

《计算机视觉》实验报告

姓名：林艺珺 学号：18120189

实验 9

任务 1

对 4 段视频中的任意一个，跟踪其中的指定目标
标识跟踪结果，画出轨迹，保存为新视频，并提交

a) 核心代码

个人笔记本为 MacOS 平台，选择开发语言为 Python，IDE 为 JetBrains 的 Pycharm。
学习 Dr. Adrian Rosebrock 发布在网站上的代码并根据要求进行改写。

GreenBall

```
1 .....
2 # 规定绿色的最高与最低HSV色域，并且初始化跟踪点轨迹的列表
3 greenLower = (29, 86, 6)
4 greenUpper = (64, 255, 255)
5 pts = deque(maxlen=args["buffer"]) # 默认值64
6 # 设置视频相关参数，包括捕获与保存
7 vs = cv.VideoCapture(args["video"])
8 fps = vs.get(cv.CAP_PROP_FPS)
9 size = (int(vs.get(cv.CAP_PROP_FRAME_WIDTH)), int(vs.get(cv.CAP_PROP_FRAME_HEIGHT)))
10 size = (600, int(vs.get(cv.CAP_PROP_FRAME_HEIGHT)/vs.get(cv.CAP_PROP_FRAME_WIDTH)*600)) # 压缩尺寸
11 fourcc = cv.VideoWriter_fourcc('M', 'P', '4', '2')
12 out = cv.VideoWriter('green_ball_result.avi', fourcc, fps, size)
13 # 循环获取当前帧并进行目标跟踪
14 while True:
15     frame = (vs.read())[1]
16     # 结尾判断
17     if frame is None:
18         break
19     # 压缩尺寸、高斯模糊并转换成HSV色域
20     frame = imutils.resize(frame, width=600)
21     blurred = cv.GaussianBlur(frame, (11, 11), 0)
22     hsv = cv.cvtColor(blurred, cv.COLOR_BGR2HSV)
23     # 为绿色建立遮罩，并通过erode和dilate去除噪声
24     mask = cv.inRange(hsv, greenLower, greenUpper)
25     mask = cv.erode(mask, None, iterations=2)
26     mask = cv.dilate(mask, None, iterations=2)
27     # 找到遮罩的轮廓并找到圆心
28     cnts = cv.findContours(mask.copy(), cv.RETR_EXTERNAL,
29                             cv.CHAIN_APPROX_SIMPLE)
30     cnts = imutils.grab_contours(cnts)
31     center = None
32     # 找到一个轮廓后进行以下步骤
33     if len(cnts) > 0:
34         # 找到最大的轮廓并计算圆及圆心
35         c = max(cnts, key=cv.contourArea)
```

```

36         ((x, y), radius) = cv.minEnclosingCircle(c)
37         M = cv.moments(c)
38         center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))
39         # 找到最小的半径
40         if radius > 10:
41             # 在当前帧上绘制圆圈及圆心，然后更新到跟踪点轨迹列表
42             cv.circle(frame, (int(x), int(y)), int(radius),
43                       (0, 255, 255), 2)
44             cv.circle(frame, center, 5, (0, 0, 255), -1)
45         pts.appendleft(center)
46         # 循环遍历跟踪点轨迹
47         for i in range(1, len(pts)):
48             if pts[i - 1] is None or pts[i] is None:
49                 continue
50             # 计算连线的粗细并且绘制
51             thickness = int(np.sqrt(args["buffer"] / float(i + 1)) * 2.5)
52             cv.line(frame, pts[i - 1], pts[i], (0, 0, 255), thickness)
53         # 显示绘制好的当前帧
54         cv.imshow("Frame", frame)
55         key = cv.waitKey(1) & 0xFF
56         # 保存当前帧
57         out.write(frame)
58         .....

