IOS 与无人机开发项目报告

题目： **物体追踪**

姓名：杨展富

学号：14301033

学院：数据科学与计算机学院

目录：

一．总述：

在本次项目中，我做出了三个版本的 app，对应版本分别有以下功能：摄像头开关控制，高斯滤波与边缘检测，物体追踪。

二．问题背景：

运动目标跟踪是在视频图像中的每一幅图像中确定出我们感兴趣的运动目标位置，并把不同帧中同一目标对应起来。智能视频监控是计算机视觉领域近几年研究较多的方向，它能够利用计算机视觉技术对采集到的视频信号进行处理，分析和理解。并伴随着无人机开发的热潮，视频监控是无人开发必不可少的研究方向之一，它能够结合具体产业，产生巨大的产业价值。作为视频监控的热门方向，物体追踪在近几年的研究中非常热门。

三，系统环境：

System：OSX 10.10

Ide：Xcode

Implement： Iphone 6.

四．功能设计与算法介绍：

版本一：摄像头开关的控制

采用 xcode 中提供的 Camera 接口，并参考 IOS 教程 3 的滤波处理.

```
@interface ViewController :UIViewController<CvVideoCameraDelegate>
{
    IBOutlet UIImageView* imageView;
    IBOutlet UIButton* button;
    CvVideoCamera* videoCamera;
}

@property(nonatomic,retain)CvVideoCamera * videoCamera;
@property(strong,nonatomic)IBOutlet UIImageView* imageView;
@property(strong,nonatomic)IBOutlet UIButton* button;
```

版本二：高斯滤波与边缘检测

介绍：高斯滤波是一种线性平滑滤波，适用于消除高斯噪声，广泛应用于图像处理的减噪过程。通俗的讲，高斯滤波就是对整幅图像进行加权平均的过程，每一个像素点的值，都由其本身和邻域内的其他像素值经过加权平均后得到。高斯滤波的具体操作是：用一个模板（或称卷积、掩模）扫描图像中的每一个像素，用模板确定的邻域内像素的加权平均灰度值去替代模板中心像素点的值。

实现：

Step 1：#import <opencv2/videoio/cap_ios.h> 并导入相关库函数

Step 2：将 videoCamera 对象与 imageView 连接

Step 3：对获取的实时图像进行处理

```
-(void)processImage:(cv::Mat &)image
{
    // Mat image_copy;
    //cvtColor(image,image_copy,CV_BGRA2BGR);
    //cv::bitwise_not(image_copy, image_copy);
    //cvtColor(image_copy, image, CV_BGR2BGRA);
    cv::Mat gray;
    // Convert the image to grayscale;
    cv::cvtColor(image, gray, CV_RGBA2GRAY);
    // Apply Gaussian filter to remove small edges
    cv::GaussianBlur(gray, gray, cv::Size(5,5), 1.2,1.2);
    // Calculate edges with Canny
    cv::Mat edges;
    cv::Canny(gray, edges, 0,   30);
    // Fill image with white color
    image.setTo(cv::Scalar::all(255));
    // Change color on edges
    image.setTo(cv::Scalar(0,128,255,255),edges);
    // Convert cv::Mat to UIImage* and show the resulting im
    self.imageView.image = MatToUIImage(image);
}

/**
```

版本三：物体追踪

算法介绍：

1.CMT：Clustering of Static-Adaptive Correspondences for Deformable Object Tracking，2015 年发表在 CVR 上的论文（计算机视觉顶级会议），其核心思想是基于物体的特征点来跟踪，就是实时的监测物体的特征点，与一开始的特征点进行匹配的方法来实现物体的跟踪。具体算法如下：

Algorithm 1 CMT

Input: $I_1, \ldots, I_n, b_1$
Output: $b_2, \ldots, b_n$

$O \leftarrow \text{detect}(I_1, b_1)$
$K_1 \leftarrow O$
for $t \leftarrow 2, \ldots, n$ **do**
  $P \leftarrow \text{detect}(I_t)$
  $M \leftarrow \text{match}(P, O)$
  $T \leftarrow \text{track}(K_{t-1}, I_{t-1}, I_t)$
  $K' \leftarrow T \cup M$
  $s \leftarrow \text{estimate\_scale}(K', O)$
  $\alpha \leftarrow \text{estimate\_rotation}(K', O)$
  $V \leftarrow \text{vote}(K', O, s, \alpha)$
  $V^c \leftarrow \text{consensus}(V)$
  $K_t \leftarrow \text{vote}^{-1}(V^c)$
  if $|V^c| \geq \theta \cdot N^O$ **then**
    $\mu \leftarrow \frac{1}{n} \sum_{i=1}^{n} V_i^c$
    $b_t \leftarrow \text{bounding\_box}(b_1, \mu, s, \alpha)$
  else
    $b_t \leftarrow \emptyset.$
  end if
end for

