

视觉组调参指南

机械装上相机之前

认真检查相机焦距和光圈是否已经固定住，通常将光圈调整到最大，然后将焦距调整到能清晰看到2-6m左右距离的装甲板数字(使用GalaxyView)。(超过这个范围，就算识别到，PNP解算误差也很大，而且操作手也不会打蛋，所以不用过多纠结识别距离)。

通常上位机与下位机的接口包含串口和相机，但是由于上位机(TX2或者NUC)的USB接口在多次插拔之后可能会导致插不紧，此时一定要给连接的USB口打上胶。不能认为平时调试时没有出现过串口断开或者相机断开的情况就认为不存在USB口松动这个问题，即便是打过胶，也应当在强烈碰撞的条件下测试(例如两车碰撞，飞坡等)。

注意看相机前的保护镜片是否刮花导致相机成像出问题(这个问题很容易忽略!!)

相机外参整定

如果你熟知坐标系变换过程，那么你应该知道，唯一需要填补的参数，就是相机坐标系到枪管坐标系的平移向量(还有一个旋转向量下面再测)。一般上在设计上相机坐标系到枪管坐标系仅仅差了一个平移，这里就需要对两个坐标系的中心点位置作分析。相机坐标系不用说中心点一定是在镜头中心位置，而枪管坐标系的中心点就需要分析了。因为枪管坐标系进行旋转就到了基坐标系，也就是最终的坐标系。我们在最终的坐标系下进行了抛物线的计算，那么枪管坐标系的中心点应恰好不受到枪管的支撑力，即在枪管出口位置。通常将x轴平移量设置为0，y轴向前，z轴向天空。y,z轴的值应当让机械在3维图纸上测，至于值的正负，你可以改变一下符号，看最终的坐标输出自行判断。

相机内参整定

推荐使用大恒自带的SDK，里面可以找到一个叫GalaxyView的可执行文件，打开这个可执行文件，可以很方便地对相机的各种参数进行调节，同时可以很方便地对标定板拍照，作为标定的图片使用。可以使用Matlab工具箱作为标定工具。至于是否需要标定出 p_1, p_2, k_3 ,至今还无定论，有兴趣的可以研究一下。

相机标定通常有很多技巧，不能随便标，否则后面PNP计算误差就会更大。一个基本原则是标定板在图片中需要占据一半以上(即使标定格清晰一点),每次拍图片时，可以动云台，也可以直接动标定板，拍摄多个角度的标定板，标定出来的参数才有普适性。另外，图片的亮度，白平衡等也会对标定结果产生影响，故标定前需要确保选择自动白平衡模式，图片亮度要至少能清晰看见方格的角点。

GalaxyView内可以看到相机的SN号，如果程序需要的话可以在GalaxyView中查看并且在程序的json中修改。

识别调参

相机识别参数整定

相机识别参数主要需要调整的有

- 曝光

曝光是相机识别参数中最为重要的一个参数，曝光调整的优劣将会直接影响你识别程序效果的好坏。调节曝光的基本原则是

1. 不能调节得太低，太低得曝光会导致灯条变得很暗，二值化会出问题不说，还会截掉一部分灯条，导致PNP计算不够准确。其次，越远的物体在相机中看到的亮度就越低，如果曝光调节得过低，识别距离就会远远减短。如果识别距离小于4m的话，那么实战中就很容易断识别。
2. 不能调节得太高，如果调节得太高的话，由于形态学对噪点的处理总是有限的。如果曝光调的很高的话，并且分类器不够给力的话，就很有可能产生误识别现象。
3. 总结一句，就是曝光应当在你程序支持不误识别的前提下尽可能往高调。

- 增益

增益通常是在曝光满足不了要求的情况下才调节，通常不应该调得很大，所以调节频率也比较低，至于具体效果怎样你可以具体实践体会一下。

至于相机得其它参数，一般影响不大，可以不用管。值得注意的是相机中内置自动曝光，自动增益这些功能，这些算法(也包括你能查到的大部分自动曝光算法)通常是以图像整体观感作为前提实现的，而我们感兴趣的区域只有装甲板那一小部分，故这些功能都是价值不大的，不太建议去探索。