```

b) 实验结果截图

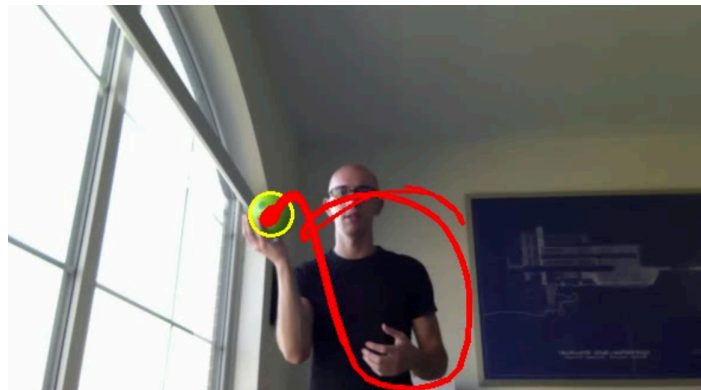


图 1: GreenBall

c) 实验小结

Dr. Adrian Rosebrock 的程序十分健全，基本考虑了所有可能出现的异常，相比之下，之前几次实验自己写的代码稍有不慎就会出现不明所以的异常终止程序，要是直接应用在商业中是会引出巨大问题的。从代码的简洁性上，也再次意识到了 OpenCV 的强大，基本功能的调用组合能够完成许多操作。这段时间在构想自己的计算机视觉大作业往什么方向进行，由于自己对摄影摄像还算比较感兴趣，于是想在滤镜方面做一些成果。但是真正开始构想的时候，发现自己还没有将所学的知识完全理解与掌握，这次目标跟踪程序学习的过程中很明显感觉到，构想功能实现的过程，就是把目标实现的路一步步分解的过程，而足够的基础知识，是帮助你分离、分解目标的利器，只有当你足够了解这些知识点的时候你才能够融会贯通加以应用。不得不说，大作业对我来说是一个挑战，在进行之前，务必要再次深入本学期的知识点。

任务 2（选做）

尝试剩余的 3 段视频

a) 核心代码

个人笔记本为 MacOS 平台，选择开发语言为 Python，IDE 为 JetBrains 的 Pycharm。
使用图像软件获取 HSV 色域的值范围，再根据任务 1 的代码进行改写。

OrangeBall

```
1 .....  
2 orangeLower = (5, 155, 155)  
3 orangeUpper = (23, 255, 230)  
4 .....  
5 out = cv.VideoWriter('orange_ball_result.avi', fourcc, fps, size)  
6 .....
```

YellowBall

```
1 .....  
2 yellowLower = (20, 125, 100)  
3 yellowUpper = (25, 255, 205)  
4 .....  
5 out = cv.VideoWriter('yellow_ball_result.avi', fourcc, fps, size)  
6 .....
```

Rat

```
1 .....  
2 ratLower = (0, 0, 0)  
3 ratUpper = (180, 35, 35)  
4 .....  
5 out = cv.VideoWriter('rat_result.avi', fourcc, fps, size)  
6 .....
```

b) 实验结果截图

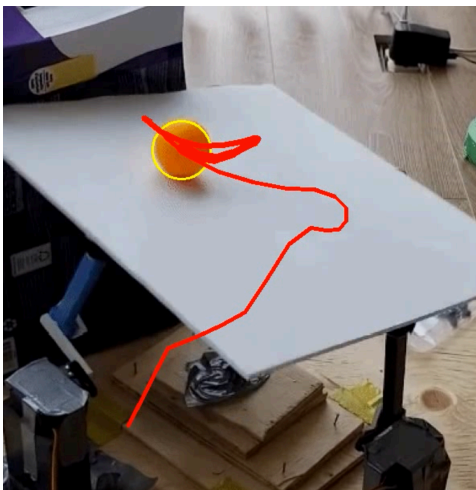


图 2: OrangeBall

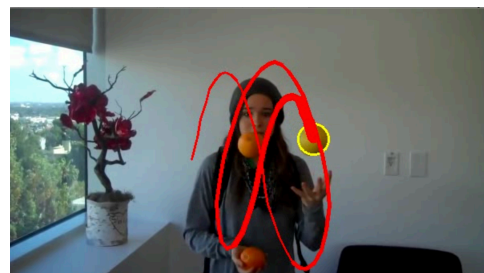


图 3: YellowBall

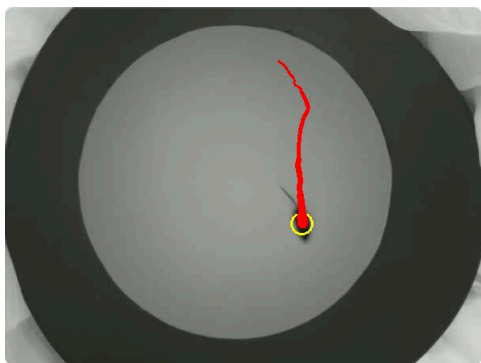


图 4: Rat

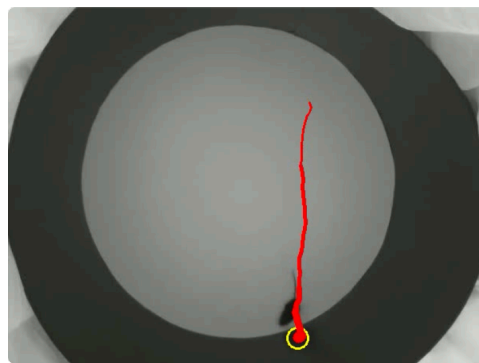


图 5: Rat 误判

c) 实验小结

橙色球和黄色球的原理与绿色球基本一致，只需要修改 HSV 色域即可。

本来认为老鼠的追踪大同小异，奈何整个画面都是灰黑色调，之前使用软件取色的方法没有用了，只能在软件一点一点调试，最终使用了误判率较少的一个值。由于老鼠走到下半部分时，在周围产生的阴影与其本身实在过于接近，很难不发生误判。