妙算程序技术文档

本文档为妙算端程序的技术说明，实现的功能包括装甲板识别、大神符识别、底盘功率检测、led程序状态显示。

# 程序流程



图0 妙算程序流程

# 装甲板识别

装甲板识别的任务是从图像中识别出装甲板，得到其具体位置。装甲板的可识别特征有：整体黑色平滑、两侧装有平行且竖直的灯条（通过红/蓝色区分敌我）。

装甲板识别的整体流程如图所示：



图1 装甲板识别流程

提取灯条的首要依据是白光周围有红/蓝光，因此实现中将红/蓝光膨胀后与白光做交集，这样得到的是灯条白光外圈，再将白光轮廓和白光外圈轮廓一一比对，只要白光外圈轮廓上任意一点位于白光轮廓内部，则表明该白光周围有红/蓝光，即为我们需要提取的灯条轮廓。之所以不能直接对红/蓝光二值图找轮廓进行矩形拟合作为候选灯条，主要原因是采集到的图像灯条周围的红/蓝光并不总是连续的，有可能拟合成多个小矩形，影响后面的装甲板识别。

单凭“白光周围有红/蓝光”这一条依据还不足以提取到正确的灯条，可以使用灯条的其它特征做进一步筛选，比如：宽高比、面积大小。

灯条两两组合经过初步筛选（灯条间夹角大小、灯条长度差异程度等诸多特征）之后得到候选的装甲板，在选出最可能的装甲板时使用的最重要的特征为装甲板内部灰度的均衡性，即计算内部所有像素点灰度的标准方差，标准方差较小的更有可能是装甲板。另外，也还可以使用其它一些特征，比如面积大的优先级更高。还有就是利用前一帧的识别结果，一般来说，两帧之间装甲板的位置以及面积都是变化不多的，特别是存在多目标的时候，利用装甲板位置信息，防止识别结果在不同目标之间来回切换，对后面的云台控制造成巨大干扰。

装甲板识别做的一个很大的性能优化是，当前一帧识别到目标时，当前帧只取前一帧识别目标周围一块区域做处理，而不是原来的整张图像。

# 大神符识别

大神符识别的任务是从以九宫格方式排列的九个图案中挑选出唯一不同的一个图案，整个九宫格每1.5s刷新一次，或者目标被击中之后立即刷新。

大神符识别的整体流程如图所示：

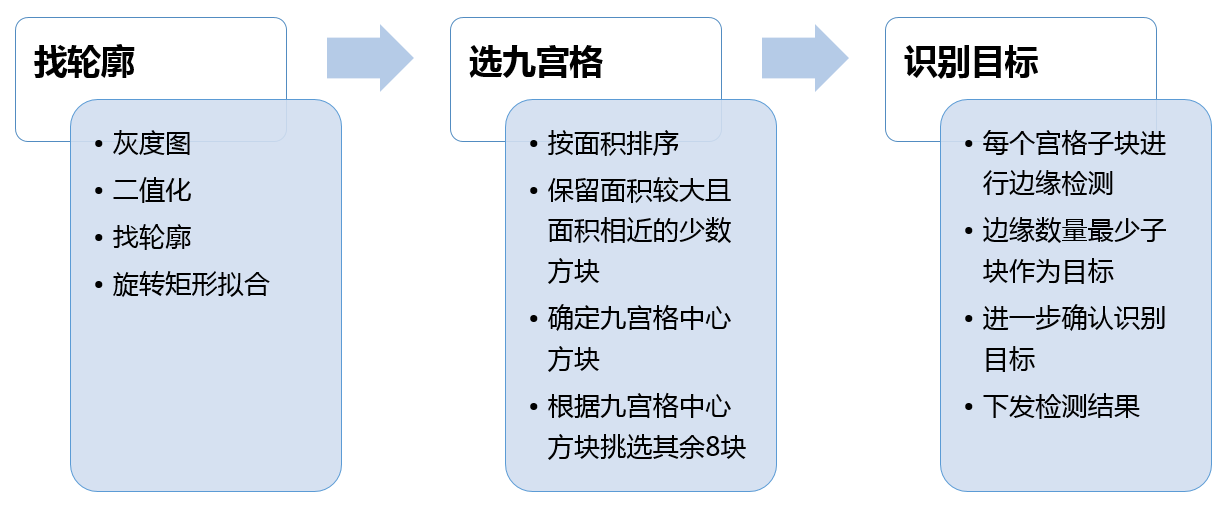


图2 大神符识别流程

对二值图找轮廓得到的结果有很多，可能有几百个，但是九宫格方块都是面积比较大并且面积相近的，又由于存在一些面积相近的干扰（不会太多），因此按照面积从大到小排序，依次遍历选出最多20个面积较大且面积相近的方块，九宫格九个方块必然包含在内。算法伪代码如图：

