

KCF 跟随功能

1. 功能简介

KCF 跟随功能是通过 KCF 算法来实现对目标的跟踪的，KCF 全称为 Kernel Correlation Filter，即核相关滤波算法。KCF 跟随功能可以利用摄像头识别我们在窗口中框选的物体，然后跟随物体进行运动。

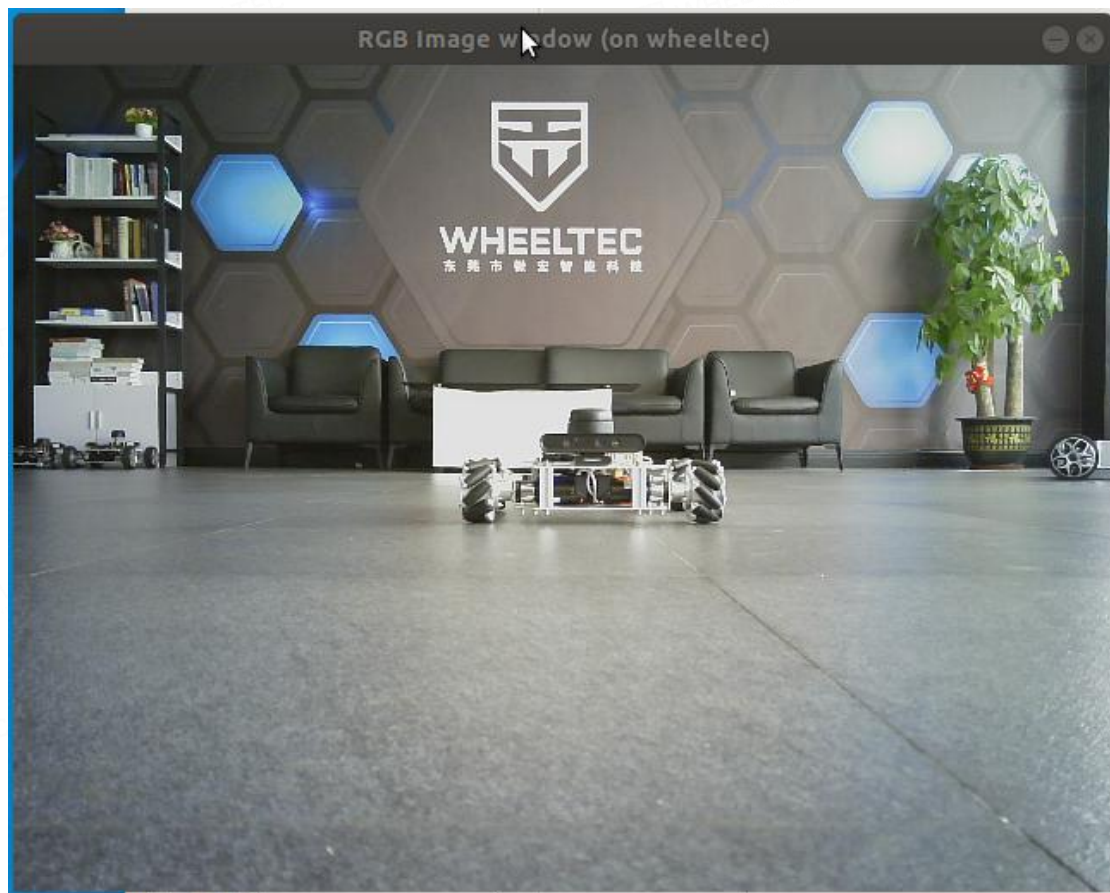
2. 使用方法

使用前需要确保上位机连接小车 WiFi，并已经 SSH 远程登录到小车端。

1) 打开终端输入启动 KCF 功能的指令

```