解释：

输出：每一帧视频的物体框

要求：后继的物体框能够保持框住初始框框住的物体 （无其他遮挡物）

步骤：

Step 1：检测初始视频帧的所有特征点及其描述(整个图像的点)

Step 2：将初始框内的特征描述赋给 K1

Step 3：从第二帧开始至最后一帧

Step 4：检测视频帧的特征点 P

Step 5：将特征点 P 与 O 匹配，获取匹配的特征点 M

Step 6：利用上帧特征点使用光流法跟踪得这帧特征点位置 T

Step 7：融合特征点 M 和特征点 T 得到这一帧总的特征点 K’

Step 8：根据 K’估计特征点相对初始帧特征的缩放比例

Step 9：根据 K’估计特征点相对初始帧特征的旋转比例

Step 10：根据 Step7，8，9 得到的数据计算每个特征点的 Vote

Step 11：采用聚类的方法选取最大的类也就是最一致的 VoteC

Step 12：将 VoteC 转换回特征点得到最后这一帧的有效特征点

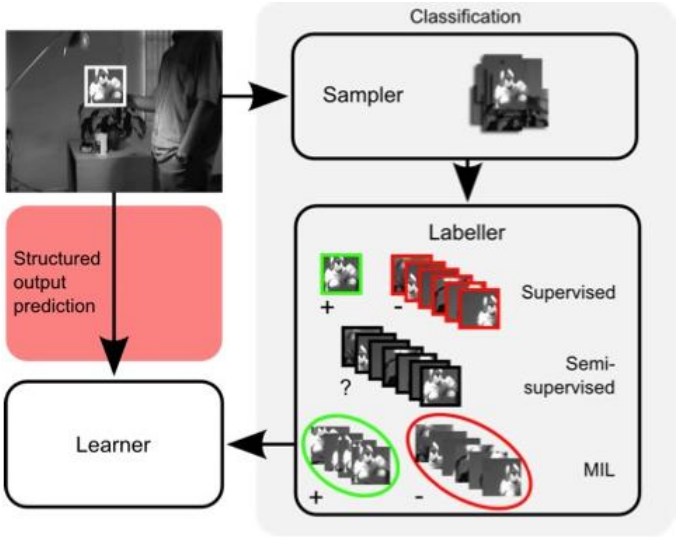
Step 13：判断 VoteC 的长度是否大于最小阈值，如果是则计算最后新的旋转矩形框的参数，如果不是也就是框太小了则输出 0.

2.TDL：

TLD(Tracking-Learning-Detection)是英国萨里大学的一个捷克籍博士生 Zdenek Kalal 在其攻读博士学位期间提出的一种新的单目标长时间（long term tracking）跟踪算法。该算法与传统跟踪算法的显著区别在于将传统的跟踪算法和传统的检测算法相结合来解决被跟踪目标在被跟踪过程中发生的形变、部分遮挡等问题。同时，通过一种改进的在线学习机制不断更新跟踪模块的“显著特征点”和检测模块的目标模型及相关参数，从而使得跟踪效果更加稳定、鲁棒、可靠。