你可能不太想使用自动白平衡，而非要在赛场上用一张白纸来进行白平衡标定，虽然可能是一个可行的办法，但我觉得自动白平衡效果已经可以了，没什么必要将调参变得这么麻烦。

识别算法参数整定

首先聊一下识别算法，自从2021年上交四点模型横空出世，很多人认为深度学习是RM装甲板识别的最优解。但是，今年哈工程使用传统视觉识别也展现出了惊人的鲁棒性，这值得令我们深思，是传统视觉不行，还是我们自己本身对传统视觉的理解并不够。深度学习虽然能避免曝光的影响，但是也会使模型不透明，导致出现一些莫名其妙的误识别(比如视觉群大佬们讨论的在前哨站附近的装甲板全部识别为前哨站)。所以我认为，我们既要开发深度学习方向的装甲板识别，也要将这个从RM诞生至此的传统算法彻底吃透。

图像预处理

图像预处理算法通常可以分为以下四种类型

1. 使用灰度图进行二值化，最后对轮廓内的像素判断颜色。
2. 使用敌方颜色的通道减去己方颜色通道(或者绿通道)。
3. 以上两者混合，即使用 (灰度图 & 通道相减图) 然后再二值化。
4. 使用HSV进行通道二值化分割。

最后预处理的方案选择了第一种，理由是使用灰度图能够避免灯条断开的问题(如果在较高曝光或者较高增益的情况下，灯条中心像素值较高，使灯条中心发白，如果使用通道相减或者HSV方法，可能会导致二值化后的图形断开)，虽然仅仅使用灰度图使后面对每个轮廓的颜色校验增加了一定的计算量，而且也会导致筛选出来的灯条变多。

灯条 & 装甲板 特征筛选

为了解决预处理带来的弊端，需要对灯条的几何条件进行严格的约束。这里我感觉在上一个赛季并没有做得很好，因为在上一个赛季中，对灯条以及装甲板特征的约束，都是给一个比较大的约束范围，但是，这些较大的约束范围而不是合理的约束范围会使几何特征较差的"灯条"以及几何特征较差的"装甲板"混入正确的灯条或装甲板中，如果将这些错误的装甲板都丢进分类器内的话，分类器又做不到完全的识别准确，必然会导致最终识别的不稳定。因此，我觉得，这些几何约束参数应该认真地调整一下!!!调整的方法也很简单，先给一个比较严格的参数约束，逐步移动目标位置(远近)，或者改变装甲板类型(大小装甲板),如果识别不到就根据输出数值再稍微扩大一点(至于这个度应该在调参过程自己感悟)。

数字分类器

目前使用的分类器是 HOG + SVM 分类器，该分类器的优点是对光照变化有一定的鲁棒性。

预处理参数

目前将装甲板数字裁剪出来并且经过仿射变换之后，由于图像在低曝光的前提之下较暗，数字可能会不太明显，故需要对图像进行一次gamma矫正，使图像整体变亮(当然也可以考虑直方图均衡化等预处理方法)。gamma值越低，数字越亮，但噪点也会越多,故gamma值需要调节到一个较为合适的值，至少肉眼能够清晰分辨数字。

训练参数

主要参数包含Hog参数以及SVM参数，此部分由于我自己都是属于玄学调参阶段，对这些参数都没有很好把握，因此就不在此班门弄斧了，详细的调参方法自行查询Hog+SVM算法原理。

其它

要注意一点，SVM算法本身是不适合大量样本的，因此如果过多样本反而会导致模型过拟合而表现不好。因此，应尽量控制每一个环境的样本数量(样本多样性)，不应太多(这个赛季对这个算法并不了解，只会堆数据集，导致算法表现比较不理想)。

卡尔曼滤波

算法调参

卡尔曼滤波属于是上个赛季研究最多的算法了，因此也是完成度最高的算法。

The specific equations for the time and measurement updates are presented below in table 4.1 and table 4.2.

Table 4.1: Discrete Kalman filter time update equations.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{4.9}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{4.10}$$

Again notice how the time update equations in table 4.1 project the state and covariance estimates forward from time step $k - 1$ to step k . A and B are from equation (4.1), while Q is from equation (4.3). Initial conditions for the filter are discussed in the earlier references.

Table 4.2: Discrete Kalman filter measurement update equations.

