

Label Code Tracking

Orin board users can directly open a web page and enter IP address:8888 to access jupyter-lab and run directly. Jetson-Nano board users need to first enter the docker container, then enter the following command in docker:

```
cd
jupyter-lab --allow-root
```

Then open a web page and enter IP address:9999 to access jupyter-lab and run the following program.

1. Function Overview

Based on the label code positioning function, combined with the robotic arm to achieve label code tracking functionality.

Code path:

```
#Jetson-Nano users need to enter the docker container to view
~/dofbot_pro/dofbot_apriltag/scripts/Apriltag_Follow.ipynb
```

2. Code Block Design

- Import header files

```
import cv2 as cv
import threading
import random
from time import sleep
import ipywidgets as widgets
from IPython.display import display
from apriltag_identify import ApriltagIdentify
from apriltag_follow import Apriltag_Follow
from dofbot_utils.fps import FPS
from dofbot_utils.robot_controller import Robot_Controller
```

- Create instances, initialize parameters

```
apriltag_Identify = ApriltagIdentify()
follow = Apriltag_Follow()
model = 'General'

robot = Robot_Controller()
robot.move_init_pose()
fps = FPS()
```

- Create widgets

```
button_layout = widgets.Layout(width='320px', height='60px',
align_self='center')
```

```

output = widgets.Output()

# 开始追踪 Start tracking
start_button = widgets.Button(description='Start', button_style='success',
layout=button_layout)
# 取消追踪 Cancel tracking
cancel_button = widgets.Button(description='Cancel', button_style='danger',
layout=button_layout)
# 退出 exit
exit_button = widgets.Button(description='Exit', button_style='danger',
layout=button_layout)

box_button = widgets.VBox([start_button, cancel_button, exit_button],
layout=widgets.Layout(align_self='center'))
# 图像控件 Image widget
imgbox = widgets.Image(format='jpg', height=480, width=640)
# 垂直布局 Vertical layout
display_box = widgets.HBox([imgbox, box_button])

```

- Mode switching

```

def exit_button_Callback(value):
    global model
    model = 'Exit'

def start_button_Callback(value):
    global model
    model = 'Start'

def cancel_button_Callback(value):
    global model
    model = 'General'

exit_button.on_click(exit_button_Callback)
start_button.on_click(start_button_Callback)
cancel_button.on_click(cancel_button_Callback)

```

- Main program

```

def camera():

    global model
    # 打开摄像头 Open camera
    capture = cv.VideoCapture(0, cv.CAP_V4L2)
    capture.set(3, 640)
    capture.set(4, 480)
    capture.set(5, 30)
    # Be executed in loop when the camera is opened normally
    while capture.isOpened():
        try:
            _, img = capture.read()
            img = cv.convertScaleAbs(img, beta=35)
            fps.update_fps()
            img, msg = apriltag_Identify.getApriltagPosition(img)
            # if len(msg):
            #     print(msg)

```

```

if model == 'Start':
    follow.follow_function(msg)
if model == 'Exit':
    capture.release()
    break
fps.show_fps(img)
imgbox.value = cv.imencode('.jpg', img)[1].tobytes()
except Exception as e:
    print("program end")
    print(e)
    capture.release()

```

- Startup

```

display(controls_box,output)
threading.Thread(target=camera, ).start()

```

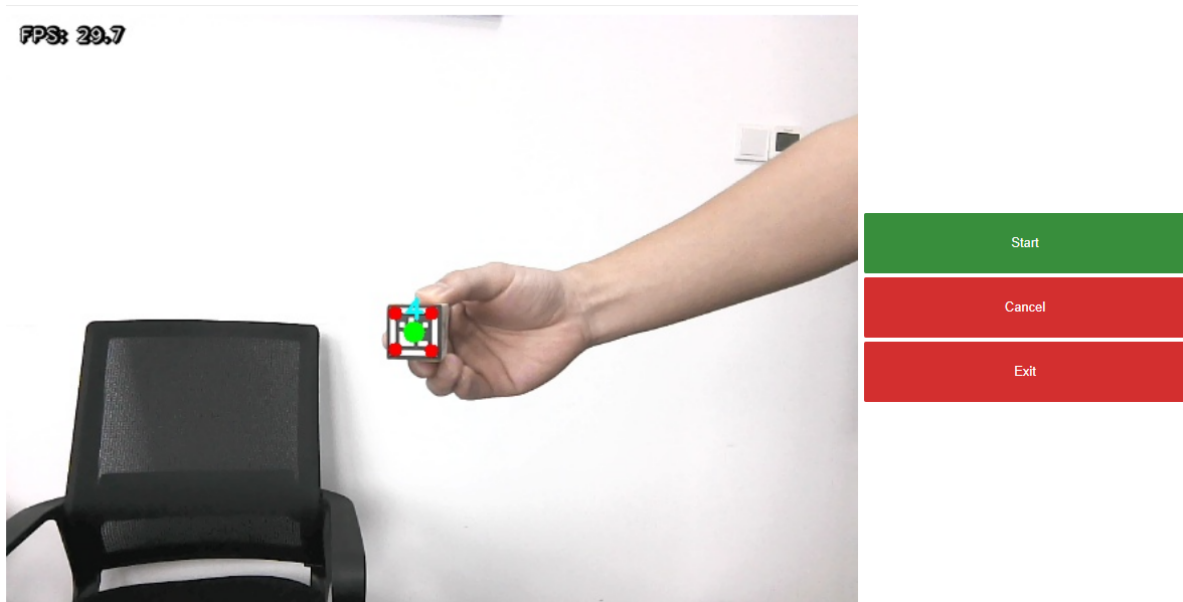
3. Run the Program

Click the run entire program button on the jupyterlab toolbar, then scroll to the bottom.



You can see the camera display. At this time, click the [Start] button to start the tracking function. Place the label code building blocks in the camera view. You can see that the label code is correctly positioned and the label code ID number is written. As you move the label code, the robotic arm will follow the label code movement.

Note that when moving the label code, the speed cannot be too fast, otherwise the robotic arm may not be able to keep up due to moving too fast.



If you want to stop tracking, please click the [Cancel] button.

If you need to end the program, please click the [Exit] button to avoid affecting other programs from calling resources.