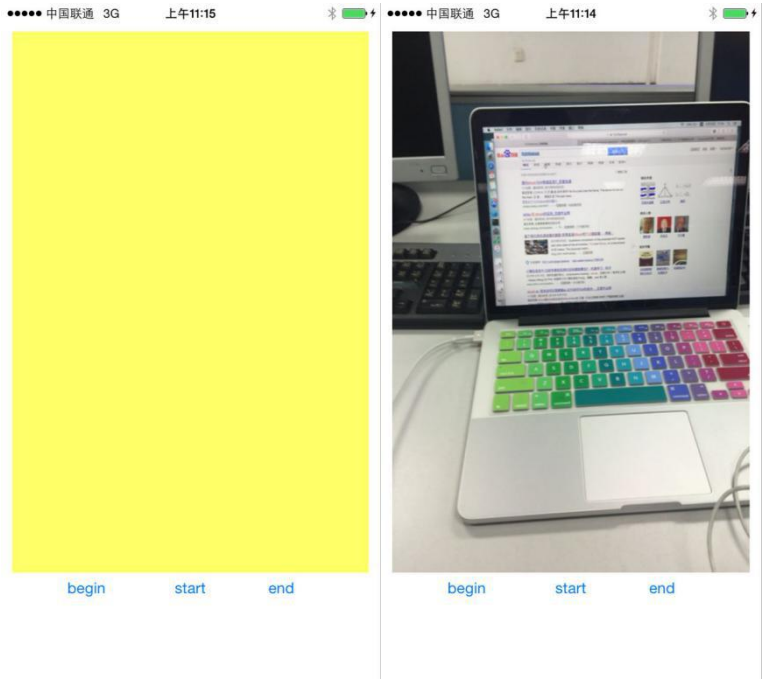
3.Struck：

《Struck:Structured Output Tracking with Kernels》是 Sam Hare, Amir Saffari, Philip H. S. Torr 等人于 2011 年发表在 Computer Vision (ICCV)上的一篇文章。Struck 与传统跟踪算法的不同之处在于：传统跟踪算法(下图右手边)将跟踪问题转化为一个分类问题，并通过在线学习技术更新目标模型。然而，为了达到更新的目的，通常需要将一些预估计的目标位置作为已知类别的训练样本，这些分类样本并不一定与实际目标一致，因此难以实现最佳的分类效果。而 Struck 算法(下图左手边)主要提出一种基于结构输出预测的自适应视觉目标跟踪的框架，通过明确引入输出空间满足跟踪功能，避免中间分类环节，直接输出跟踪结果。同时,为了保证实时性，该算法还引入了阈值机制，防止跟踪过程中支持向量的过增长。



五．实现效果：

版本一：摄像头控制



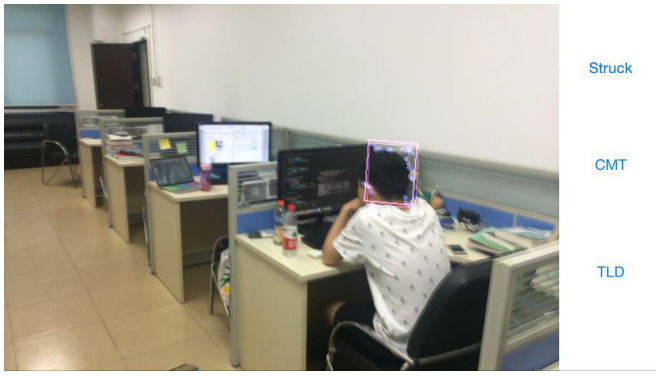
（关）　　（开）

版本二：高斯滤波与边缘检测：

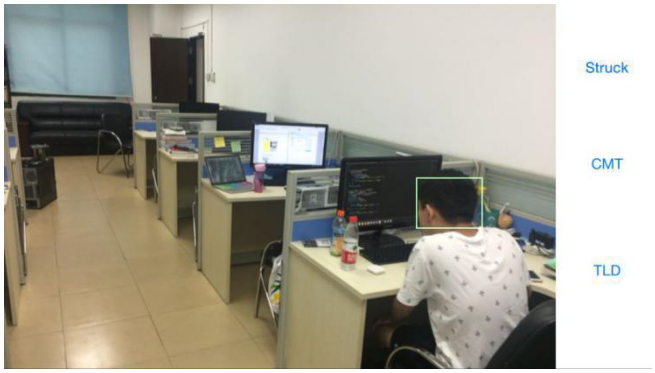
（滤波）　　（边缘检测）

版本三：物体跟踪：

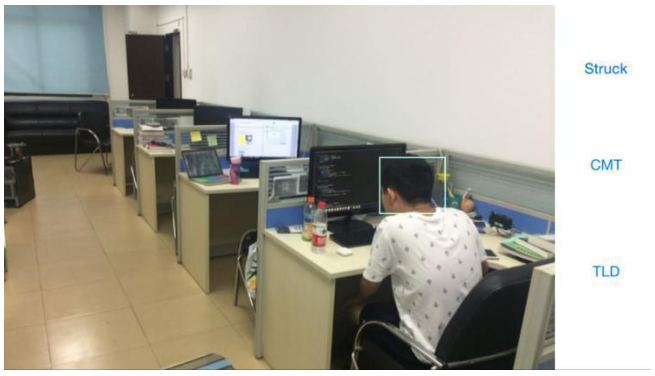
（开启等待页面）



（CMT，反应快，遮挡物影响大）



（TDL，反应慢）



（struck，反应慢）

六．总结

视觉是人类获取信息的重要途径。计算视觉的研究成果使机器具备了人类视觉的某些能力，同时也使人类在机器的协助下看得更清楚、更准确。在无人机开发过程中，只能检测更是其中的重中之重，在这次项目中，我稍微理解了一下视觉的一些算法以及实现了一些 ios 相关的应用，希望在今后 ios 或者无人机视觉计算的开发和学习中，我能继续深入了解技能，不断创新进步。

而且在这次项目中，我所实现的大多为单目标的物体追踪，也希望在今后的学习中，能够跟深入了解多一些多目标等一些更贴近实际生活中大数据应用的算法以及实现。