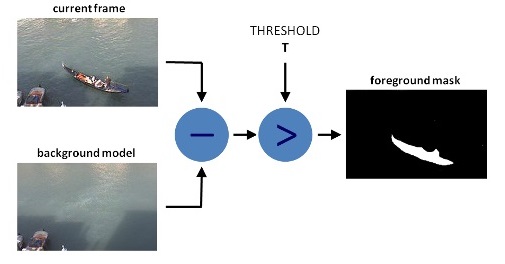
**基于高斯建模的背景分割**

一、概要

在很多基础应用中背景检出都是一个非常重要的步骤。例如顾客统计，使用一个静态摄像头来记录进入和离开房间的人数，或者是交通摄像头，需要提取交通工具的信息等。在所有的这些例子中，首先要将人或车单独提取出来。也就是说，我们需要从静止的背景中提取移动的前景。

如果你有一张背景（仅有背景不含前景）图像，比如没有顾客的房间，没有交通工具的道路等，那就好办了。我们只需要在新的图像中减去背景就可以得到前景对象了。但是在大多数情况下，我们没有这样的背景图像，所以我们需要从我们有的图像中提取背景，即进行背景建模。



**图1 背景减除**

本文主要介绍了三种背景建模的方法，并利用Python和OpenCV来编写程序，对每个方法进行测试。

二、单高斯背景建模

单高斯模型的基本思想是：将图像中每一个像素点的颜色值看成是一个随机过程X，并假设该点的某一像素值出现的概率服从高斯分布。令I(x, y, t)表示像素点(x, y, t)在



其中和分别为t时刻该像素高斯分布的期望值和标准差。

具体步骤如下：

1. 初始化背景模型，为了计算偏差，增加一个方差



其中std\_init通常设置为20

1. 检测前景像素和背景像素

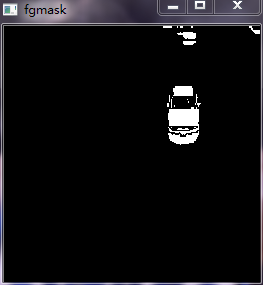


1. 对背景模型进行更新

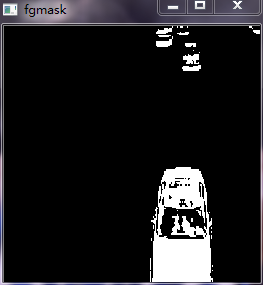
对那些被判定为背景的像素的背景模型进行更新：



利用Python，按照以上的步骤编写程序，程序中的学习率取值为0.03。实验结果如下图所示。



**图2 第25帧的背景图与前景图**



**图3 第55帧的背景图与前景图**

从以上结果可以看出能较好的提取出前景，但对阴影也当成了动态处理，无法很好的处理阴影部分。该方法受光照变化的影响较大。

三、混合高斯背景建模

混合高斯背景模型是 P.KadewTraKuPong和 R.Bowden 在 2001 年提出的。它使用 K（K=3 或 5）个高斯分布混合对背景像素进行建模。即用K个高斯模型来表征图像中各个像素点的特征：

其中：K为高斯分布的个数，为t时刻第i个高斯分量的加权系数

对于一个像素的K个高斯分量，根据 的值对它们从大到小进行排列，对于满足下式的前B个高斯分布被当作是背景模型 ：

其中T是背景模型占有高斯分布的最小比例，通常为0.7。

检测背景与前景：

2531780_12957503243ISi

若该像素被认为是前景，即前B个高斯分布中没有一个与之匹配，则用一个新的高斯分布取代权重最小的那个高斯分布。新的分布的期望值即为当前的像素值，同时为它分配一个较大的初始偏差std\_init和较小的初始权重值weight\_init 。

若该像素被认为是背景，则对该像素的各个高斯分布的权重做如下调整 ：

2531780_1295750325ZmWJ

其中α为学习率，值为0到1；如果第i个高斯分布与当前像素匹配，；否则 。

同时对于与当前像素匹配的高斯分布，更新它们的期望值和偏差值：

2531780_1295750325YuDB

**2531780_1295750326yepS**

**2531780_129575032624Cz**

调用opencv中cv2.BackgroundSubtractorMOG(history,nmixtures,backgroundRatio[,noiseSigma])函数创建一个背景对象，其中：

history: 进行建模场景的时间长度

nmixtures: 混合高斯的个数

backgroundRatio: 背景比

noiseSigma: 噪声强度

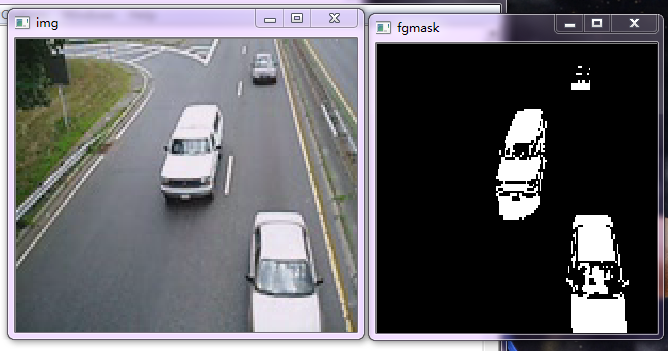
然后利用 BackgroundSubtractorMOG.apply(image [, fgmask, [learningRate]]) 函数计算前景掩模，其中：

