com/sudrag/line_follower_turtlebot
cd ~/catkin_ws
catkin_make
```

optionally, 若想指定Opencv版本，可在`line_follower_turtlebot`中修改`CMakeLists.txt`

```
set(OpenCV_DIR ${your_opencv_dir}/share/OpenCV)
```

Modify

1. **修改gazebo_model的调用**

```
cd ~/catkin_ws/src/turtlebot3_simulations/turtlebot3_gazebo/launch
gedit turtlebot3_empty_world.launch
```

改为如下代码：

```
<launch>
  <arg name="model" default="$(env TURTLEBOT3_MODEL)" doc="model type [burger,
waffle, waffle_pi]"/>
  <arg name="x_pos" default="0.0"/>
  <arg name="y_pos" default="0.0"/>
  <arg name="z_pos" default="0.0"/>
  <arg name="world_file" default="$(find turtlebot3_gazebo)/worlds/empty.world"/>

  <include file="$(find gazebo_ros)/launch/empty_world.launch">
    <arg name="world_name" value="$(arg world_file)"/>
    <arg name="paused" value="false"/>
    <arg name="use_sim_time" value="true"/>
    <arg name="gui" value="true"/>
    <arg name="headless" value="false"/>
    <arg name="debug" value="false"/>
  </include>

  <param name="robot_description" command="$(find xacro)/xacro --inorder $(find
turtlebot3_description)/urdf/turtlebot3_$(arg model)_for_autorace.urdf.xacro" />

  <node pkg="gazebo_ros" type="spawn_model" name="spawn_urdf" args="-urdf -model
turtlebot3_$(arg model) -x $(arg x_pos) -y $(arg y_pos) -z $(arg z_pos) -param
robot_description" />
</launch>
```

2. **修改topic指向**

```
cd ~/catkin_ws/src/line_follower_turtlebot/launch
gedit lf.launch
```

改为如下代码：

```
 <launch>

<arg name = "StartRec" default = "false" />

<include
file = "$(find turtlebot3_gazebo)/launch/turtlebot3_empty_world.launch">
```

```
<arg name = "world_file" value =  "$(find
line_follower_turtlebot)/Maps/lfm1.world"/>
</include>

<!-- Detection node -->
<node pkg="line_follower_turtlebot" name="detection" type="detect" cwd="node"
output="screen">
<remap from="/camera/rgb/image_raw" to="/camera/image" />
</node>

<!-- Robot commands node -->
<node pkg="line_follower_turtlebot" name="Velocity" type="navig" cwd="node"
output="screen">
<remap from="/cmd_vel_mux/input/teleop" to="/cmd_vel" />
</node>

<node
pkg = "rosbag"
type = "record"
name = "rosbag_record_all"
respawn = "true"
output = "screen"
args = "-a -O BagFile -x /camera/(.*) "
if = "$(arg StartRec)"
/>

</launch>
```

Run

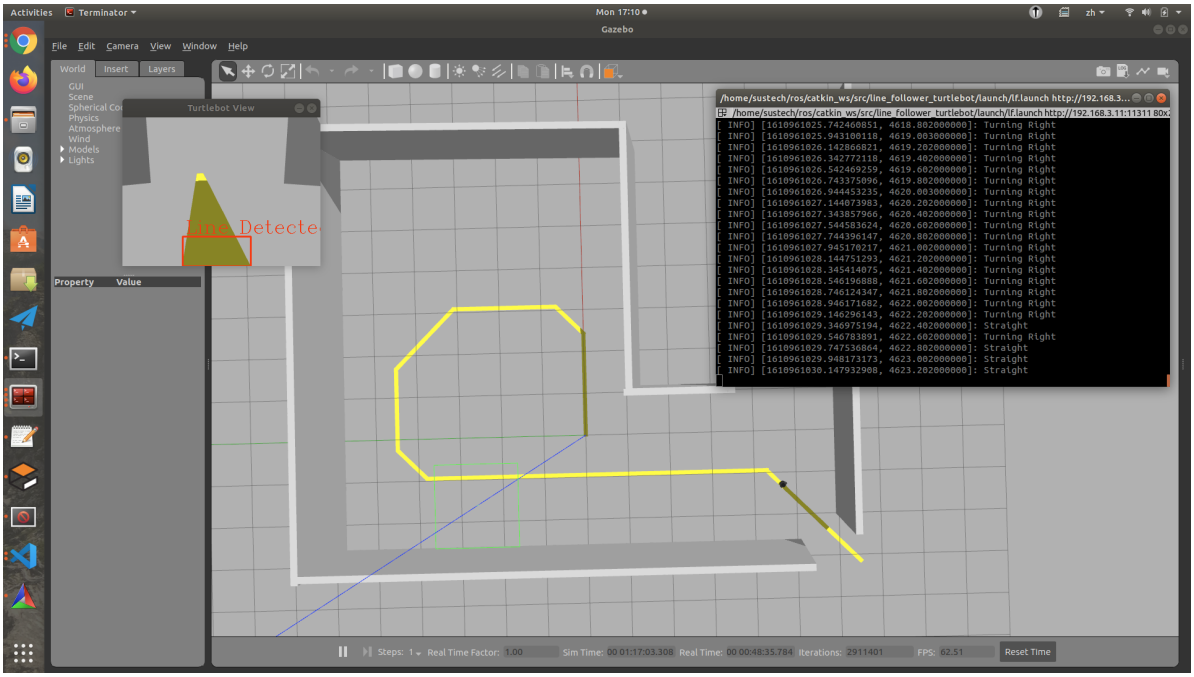
在终端运行如下代码，或写入`~/.bashrc`永久有效

