

# 一种基于匹配次数的运动目标检测算法

段胜业, 李俊红, 许文培, 柳 香, 尚志恩

(河北师范大学数学与信息科学学院, 石家庄 050024)

**摘 要:** 运动目标检测中高斯混合模型计算量大、检测实时性较差。为此, 提出一种基于匹配次数的运动目标检测算法。利用高斯混合模型构建背景, 统计每个像素的观测值与背景模型的匹配次数, 根据匹配次数将检测场景分为静态区和动态区, 对静态区的像素点进行隔帧检测, 对动态区的像素点实行逐帧检测, 并结合检测质量和实时性要求研究匹配次数阈值和间隔帧数。实验结果表明, 当静态区面积占整帧面积的 50% 左右时, 该算法室内和室外场景每帧图像的检测时间分别为 27 ms 和 20 ms, 检测效率较高。

**关键词:** 高斯混合模型; 目标检测; 匹配次数; 实时检测; 智能视频监控

## A Moving Object Detection Algorithm Based on Match Time

DUAN Sheng-ye, LI Jun-hong, XU Wen-pei, LIU Xiang, SHANG Zhi-en

(College of Mathematics and Information Sciences, Hebei Normal University, Shijiazhuang 050024, China)

**【Abstract】** The disadvantage of Gaussian Mixture Model(GMM) is that it requires large amount of computation and its poor real-time detection performance in the process of moving object detection. Aiming at this problem, a moving object detection algorithm based on match times is proposed. It constructs background by creating Gaussian models, counts the number of match times of the observed values of each pixel with the background model, according to the number of match times, divides the detected scene into two areas, such as a static area and a dynamic area. It detects every discontinuous frame for static pixels, does every frame detection for dynamic pixels, and is combined with testing quality and the real-time requirements to study the match threshold value and the interval between frames. Experimental results show that when the static zone area is about 50% for the area of the whole frame, the image detection times of this algorithm for each frame in indoor and outdoor scene are 27 ms and 20 ms, and it can improve the efficiency of detection.

**【Key words】** Gaussian Mixture Model(GMM); object detection; match time; real-time detection; intelligent video surveillance

DOI: 10.3969/j.issn.1000-3428.2013.05.042

### 1 概述

运动目标检测常用的方法是背景减除法。文献[1]提出高斯混合模型(Gaussian Mixture Model, GMM)的原理, 并由文献[2-3]发展成为最常用的解决复杂背景问题的方法, 每个像素建立多个高斯模型, 在检测过程中, 动态更新各高斯模型的参数, 用权重与标准差之比较大的几个高斯模型构建该像素的背景。高斯混合模型可以模拟多模态背景, 在一定程度上克服单高斯模型问题, 但尚存在模型初始化速度慢、学习率固定、计算复杂度、丢失空间信息等问题。针对这些问题, 文献[4-5]采用调整模型个数和学习速率的方法提高模型对背景变化的适应能力。文献[6]对高斯混合模型设计了初始化算法, 提高了高斯混合模型的初始

化速度。文献[7]在利用背景差法判决当前帧的前景目标时, 融合当前像素点周围的像素特征综合判决, 提高了目标检测的鲁棒性。文献[8]利用每个像素的匹配次数在线更新学习率的方法, 提高了检测的正确率。文献[4-8]侧重于提高高斯混合模型的检测质量, 对检测效率并没有做明显的改进。文献[9-10]通过自适应高斯模型个数、提高了检测效率, 文献[11-12]通过偏差均值、调整参数的更新等方法提高了检测效率。

现阶段基于高斯混合模型的检测效果已经能达到实际要求, 但是仍存在计算量大、检测实时性差的问题。对此, 本文提出一种基于匹配次数的运动目标检测算法。根据每个像素的历史观测值对后续观测值进行预测, 以达到提速效果进而满足实时性要求。

**基金项目:** 国家自然科学基金资助项目(10926719); 河北省科技厅重点实施科技支撑计划基金资助项目(10243554D)

**作者简介:** 段胜业(1988—), 男, 硕士研究生, 主研方向: 智能视频监控, 嵌入式系统; 李俊红, 副教授; 许文培、柳 香, 硕士研究生; 尚志恩, 高级工程师

**收稿日期:** 2012-05-21 **修回日期:** 2012-08-16 **E-mail:** shengye\_205@163.com

## 2 基于传统高斯混合模型的运动目标检测

传统高斯混合模型认为每个像素点的观测值在时间域上服从高斯分布, 为每个像素点建立  $W$  个高斯模型, 由这  $W$  个高斯模型来构建该像素点的背景。首先用采集到的第 1 帧图像中的各像素点的观测值作为  $W$  个高斯模型的期望, 并为每个高斯模型分配一个比较大的方差和相同的权重。