image: 下一视频帧

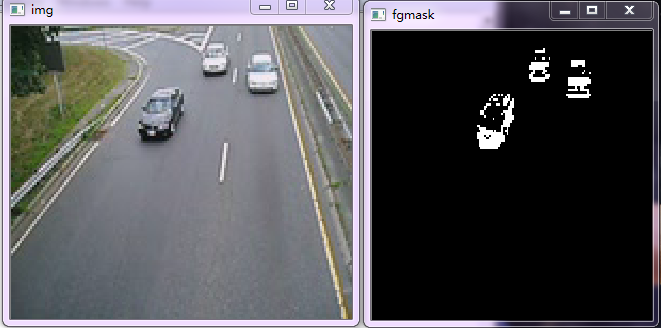
fgmask: 输出前景掩模（8为二进制图）

learningRate: 学习率

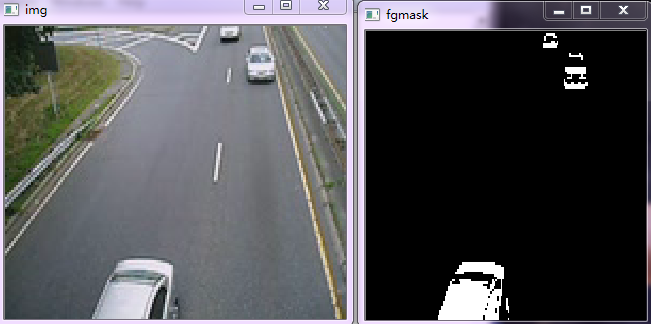
实验中选择的混合高斯个数K=5；背景比T=0.7，学习率为0.02。实验的结果如下图所示：



**图4 第40帧的原图和前景图**



**图5 第65帧的原图和前景图**



**图6 第80帧的原图和前景图**

可以看出该方法也能较好的提取出前景，但对阴影也当成了动态处理，无法很好的处理阴影部分；也会出现一些白噪声影响处理效果，对于光照的影响有了一定的抑制。

四、自适应高斯背景建模

自适应高斯模型是对高斯混合模型的改进，它是以 2004 年和 2006 年 Z.Zivkovic 的两篇文章为基础的。这个算法的一个特点是它为每一个像素选择一个合适数目的高斯分布（混合高斯模型使用是 K 高斯分布）。这样就会对由于亮度等发生变化引起的场景变化产生更好的适应。

调用opencv中的cv2.BackgroundSubtractorMOG2(history, varThreshold [, bShadowDetection])函数创建一个背景对象，其中：

history: 进行建模场景的时间长度

varThreshold: 阈值决定它是否能很好地描述背景模型（该参数不影响背景更新）

bShadowDetection: 定义是否检测阴影的参数（True / False ）

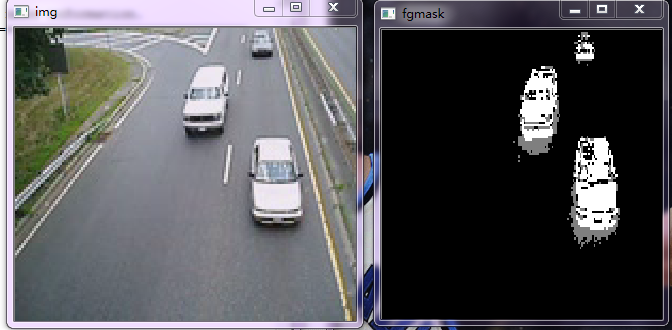
然后利用BackgroundSubtractorMOG2.apply(image [, fgmask, [learningRate]]) 函数计算前景掩模，其中：

image: 下一视频帧

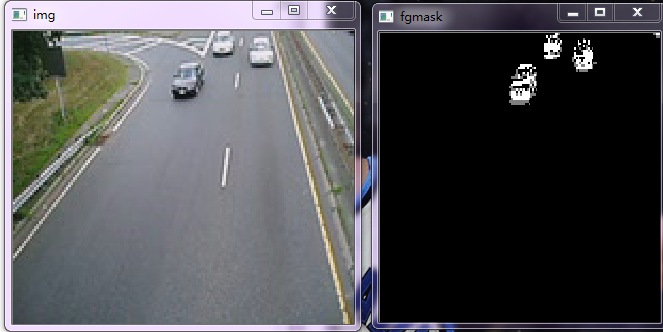
fgmask: 输出前景掩模（8为二进制图）

learningRate: 学习率

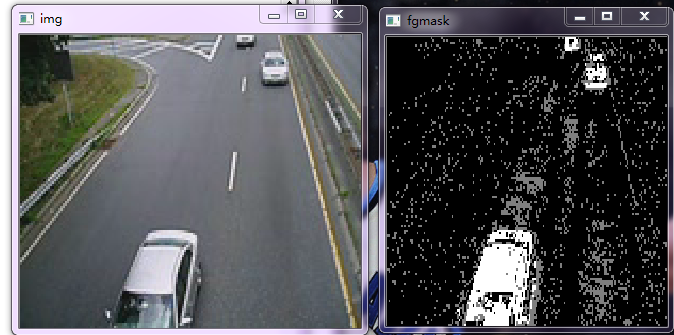
实验中设bShadowDetection = True 即检测阴影，学习率为0.02。实验的结果如下图所示：



**图7 第35帧的原图和前景图**



**图8 第60帧的原图和前景图**



**图9 第80帧的原图和前景图**

可以看出该方法也能较好的提取出前景；同时，对阴影也进行了较好的检测，但也产生了很多的阴影噪声；也降低了处理的速度。其能对光照变化引起的场景变化产生更好的适应。

当然影响背景建模的因素还有很多，如：

相机抖动：在一些条件下，风会引起摄像机前后抖动 ；

光照变化：这是最常见的；

动态背景：例如小目标抖动、树叶抖动、水面涟漪；

进入休眠的前景目标：一些前景目标进入场景后静止不动的情况等等；会在接下来的学习工程中去不断的学习和解决。