



程序框架

一、红色检测（约 5ms）

- 1、颜色滤波：为了得到目标投掷模块的红色区域
- 2、开运算：仅仅通过颜色滤波容易造成轮廓断点，合理地选择形态学滤波器能够对图像作一定程度上的补偿。
- 3、取轮廓：将连通域的外围勾画出来，便于形成外接矩形。
- 4、面积判断：对轮廓的最小外接矩形进行面积判断。
- 5、去除中心干扰：由于灯光颜色分布在格子的中间区域不均匀，导致颜色滤波后格子中间也会出现无检测轮廓，故此步消除干扰，以便锁定红色九宫格。
- 6、取中心：对九宫格最小外接矩形求中心。
- 7、判断两次中心距离，滤除跳变，锁定 10 次。
- 8、返回锁定红色九宫格的中心位置。

二、取球框检测

- 1、颜色滤波：颜色滤波比边缘检测靠谱，因为定位下降的时候摄像头太低，不一定能完整检测出取球框。
- 2、开运算：选择形态学滤波器能够对图像作一定程度上的补偿。
- 3、取轮廓：对轮廓的最小外接矩形进行面积判断
- 4、面积判断：最大面积比第二大面积，若大于两倍则判定为取球框
- 5、取质心
- 6、返回锁定去球框的中心位置。

三、蓝色检测

- 1、颜色滤波：颜色滤波比边缘检测靠谱，因为定位下降的时候摄像头太低，不一定能完整检测出取球框。
- 2、开运算：去除部分噪声
- 3、取轮廓
- 4、面积判断：判断轮廓的最小外接矩形面积。
- 5、取质心
- 6、判断 $width/4$ 、 $height/4$ 、 $3*width/4$ 、 $3*height/4$ 为阈值判断蓝色格子个数，继而判断方向。

四、二维码定位方案

虽然对角线上的二维码是通过旋转得到的,但若把相邻的四个二维码组合起来,从下图可以看出,从四个不同方向得到的图像是互不相同的.因此本方案的二维码定位识别是通过把四个二维码组合成一个大二维码,进而识别判别方向实现.具体技术方案如下:



1. 颜色滤波: 由于地面除了二维码区域为黑色或白色外,其他区域均为彩色,故颜色滤波得到的结果理论上是理想的.具体的滤波器可选择 HSV 颜色空间或 RGB 颜色空间.实验结果表明,在 HSV 空间下,效果更好,为了更好地克服亮度对颜色的影响,可选择采用自适应滤波.

2. 形态学滤波:仅仅通过颜色滤波容易造成轮廓断点,合理地选择形态学滤波器能够对图像作一定程度上的补偿.通过实验验证,闭运算(先膨胀再腐蚀)能够得到优秀的效果.

3. 取轮廓:将连通域的外围勾画出来,便于形成外接矩形.

4. 获取最小外接矩形.

4. 尺寸判断:为了排除干扰矩形的影响,可利用矩形的面积和纵横比排除大多数不符合要求的噪声矩形.

5. 大二维码的获取:此时图像中只剩下了小二维码的矩形轮廓,此步所做的工作可以理解为把四个 Blob 融合为一个大 Blob.由于四个 Blob 之间存在一定的间隔,因此融合时应注意横向 gap 和纵向 gap 的选取,同时还应注意如何通过四个小 Blob 的数据获取大 Blob 的最小外接矩形.这对大二维码的识别很关键.

6. ROI 图像提取.提取二维码图像的同时,还需记录二维码的偏转角度.

7. 最近邻插值重构图像.由于二维码已经固定,故本方案并不打算采用二维码识别库做处理,而是通过建立哈希值数据库识别待测二维码的信息.重构图像大小是为了能够快速高效地提取二维码的哈希值.

8. 转换灰度图.目的是为了减少计算量.

9. 计算 DCT: 得到 DCT 系数矩阵.

10. 缩小 DCT:由于图像的信息主要集中在低频区域,因此可只选取低频 DCT 系数矩阵.

11. 计算哈希值:对 DCT 系数取均值,若大于均值,则取 1;若小于均值则取 0.

12. 汉明距离计算:通过计算截取图像的哈希值与数据库中的哈希值的汉明距离.若小于一定的阈值,则可匹配出 M100 机头大致朝向,再根据二维码的偏转角度,得出准确的机头朝向.

13. 飞行距离计算:通过计算通过图像中心范围二维码数,可判断出飞机相对运动距离。