

# 作业说明：

Main.py 为程序代码  
Main 为 Linux 下可执行程序  
Balls.avi 为待跟踪视频  
Result.mp4 为跟踪结果视频

## 使用说明

Main 不依赖任何库，可直接运行在 Linux 环境下（Ubuntu16 亲测可行），只需将 Ball.avi 与 Main 放于同一目录下，最好使用 shell 运行  
假设当前目录下包含 Main 与视频文件，命令如下： ./Main，  
此时会弹出视频的第一帧，**提示勾选三个小球，此时需要使用鼠标左键，点击->拖拽至合理大小，请勿将选择框与其他小球有重叠**，选完 3 个小球后，视频会自动播放，小球的实时跟踪框也会展现在视频中。  
运行结束后，终端会出现“finish!! press Esc to exit!”，此时按 Esc 就能退出。  
**注：由于运行程序较大，启动可能需要几秒，请耐心等待。**

## 算法说明

主要思路如下：

1. 勾选小球，对小球跟踪框转 HSV，然后通过阈值过滤掉背景
2. 计算跟踪框的直方图
3. 播放视频后，对每一帧，都转 HSV，通过 H 信道过滤掉背景，再将最初的小球直方图反向投影到当前帧，这样就能够将背景、三个小球分别区分开
4. 使用 meanshift 从反向投影图中找到当前帧的小球跟踪框。  
此 meanshift 为在 opencv 上自己改善过的，原因在 opencv 提供的 meanshift 在小球距离过大时，会跟丢，我在这之上，套了一个广度搜索，先垂直方向，再水平方向，每次迭代移动 5 个像素，直到找到小球。

## 实验结果

见 result.mp4

# 小组成员

刘方超 201628015029012