roslaunch kcf_track kcf_tracker.launch
```

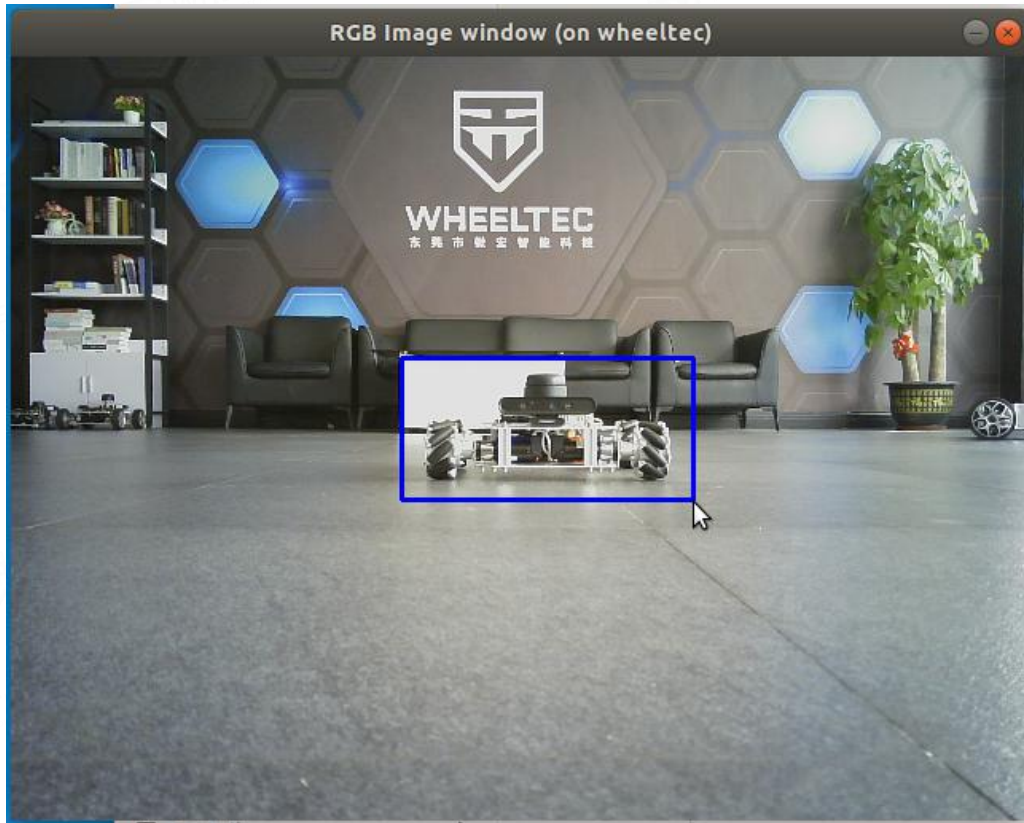
KCF 跟随节点开启后，会弹出一个框，框内是小车摄像头所看到的实时画面，



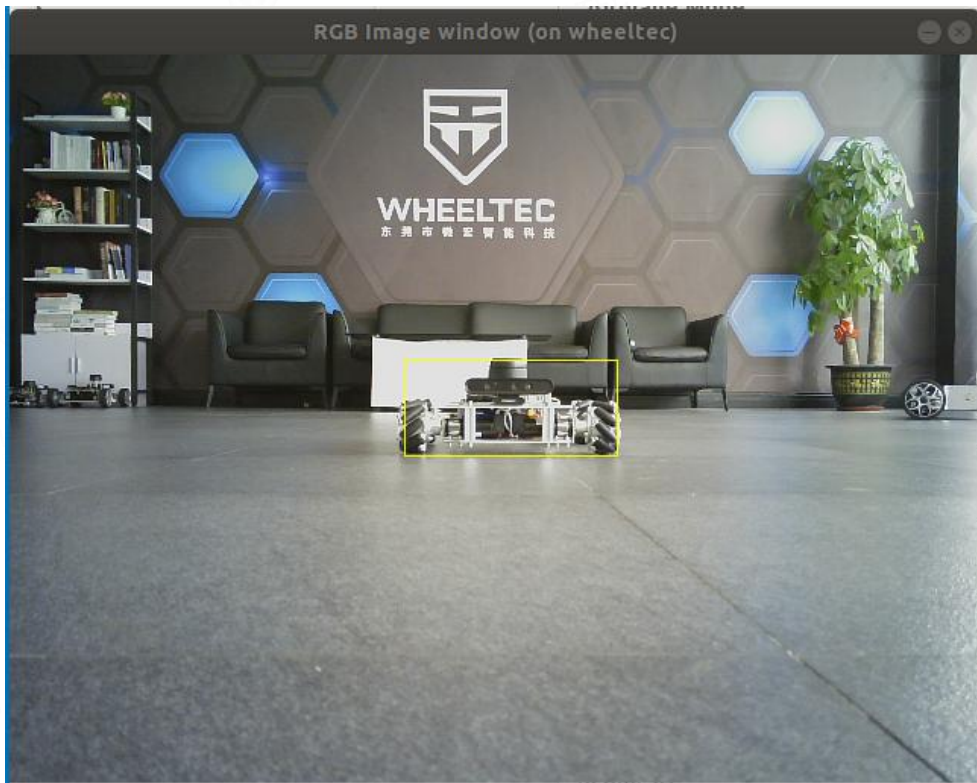
RGB image window 显示摄像头实时画面

我们按住鼠标，拖动框选住想要让小车跟随的物体，此时框边为蓝色，松开

鼠标后，小车就会运动跟随被选物体，此时框边为黄色。



框选跟随目标物体



