

Robomaster 2018 DJTU 步兵视觉开源

声明

本次开源的程序为 Robomaster 2018 赛季中，大连交通大学 TOE 战队步兵机器人的视觉识别程序。开源的目的是将我们的方案提供给 robomaster 各参赛队作为参考，促进大家的技术进步。

本次开源的程序有三个程序，分别为：

InfantryVisionFinal：能量机关打击+装甲识别程序

Stereo_Calib：双目视觉系统标定程序

Vision_watchdog_proc：看门狗程序

这三个程序的开发环境均为：

系统：ubuntu14.04

开发平台：Intel nuc7i7bhn

IDE：Qt 5.9.1

框架：OpenCV 3.3.0 Caffe1

注意：摄像头选型为三个相同型号的无畸变相机，或进行矫正过的有畸变相机。双目视觉系统由一块碳纤维板（保证两个相机在同一平面上）及两个放置在同一高度，与碳纤维版的中心镜像对称的两个相机。

这篇文档给出了各程序的开发思路及使用说明。如有疑问，欢迎大家交流。

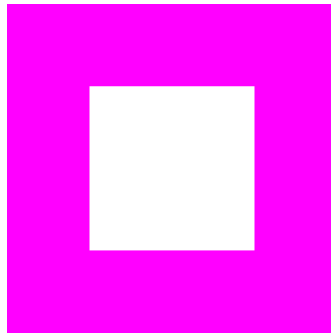
程序作者：丁沛然
联系方式：QQ:438253351

Stereo_Calib

本程序不是 opencv 自带的棋盘标定程序

使用方法：

制作一个 10cm x 10cm 大小的标定块，颜色及样式如下：



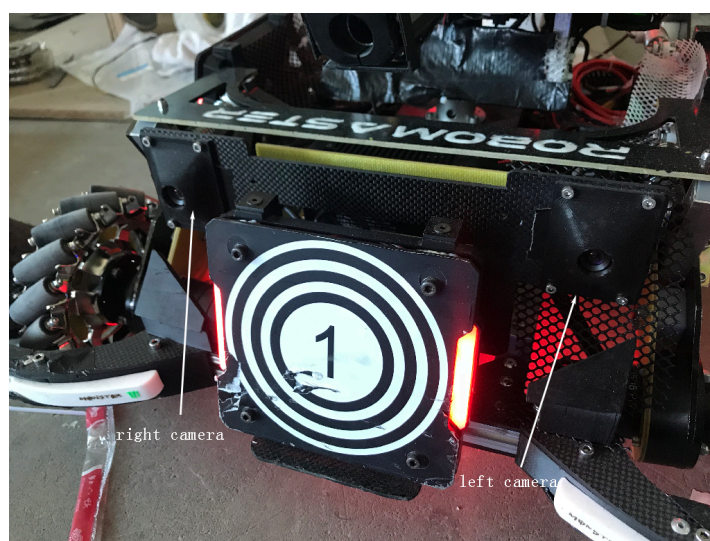
外围颜色 rgb 值为(255,0,255)，内部正方形为(255,255,255)。

该程序可以自动识别出该标志物的中心，假设双目中所同时看到的该物体中心为匹配点，通过简单的双目视觉测量原理，反推出系统所需要的内参。双目视觉原理本文档不做说明。

标定过程：

将双目视觉系统垂直于地面摆放，在地面上垂直于双目视觉板的平面摆放一把钢尺，将该标定块摆放在钢尺上方，垂直于地面。保证标定块同时在双目相机的视野中，通过程序提示来移动标定块的位置，程序将自动计算出标定的内参，并以名为 stereo_calib.xml 的文件保存在对应路径下。

双目视觉在机器人上搭载的方式：



InfantryVisionFinal

想要运行该程序，至少在 pc 端接入两个摄像头及一个 usb 转 ttl 模块，最多安装三个摄像头(单目+双目)。在调试模式下，只接入一个 usb 转 ttl 模块即可运行。

整体架构

文件名	类	作用
armorfind.cpp	ArmorFind	识别装甲
armorpredict.cpp	ArmorPredict	解算自动打击云台角度
RMVideoCapture.cpp	RMVideoCapture	夏令营开源的摄像头调用类
StereoXML.cpp	StereoXML	双目视觉匹配估计三维坐标
CRC_Check.cpp	-	官方的 crc 校验函数
serial.cpp	-	串口通信相关代码
caffe_prototype.cpp	Classifier	调用 caffe 训练得到的模型做分类
HandWritingSolver.cpp	NumberPredict	识别大小能量机关击打区域数字
ImageProcess.cpp	SortRects SortSudokus ImageProcess	对图片进行预处理，得到大小能量机关的击打区域及 led 数码管区域的 roi 并进行排序
LedNumberSolver.cpp	LedNumberSolver	识别 led 数码管区域的数字
LogicProcess.cpp	LogicProcess	判断当前应该击打的能量机关位置
InitParam.cpp	-	初始化参数
TString.cpp	TString	一个自定义的字符串类
Header.h	-	公用头文件
WatchDog.cpp	WatchDog	看门狗喂狗类
sudokuposcal.cpp	SudokuPosCal	计算能量机关击打点位置

