**项目各个模块功能说明**

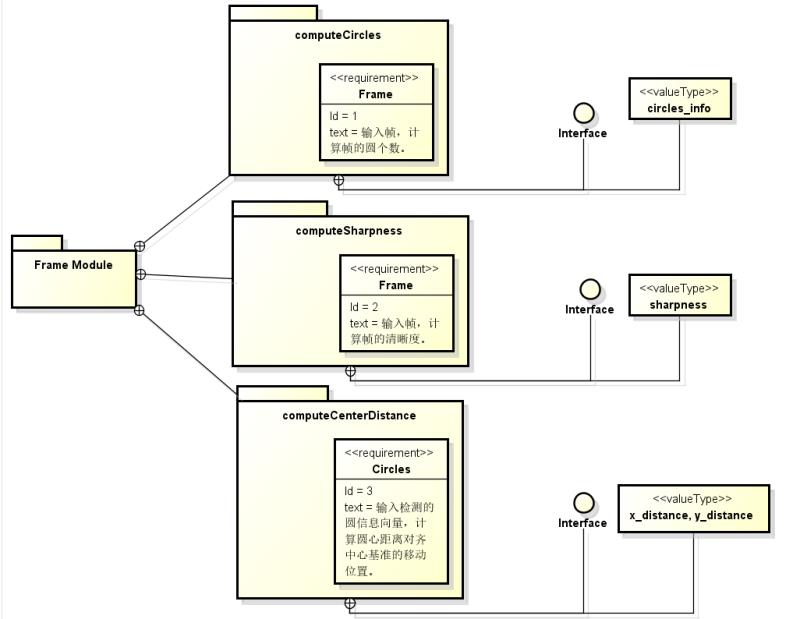
针对近期已完成的模块作出总结：

**视频模块设计**

图像相似度计算

### 图像模块设计

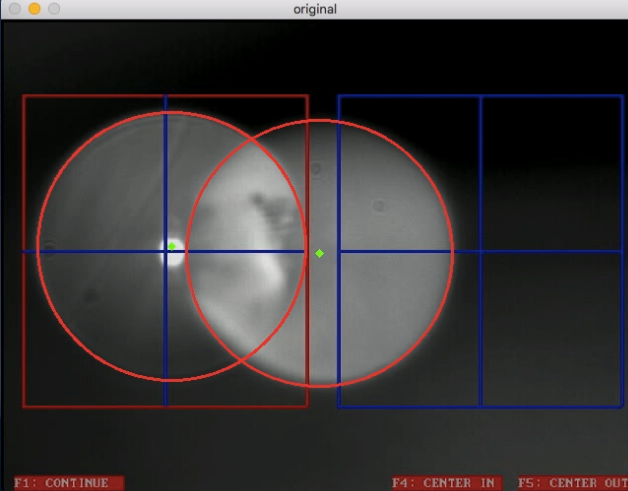
包括圆检测，清晰度检测，圆心距离计算三个子模块，分别输出相应的内部API接口，以供控制模块调用。



1. 圆检测

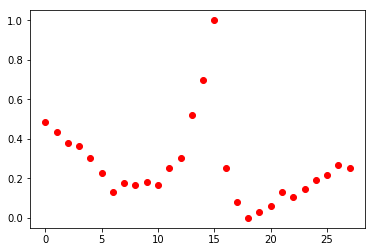
在许多应用场合中需要快速准确地检测出直线或者圆。其中一种非常有效的解决问题的方法是霍夫变换，Hough变换的基本原理在于利用点与线的对偶性，将原始图像空间的给定的曲线通过曲线表达形式变为参数空间的一个点。这样就把原始图像中给定曲线的检测问题转化为寻找参数空间中的峰值问题。

