

## 6、Face Tracking

---

### 6、Face Tracking

#### 6.1、Face detection Introduction

#### 6.2、Operating steps

##### 6.2.1、Start up

##### 6.2.2、PID tuning

##### 6.2.3、Face tracking

### 6.1、Face detection Introduction

In OpenCV, face detection in video only reads each frame of image from the camera, and then uses the static image detection method for detection. Face detection requires the classifier:

@ Face detector(default): haarcascade\_frontalface\_default.xml

@ Face detector(fast Harr): haarcascade\_frontalface\_alt2.xml

@ Face detector(Side view): haarcascade\_profileface.xml

@ Eye detector(left eye): haarcascade\_lefteye\_2splits.xml

@ Eye detector(right eye): haarcascade\_righteye\_2splits.xml

@ Mouth detector: haarcascade\_mcs\_mouth.xml

@ Nose detector: haarcascade\_mcs\_nose.xml

@ Body detector: haarcascade\_fullbody.xml

@ Face detector(fast LBP): lbpcascade\_frontalface.xml

@ Only open eyes can be detected: haarcascade\_eye.xml

@ Only person with glasses can be detected: haarcascade\_eye\_tree\_eyeglasses.xml

@ <https://github.com/opencv/opencv/tree/master/data> Download classifier file link

haarcascade\_profileface.xml is the cascading data of Haar. This xml can be obtained from this link [https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade\\_profileface.xml](https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_profileface.xml)

Next, we can start face detection by `face_cascade.detectMultiScale()`. We need to convert the image into a grayscale, then, transfer each frame of the image obtained by the camera into `.detectMultiScale()`.

(Note: we need to ensure enter the correct location of `haarcascade_profileface.xml` correctly.)

OpenCV API function:

**`detectMultiScale(const Mat& image, vector& objects, double scaleFactor=1.1,int minNeighbors, int flag,cvSize)`**

Parameter analysis:

image --- Input grayscale image

objects --- The rectangular box vector set of the detected object

scaleFactor --- Each scale parameter in the image scale, the default value is 1.1.

minNeighbors --- Default is 3.

minNeighbors --- The default value of 3 indicates that there are at least 3 overlap detections, so we think that the face is exist.

minSize --- Target minimum size

maxSize --- Target maximum size

## 6.2、Operating steps

### 6.2.1、Start up

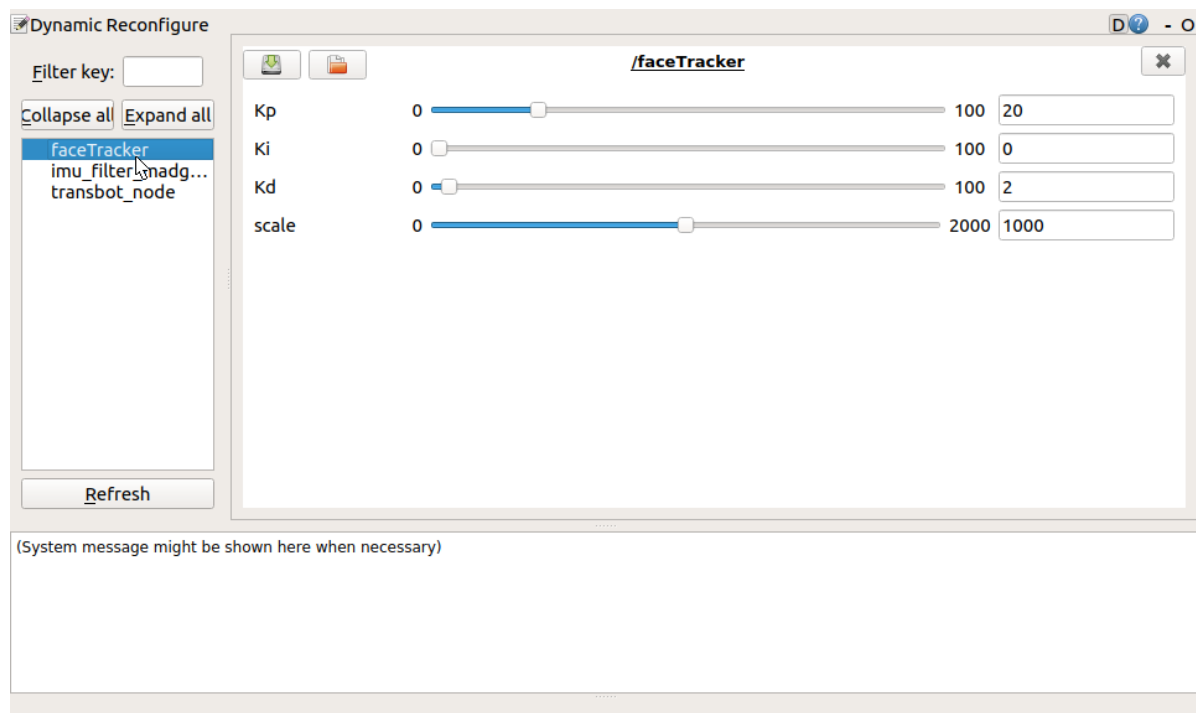
Start the face tracking. (Jetson nano side)

```
roslaunch transbot_bringup bringup.launch  
rosvrun transbot_facetracker face_follow.py
```

### 6.2.2、PID tuning

Dynamic parameter debugging

```
rosvrun rqt_reconfigure rqt_reconfigure
```



Select the 【faceTracker】 node to adjust 【Kp】 , 【Ki】 , 【Kd】 and 【scale】 . When the slider is in the dragging state, the data will not be passed into the system, only after release; You can also select a row and then scroll the mouse wheel.

### **6.2.3、 Face tracking**

After running the program, when the face is displayed in the picture, when a square surrounds the face, the Yuntai camera will move with the movement of the face.