

ZJUDancer软件部分题目

注意使用C++，并使用Git管理代码，建议使用Ubuntu系统。

关于Git的学习，可以参考：

- 廖雪峰的[中文教程](#)
- Github的[互动式课程](#)
- Atlassian也有一个[简明的Git教程](#)
- 如果需要深入了解Git是如何进行版本管理的，可以参考[A Hacker's Guide to Git](#)

【1】固定白平衡（提供[空项目](#)）

使用相机的自动白平衡会导致低头看草地和抬头看远处（图像中出现白色背景）时参数差距过大，而相机驱动接口的手动白平衡仅开放一个接口参数，无法对RGB的gain进行分别调整，效果极差。



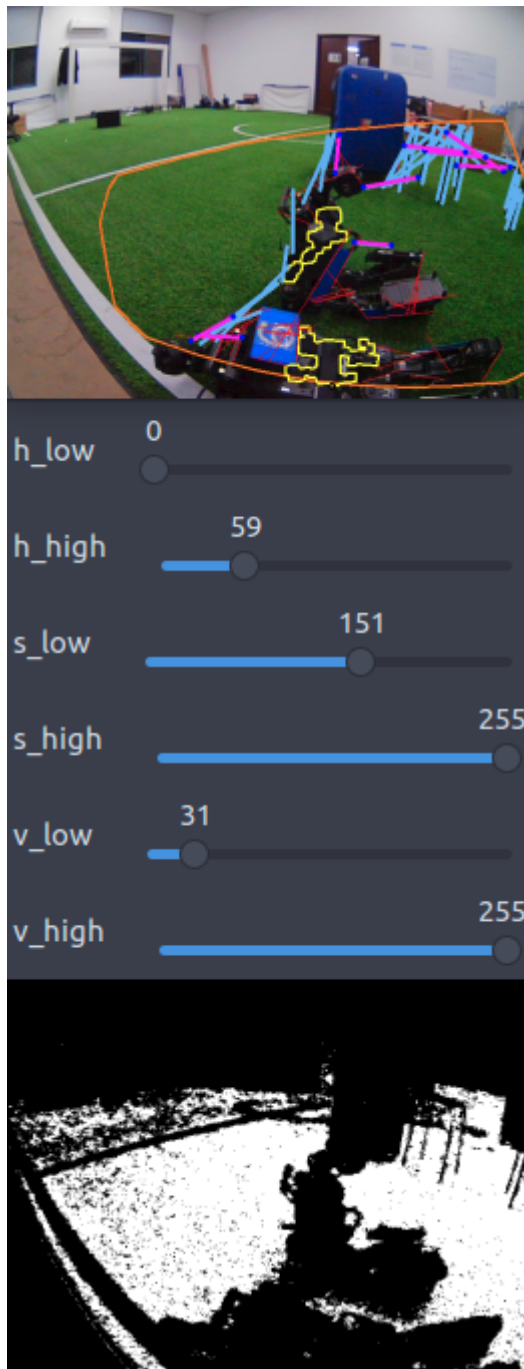


如上图，分别为抬头与低头时的图像，低头状态下自动白平衡调整导致整体图像颜色偏红。

要求：需通过底层接口将相机白平衡固定在一个较好的参数，或者在现有驱动的接口下，固定一个白平衡数值，并通过标定的白平衡参数对每帧图像手动进行白平衡处理。

如使用空项目请将项目仓库fork到自己帐号下。测试用视频发布于空项目的release中。

【2】自适应hsv阈值（或者选框，提供[空项目](#)）



目前的检测需设定hsv通道下的阈值，在比赛（换场地、光线变化等）调试时十分繁杂。

需要根据场地使用自适应的方法确定阈值（给定图像确定阈值），或者在GUI的图像中框选部分图像确定阈值，简化参数调整。

如使用空项目请将项目仓库fork到自己帐号下。测试用图片、视频发布于空项目的release中。

【3】白线检测的并行加速

使用cv::cuda、openmp或tbb的并行方法进行加速。详见补充材料，请联系队员获取。

【4】Vision Compass

事先对环境构建全景地图，通过特征匹配等方式，将当前帧与地图进行匹配，获得当前机器人yaw角。

【5】罚球点检测

用yolo等目标检测方法对罚球点初步检测，并用白线、位置等对检测结果进行限制，获得罚球点位置，辅助定位。

题【1】、【2】的空项目仓库将于2019.7.21前公开。

针对以上题目描述，如有疑问请联系邮箱zjudancer+joinus2019@gmail.com。