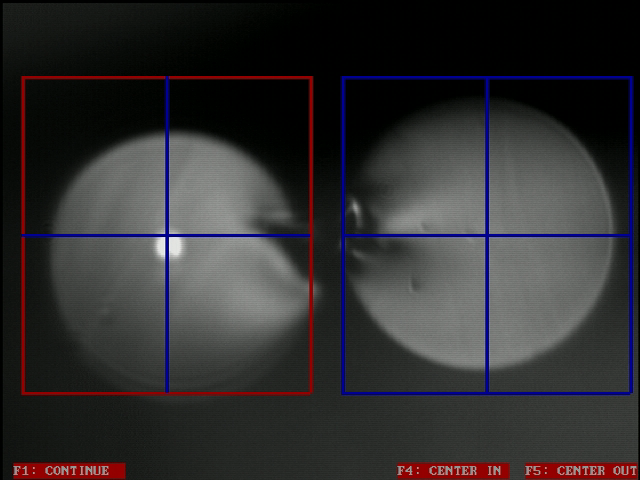
主要利用cv2.HoughCircles()函数计算图像中出现的圆数目。



(2) 清晰度检测

利用tenegrad（）函数，计算清晰度，结果越小，清晰度越好。



根据最小值可以基本判定锐度值最高图像。

根据图像来看，也可以看出其为最清晰图像。

## （3）圆心距离基准距离

项目要求：对于聚焦好的圆的分析可以给出拟合圆的直径像素数（或拟合圆边界曲线），圆心的屏幕坐标，圆心与方框图在上下、左右两方向的正负像素差值。

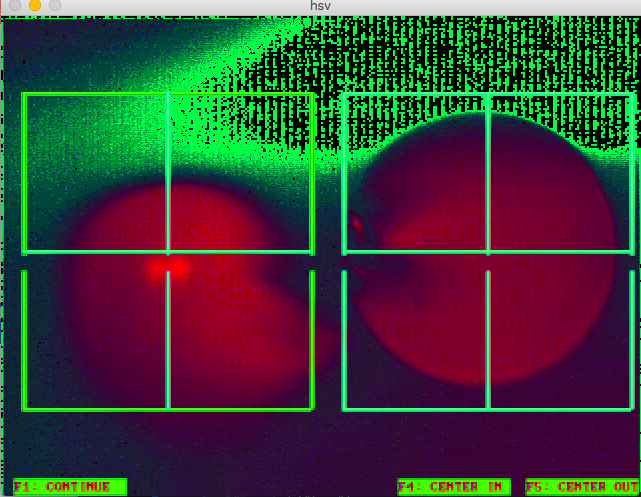
考虑到圆心坐标距离方框图在上下左右两方向距离等价于与方框图中心的距离。所以先考虑与基准点距离。

