

Final Practical Project: Object Tracking

1. General Description.

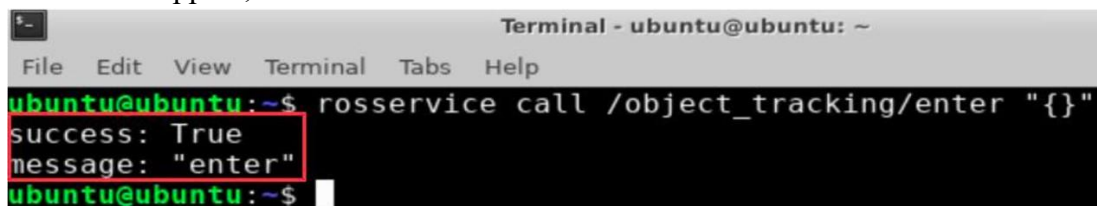
First, we need to identify the color, and we use the LAB color space to process it. We convert the RGB color space to LAB, then perform binarization processing and operations such as dilation and erosion to obtain a contour that only contains the target color, and then frames the color contour with a circle to achieve object color recognition.

Then, process the height of the robotic arm after the recognition, take the x, y, z coordinates of the center point of the image as the set values, and use the currently acquired x, y, z coordinates as the input values to update the PID.

Last, it will be calculated according to the feedback on the image position. Finally, the coordinate value will change linearly through the change of position to achieve tracking.