被跟随物体已被框选

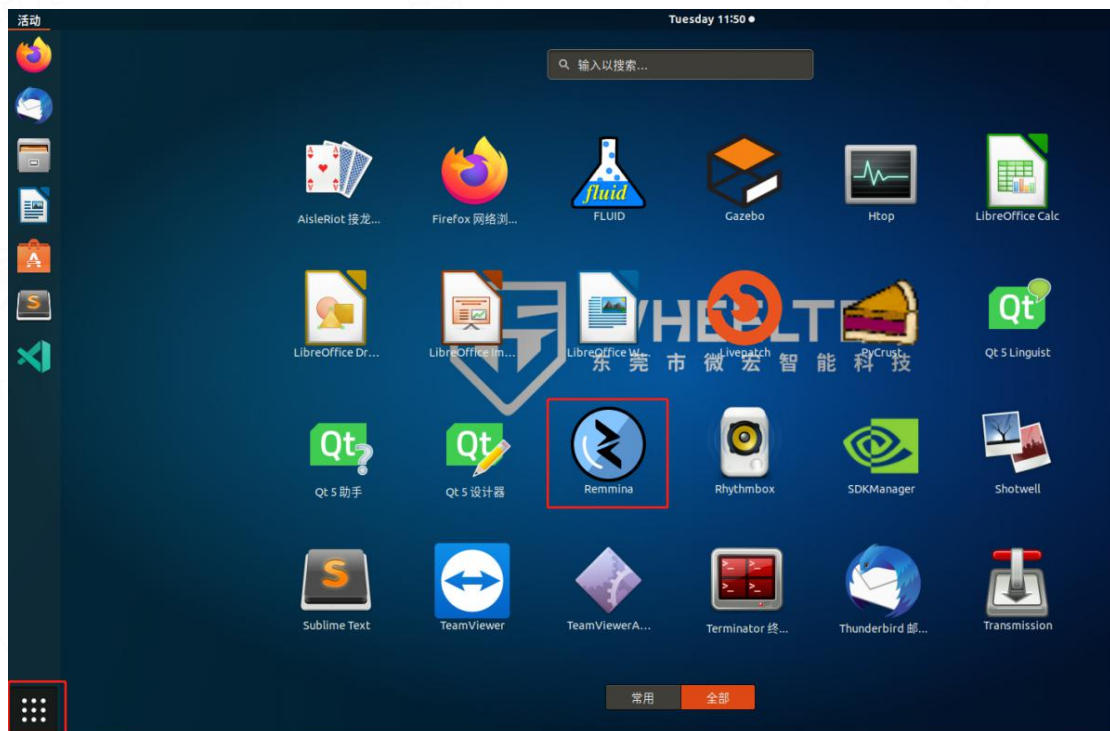
3. 注意事项

① 小车跟随问题

框选被跟随物体时，尽量使被跟随物体与背景物体在视觉上有一定差异，以防止小车在运动过程中因识别到相似物体而导致跟随目标改变，同时被跟随物体的运动也不要过快，过快小车也容易跟丢目标，可能导致小车失控。当 KCF 功能运行在部分主控上如 Jetson nano 时，可能有较大的时延，跟随效果不理想，这时我们可以选择 VNC 远程登录到小车 ROS 主控桌面在本机运行 KCF 功能。

