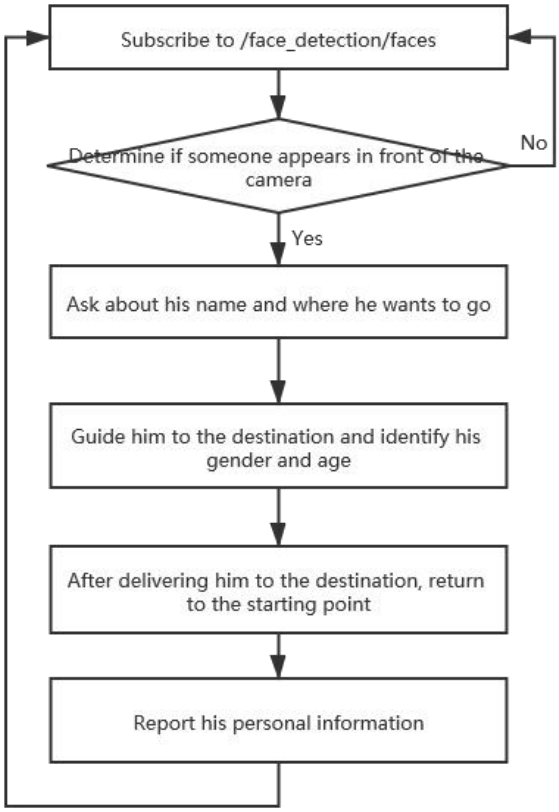
Robot Software Engineering Technical Report

Zhai Xiaolin 1711533

1. Function description:

The robot is responsible for receiving guests at the door. If someone appears in front of the camera, the robot asks the guest his name and where he wants to go. If no one appears in front of the camera, the robot continues to wait. After getting the guest's information, the robot takes the guest to the place they want to go. In this process, the Baidu API is used to identify the guest's gender and age. After the guest is taken to the destination, the robot returns to the starting point and will telll the owner of the guest information.



2. Technical realization:

2.1 farewell_vision.py node

2.1.1 Import package

Import vision related function packages, including image from sensor_msgs.msg, FaceArrayStamped from opencv_apps.msg, CvBridge, CvBridgeError from cv_bridge. Import and call API related function packages, including request, base64, time, urllib, urllib2, ssl, json, base64.

2.1.2 Identify whether someone appears in front of the camera

Subscribe to /face_detection/faces, the message type is FaceArrayStamped, and the callback function is faceCallback, to determine whether the message msg.faces is empty, if it is empty, it means that there is no one in front of the lens, if it is not empty, it means that there is someone in front of the lens, then go to / farewell_speech sends a message and the voice node asks the guest's

name and where they want to go.

2.1.3 Call Baidu API to identify gender and age

First call the take_photo function to take a picture of the guest and save it. Then call Baidu API for identification, send the identified gender to /farewell_age, and send the identified age to /farewell_sex.

Steps to call Baidu API:

① Obtain the token and use it for verification. Pay attention to replace it with the API Key and Secret Key you applied for. The code is as follows.

```
def getToken():
    global token

    host =
'https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&cl
ient_id= &client_secret= '
    request = urllib2.Request(host)
    request.add_header('Content-Type', 'application/json; charset=UTF-8')
    response = urllib2.urlopen(request)
    content = response.read()
    if (content):
        token = json.loads(content)['access_token']
```

②The uploaded pictures for inspection must be Base64 encoded. It should be noted that the base64 encoding of the picture does not include the picture header.The code is as follows.

```
def imgToBase64(imgPath):
    with open(imgPath, "rb") as f:
        base64_data = base64.b64encode(f.read())
        return base64_data
```