对读入的一帧图像, 将其中的每个像素的观测值与该像素所对应的  $W$  个高斯模型进行匹配检验<sup>[3]</sup>, 若匹配则对匹配的模型进行参数更新, 若不匹配则进行高斯模型的替换。每次对  $W$  个高斯模型根据权重由大到小进行排序, 并选择其中的前  $B$  个作为背景模型, 若当前像素点与此  $B$  个高斯模型均不匹配, 则将次像素点判为前景点, 否则为背景点。

## 3 基于匹配次数的运动目标检测

在基于背景减除法的运动目标检测算法中往往对每一帧中的每个像素都进行检测, 判断其是否与背景模型匹配, 若匹配则认为是背景, 反之则为前景。但是在实际场景中, 并非整帧图像的每个像素点都有前景频繁出现, 而是有一部分区域很少, 甚至不会出现前景, 如天空、马路旁边的草丛、楼道的墙壁等, 所以, 这部分区域不是目标检测的敏感区域, 称为静态区; 前景经常出现的区域称为动态区。显然每帧都对静态区的所有像素点进行检测是没有必要的。

### 3.1 基本原理

首先利用每个像素与背景模型的匹配次数将一帧中的像素点划分为静态区和动态区, 然后对静态区中的像素点进行隔  $K$  帧检测, 对动态区的像素点进行逐帧检测, 减少对静态区像素点的计算量, 提高检测效率, 最后为保证检测质量, 对边缘进行特殊处理。

划分静态区和动态区: 设  $S$  为在视频序列中同一像素点在每帧中的观测值与背景模型连续匹配的次数。若当前帧中该像素点的观测值与背景模型匹配, 则对  $S$  加 1; 否则将  $S$  置 0, 重新计数。因为静态区极少出现前景, 像素观测值与背景模型能够长时间匹配, 所以  $S$  将累加到一个很大的值; 而动态区频繁出现前景,  $S$  会反复被置 0, 所以, 其值将在一个较小的范围内波动。设  $C$  为划分静态区与动态区的匹配次数阈值。当某像素点的  $S$  大于  $C$  时, 将该像素点划分到静态区, 否则划分到动态区。

设  $K$  为静态区像素点连续 2 次检测所间隔的帧数, 对静态区中的像素点进行每隔  $K$  帧检测一次, 对动态区的检测次数进行逐帧检测, 这样就实现了对静态区域的检测次数减少, 而对动态区的检测次数不变, 大大降低了检测算法的时间复杂度。

为了避免在静态区突然出现前景时漏检前景目标, 采

取如下方案: 摄像头拍摄的图像帧为一个矩形, 将该矩形的上下 2 条边的中点相连, 左右 2 条边的中点相连, 称此 6 条边为图像帧的边缘, 边缘上的像素点也同普通像素点一样进行匹配次数的累加、更新等操作, 同样也被划分到静态区或动态区, 但对边缘像素点进行逐帧检测, 若检测到边缘上的某像素点属于静态区, 但又与背景模型不匹配, 则认为静态区将会有前景出现, 尤其是该像素的邻域, 所以, 对该帧进行逐像素检测, 并将其  $N \times N$  邻域的像素点的匹配次数置 0, 确保不会漏检前景目标。这样就保证了在前景突然进入静态区时能够及时检测出来。

### 3.2 算法描述

算法的描述如下:

(1) 参数设定:  $match[i]$  表示第  $i$  个像素与背景连续匹配的次数;  $interval[i]$  表示第  $i$  个像素从最近一次检测以来, 没有检测的帧数;  $m$  表示动态区像素点的匹配次数的最大值。