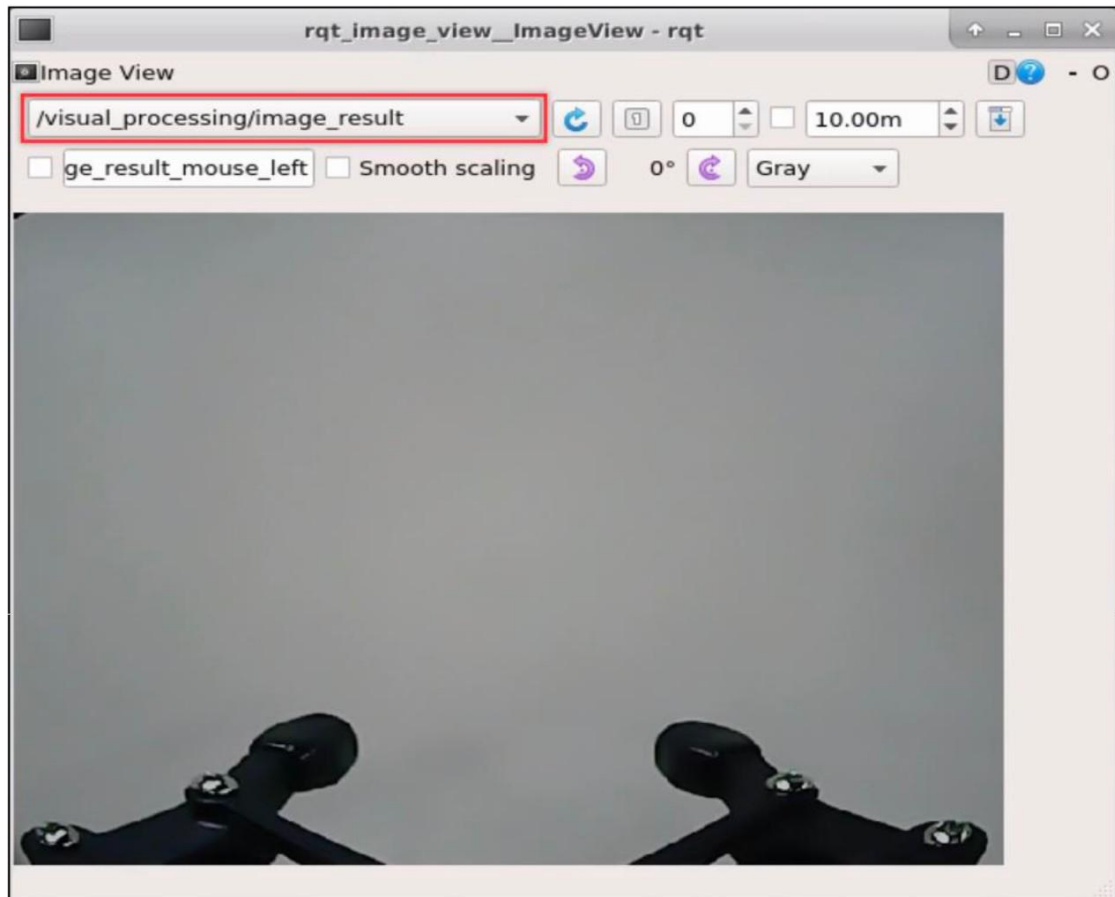
2. Experimental Procedure

- (1) Power on the ArmPi Pro, select the WIFI host to connect the robot and control the system desktop remotely via No Machine.
- (2) Click the Applications icon in the lower left corner and select "Terminal Emulator" in the popup interface to open the command line.
- (3) Enter the command "**roslaunch object_tracking object_tracking_node.py**". Then, press "Enter" to activate the object tracking node.
- (4) Do not close the previously opened terminal, then open a new terminal, and enter the command "**rosservice call /object_tracking/enter "{}"**", and then press "Enter" to enter the target tracking game. After entering, a print prompt will appear, as shown below:



```
Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~$ rosservice call /object_tracking/enter "{}"
success: True
message: "enter"
ubuntu@ubuntu:~$
```

- (5) Do not close the previously opened terminal, then open a new terminal, enter the command "**rqt_image_view**", press "Enter", and wait a moment to open the rqt tool.
- (6) Click the position of the red box shown in the figure below to select the topic option (topic) of target tracking. The option is "**/visual_processing/image_result**", and the rest of the settings remain unchanged.



- (7) Enter the command **"rosservice call /object_tracking/set_running "data: true"** in the same way. If the box in the figure below appears, the startup is successful.

```
ubuntu@ubuntu:~$ rosservice call /object_tracking/set_running "data: true"
success: True
message: "set_running"
ubuntu@ubuntu:~$
```

- (8) After starting the game, we also need to set the parameters to select the target color of the tracking. Here we take the tracking blue as an example, enter the command **"rosservice call /object_tracking/set_target "data: 'blue'"**. (Note: To track green and red, you can fill in green or red in data: '. (Strictly case sensitive))

```
ubuntu@ubuntu:~$ rosservice call /object_tracking/set_target "data: 'blue'"
success: True
message: "set_target"
ubuntu@ubuntu:~$
```

- (9) To stop, enter the command **"rosservice call /object_tracking/set_running "data: false"**.

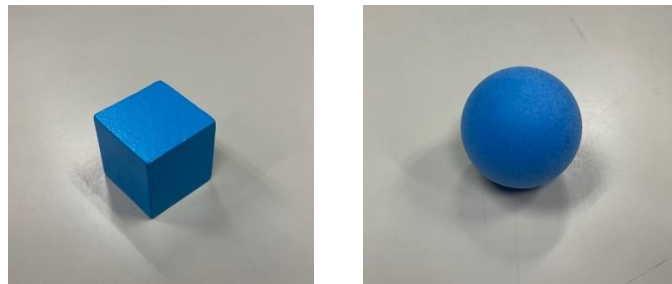
```
ubuntu@ubuntu:~$ rosservice call /object_tracking/set_running "data: false"
success: True
message: "set_running"
ubuntu@ubuntu:~$
```

- (10) To exit the game, enter the command “`rosservice call /object_tracking/exit "{}"`” to exit.

```
ubuntu@ubuntu:~$ rosservice call /object_tracking/exit "{}"  
success: True  
message: "exit"  
ubuntu@ubuntu:~$
```

3. Tasks

- (1) Based on the provided object tracking script, “`object_tracking_node.py`”, add **your own code** in the script so that the robot will **only track the blue ball** and ignore blue objects with other shapes, such as a cube.



Examples of Blue Cube and Ball

- (2) Based on the object tracking and visual processing script, modify the code to make the robot **track the “Apriltag” marker**.



Example of Cube with “Apriltag” Marker

Hint: The detection of the “Apriltag” marker has been implemented in the visual processing package. You don’t need to write extra code. This task can be achieved by modifying the original code.

- (3) Draw a **flow diagram** of this object-tracking application based on your understanding of the code.

4. TAs

If any guidance or assistance is expected, please make appointments with TAs via

Email:

- Yiming Huang: yhuangdl@link.cuhk.edu.hk