### 主要方法：

## 寻找基准点坐标

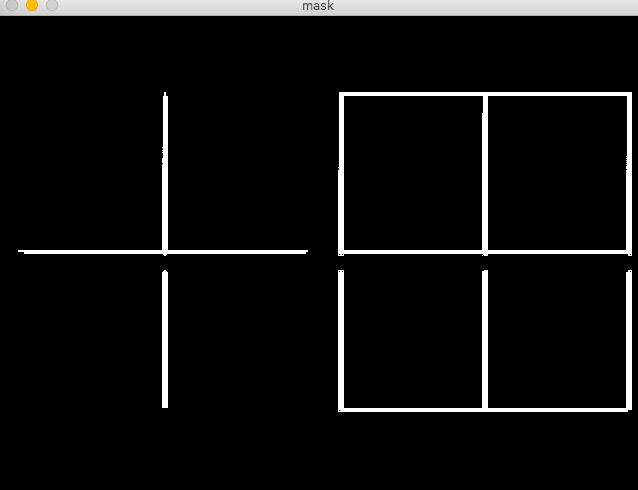
* 转换hsv颜色空间

利用cvtColor()将BGR格式转成HSV格式，参数为cv2.COLOR\_BGR2HSV。



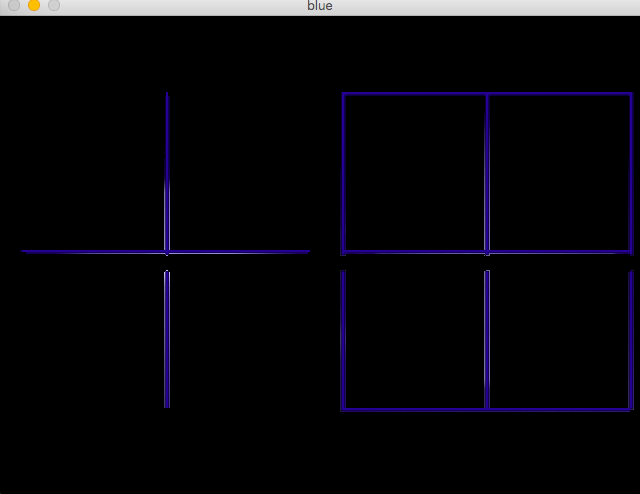
* 获取mask

利用inRange()函数和HSV模型中蓝色范围的上下界获取mask，mask中原视频中的蓝色部分会被弄成白色，其他部分黑色。

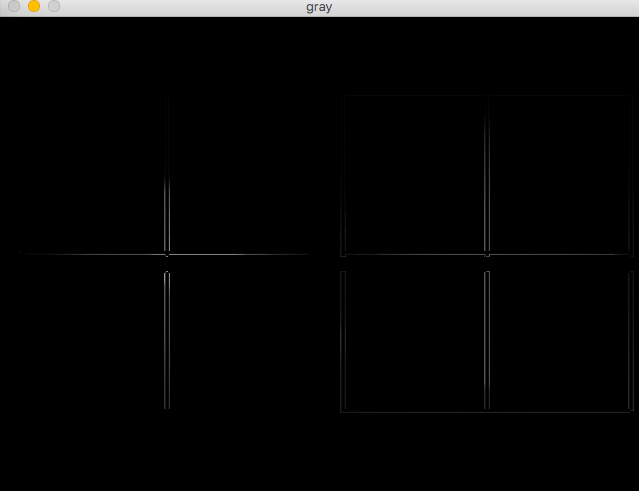


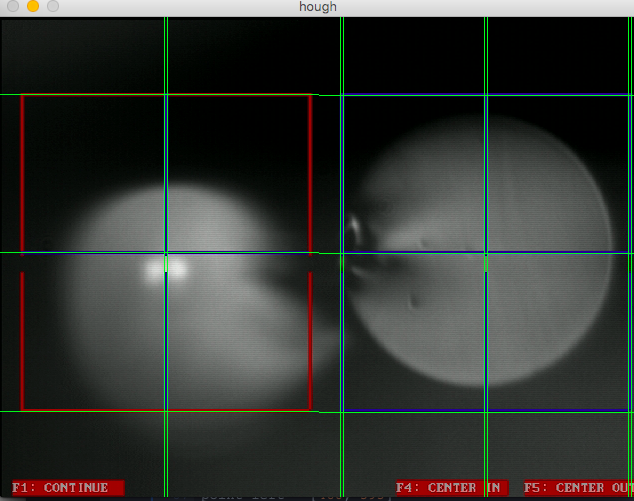
* 图像按位操作

利用cv2.bitwise\_and函数将mask于原视频帧进行按位与操作，则会把mask中的白色用真实的图像替换：



* 从hsv颜色空间转换到rgb再到灰度空间



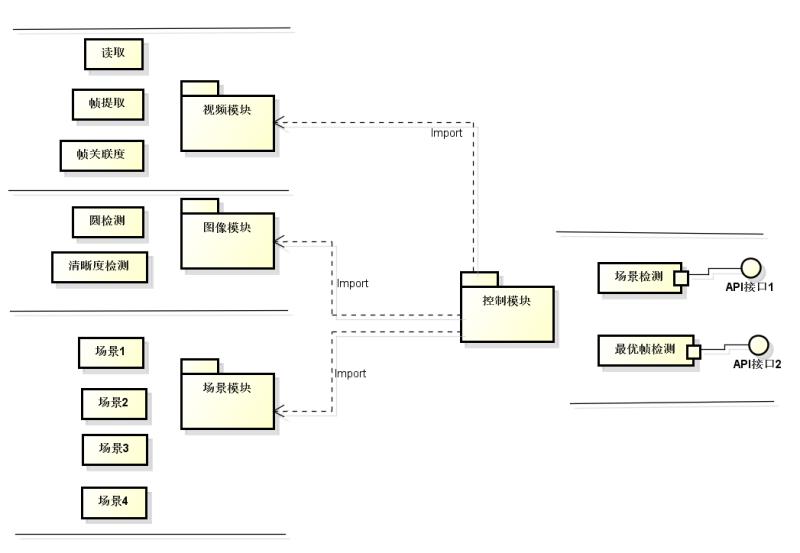
* 霍夫直线检测 利用cv2.HoughtLines()检测直线，并画出直线。 

## 计算圆心与基准点距离

* computeCenterDistance()

根据直线坐标，可以找出基准点的x,y坐标。目前的问题主要有，只能显示全部直线，然后通过自己推断选择出坐标。然后考虑在每一帧基准点位置不变，所以将其作为存储为参数，直接传入computeCenterDistance()函数。然后简单利用圆心坐标与基准点坐标相减，可以算出距离。

### 控制模块设计



控制模块导入视频模块、图像模块、场景模块，利用内部API接口做场景检测与最优帧检测。这两个API接口面向用户。

**# TODO**

1. 视频图像间的关联度。
2. 系统整合阶段。
3. 算法优化。