

姓名

- 倪诗宇

学号

- 201900180065

实验日期

- 2020.10.17

实验题目

- 基于直方图的目标跟踪

实验过程中遇到的问题和解决方法

- 问题一：
 - 问题：刚开始写的时候，完全没有思路
 - 解决：看了一篇网上的代码，有了思路。主要过程就是
 - 首先打开摄像头，对视频的每一帧都进行分析
 - 然后按住鼠标左键选取自己想要进行跟踪的部分
 - 这里主要用到了一些没有接触过的函数

```
void onMouse(int event, int x, int y, int flags, void *param)
```

event: 鼠标回传事件的号码

x, y: 鼠标的坐标

flags: 代表拖拽事件，和event构成组合事件

param: 用户定义的传递到setMouseCallback函数调用的参数。

```
void setMousecallback(const string& winname, MouseCallback  
onMouse, void* userdata=0)
```

winname: 窗口名称

onmouse: 鼠标响应函数，回调函数。指定窗口内每次发生鼠标事件的时候，被调用的函数。这个函数的结构就是上面说的onMouse()

userdata: 传递给回调函数的参数

```
rectangle(image, startPoint, endPoint, Scalar(255, 0, 0),  
thickness)
```

image: 要进行画框的图像

startPoint, endPoint: 画的框的左上角和右下角两个点的坐标

Scalar: 选颜色

tickness: 框的线宽, 如果是 -1, 则代表填充整个矩形

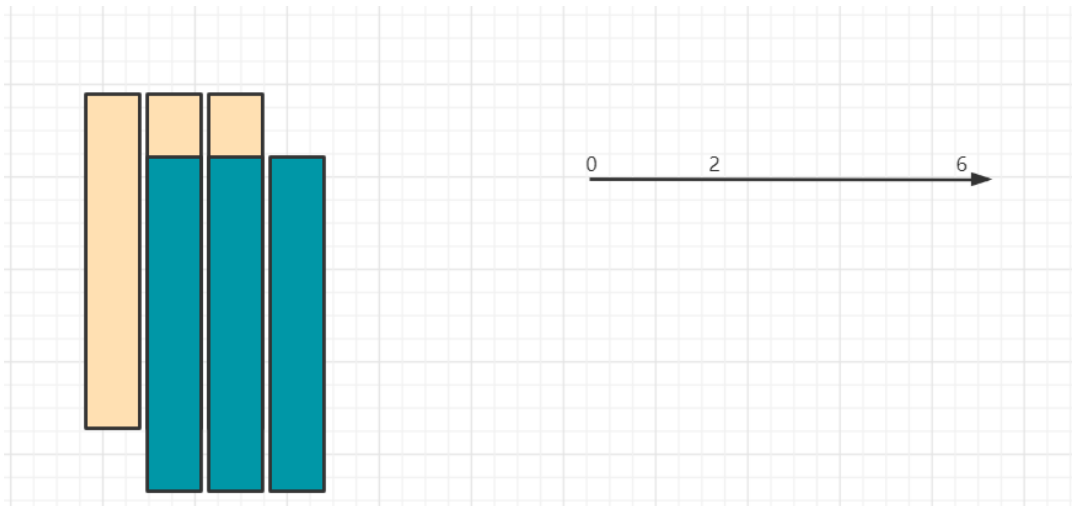
- 计算选取部分图像的直方图, 为source
 - 进行图像增强
 - 扫描选择图像的周围区域, 计算直方图, 进行图像增强
 - 将两个直方图进行比对
 - 若很相似, 则画框
 - 若较相似, 改变source的值为当前图像的直方图的值。增强连续性
- 问题二:
 - 问题: 在画矩形框的时候, 很容易在框画到边界的时候, 框变小, 甚至挤在角落里, 变成一条线
 - 解决: 在这种情况下是没有对直方图进行归一化的。如果不归一化, 纯粹去数不同像素值对应的像素个数, 那么图像大小也会对结果造成影响。考察的图像小, 像素值就都少, 在与source进行比对的时候, 会对结果造成影响。可以通过计算完直方图之后, 对每个值, 除以它所在的图像的像素个数来进行归一化, 忽略图像大小对相似度判断的干扰

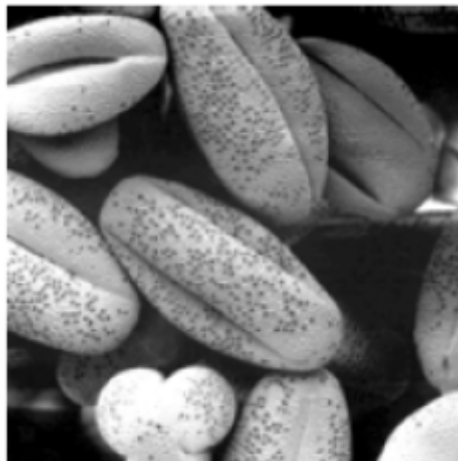
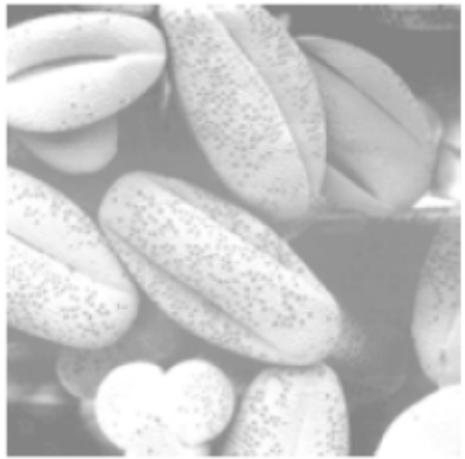
结论分析与体会

- 问题: 若两个图相似, 但是亮度不同, 是不是就不容易辨认了
- 解决:

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j)$$

- 对于这两个图, 三个像素值分别都有一个像素, 绿色的为黄色的向右 +1
 - 如果两个图像的像素完全不重叠
 - 不优化的情况下, 差距为6
 - 优化情况下, 差距为4以上
 - 图中情况
 - 不优化, 差距为2, 占1/3
 - 优化, 差距为0
 - 针对两个图片中物体相同, 但整体亮度不同的情况, 能很好地判断相似。





- 本次实验趣味性比较高，但明显比以前难了。不过学到了鼠标交互方法，应用了课内知识，有不错的收获
- **我这里这个公式并没有改变像素值，其实是一个错误使用。**