**关于“定点寻找指定人物”功能实现的技术报告**

1710675

杨宇航

一、功能描述

机器人接受寻找某个人的语音命令，根据导航自主前往预先设定的三个地点，到达一个寻找点后，机器人旋转一圈并利用摄像头和人脸识别功能识别目标人物的人脸，若找到待寻找人物则返回初始点进行报告；若该地点没有则前往下一个地点寻找；若所有寻找点都没有找到则返回初始点报告没有发现目标人物。

二、技术实现

1.自主导航结点navi\_point

1.1导入功能包

导入移动机器人导航的相关功能包，比如tf.transformations, move\_base\_msgs.msg, geometry\_msgs.msg, actionlib等，并利用如下两条命令连接到机器人相关服务并等待初始化完成：

self.move\_base = actionlib.SimpleActionClient("move\_base", MoveBaseAction)

self.move\_base.wait\_for\_server(rospy.Duration(120))

1.2确定目标点机器人位姿

根据已经建立的地图确定三个搜索点和原点的机器人位姿，并用一个列表储存。在本功能中，原点和三个搜索点分别为：

1.3与机器人语音结点talkbot通信

导航结点navi\_point和机器人语音交互结点talkbot结点直接需要通过通信建立联系，可使用topic方式，通信格式使用String类型，具体连接方式如图1所示。其中，navi\_to\_point话题接受机器人前往某个目的地的具体命令，navigation\_feed\_point话题反馈机器人十分到达目标点的信息。

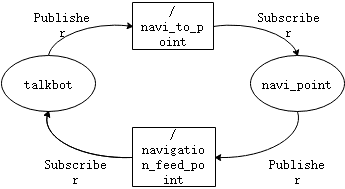


图1 导航结点和主节点的通信

1.4机器人自主移动

首先将地图和时间戳信息传输给机器人，具体命令如下所示：

self.goal.target\_pose.header.frame\_id = 'map'

self.goal.target\_pose.header.stamp = rospy.Time.now()

之后根据navi\_to\_point接收到的具体命令实现具体的导航功能，此次以导航至study(A点)为例说明。第一步将目的地列表中与命令相对应的点的坐标发送给机器人作为目标；第二步向机器人发布移动命令并等待移动完成；第三步机器人移动完成，向navigation\_feed\_point发送移动完成的消息。示例代码如下：

# Robot will go to the study

if data.data == "go to the study":

rospy.loginfo("Going to the study")

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 study")

self.navigation.publish('Reached the study')

2.talkbot语音交互结点实现

2.1导入功能包。

由于该节点是实现机器人功能的基础结点，所需功能包较多，包括pocketsphinx语音包中相关结点、opencv\_app包中人脸识别结点，以及一些机器人所需位置信息的文件。导入完毕，初始化语音合成相关功能。

2.2建立通信

建立和语音识别结点、人脸识别结点、导航结点的通讯，与导航结点的通讯在上一小节已经叙述，此处不再重复。

（1）与人脸识别结点的联系。人脸识别结点中话题/face\_recognition/output输出了人脸识别的结果，包括人脸x, y位置、人脸标签和置信度信息，其类型为FaceArrayStamped，通过订阅该话题即可获得需要信息。注意该话题输出的是列表类型，每个列表元素是一帧人脸信息，因此读取数据时要具体到每一帧，代码示例如下：

def recogniton\_back(self,face\_rec):

rec = face\_rec.faces

if rec:

self.recognition\_x=rec[0].face.x

self.recognition\_y=rec[0].face.y

self.recognition\_label=rec[0].label

self.recognition\_confidence=rec[0].confidence

else:

self.recognition\_x=0

self.recognition\_y=0

self.recognition\_label=""

self.recognition\_confidence=0

图2为talkbot结点订阅face\_recognition/output话题的示意图。

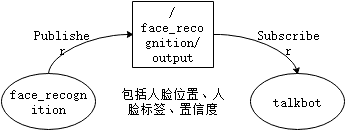


图2 订阅人脸识别结果话题

（2）与语音结点的联系。语音识别的结果输出到话题/lm\_data中，类型为String，通过订阅该话题并识别具体内容，并结合语音合成功能即可实现与机器人的语音交互。

2.3向机器人发布寻找命令

机器人通过话题/lm\_data识别到寻找命令，开始实现。通过一个循环机器人依次访问3个目标点，具体流程是：

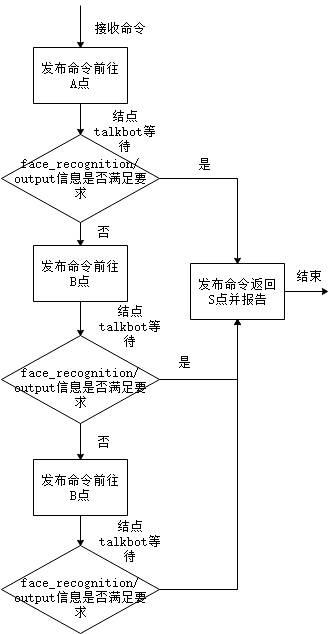
（1）talkbot结点向话题navi\_to\_point发布命令“go to the study”（简称A点），机器人导航结点navi\_point接收命令前往A点，此时talkbot结点通过rospy.wait\_for\_message()函数等待话题navigation\_feed\_point反馈的消息。talkbot结点收到反馈消息，表示已到达目标点，机器人开始旋转寻找人物，查看订阅话题/face\_recognition/output所获得的人脸识别信息是否满足待寻找人的特征，若满足则停止旋转并再次向navi\_to\_point发包命令“go back”使机器人返回原点（S点），此时talkbot结点再次阻断，待机器人返回，输出语音；若在A点旋转一周没有找到目标人物，转动停止，进行步骤2。步骤2、 3到达指定点后寻找时执行同样的旋转。

图3 搜索流程

（2）talkbot结点向话题navi\_to\_point发布命令“go to the bedroom”（B点），机器人前往B点，talkbot结点阻塞等待导航结果。到达B点后检查/face\_recognition/output信息是否满足，满足则向话题navi\_to\_point发布命令使机器人回到S点报告，不满足则转到步骤3。

（3）talkbot结点向话题navi\_to\_point发布命令“go to the living room”（C点），机器人前往C点，talkbot结点阻塞等待导航结果。到达C点后检查/face\_recognition/output信息，无论是否满足均返回S点进行语音报告。图3所示为搜索命令执行流程。

2.4 机器人旋转的实现

建立Twist消息类型(表示线速度角速度)的结构变量move，设定角速度move.angular.z，建立Publisher以指定频率rate传输move至话题/cmd\_vel\_mux/input/teleop（采用了用于键盘遥控的话题），可以使机器人旋转，停止发布则机器人停止转动。转动一周需要发布的次数是(1/move.angular.z)\*rate\*2\*pi.