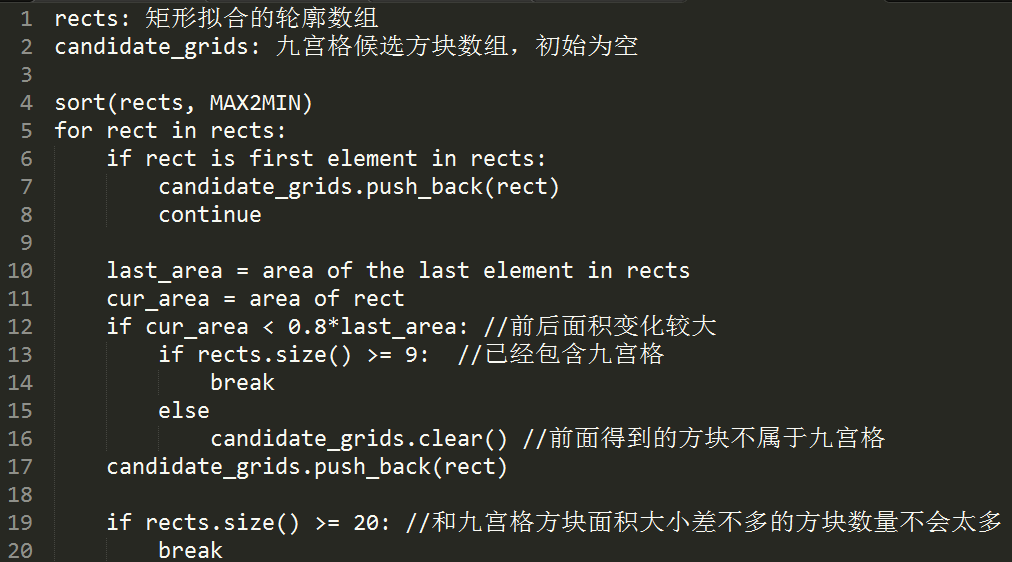


图3 算法伪代码

通过九宫格上下左右的排列方式，就能很容易地从仅剩不多的候选方块中确定九宫格中心方块，再通过与中心方块的距离信息便能挑选出剩下的8个方块。

目前规则中采用的九宫格图案为8个可任意旋转的战车和1个挥剑的战士，战士的图案要比战车简单，因此通过图案的边缘多少基本能区分战车和战士，边缘最少的是战士。

而最大的问题在于，云台移动过程中摄像头采集到的图像是模糊的，根本没办法区分目标图案和其它图案，选择边缘最少的图案作为目标，则必然会出错。为了规避这种情况，需要进一步判断是否8个图案边缘数量相差不多，而另外一个图案边缘数量明显比较少。

视觉算法之外，还需要考虑云台的控制方式，大符每次刷新的时候，妙算给单片机发送目标的编号（1-9），云台直接一次性调整到位发射子弹。因此，大符刷新之后，对新目标的识别准确率要求就很高了，然而单帧识别准确率是不可能达到100%的，这里采用了连续3次累计确认的方法，即连续3次都识别到新的同一个目标。另外，为了避免云台移动造成误识别，可以假设大符刷新发送目标之后400ms（云台移动并发射子弹的时间要大于400ms）内不会有新的识别目标。