③Call the interface for face recognition. The input of this function is the base64 encoding of the picture. The more important parameter in the request is the face_field. By default, it only returns the position, probability and rotation angle of the face frame, age (age prediction), gender (gender) and other attributes. , Need to be added in this parameter.

```
def faceDetect(imgBase64):

    request_url = "https://aip.baidubce.com/rest/2.0/face/v3/detect"
    request_url = request_url + "?access_token=" + token
    request = urllib2.Request(request_url)
    request.add_header('Content-Type', 'application/json')
    data = {"image": imgBase64, "image_type": "BASE64", "face_field":
"age,beauty,expression,face_shape,gender"}
    response = urllib2.urlopen(request, urllib.urlencode(data))
    content = response.read()
    if content:
        return content
```

④ After face recognition, the json data is returned, and then the json data is parsed.

```
def getFeature(self):
    getToken()
    imgPath = "/home/zhaixiaolin/"+self.img_title
    result = json.loads(faceDetect(imgToBase64(imgPath)))['result']
    face_list = result['face_list'][0]
    location = face_list['location']
    age = face_list['age']
    beauty = face_list['beauty']
    expression = face_list['expression']['type']
    gender = face_list['gender']['type']
    self.feature=gender.encode("utf-8")+' '+str(age)
    print(self.feature)
```

2.2 farewell_speech.py node

2.2.1 import package

Import voice-related function packages, import String from std_msgs.msg, Int32 from std_msgs.msg, and SoundClient from sound_play.libsoundplay.

2.2.2 ask the guest's name and where to go

Subscribe to the /farewell_speech topic. If the message content is 1, ask the guest's name and the place to go. Subscribe to the /lm_data topic, get the content of the voice, and assign values to the name and place variables.

2.2.3 communicate with navigation nodes

Send the obtained location information to the /farewell_nav topic, and the farewell_nav node starts to move after receiving the information.

2.2.4 feedback guest information to host

Subscribe to /farewell_age and /farewell_sex two topics to get guests' age and gender information. Subscribe to the /farewell_feature topic at the same time. When the navigation node sends a message to the topic with a content of 1, it indicates that the robot has sent the guest to the destination. The voice node will say "We arrived." after receiving the message; send it to the topic at the navigation node When the content of the message is 2, it indicates that the robot has returned to the starting point, and it should report the guest's information to the host. After receiving the message, the voice node says the guest's information "name is a sex of age.".

2.3 farewell_navigation.py node

2.3.1 import package

Import related function packages for mobile robot navigation, such as tf.transformations, move_base_msgs.msg, geometry_msgs.msg, actionlib, etc., and use the following two commands to connect to robot-related services and wait for initialization to complete:

```
self.move_base = actionlib.SimpleActionClient("move_base", MoveBaseAction)
self.move_base.wait_for_server(rospy.Duration(120))
```

2.3.2 guide guests to their destination

Subscribe to the /farewell_nav topic, get destination information, and guide guests to the destination. After reaching the destination, send a message to the /farewell_feature topic. After the voice node receives the message, the robot will say "We arrived." The sample code is as follows:

```
if data.data == 1:
    rospy.loginfo("Going to the bar")
    rospy.sleep(2)
    self.goal.target_pose.pose = self.locations['A']
    self.move_base.send_goal(self.goal)
    waiting = self.move_base.wait_for_result(rospy.Duration(300))
    if waiting == 1:
        rospy.loginfo("Reached the bar")
        self.pub.publish(1)
        rospy.sleep(2)
        tag=1
```

2.3.3 the robot returns to the starting position

After the robot returns to the starting position, it sends a message to the /farewell_feature topic. After the voice node receives the message, the robot will report the guest information and send a message to the /farewell_reset topic to initialize the code and prepare to welcome the next guest. The sample code is as follows :

```
if tag==1:
    rospy.loginfo("Going back home")
    rospy.sleep(2)
    self.goal.target_pose.pose = self.origin
    self.move_base.send_goal(self.goal)
    waiting = self.move_base.wait_for_result(rospy.Duration(300))
    if waiting == 1:
        rospy.loginfo("Reached home")
        self.pub.publish(2)
        self.pub1.publish(1)
        rospy.sleep(2)
    tag=0
```