

图像处理与系统建模

任务说明

本任务旨在培养组员在已经学会Opencv的基本语法的情况下，对实际的计算机视觉工程与任务进一步了解；同时在比较理想的仿真条件下完成Robomaster比赛中比较重要的一项视觉任务，为未来进入队伍中进行视觉开发进行一定的学习和准备。

在任务中，组员需要掌握Opencv的基本使用，对C++STL的使用，对数据和信号的一些处理方法，以及基本的程序架构思想与能力(相对次要，但必须有)。

任务目标与任务要求

本内容可见于task/readme.md

需要提醒的是，做法可以八仙过海各显神通，最后考察的是你如何从运动的图形中提取有用有效的数据，以及这个数据的准确性等等。而如何体现你对数据的提取结果可以作为相对附加的项目。

关于代码结构

在Ubuntu的终端中输入tree指令,可以得到以下的目录结构

```
$ tree
.
├── CMakeLists.txt
├── image
│   ├── R.png
│   └── target.png
├── include
│   └── windmill.hpp
├── main.cpp
├── readme.md
└── src
    └── windmill.cpp
```

在我们目前的项目工程中，在编写C++时，一般使用include和src的结构，即include目录中放置头文件，src目录中放置同名的源文件。同时，我们在hpp与cpp文件中的编码有一定的讲究，感兴趣的同学可以通过观察，询问，查阅资料的方法来适应这种风格，同时形成更好的编码习惯。

提交内容

整体的 task01项目需提交至 Github，其中应包括，符合 项目结构 的内容，即至少包括以下内容：

1. 源代码
2. 结果展示
 - 使用合适的方法展示你的结果，可以是视频或者曲线图等图表(推荐)，也可以是纯粹的数据与表格(不直观，不推荐)。
 - 程序运行时终端的输出截图。
3. README.md

评价标准

- 功能实现：最起码实现三个档次中的一个。
- 提交规范：所有要求的提交内容均已齐全，并按照指定方式提交。
- 文档完整性：提交的报告内容完整，能够清晰描述项目的实现过程和结果。
- 代码质量：代码结构合理，注释清晰，变量命名规范。

参考资料

- [OpenCV 官方文档](#)
- [C++ 官方教程](#)