```
export TURTLEBOT3_MODEL=burger
source ~/catkin_ws/devel/setup.bash
```

运行：

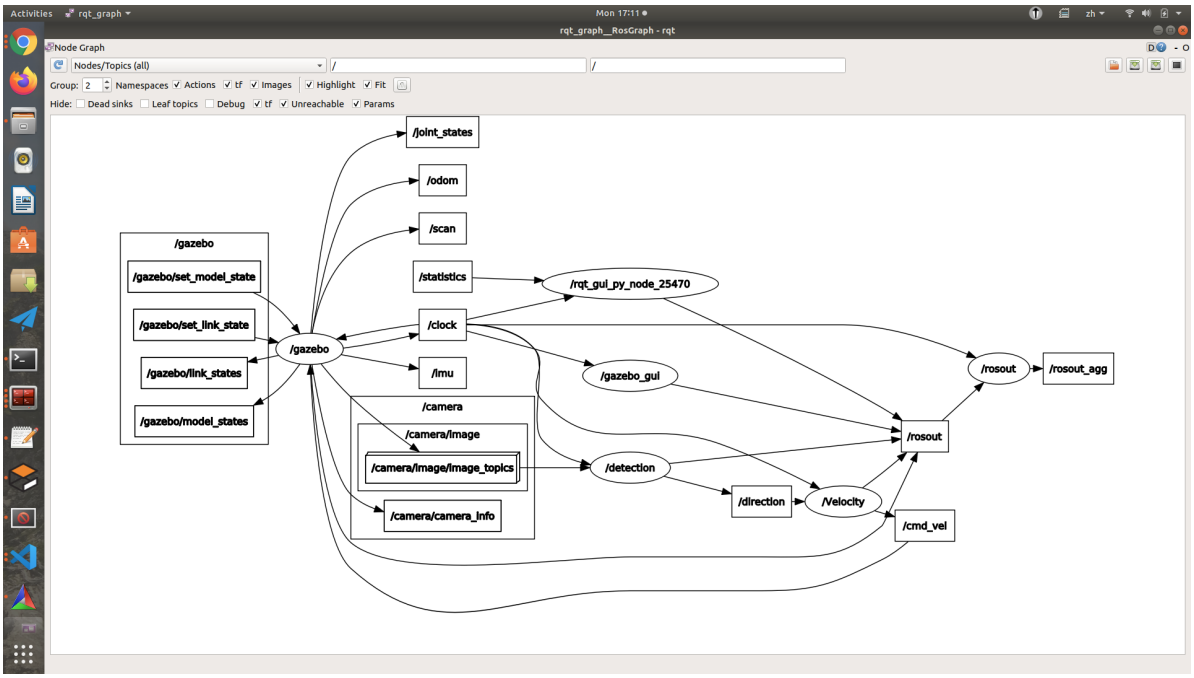
```
roslaunch line_follower_turtlebot lf.launch
```

效果图：



话题关系图：

```
rqt_graph
```



With Python Script

1. Add the python file and endow it.

```
cd ~/catkin_ws/src/line_follower_turtlebot/
mkdir scripts  # and add your lf.py to here
sudo chmod +x lf.py
```

2. create a launch.

```
cd ~/catkin_ws/src/line_follower_turtlebot/launch
gedit lf_py.launch
```

and add this to the launch file:

```
<?xml version="1.0"?>
<launch>
<include
file = "$(find turtlebot3_gazebo)/launch/turtlebot3_empty_world.launch">
<arg name = "world_file" value =  "$(find
line_follower_turtlebot)/Maps/lfm1.world"/>
</include>
<node pkg="line_follower_turtlebot" type="lf.py" name="line_follower"
output="screen" cwd='node'>
<remap from="/camera/rgb/image_raw" to="/camera/image" />
<remap from="/cmd_vel_mux/input/teleop" to="/cmd_vel" />
</node>

</launch>
```

3. source your `setup.bash`, and u can run it.

```
source ~/catkin_ws/devel/setup.bash
roslaunch line_follower_turtlebot lf_py.launch
```