# 功率检测

裁判系统功率检测原理如图所示：

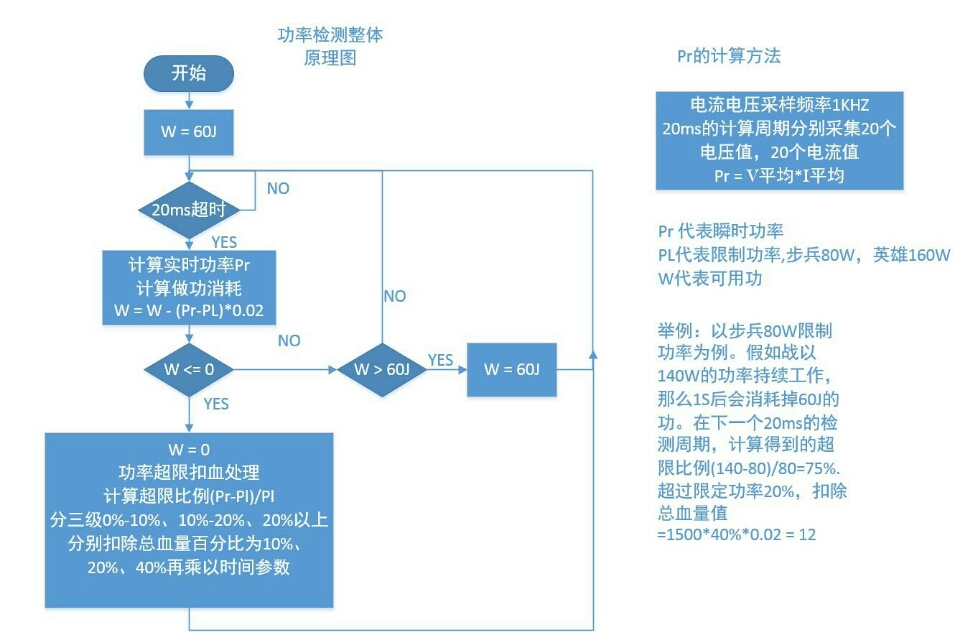


图4 裁判系统功率检测原理图

妙算负责计算W值，通过与单片机的串口连接发送给单片机，再由单片机做进一步的控制操作。

实际中有几个需要注意的地方：

1. 底盘不动的时候也会有功率值，因此裁判系统在做功率检测的时候，使用的瞬时功率Pr不同于显示屏上面显示的功率值，实际为显示的功率值减去某个固定的值。不过，使用妙算做功率检测的时候，可以忽略这一点，因为这样只会把功率限得更多。
2. 妙算串口接收数据，并不能每次都刚刚好接收到完整的一帧数据，会有各种情况：1帧数据分开2次接收、2帧数据一次性接收、丢失个别字节、CRC校验不过。因此，每次接收到数据要先放到缓冲区，然后再进行连续处理。
3. 数据帧无效的时候，无法确认当前帧的实时功率，则根据前2帧有效数据通过微分关系推测当前帧的实时功率。因为推测不一定准确，所以为了确保计算得到的W值比实际W值小，这种情况下要使W值有减无增，计算时采用的功率是推测所得功率的1.x倍（经过反复测试，1.3倍的时候既能兼顾速度，又能保证基本不掉血）。

# led状态显示

led供电连接妙算扩展接口的14引脚，接地可选16引脚。

程序功率检测功能正常运行之后，点亮led，之后如果摄像头正常，则每处理10帧图像反转led的状态，以达到闪烁的效果，否则主线程、图像采集线程睡眠，功率检测相关的串口线程依然正常运行，led长亮。

# 通信协议

妙算UART2连接单片机，UART3连接裁判系统比赛数据读取接口。

## 单片机->妙算

每帧数据只有1个字节，表示妙算程序要切换的状态。

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Index(byte) | Size(bit) | Description |
| STATE | 0 | 8 | 妙算程序要切换的状态   * 0xf0: 无操作 * 0xf1: 蓝色装甲板识别 * 0xf2: 红色装甲板识别 * 0xf3: 大神符九宫格识别 * 0xf4: 九宫格中心方块标定 * 0xf5: 九宫格右下方块标定 * 0xff: 大神符九宫格识别（云台闭环控制） |

## 妙算->单片机

每帧数据固定为6个字节。

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Index(byte) | Size(bit) | Description |
| TYPE | 0 | 8 | 帧类型   * 0xf1: 蓝色装甲板识别结果 * 0xf2: 红色装甲板识别结果 * 0xf3: 大神符九宫格识别结果 * 0xf4: 九宫格中心方块标定数据 * 0xf5: 九宫格右下方块标定数据 * 0xf6: 已发射子弹数据 * 0xf7: 底盘功率数据 * 0xf8: 装甲掉血数据 * 0xff: 大神符九宫格识别结果（云台闭环控制） |
| DATA | 1 | 40 | 帧数据 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TYPE ID | Field | Index(byte) | Size(bit) | Data Type | Description |
| 0xf1  0xf2 | DIS | 1 | 8 | int8\_t | 装甲板到摄像头距离的10倍，单位：m |
| X | 2 | 16 | int16\_t | 装甲板中心x轴坐标 |
| Y | 4 | 16 | int16\_t | 装甲板中心y轴坐标 |
| 0xf3 | IDX | 1 | 8 | int8\_t | 大神符九宫格识别目标编号（1-9） |
| RES | 2 | 32 | --- | 保留，固定为0 |
| 0xf4  0xf5 | RES | 1 | 8 | --- | 保留，固定为0 |
| X | 2 | 16 | int16\_t | 装甲板中心x轴坐标 |
| Y | 4 | 16 | int16\_t | 装甲板中心y轴坐标 |
| 0xf6 | RES | 1 | 8 | --- | 保留，固定为0 |
| POWER | 2 | 16 | int16\_t | 实时底盘功率 |
| WATT | 4 | 16 | int16\_t | 底盘剩余可用功 |
| 0xf7 | RES1 | 1 | 8 | --- | 保留，固定为0 |
| COUNT | 2 | 16 | int16\_t | 已发射子弹数量 |
| RES2 | 4 | 16 | --- | 保留，固定为0 |
| 0xf8 | RES1 | 1 | 8 | --- | 保留，固定为0 |
| VAL | 2 | 16 | int16\_t | 血量变化值 |
| RES2 | 4 | 16 | --- | 保留，固定为0 |
| 0xff | IDX | 1 | 8 | int8\_t | 大神符九宫格识别目标编号（1-9） |
| X | 2 | 16 | int16\_t | 大神符九宫格识别目标中心x轴坐标 |
| Y | 4 | 16 | int16\_t | 大神符九宫格识别目标中心y轴坐标 |

## 裁判系统->妙算

详见Robomasters官方文档《裁判\_系统协议整理(学生接口).pdf》。

# 注意事项

妙算串口每次发送完数据之后，需要等待一段时间才能发送下一次数据，否则单片机以DMA中断方式接收数据的时候有可能一次性接收到不止一帧数据而被丢弃。

妙算从裁判系统读取数据会出现很多丢失个别字节、CRC校验不过的情况。