Robotics competition report

20165144 이종원,  20165192 허서연, 20175052 김호영

1.     Overview

1)     Platform

Main platform : Turtle Bot3 and Open Manipulator

 Turtlebot3 is used for a base of robot and enable robot to move using its two motors. Open manipulator is used for a gripper. In turtlebot3, Raspberry Pi is mounted so that control the OpenCR board for maneuvering turtlebot3 and open manipulator. In detail, Raspberry Pi B+ is used for mainboard to order the command to OpenCR board. And Open CR is used to give all sensor’s information to Raspberry Pi B+.

2)     Sensor

We use overhead camera. It can detect the location of robot, trash, and trash bin by using Open CV.

2.     Methodology/Strategy

1. Because the initial location and orientation of robot is unknown, the robot’s own coordinate is unknown. Therefore, we find the robot’s own coordinate by generating transformation matrix between the absolute coordinate(overhead camera) and robot’s coordinate. In this case, we convert pixel value to robot’s coordination.
2. Using OpenCV, find the trash using the detection of color and shape.
3. Using filter, remove the noise and transform the pixel value to robot’s coordinate so that to tell the robot where to move.
4. Move the robot to corresponding location(the location of trash) be ready to grab it using Open Manipulator.
5. For grabbing precisely, rotate the robot for locating the trash in front of the open manipulator. For that, we plan to use YOLO for detecting the trash.
6. After grabbing it, move the robot in front of the bin and throw it away.
7. The location of the trashes might be changed because of the collision with robot and trash. Therefore, detecting the trash again and follow the step between 2~7 again.

3.     Results

Task 1) As described below at Task2, we detect the robot using the color value of robot. Also by receiving the odometry information generated by “/om\_with\_tb3/odom” node, we can easily know where the robot headed. By moving robot 1m for each x and y direction, we can know the basis of robot’s coordinate and convert it to global coordinate which means we can know where the robot is and headed using the global coordinate.

Task 2)  We first detect the positions of 6 trashes(3 color blocks and 3 cups) giving a highly accurate position compare to the water bottle because water bottle had been shown small in the view of overhead and the color was not noticable. Also, we increased the resolution of camera to improve accuracy from 640x480 to 1280x720, and also cut the frame to reduce the environmental impact. Then, values were filtered using the area of the detected pixel. For example, we didn’t detect the object whose detected area is less than 50 and bigger than 150. Through several trials, we found the appropriate values of area, color range that reduces the error for each object. After on filter using the area, we used the mode(the most common value) of positions of trashes captured by many frames(about 30 frames) to reduce error. As a result, six trashes(3 color blocks and 3 cups) are successfully detected with very high accuracy and returned to the list. However, despite efforts to change the resolution and using the shape of object, the position of the water bottle could not be accurately detected. (Because it is very similar to the surrounding color and it’s size is very small in the view of overhead.

Task 3) We know the location of robot and trash. So we can change position of turtle bot3 by using cmd\_vel node provided by ROS. First we make the robot head the trash using arctan function to give the robot radian value to turn. Radian is calculated based on the two positions of robot and trash. After it, we move robot using Publish(cmd\_vel) node. Robot moves until the distance is smaller than 25cm. We maintain this distance for grabbing. We neglect the avoid-obstacle planning because it’s very hard and the change of location of trash could be detected by overhead camera.

Task 4) First we plan to use YOLO which is based on deep learning, but YOLO didn’t perform well. We used pre-trained model and it detected well in case of bottle and green block, but cannot detect well in case of yellow, blue, red block and white water cup. So we didn’t use YOLO. Instead, we just trying to head the trash by using radian value which was calculated initially to move the robot.

For picking up and grabbing, we used robot’s geometry message(geometry\_msgs). This message make us possible to set the pose of open manipulator. Using this method, first, we set the pose of pick the trash, second, we set lift it up pose. When we throw away the trash, same as pick up the trash, we set the pose to reach into the trash bin and open the gripper. The code we referenced is commented at reference.

4.     Appendix - screenshot 필요

1)     How to start & run our code

1.     Robot bring up

First, launch roscore. And access robot by using ssh. Second, setting robot can be able to run our code.

$ export TURTLEBOT3\_MODEL=${TB3\_MODEL}

$ ROS\_NAMESPACE=om\_with\_tb3 roslaunch turtlebot3\_bringup turtlebot3\_robot.launch multi\_robot\_name:=om\_with\_tb3 set\_lidar\_frame\_id:=om\_with\_tb3/base\_scan

Launch rosserial and lidar node

$ ROS\_NAMESPACE=om\_with\_tb3 roslaunch open\_manipulator\_with\_tb3\_tools om\_with\_tb3\_robot.launch

Launch robot state publisher node

2.     Run Moveit! To control open manipulator

$ export TURTLEBOT3\_MODEL=${TB3\_MODEL}

$ roslaunch open\_manipulator\_with\_tb3\_tools manipulation.launch use\_platform:=true

3.     Run our code

2)     Software

Ubuntu, ros kinetics, python

Reference

“Move Group Python Interface”, June 20.2019, <https://ros-planning.github.io/moveit_tutorials/doc/move_group_python_interface/move_group_python_interface_tutorial.html>

<https://github.com/webnautes/nudapeu/blob/master/opencv-python-006.py>