Task 1.2 - Pollinator Bee

Problem Statement

Waypoint Navigation of the drone and image processing within V-REP

Problem Statement

- The drone should visit the given waypoint coordinates in the Scene.ttt simulation using the PID control algorithm. Waypoints are in the form (x, y, z) :
  [(-5.63, -5.63, 30), (5.57, -5.63, 30), (5.55, 5.54, 30), (-5.6, 5.54, 30), (0.0, 0.0, 30)]
- Within the same python node you must write code that would detect the colors placed in the simulation scene and publish them on their respective topics.

Procedure

1. Load ***Scene.ttt*** in V-REP simulator after launching *roscore* in a terminal. It is the same scene as given in the Task 1.1 folder.
2. Open task_1.2.py within the scripts folder of the package you cloned in Task 1.1.
3. Write a PID script that would help the drone is visiting the waypoints given in this file.
   You must place the given waypoint list in your Python script and move the drone to these waypoints one after another. **The drone needs to only visit the waypoints and not hover at them**. Each element of the list is a coordinate in 3D space represented as (x, y, z). A maximum error of **0.2** from the desired waypoint is acceptable in both the x and y coordinates. Maintain z coordinate of the drone between 28.5 and 31.5 during flight. After traversing over the four waypoints as per expected flight path, the drone must land on the fifth waypoint indicating end of task.
4. Within the same script, you must write an image processing code that detects the colors Red, Green and Blue placed in the simulation. You must create publisher topics **'/red'**, **'/green'** and **'/blue'**. The number of Red patches detected within the simulation must be published under the topic /red. Do the same for the rest. A boiler plate code is provided to you in the scripts folder titled **task_1.2.py**. **The boiler plate code helps you convert your ROS Image message into an OpenCV image upon which you can run your Image Processing code thereby reducing your work. You only need to run Image Processing on the opencv image after conversion**.
5. Expected output, if the simulation scene has 3 Red patches, 2 Green patches and 3 Blue patches, after detecting them, you must publish 3 under the topic /red, 2 under the topic /green and 3 under the topic /blue. **Your code will be thoroughly scrutinized and will be passed through multiple V-REP scenes in order to verify your color detection.**

From the tutorials, you have learned

- Basics of Image Processing

Scene Description

Load the given scene *Scene.ttt* in V-REP simulator. The scene looks as shown in Figure 1:

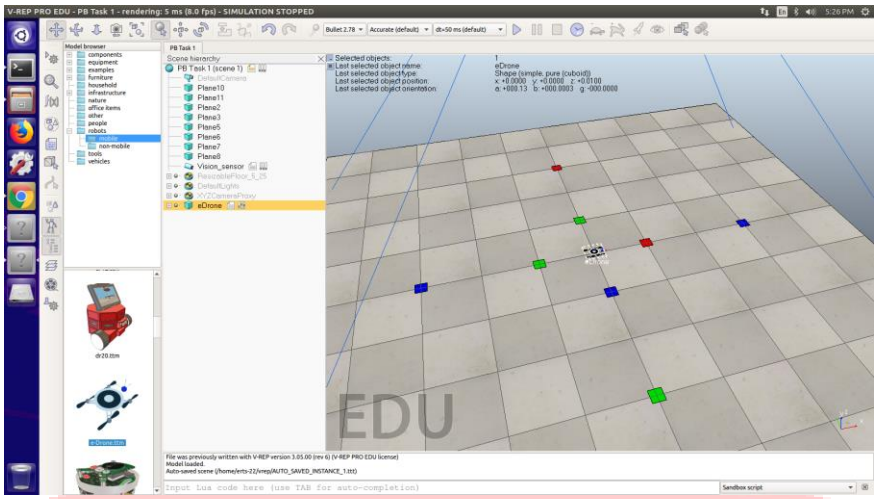

Figure 1: Scene.ttt

Following are the various objects in the scene:

Pluto Drone: There is one Pluto Drone model with the name eDrone.
Vision_sensor: This gives an image within the blue region with a resolution 640 x 480.
The image of the vision sensor is visible in the floating view window named Vision_sensor present in the side of the scene on running the simulation.
Color Patches: These are patches of colors Red, Green and Blue. You must detect these patches using Image Processing.

Points to remember

- Simulation setting should not be changed
  - Dynamics engine : Bullet 2.78
  - Dynamics settings : Accurate (default)
  - Simulation time step : dt = 50 ms (default)
- You are not supposed to change any of the templates
- **Create a PID script or edit the given example script by adding your PID code. Finally, run it to control the drone. The drone must visit the given waypoints.**
- **Within the same script write your Image Processing code to detect the patches.**
- Run "*rostopic type /topic_name*" to see what the message type of the corresponding topic is
- Run "*rosmsg show topic_type"* to see what the message structure is
- You cannot run two nodes with the same name. Look into the group tag in launch file to understand how you can run two nodes with same name.

Submission Instructions:

Follow the instructions below to submit your Task.

1. **Bag File:**
   a. First launch your Pollinator Bee package by running the following command **after loading your V-REP scene**:

roslaunch pollinator_bee pollinator_bee.launch

Make sure your launch file has a node of task_1.2.py. Comment out task_1.1.py node.

b. Next run the rosbag command to record your work. The following command records your work for 15 seconds and saves it with a .bag extension in the directory from where you executed the command:

rosbag record /whycon/poses /whycon/image_out /red /blue /green --duration=15s --chunksize=10

c. Next step is to compress the .bag file that is created before you can upload it. Run:

rosbag compress –j ~file_name.bag

d. Rename the compressed bag file as **<team_id>.bag**

2. **Python Code:**
   - **a.** You must submit your PID script that you developed and the image processing script.
   - b. Rename the python script as **<team_id>.py**

3. **Video Submission:**
   - a. Using screen record, record video of the drone performing the task.
   - b. Name the video as PB_eYRC#<team_id>
   - c. Upload instructions will be provided on the portal.

Store the files mentioned above in a folder and compress the folder into .zip file and rename the folder as <team_id>. Do not place the video within the .zip folder. You must upload that separately on YouTube. Instructions for uploading this video are given on the portal.

NOTE: You must upload all of the following: (i) bag file and (ii) Python code and (iii) Think and Answer.pdf. Please place all these files inside a .zip file before uploading. You must also upload the video to YouTube.

Please follow the naming convention strictly as specified in each step. Failure to do so may lead to repercussions.

Your final .zip output must be of the following structure:

<team_id>.zip

<team_id> *[folder]*

- **<team_id>.bag**
- **<team_id>.py**
- **<team_id>.pdf**

Instructions for uploading the folder will be provided on portal

"Float like a butterfly and sting like a bee"

- **Muhammad Ali**

Good Luck!!!