Voice Control Tracking and Grasping Color Block

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 runs, voice commands are issued to instruct the robotic arm to track and grasp color blocks. The program will control the robotic arm to track color blocks and grasp color blocks when conditions are met according to the command.

2. Startup and Operation

2.1. Startup

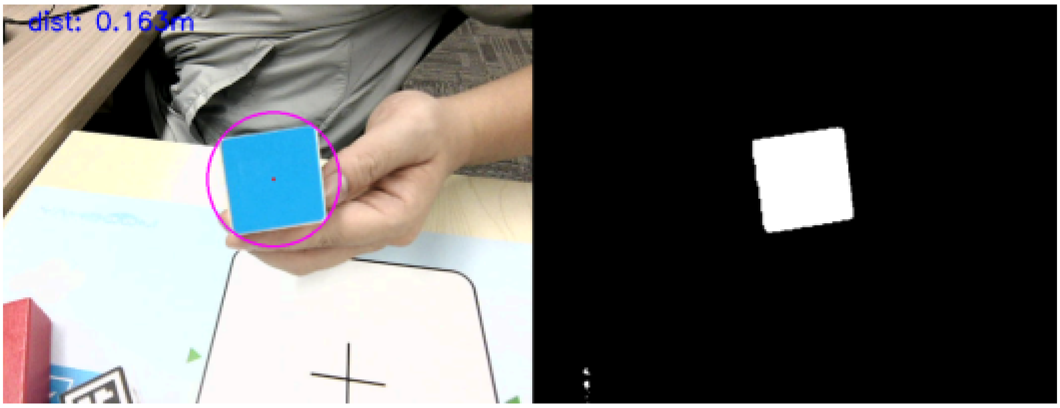
Users with Jetson-Nano board version need to enter the docker container and input the following commands. Orin board users can directly open the terminal and input the following commands,

```
#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
#Start voice control tracking and grasping color block
ros2 run dofbot_pro_voice_ctrl color_follow_VC
```

2.2. Startup Steps

2.2.1. Calibrate Color Block

After the program starts, the robotic arm will move to the tracking posture. Then hold a color block and make it appear in the image. Select an area of the color block with the mouse and release to complete calibration.



2.2.2. Voice Control

After color calibration is completed, say "Hello, yahboom" to the speech recognition module, and the speaker will broadcast "here". Then say "Track and clip color blocks" to the speech module. The robotic arm will then track the held color block. Slowly move the color block, and the robotic arm will move correspondingly. When the robotic arm becomes stationary, tracking ends. After a "beep" sound, the robotic arm will grasp the held color block and place it at the designated position. Finally, the robotic arm returns to its initial posture.

3. Core Code Analysis

3.1. Tracking and Grasping Color Block Node color_follow_VC

`Source code path:` `~/dofbot_pro_voice_ctrl/dofbot_pro_voice_ctrl/Color/color_follow_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, if the received speech recognition result data is 106, it
means the control command is "Track and clip color block", then change
self.start_flag value to True indicating that color block position information
can be published and publish voice broadcast topic, the broadcast audio is "OK"
def getVoiceResultCallback(self,msg):
    if msg.data == 106:
        self.start_flag = True
        play_id = Int8()
        play_id.data = 45
        self.pub_playID.publish(play_id)
```