实验六 视觉跟踪

1. 实验目的
   1. 理解视觉跟踪的原理。
   2. 学习Python中函数的写法和文件之间的函数调用。
   3. 掌握从Github下载源代码的方法，创建自己的Github源代码网站。

4. 能够调用基于MOSSE模型的相关滤波的视觉跟踪源码。

1. 实验内容
   1. 运用断点调试的方法解读代码中主要函数的用途。
      1. 通过断点调试，已大略了解代码中各个函数的用途，并以代码注释的形式记录在了原代码中，其主要函数包括读取图片，获取初始的滤波器，每一帧处理前对滤波器进行预处理，以及在命令行对代码进行控制的语句。
   2. 将跟踪的结果以视频的方式保留。
      1. 在命令行中输入保存的指令，使得跟踪时每一帧图片得以保存，再利用VideoWrite函数将这些图片转化为视屏文件。
   3. 统计每一帧计算的时间。
      1. 在源代码循环对每一帧进行遍历的过程中，以cv2.getTickCount()函数对运算时间进行统计，并将每一帧计算结果加入到一个list当中。另统计了list的均值，得到每帧计算时间约为12.196ms.

4. 对代码进行运用。

* + 1. 通过前几次实验所学，利用摄像头拍摄了一段视屏并保存每帧图像，利用Mosse代码对自己录制的视频中物体进行了跟踪。

1. 实验结论
   1. MOSSE算法是依据上一帧的滤波器位置在其中找到本帧的响应最大点，并以此为基点重新构造滤波器，这就造成了两个问题。

1.是当所跟踪的物体运动的足够快时，在上一帧滤波器的位置上会完全没有属于跟踪物体的部分，从而使得下一帧所构造的滤波器与所跟踪的物体没有任何联系，极易出现一直留在原地的情况。

2.是滤波器的大小一直保持恒定，除非是跟踪物体运动到了画框的边缘，否则不变，这就导致了当跟踪物体与拍摄点的距离发生变化时，无法进行准确地跟踪。