Palm control of the two-dimensional gimbal servo

Note: The ROS_DOMAIN_ID of the Raspberry Pi and the microROS control board need to be consistent. You can check [MicroROS Control Board Parameter Configuration] to set the microROS control board ROS_DOMAIN_ID. Check the tutorial [Connect MicroROS Agent] to determine whether the IDs are consistent.

1、Introduction

MediaPipe is an open source data stream processing machine learning application development framework developed by Google. It is a graph-based data processing pipeline for building data sources using many forms, such as video, audio, sensor data, and any time series data. MediaPipe is cross-platform and can run on embedded platforms (Raspberry Pi, etc.), mobile devices (iOS and Android), workstations and servers, and supports mobile GPU acceleration. MediaPipe provides cross-platform, customizable ML solutions for live and streaming media.

2、Program Description

After the program is started and the camera captures the image, the car's 2D gimbal will follow the movement of the palm in the screen. The palm movement speed here should not be too fast, otherwise the image processing will not be able to keep up, which will cause lag.

2.1、Program code reference path

After entering the docker container, the source code path of this function is as follows.

```
/root/yahboomcar_ws/src/yahboomcar_mediapipe/yahboomcar_mediapipe/
```

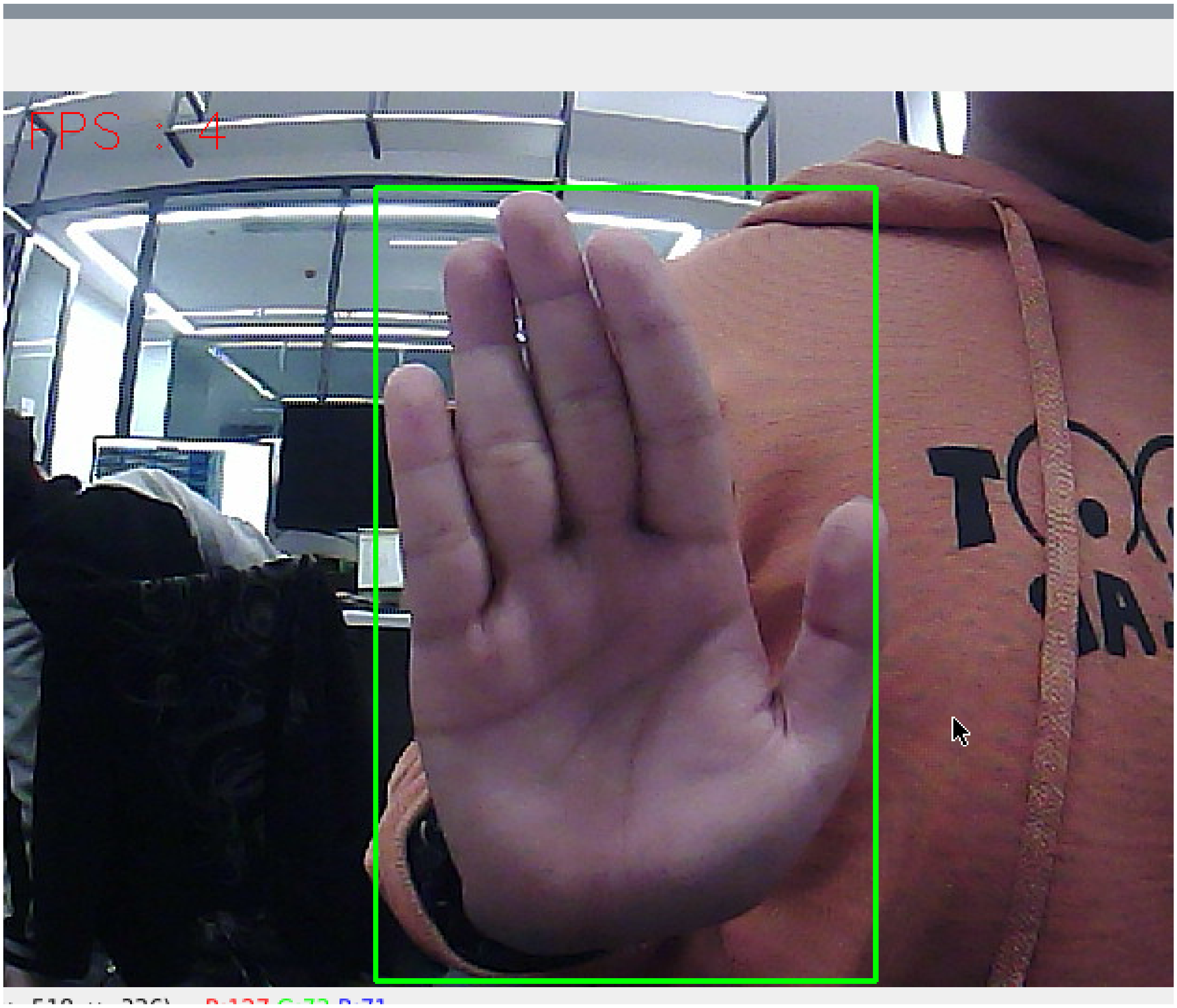
3、Program start

3.1、Start command

After entering the docker container, enter the following command in the terminal

```
ros2 run yahboomcar_mediapipe control_shape
```

After the function is turned on, the car's 2D gimbal will follow the movement of the palm.