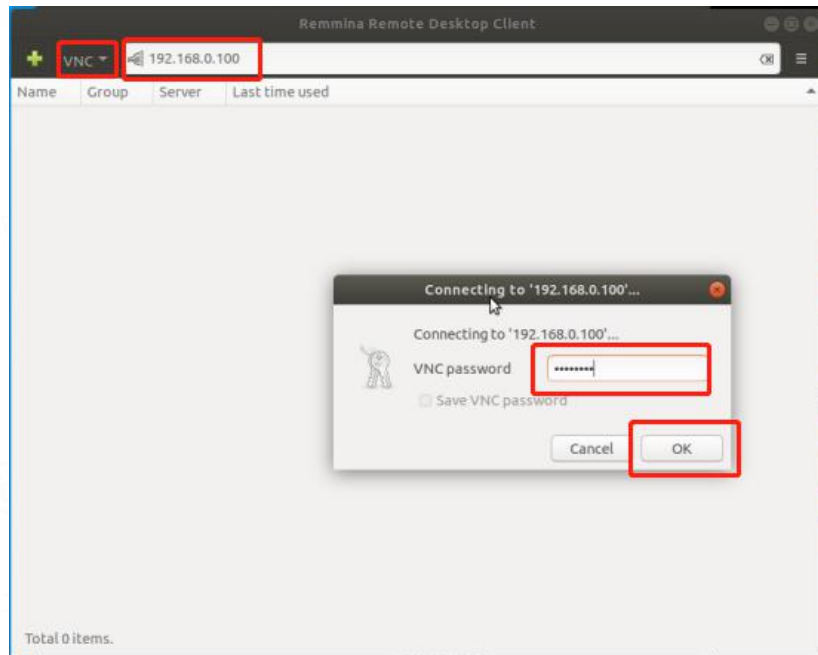
② VNC 远程登录

VNC 远程登录可以实现直接在虚拟机端操控小车主控的桌面，而不用将小车主控连接屏幕。VNC 操作需要先连接小车 WiFi，接着我们运行 ubuntu 系统中的 Remmina 远程软件，点击虚拟机界面左下角打开虚拟机中安装的应用，再点击 Remmina 软件，



使用 Remmina 进行 VNC 操作

打开之后，我们点击左上角选择 VNC，在右边输入小车的 ip 地址，这里输入小车默认 ip 地址 192.168.0.100 为例，等待一段时间后需要输入 vnc 密码，vnc 密码我们默认配置为 dongguan，输入密码后点击 OK 即可进入远程桌面。



进行远程连接

现在我们就可以在远程桌面上运行命令了，因为此时我们已经是小车端运行，所以不需要再进行登录，直接打开终端运行命令行即可。

③ 远程连接的画质问题

远程桌面的画质可能比较差，我们可以选择更清晰一点的画质，点击左边设置，默认画质为 **poor**，我们可以选择 **medium**，注意，画质选择越高，运行时画面的刷新就会更慢，画面显示延迟会更大，所以这里建议不要选择过高画质。