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{4.11}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{4.12}$$

$$P_k = (I - K_k H)P_k^- \tag{4.13}$$
 @涅索斯衬衫

对于卡尔曼滤波，我们要先明白各个参数的含义， x_k 为状态估计量，也即你需要通过卡尔曼滤波得到的参数，在我们的卡尔曼滤波中，位置和速度是我们需要的，由于需要在三维空间中进行状态估计，故 x_k 有六维。

z_k 为测量值，测量值说白了就是传感器测量得到的值，我们使用的视觉传感器能得到图像的像素点位置，理论上应该将像素点作为测量值。假设像素点的离散程度可以用高斯分布表示，那么对像素点进行线性变换之后，所得到的三维点坐标噪声散布也应当符合正态分布(当然这里忽略了畸变，否则模型建立十分麻烦)。但为什么对三维点坐标作非线性变换，最后用yaw,pitch,distance作为测量量呢？那是因为，枪管正对的角度不一定是陀螺仪的0度(你可以想象一下敌方装甲板在你枪管的正前方，但你计算得到的y值为负),计算得到的x,y值方差实际上是会跟随枪管正前对着的陀螺仪角度变化的(想想为什么)。并且，直接拿yaw,pitch当测量值，最后也要把状态估计量 \hat{x}_k 转换为yaw,pitch再发送给下位机，这个模型把你最后转换为yaw和pitch过程的估计误差也考虑在内了，理论上模型更加完整合理。

我们需要调节的卡尔曼滤波矩阵理论上有三个，以下一一分析

1. P 矩阵

P矩阵是状态估计协方差，其初值会影响迭代的收敛速度，但一般设定为单位阵让它自己收敛即可。

2. R 矩阵

R 矩阵在该模型中为测量噪声矩阵，大小为3x3,一般只需要设置其对角元素，即认为测量量之前无相关性。在我们的模型之中，测量量分别为yaw,pitch,distance,根据R矩阵的定义,对角元素的值应当为你测量量的方差，也就是说，这个值可以有两个来源，一是通过传感器说明书直接获得，但这并不适合我们的模型；二是利用数学上统计的方法计算这三个值的测量误差。通常不会将R矩阵的对角线元素设为方差 σ^2 ,而是设置为 $(3\sigma)^2$ (这是我从网络上看到的一种说法，可以理解为几乎不可能有数在 $(3\sigma)^2$ 之外)。

于是，一种我认为比较合适的做法是，将敌方装甲板不动，自己动云台，保存相关的原始坐标值，然后利用Matlab或者Python计算出方差，最后填入该矩阵中。

值得注意的是,由于相机的小孔成像模型，如果误差一个像素的话，敌方装甲板在离自己不同距离所计算得到的distance方差是完全不同的。因此，一种显然的想法是根据distance的计算值动态改变distance的方差。目前的做法是测出不同距离的distance方差，然后用一条直线拟合出来，每次都不断计算更新distance的方差。这种方法虽然并没有合理的理论支撑，但我认为至少比设定一个固定的值要好上不少。之于yaw和pitch的方差与距离无关，故设置固定值即可。

3. Q 矩阵

Q矩阵是一个6x6的矩阵，那么应当有六个值需要调，但是，如果你仅仅设置Q矩阵的对角阵的话，有一个很致命的问题就是，你默认了一个轴的速度与那个轴的位置值无关，这个命题显然是不成立的。那么你应该能想到速度值和位置值的过程噪声必然存在一些关联。这些关联你可以通过搜索"CV运动模型"来了解相关内容。在你得到同一个轴之间的关联之后，需要调节的参数就只剩下三个了，分别是x,y,z轴的过程噪声。如果你仔细分析这个模型，你会发现x,y轴应该是等效的(因为存在陀螺仪绕z轴的旋转)，那么x,y轴的过程噪声也应当设置为一致的，即此时，你需要调节的仅仅只有两个参数。

4. 调参流程

在你将需要统计计算得到的结果统统得到以后，便可以开始进行调参，通常需要调节的只有两个过程噪声参数而已。调参过程可以分为两个部分

1. 滤波部分:即需要将你得到的位置曲线与滤波得到的位置曲线基本重合(这里的重合不是指完全重合，因为原位置曲线附带噪声,滤波得到的位置曲线应当穿过原位置曲线的平均值)。
2. 预测部分:在你得到一条拟合度比较好的位置曲线之后，由于你仍然未对你速度的准确性进行评估，于是，一个很好的方法是，将你预测一段时间(比如0.3s)后的x,y,z,yaw,pitch信息与实际时间线上x,y,z,yaw,pitch信息作对比，很显然，如果这两种对比图像都画在同一幅图上，两条曲线重合的地方，便是卡尔曼滤波计算正确的地方。通过这个方法，微调过程噪声矩阵，使敌方机器人变速之后能以最快速度收敛到两条曲线重合(这在图像中应该是相当明显的)。调节参数到最快收敛(一般有一个极限值，过程噪声大于该极限值之后收敛速度不再明显加快)

卡方检验

预测时间

其它状态估计

位姿估计中丢失的旋转调节

使用ROI加速

优先级判别

反陀螺算法初步

大符算法

优先级判断

算法存在问题总结

未来发展方向