Voice Control Color Block Sorting

Before running the function, you need to close the App and large programs. For the closing method, refer to [4. Preparation] - [1. Manage APP control services].

1. Function Description

After the program starts, voice commands are issued to sort color blocks. After receiving the command, the program controls the robotic arm to grasp the corresponding color block and place it at the designated position according to the command content.

2. Startup and Operation

2.1. Startup

Users with Jetson-Nano board version need to enter the docker container and input the following commands. Orin board users can directly open the terminal and input the following commands,

```
#Start camera and inverse kinematics
ros2 launch dofbot_pro_info camera_arm_kin.launch.py
#Start speech recognition and broadcast
ros2 launch yahboom_speech speech.launch.py
#Color block color recognition
ros2 run dofbot_pro_voice_ctrl color_detect_VC
#Color block sorting and grasping
ros2 run dofbot_pro_voice_ctrl color_grasp_VC
```

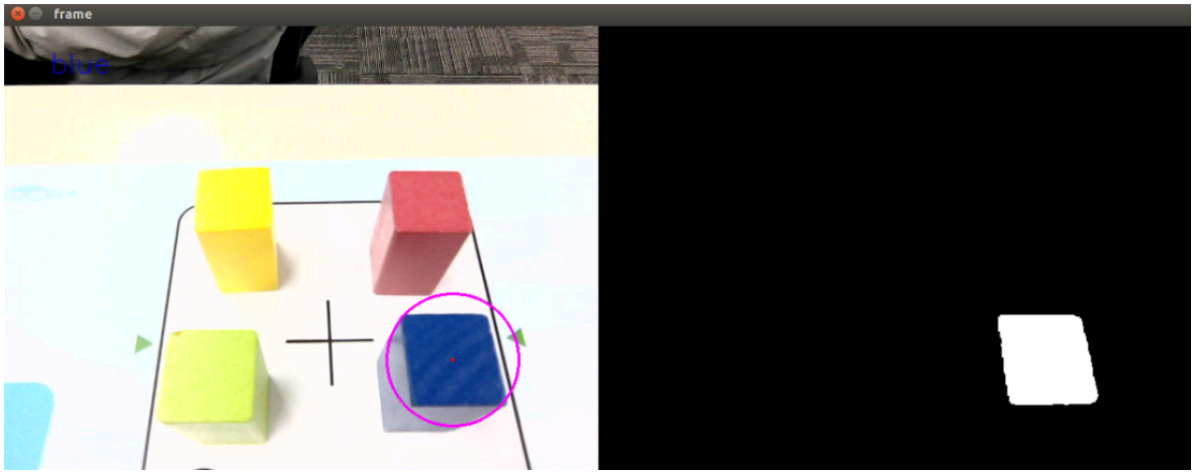
2.2. Operation Steps

2.2.1. Calibrate Color Block Colors

After all programs are running, you need to calibrate the color block colors. The upper left corner of the image will display the currently recognized color block color, initially red. The program starts the color calibration process. Select the top area of the red block with the mouse and release to complete calibration; then press the f key to start calibrating the green block. Similarly, select the top area of the green block with the mouse and release to complete calibration; press the f key to start calibrating the blue block, select the top area of the blue block with the mouse and release to complete calibration; press the f key to start calibrating the yellow block, select the top area of the yellow block with the mouse and release to complete calibration.

2.2.2. Voice Control

After completing the color block color calibration, say "Hello, yahboom" to the speech recognition module, and the speaker will broadcast "here". Then say "Sort the red/green/blue/yellow blocks" to the speech module. The robotic arm will lower its gripper to grasp the target color block and the speech module will broadcast "OK". After grasping the target color block, it will place it at the designated position. Finally, the robotic arm returns to its initial posture, and the speech broadcast module will announce "Placement complete".



3. Core Source Code Analysis

3.1. Color Recognition Node color_detect_VC

`Source code path:`
`~/dofbot_pro_voice_ctrl/dofbot_pro_voice_ctrl/Color/color_detect_VC.py`

Mainly explains how to subscribe to and process speech recognition result topics and publish voice broadcast topics.

```
#Create subscriber for speech recognition result topic
self.sub_voice =
self.create_subscription(Int8,"voice_result",self.getVoiceResultCallback,1)
#Create publisher for voice broadcast topic
self.pub_playID = self.create_publisher(Int8,"player_id", 1)

#Callback function, processes received speech recognition result topic data. 100-
103 respectively represent "grasp red/green/blue/yellow block" commands.
According to the command, modify self.target_color, then send voice broadcast
topic data, broadcasting "OK", finally, modify self.Track_state to "identify"
indicating start color recognition and self.start_flag value to True indicating
that target color block position information can be sent
def getVoiceResultCallback(self,msg):
    if msg.data == 100:
        self.target_color = "red"
        self.pubPos_flag = True
    elif msg.data == 101:
        self.target_color = "green"
        self.pubPos_flag = True
    elif msg.data == 102:
        self.target_color = "blue"
        self.pubPos_flag = True
    elif msg.data == 103:
        self.target_color = "yellow"
        self.pubPos_flag = True
    play_id = Int8()
    play_id.data = 45
    self.pub_playID.publish(play_id)
    print("Get the target color is ",self.target_color)
    self.Track_state = "identify"
    self.start_flag = True
```

3.2. Robotic Arm Color Block Grasping Node color_grasp_VC

`Source code path:`

`~/dofbot_pro_voice_ctrl/dofbot_pro_voice_ctrl/Color/color_grasp_VC.py`

Mainly explains how to publish voice broadcast topics.

```
#Create publisher for voice broadcast topic
self.pub_playID = self.create_publisher(Int8,"player_id", 1)
#At the end of the grasp function, publish voice broadcast topic data, the
broadcast content is "Placement complete"
play_id = Int8()
play_id.data = 81
self.pub_playID.publish(play_id)
```