识别原理：

小能量机关：

1. 自适应二值
2. 查找轮廓
3. 用最小矩形包围轮廓，根据矩形的长宽比进行初筛
4. 根据轮廓面积进行第二次筛选，判定条件为 $area[i] > area_max / 3.5$ ， $area[i]$ 为第 i 个轮廓的面积， $area_max$ 为该组轮廓中的最大面积。将结果保存在一组 `vector` 中。
5. 根据最上方的三个九宫格位置，得出 led 数码管所在的 roi。
6. 对 led 的 roi 进行二值化，寻找轮廓，通过长宽比确定数码管数字区域，保存在一组 `vector` 中。
7. 使用 `caffe` 处理手写体数字，使用 `hog+svm` 处理 led 数字。
8. 对结果进行判断，匹配左右目标区域中心点，输出目标点的三维坐标，转化为 yaw，pitch 角度输出。

小能量机关需要调整的参数有两个：

`sudoku_exp`:该模式下的相机曝光。

sudoku_led_thres: led 数码管区域二值化过程的预值。

大能量机关：

1. 二值化
2. 寻找记分板，若寻找到十个记分板，则根据记分板区域选出击打数字所在的 roi。若未寻找到，则对原图进行一系列处理，得到一张较理想的二值图，在该图中直接寻找数字区域。该代码在 ImageProcess::ThresholdProcess 函数中
3. 利用最小矩形选出九个数字的 roi，若 roi 的区域多或少，则利用相互的位置关系将多余的筛掉，将缺失的预测出来，保证最终结果是 9 个。
4. 利用最上面三个数字的位置选出 led 数码管所在的 roi。
5. 对 led 的 roi 进行二值化，寻找轮廓，通过长宽比确定数码管数字区域，保存在一组 vector 中。
6. 使用 caffe 处理火焰数字，使用 hog+svm 处理 led 数字。
7. 对结果进行判断，匹配左右目标区域中心点，输出目标点的三维坐标，转化为 yaw, pitch 角度输出。

大能量机关需要调整的参数有三个：

sudokunew_exp:该模式下的相机曝光。

sudokunew_led_thres: led 数码管区域二值化过程的预值。

sudokunew_board_thres:全图二值化的预值。

装甲板识别：

1. 二值化
2. 寻找轮廓，通过轮廓的长宽，角度等特征，筛出灯条，进行排序。
3. 确认出装甲板中心位置。
4. 对每个识别到的装甲中心进行匹配，通过距离进行排序，输出距离最近的装甲板三维坐标。
5. 若双目系统识别到装甲板，则输出。若没有识别到，则对云台上的单目相机获取到的图像进行处理，

装甲板识别需要调整的参数有四个：

armor_thres_red:识别红色装甲二值化的预值

armor_thres_blue:识别蓝色装甲二值化的预值

armor_exp_red:识别红色装甲相机的曝光

armor_exp_blue:识别蓝色装甲相机的曝光

通信协议：

通信使用的硬件是 usb 转 ttl 模块，相关函数在 Serial/serial.cpp 中。

stm32->pc:

byte0	byte1	byte2	byte3	byte4	byte5
0xA5	mode	robotlevel+50	CRC8	'\r'	'\n'

mode:

0:击打红色装甲

1:击打蓝色装甲

50:打小能量机关

51:打大能量机关

52:补给站补弹视觉（已弃用）

53:单目识别红色装甲

54:单目识别蓝色装甲

单目模式在机器人左右摇摆时使用

pc->stm32:

识别装甲模式:

byte0	byte1	byte2	byte3	byte4	byte5	byte6	byte7		
0xA5	mode	pitch_angle		yaw_angle		yaw_speed			
byte8	byte9	byte10	byte11	byte12	byte13	byte14	byte15	byte16	byte17
shoot_speed				distance_z				CRC16	

pitch_angle: pitch 轴相对中间位置的角度偏移量

yaw_angle:: yaw 轴相对中间位置的角度偏移量

shoot_speed:射速

distance_z:目标与机器人云台距离

识别能量机关模式:

byte0	byte1	byte2	byte3	byte4	byte5	byte6	byte7		
0xC6	mode	pitch_angle		yaw_angle		-	-		
byte8	byte9	byte10	byte11	byte12	byte13	byte14	byte15	byte16	byte17
which	pos	-	-	distance_z				CRC16	

which: 当前击打第几个数字

pos: 当前击打数字在第几个格子

参数配置文件:

该文件为 ExtraFiles/Params.xml, 参数如下:

x/y/z: 双目左相机位置与云台 yaw, pitch 轴交点处在 x/y/z 方向的偏移量

debug_flag:debug 模式, 1 为开启, 0 为关闭

debug_mode:debug 模式下的测试内容, 与通信协议中的 mode 规定一致

writemov:录像功能, 1 开启, 0 关闭。影像为双目左相机的图像

usevideo:debug 模式下若使用视频调试则置 1, 否则置 0

video_path:视频调试模式的视频路径

mono*_armor*_const:装甲距离计算常数 C, 原理为 $d=\sqrt{C/\text{armor_area}}$

mono_f*:单目焦距

mono_base_*:单目光轴位置

mono_x/y/z:单目相机位置与云台 yaw, pitch 轴交点处在 x/y/z 方向的偏移量