(2) 目标检测流程如图 1 所示。

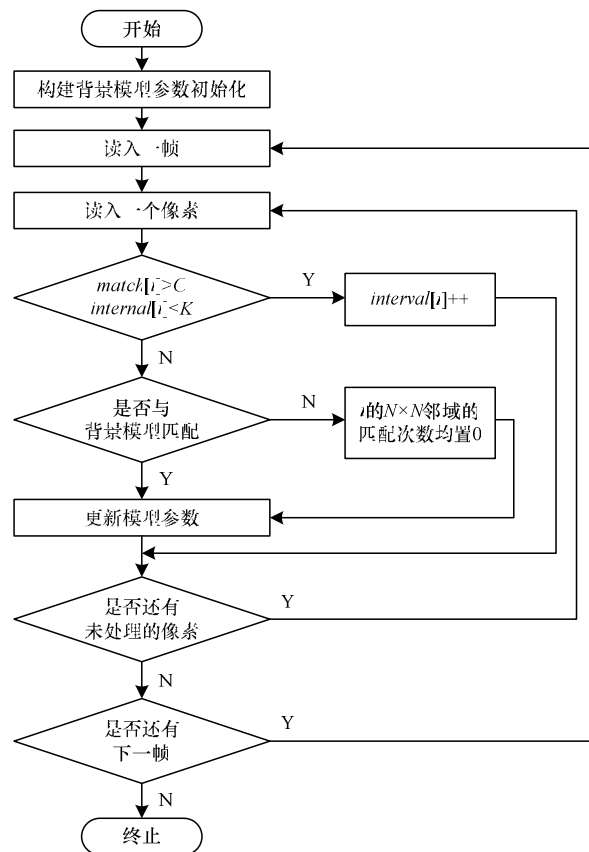


图1 目标检测流程

在图 1 中各步骤具体内容如下:

**Step1** 构造背景模型, 假设初始化参数  $C = m$ ,  $match[i] = 0$ ,  $interval[i] = K$ 。

**Step2** 读入一个像素值, 判断是否匹配次数大于阈值  $C$ , 同时间隔帧数小于  $K$ , 若是则转 Step3, 否则转 Step4。

**Step3** 不检测此像素点, 并将间隔帧数  $interval[i]$  加 1, 转 Step7。

**Step4** 进行高斯混合模型的匹配检验<sup>[8]</sup>, 如果与背景匹配则将匹配次数  $match[i]$  加 1, 并转 Step5, 否则转 Step6。

**Step5** 进行高斯混合模型的参数更新<sup>[8]</sup>。

**Step6** 如果有前景出现, 将该像素与其  $N \times N$  邻域内所有像素点的匹配次数都置为 0, 重新计数, 这是因为一个像素点有前景出现, 则其邻域内也很可能有前景出现, 所以对其邻域进行逐像素检测, 并判断该点是否为边缘像素点, 若是, 则对该帧的所有像素点进行检测, 防止漏检。

**Step7** 判断是否还有其他像素, 若有转至 Step2, 否则执行 Step8。

**Step8** 判断是否还有下一帧, 若有转至 Step2, 否则算法终止。

### 3.3 参数设置

#### 3.3.1 间隔帧数 $K$ 的设置

设静态区占总场景的比例为  $r$ , 混和高斯模型处理每帧所需时间为  $T$ , 则有:

$$\begin{cases} T_B = \frac{r}{K}T + (1-r)T \\ \Delta T = T(r - \frac{r}{K}) \\ H = \frac{\Delta T}{T} = r(1 - \frac{1}{K}) \end{cases} \quad (1)$$

其中,  $T_B$  表示 BMTD 算法所用时间;  $\Delta T$  表示 BMTD 算法节省的时间;  $H$  表示加速比。

由式(1)可以看出,  $H$  与  $r$ 、 $K$  有关, 在一个场景中  $r$  是确定的, 所以,  $H$  主要取决于  $K$ 。间隔帧数与加速比的关系如图 2 所示, 其中,  $r=1/3$ 。

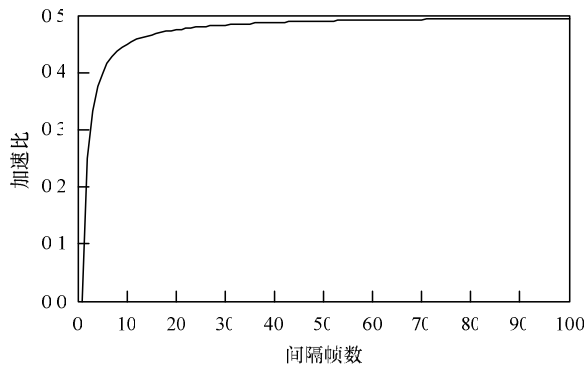


图 2 间隔帧数与加速比的关系

由图 2 可知, 当  $r$  一定时,  $H$  随  $K$  的增大而增大, 即在确定的场景中, 加速比  $H$  随  $K$  的增大而增大。当  $K$  增大到一定程度后, 加速比的变化不再明显, 下面以实际数据进行说明。

设处理每帧图像所用时间为  $T$ , 静态区占总场景的比例  $r = \frac{1}{2}$ , 则有:

(1) 当  $K=100$  时:

$$H = \frac{\Delta T}{T} = r(1 - \frac{1}{K}) = 49.5\% \quad (2)$$

(2) 当  $K=1000$  时:

$$H = \frac{\Delta T}{T} = r(1 - \frac{1}{K}) = 49.95\% \quad (3)$$

由式(2)、式(3)可知  $K=1000$  与  $K=100$  相比, 加速效果并无明显提高, 由式(2)计算可知,  $r$  越小,  $H$  对  $K$  越不敏感, 且一般场景下  $r < \frac{1}{2}$ 。所以, 在大部分场景中  $K$  取 100 时基本接近了加速极限, 本文取  $K=100$ 。

#### 3.3.2 阈值 $C$ 的设置

$C$  为划分某个像素属于静态区或动态区的阈值。  $C$  值过小, 会将部分动态区像素点划分到静态区, 可能会导致对前景的漏检, 所以, 应该确保  $C$  值大于动态区任何像素点的匹配次数, 从而保证检测质量。经过长时间对某固定场景进行监控, 静态区像素点的匹配次数会变得很大, 所以,  $C$  值设的较大不会影响静态区的像素点。

设  $S$  为在视频序列中, 同一像素点在每帧的观测值与背景模型连续匹配的次数, 经过长时间的检测, 静态区像素点的  $S$  将会累加到非常大的一个值, 而动态区像素点的  $S$  将在一个范围内波动, 设动态区像素点的匹配次数的峰值为  $m$ , 那么  $C$  应大于  $m$ 。设  $m$  为监控场景中未出现目标的最大持续时间(单位为 s),  $time$  值与场景有关, 可以根据先验知识获得, 假定摄像头每秒拍摄 25 帧, 得  $m = time \times 25$ , 从而得出  $C$  的值。

综上所述, 合理的  $C$  值可以准确地将静态区与动态区分开, 从而实现只对静态区进行隔帧检测, 减少检测的像素数。

## 4 实验结果及分析

### 4.1 实验数据

为了验证本文算法的有效性, 分别对室内、室外场景进行了测试, 测试的硬件平台为: CPU 为 Intel™ Core™2 Quad CPU Q9400 @ 2.66 GHz, 内存为 4 GB 的 IBM PC 机; 软件平台为: Windows XP Professional SP3 操作系统, Visual Studio 2008 开发平台。

室内场景的目标检测结果如图 3 所示, 其中, 图片格式为 AVI 格式, 320×240 像素, 25 f/s。图 4~图 6 分别是对室外场景动态背景下运动目标检测的结果对比, 其中, 图片格式为 AVI 格式, 320×240 像素, 25 f/s。