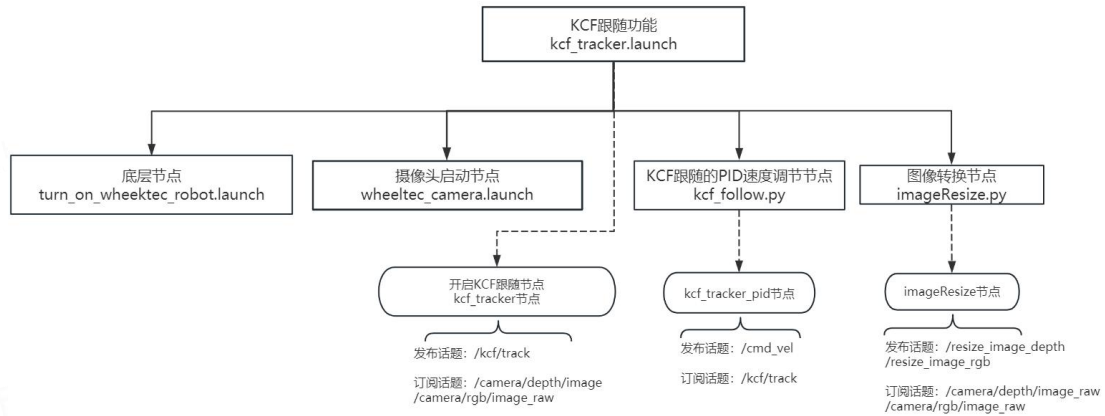


远程桌面显示界面

4. 功能讲解

KCF 跟随功能是通过 `kcf_track.launch` 启动文件的运行来开启的

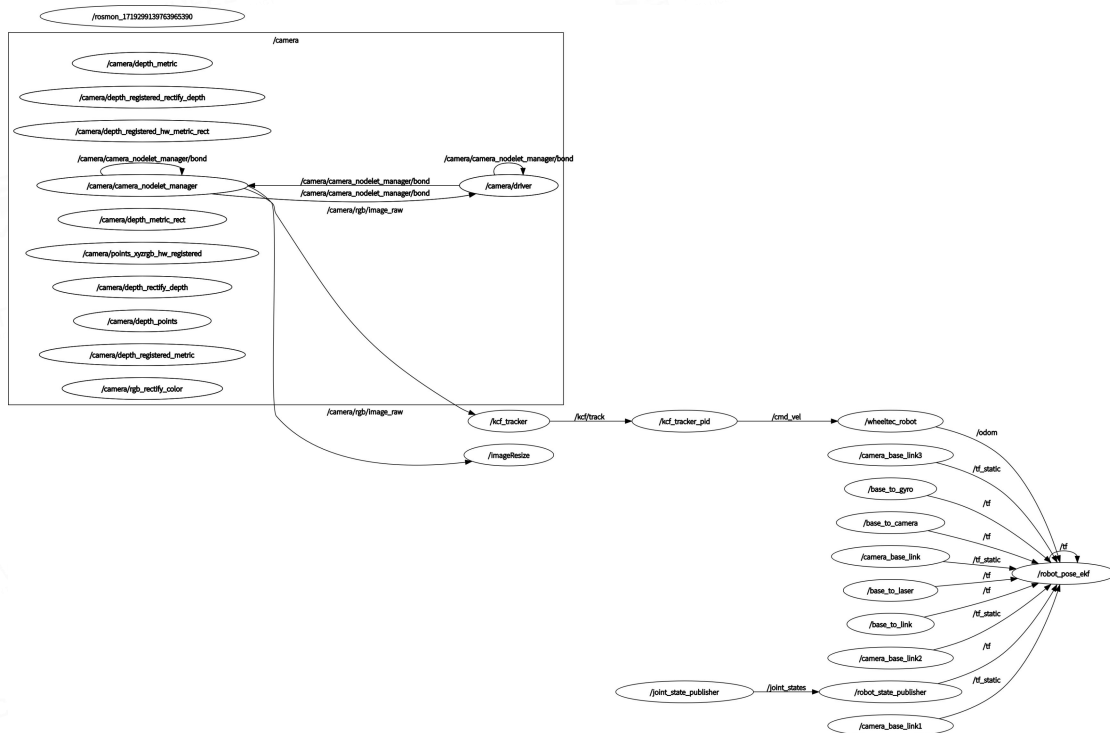
① 启动 KCF 功能



KCF 功能启动框架

② KCF 功能节点关系

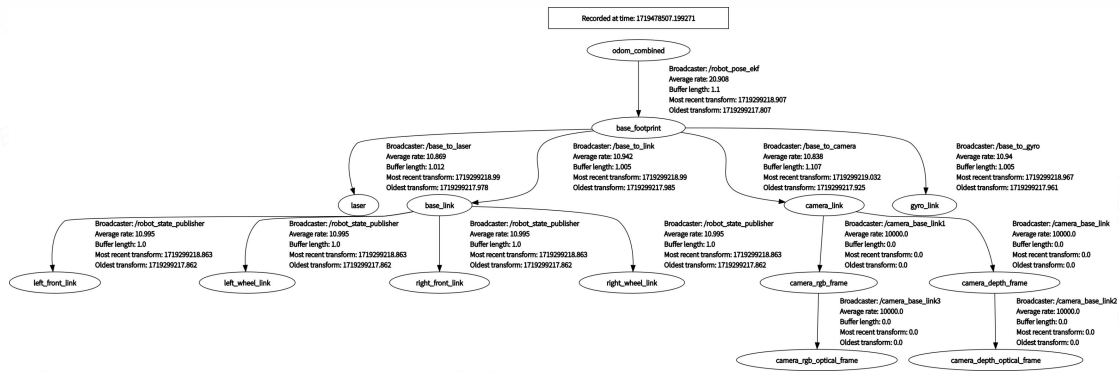
rqt_graph



KCF 节点关系图

③ KCF 功能 TF 树

`roslaunch rqt_tf_tree rqt_tf_tree`



KCF 功能 TF 树

④ KCF 功能启动文件介绍

Launch 文件可以简单的分为四部分，第一部分是开启底层运动节点与摄像头，

```
<include file="$(find turn_on_wheeltec_robot)/launch/turn_on_wheeltec_robot.launch" />
<include file="$(find turn_on_wheeltec_robot)/launch/wheeltec_camera.launch" />
```

第二部分调用了用于进行图像转换的 imageResize.py 文件，将 ROS 图像转换为 OpenCV 图像，以便后面做图像处理，

```
<node name='imageResize' pkg='kcf_track' type='imageResize.py'>
</node>
```

第三部分是 KCF 跟随节点的开启，同时使用了 rgb 与深度话题的信息，

```
<node name='kcf_tracker' pkg='kcf_track' type='kcf_node'>
  <param name='rgb_topic' value='/camera/rgb/image_raw' type='string' />
  <param name='depth_topic' value='/camera/depth/image' type='string' />
