SJTU JiaoLong RM2018 Armor Detection

环境要求

- Ubuntu 14.04 或更高
- CMake
- OpenCV（推荐OpenCV3以上版本）
- libMVSDK.so（全局曝光相机驱动）

状态机

- Fast Explore（默认状态）：在机器人快速移动的时候，寻找目标，有一定概率误识别
- Fast Track：追踪在Fast Explore中找到的可能装甲板
- Slow Explore：机器人慢下后，重新寻找目标，此时画面稳定，不容易误识别
- Slow Track：追踪在Slow Explore中找到的可能装甲板

寻找算法

- 筛选出高亮区域（对于黑白摄像头）或红/蓝区域（对于彩色摄像头）
- 使用 `findContour` 寻找轮廓
- 寻找长条形的轮廓，可能是两侧的灯条
- 配对灯条（根据长度、角度等）

追踪算法

- KCFTracker：传统方法的追踪器，可以达到速度（60fps）和性能的平衡

加速算法

- OpenMP：并行计算，在处理当前图像的同时，获取下一张图像

可改进

- 使用机器学习配对灯条
- 更进一步，直接使用目标检测查找装甲板
- 识别中间贴纸的数字
- 同时使用两个摄像头

PS

- 我们使用了MindVision的全局曝光相机，驱动是专有的。在 `include/GlobalCamera.h` 里对驱动进行了包装。
- 为了提高速度，没有使用OpenCV自带的KCFTracker，另外找了一个CPP版本，被包装在 `src/KCFTracker.cpp` 里。
- 为了便于区分自己电脑的开发环境和妙算上的部署环境，在 `precom.h` 里设置了宏，通过是否ARM架构来区分。
- `precom.h` 里面还有一些宏来控制OpenMP，显示中间图像和录像。其中OpenMP和显示中间图像是互斥的。
- 为了优化彩色相机的速度，直接处理了原始的拜耳阵列，可以通过宏 `BAYER_HACKING` 控制。

Requirements

- Ubuntu 14.04 or higher
- CMake
- OpenCV (version 3 recommended)
- libMVSDK.so (Global Shutter Camera Driver)

State Machine

- Fast Explore(default state): Explore the armor when the robot is moving fast
- Fast Track: Track the armor detected in Fast Explore
- Slow Explore: After the robot slows down, explore the armor again
- Slow Track: Track the armor detected in Slow Explore

Explore Algorithm

- Find the light region(for Gray Camera) or the blue/red region(for Color Camera)
- Use `findContour` to find the contours
- Find the thin and long contours, which may be the side light of armor
- Pair the lights to find the armor(length, angle and so on)

Track Algorithm

- KCFTracker: the balance of speed(60 fps) and accuracy.

Speedup Algorithm

- OpenMP: Process the image and fetch next image at the same time.

Ways to improve

- Use Machine Learning to pair the lights
- Use Object Detection to explore the armor
- Recognize the digit in the center
- Use two camera at the same time

PS

- We use a special camera with **Global Shutter**, so the driver is special. It is wrapped in `include/GlobalCamera.h`.
- We find a kcftracker of cpp version instead of the one in the OpenCV, because it runs much faster. It is wrapped in `src/kcftracker.cpp`.
- To distinguish with the PC platform and MiniPC platform, there is a macro in the `precom.h`. If the CPU is ARM architecture, then it is regarded as the MiniPC platform.
- There are some macros in the `precom.h` to switch the **OpenMP**, **Show Image** and **Record Videos**. And when the OpenMP is on, there is no way to show the image.
- To improve the performance of RGB camera, we process the raw data with BAYER format. To learning more, search the Bayer. It is opened by macro `BAYER_HACKING`.