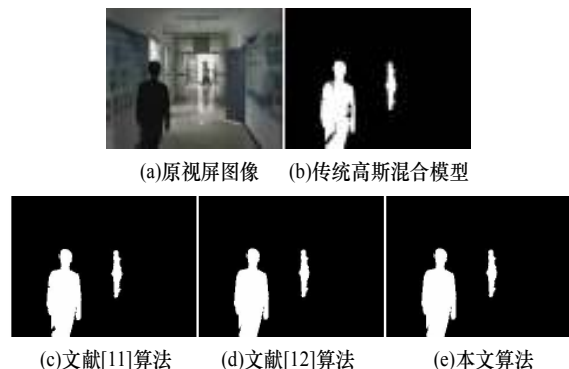


图 3 室内场景的目标检测结果

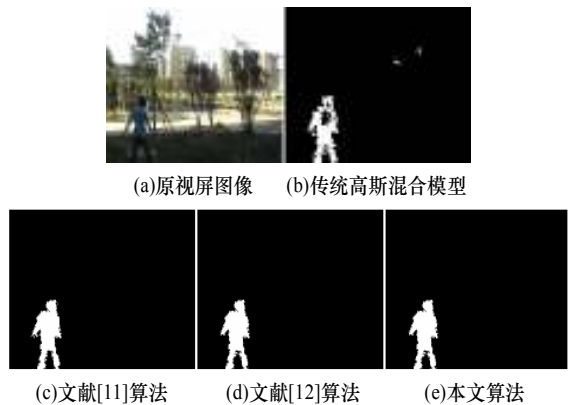


图 4 单个行人室外场景动态背景目标检测结果

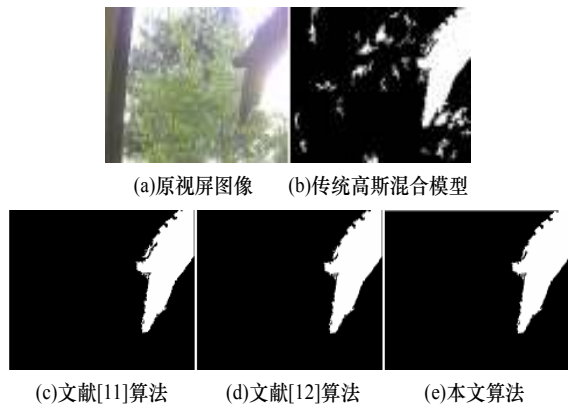


图 5 晃动手掌室外场景动态背景目标检测结果

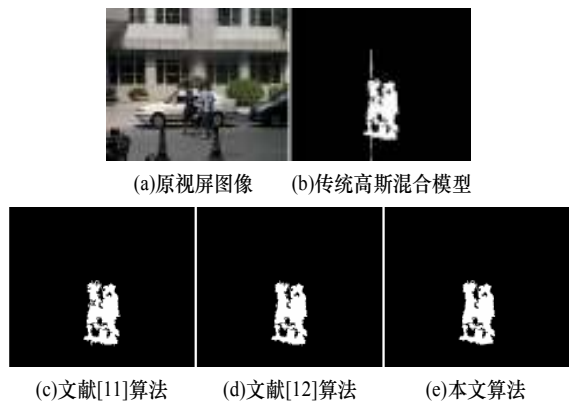


图 6 室外场景动态背景多目标检测结果

4.2 检测效果对比

对室内场景进行检测时, 算法中设置的参数为:  $C=18\,000$ ;  $K=100$ 。由于室内背景相对比较固定, 除了灯光变化的条件外, 背景不会出现大幅度波动, 因此背景比较容易获取, 图 3 是检测的效果图, 由效果图可以看出, 本文算法能够较好地将运动目标检测出来。

图 4~图 6 是室外检测的效果图, 选择动态背景的场景, 场景中有摇动的树叶, 算法的参数设置为:  $C=54\,000$ ;  $K=100$ 。可以看出, 在室外条件下本文算法也能够较好地将背景与前景区别开来。

4.3 检测效率对比

在视频中静态区面积占整帧面积的 50%左右时, 室内、室外场景每帧图像花费检测时间对比如表 1 所示。

表 1 室内、室外场景每帧图像的检测时间对比 ms

| 场景   | 每帧图像的检测时间 |          |          |      |
|------|-----------|----------|----------|------|
|      | 传统高斯      | 文献[11]算法 | 文献[12]算法 | 本文算法 |
| 室内场景 | 45        | 38.0     | 37.6     | 27   |
| 室外场景 | 47        | 36.5     | 35.2     | 20   |

由表 1 可知, 文献[11]、文献[12]及本文算法均在原有的基础上提高了检测效率, 由式(1)计算可得, 本文算法在室内、室外场景下的加速比分别为:  $H_1=0.4$ 、 $H_2=0.57$ 。

因为本文算法以减少对静态区像素点的检测次数来提高检测速度, 所以加速比因场景的不同而异。在整个场景中, 静态区占的比例越大, 每帧检测的像素点越少, 加速效果就越明显, 所以, 该算法更适合于长时间对某场景进行检测的情况, 因为检测时间越长算法对静态区和动态区的划分就越明朗, 各参数的设定就更合理。这样既保证了动态区的检测质量, 又提高了对静态区的检测速度。

5 结束语

本文提出一种基于匹配次数的运动目标检测算法。分析传统以及一些改进高斯混合模型算法的建立、参数的更新以及目标检测的方法, 结合实际场景中存在的静态区和动态区的特点, 设计隔帧检测方法。实验结果表明, 该算法在保证检测效果的前提下, 能显著提高检测速度, 满足视频监控的实时性要求。今后将考虑将其与大部分基于背景减除的运动目标检测算法相结合, 进一步提高检测效率。

参考文献

[1] Friedman N, Russell S. Image Segmentation in Video Sequences: A Probabilistic Approach[C]//Proc. of the 13th Conference on Uncertainty in Artificial Intelligence. Providence, USA: [s. n.], 1997.

[2] Stauffer C, Grimson W E L. Adaptive Background Mixture Models for Real-time Tracking[C]//Proc. of IEEE Conference on Computer Vision and Pattern Recognition. Fort Collins, USA: [s. n.], 1999.

[3] Grimson W, Stauffer C, Romano R. Using Adaptive Tracking to Classify and Monitor Activities in a Site[C]//Proc. of IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Computer Society, 1998.

[4] Shyang L D. Effective Gaussian Mixture Learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(5): 827-832.

[5] Zivkovic Z. Recursive Unsupervised Learning of Finite Mixture Models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(5): 651-656.