</node>
```

最后第四部分是 KCF 跟随的 PID 速度调节节点，对 PID 参数进行赋值。

```
<node name='kcf_tracker_pid' pkg='kcf_track' type='kcf_follow.py'>
  <param name='line_maxSpeed' value='0.3' type='double' />
  <param name='angular_maxSpeed' value='0.4' type='double' />

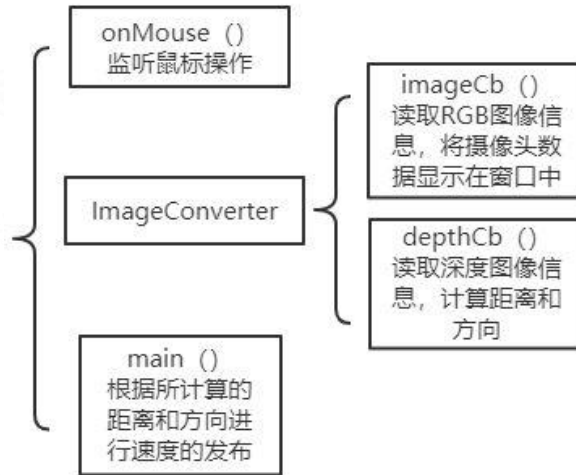
  <param name='dis_Kp' value='0.1' type='double' />
  <param name='dis_Kd' value='0.5' type='double' />
  <param name='dis_setPoint' value='1.2' type='double' />

  <param name='ang_Kp' value='0.002' type='double' />
  <param name='ang_Kd' value='0.001' type='double' />
  <param name='ang_setPoint' value='320' type='int' />
</node>
```

Launch 里面的内容主要都是节点的开启，下面首先讲解一下 KCF 跟随当中

最主要的 kcf_tracker 节点。

kcf_tracker 节点是在 runtracker.cpp 文件中进行初始化定义的，runtracker.cpp 文件的主要内容是进行所选框内图像的处理然后做速度的发布，



runtracker.cpp 文件函数结构与作用

除主函数外，该 cpp 文件中定义了三个函数，函数作用如上图所示。 另外还有一个节点是 kcf_tracker_pid，这个节点主要是小车速度的 PID 调，节点是在 kcf_follow.py 中定义的。