4、Code analysis

4.1、control_shape.py

- Code reference location

```
/root/yahboomcar_ws/src/yahboomcar_mediapipe/yahboomcar_mediapipe
```

- code analysis

  1）、Import important library files

```
from media_library import *
```

  2）、Detect hands and obtain finger information

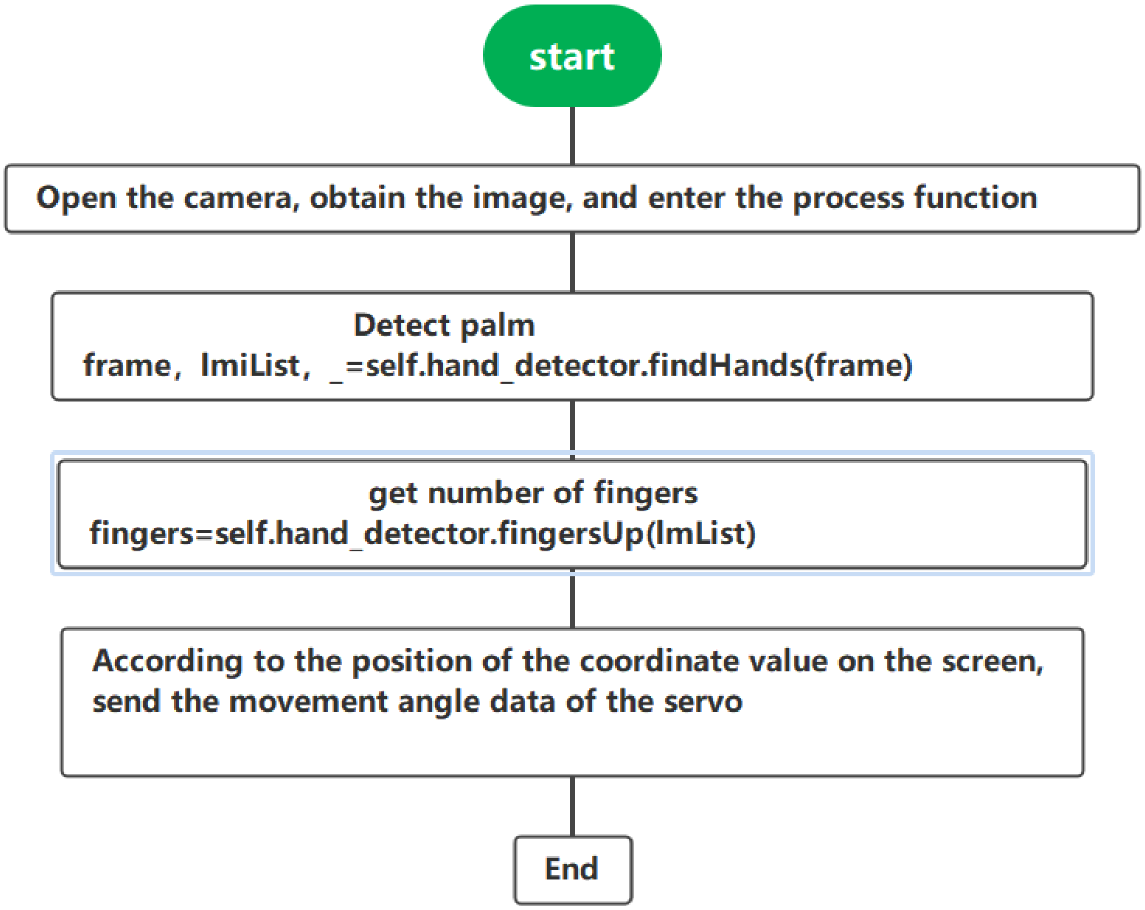
```
frame, lmList, bbox = self.hand_detector.findHands(frame)
#bbox is the minimum and maximum xy value of the frame that frames the detected
hand. This value is very important. By calculating the center coordinates, the
position of the palm on the screen can be determined. The source code is in
media_library.py
angle = self.hand_detector.ThumbTOforefinger(lmList)
#Here you can calculate the bending angle of your thumb to control the opening
and closing of the gripper.
```

3）、Calculate the angle of each servo

```
point_x = lmList[9][1]
point_y = lmList[9][2]
if point_y >= 270: self.y -= 2
elif point_y <= 210: self.y += 2
if point_x >= 350: self.x -= 2
elif point_x <= 290: self.x += 2
if self.x <= -90: self.x = -90
elif self.x >= 90: self.x=90
if self.y <= -90: self.y = -90
elif self.y >=20: self.y = 20
```

It can be seen that indexX and indexY are the coordinates of the center point of the frame. Through judgment, the angle at which the servo should rotate is calculated. The parameters of each angle calculation method are calibrated based on the actual angle that can be rotated and the visual range of the camera. .

4.2、flow chart


start
Open the camera, obtain the image, and enter the process function
Detect palm
get number of fingers
According to the position of the coordinate value on the screen, send the movement angle data of the servo
End

```
frame，ImiList，_=self.hand_detector.findHands(frame)
fingers=self.hand_detector.fingersUp(ImList)
```