

Project-5：背景建模

1.实验目的

(1)利用 Scene_Data 文件夹中的视频帧序列实现基于 GMM（高斯混合模型）的背景建模

(2)对实时性进行分析（即每秒能处理多少帧）

2.实验原理

背景中的像素点的值在短时间内变化不大，可以看成是在某一中心点的邻域内上下浮动。当数据点足够多的时候，我们认为背景中像素点的值服从高斯分布。某一时刻像素点的值偏离中心值较远，认为该点属于前景，偏离中心值较近时，认为该点属于背景。

由于现实环境中像素点的值经常受光照变化、风的吹动、图像边缘位置抖动等因素的影响，背景像素点存在多个中心位置，因此用混合高斯模型（GMM）来描述具有多峰特性的分布。

3.实验步骤

Step 1：初始化参数。

参数变量	含义	初始值
C	单模型个数	3
D	方差的阈值系数	2.5
sd	模型标准差	[6,6,6]
mean	模型均值	[第一帧的像素值,0,0]
w	模型权重	[1,0,0]
alpha	学习率	0.01
thresh	阈值	0.75

Step 2：若当前帧的像素值与模型均值的距离小于 D 倍标准差，则表示匹配。如果匹配则更新该模型的权值，均值，方差

$$w_{k,t} = (1 - \alpha)w_{k,t-1} + \alpha$$

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^2$$

其中， $\rho = \alpha * N(X_t | \mu_{t-1}, \sigma_{t-1}^2)$

如果不匹配则更新该模型的权值

$$w_{k,t} = (1 - \alpha)w_{k,t-1}$$

Step 3：如果 step2 中没有任何模式匹配，建立一个新模型，权值为较小初始值，

方差为较大初始值，均值为当前像素值，用新模型替换权重最小的模型。

Step 4: 权值 w 进行归一化处理，对 GMM 的各个单模型按 w/std 从大到小排序

Step 5: 前 B 个模型的权值之和大于阈值 $thresh$ ，那么前 B 个分布为背景模型，剩余的为前景模型。

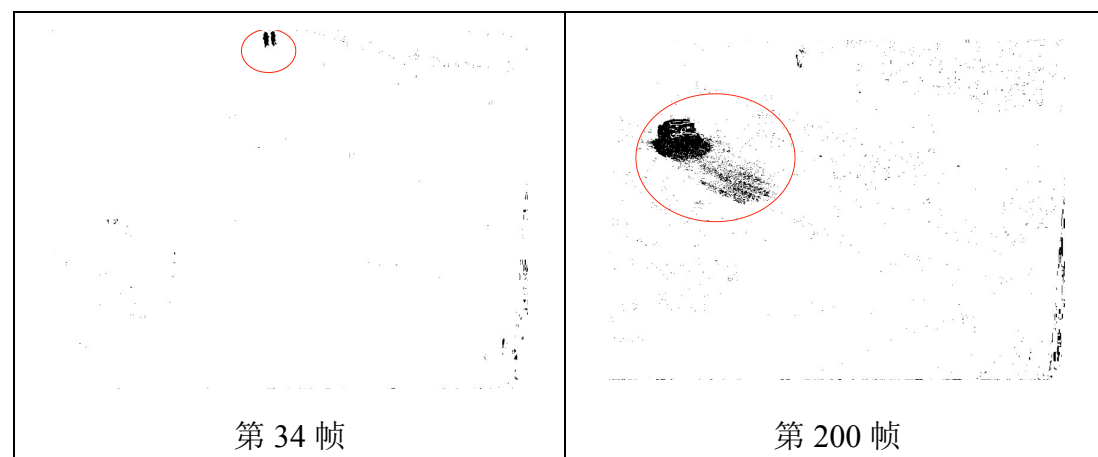
Step 6: 若当前帧像素值与前 B 个模型中均值的距离小于 D 倍标准差，则判断该点是背景，否则为前景。

4.实验结果

参考文献^[1]和 OpenCV 库中 BackgroundSubtractorMOG/MOG2 函数^[2,3]，用 MATLAB 实现了基于高斯混合模型的背景建模算法。实验结果见 result.avi。

视频中可以看到：（1）开始阶段模型还不是很稳定，有几帧只能标记出人的轮廓，后面随着训练数据的增多，算法能将人体全部识别为前景。

（2）虽然能分割出运动的目标和背景，但画面中（特别是右下角）仍存在很多噪声点，尤其当汽车经过时。此外，分割的结果也有留滞现象（前一帧物体所在的位置在当前帧仍被标记为目标），如下图所示。



5.实验分析

（1）与基于直方图的阈值方法进行目标分割不同，前者在整张画面上采用同一阈值，而 GMM 在画面的每个部分都采用不同的分割阈值进行分割，因此更加精确，鲁棒性更强。

（2）实时性分析。测试数据共 200 帧，全部在程序上跑完的时间是 654.776024 秒，相当于每帧的处理时间为需要约 3.27 秒。在算法的实现过程中已经采用了一些手段减少计算量，比如 ρ 的值用 $\alpha/w(I,j,k)$ 来近似。但由于针对每个像素点都要构造 C 个高斯模型，共有四层循环，中间还涉及排序问题，所以复杂度还是很高。

（3）针对结果中出现的当车驶过时噪声变大问题，可以推测其出现原因是，建

筑物作为背景，波动的峰值个数要比其他背景部分的多。由于算法中统一将所有像素点的单个模型个数设为 3，所以当像素点变化超出这 3 个模型的描述范围之后，就将一些背景像素点误识为目标。针对滞留现象，推测出现的原因可能是运动物体的阴影影响。OpenCV 库中 BackgroundSubtractorMOG2 函数用的是自适应高斯混合模型，相对于传统 GMM 模型，增加阴影检测功能，每个像素点的高斯分量个数不固定，能很好地解决这一问题。同时也提高了算法的效率。

6.参考资料

- [1] 《Adaptive background mixture models for real-time tracking》
- [2] [OpenCV 中的混合高斯算法原理](#)
- [3] [基于高斯混合模型的背景建模 BackgroundSubtractorMOG/MOG2](#)