

基于 HSV 颜色空间和码本模型的运动目标检测

张 军, 代科学, 李国辉

(国防科技大学信息系统与管理学院系统工程系, 湖南长沙 410073)

摘要: 码本模型的思想就是根据像素的颜色失真程度和亮度范围将背景像素值量化后用码本表示, 然后利用减背景思想对新输入像素值与其对应像素码本做比较判断, 从而提取出前景运动目标像素。提出将像素从 RGB 空间转换到 HSV 空间来计算颜色失真度, 多种像素亮度和颜色失真度计算方法的对比实验结果表明, 该方法能取得更好的目标检测效果。

关键词: 减背景; 码本; 监控视频; 目标检测

中图分类号: TP 391 **文献标志码:** A

HSV color-space and codebook model based moving objects detection

ZHANG Jun, DAI Ke-xue, LI Guo-hui

(Dept. of System Engineering, School of Information System and Management,
National Univ. of Defense Technology, Changsha 410073, China)

Abstract: The codebook model represents each background pixel with some codeword based on its color distortion and brightness range. Then input pixel values of new frame are compared with the codebooks for identifying foreground pixels. Pixels are converted from RGB space to HSV space to compute pixel color distortion. Several methods of computing color distortion and brightness are experimented for contrast. The results show the new idea achieves better effectiveness for detecting moving objects.

Keywords: background subtraction; codebook; surveillance video; object detection

0 引言

在智能视频监控中, 提取运动目标的一个较为普遍的方法是减背景技术。该技术的思想是将每个输入视频帧和一个背景模型做比较, 如果同位置的像素特征、像素区域特征或其他特征存在一定程度的差别, 则新视频帧中这些位置的像素点或像素区域就构成前景运动目标区域。对这些前景像素点作进一步处理, 即可得到运动目标位置、大小、形状等信息, 以便实现目标跟踪和视频内容理解。

显然, 建立好的背景模型是关键。虽然已有很多关于减背景算法的文献^[1], 但远还没有完全解决复杂环境中的运动目标检测。主要是由于不论在室内还是在室外环境, 随时都有很多引起背景变化的情况产生, 比如光照变化、背景扰动、以及运动目标自身引起的背景变化等。

由于简单的差分法^[2]和均值滤波法^[3]有着占内存大、自适应能力差等缺点, 一些背景模型假设一个像素的亮度变化值可由一个单峰分布模型建模^[4,5]。但是单模态模型不能对含摇动的树枝等背景建模。混合高斯模型虽能处理

复杂、非静止的多模态背景^[6], 但它不能适应快速的背景变化, 因而不适合敏感性内容检测。为此人们提出非参数技术^[7], 用内核密度估计的方法估计每个像素处的背景概率。但是非参数技术需要大量内存来存储先前的视频数据, 比中值滤波还耗时, 且不能解决背景中有较长时间的周期运动。这些方法都基于独立的单个像素, 当然也有利用区域或帧特征的方法^[8,9]。

码本模型是 Kim 等提出的一种新颖的运动目标检测思想^[10]。该模型针对彩色监控视频, 根据像素点的连续采样值的颜色失真程度及其亮度范围将背景像素用一个或多个码本表示, 然后利用减背景思想对新输入像素值与其对应码本做比较判断, 从而提取出前景运动目标像素。它既不同于前述的差分和均值滤波等方法, 被比较的背景模型值不只有一个, 也不同于混合高斯和非参估计等方法, 不用计算概率分布。

考虑到 RGB 颜色空间与人眼感知的差异很大, 本文将像素从 Kim 等采用的 RGB 空间转换到 HSV 空间进行颜色失真比较, 并在实验中以多种方法计算像素亮度和颜色失

收稿日期: 2007-03-21; 修回日期: 2007-05-29.

作者简介: 张军(1975-), 女, 副教授, 博士研究生, 主要研究方向为多媒体信息系统, 数字图像处理, 信号处理。E-mail: aiyao215@263.net
(C)1994-2020 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

真度,进行目标检测效果的对比评价。结果表明,虽然由于转换到 HSV 颜色空间引入了浮点运算,使得处理速度比 Kim 等采用的方法稍有下降,但能取得更为准确的目标检测效果。本文的实验还表明,亮度范围的确定方法对目标检测效果的影响也比较大,但码本模型总体上比混合高斯模型更具优势,可以在不同监控视频应用中取得比之更好的检测效果。

1 码本模型描述

1.1 构建像素码本

假设 $\chi = \{x_1, x_2, \dots, x_N\}$ 是一个像素的序列采样值,其中 $x_t (t=1, \dots, N)$ 是 RGB 向量。并设 $C = \{c_1, c_2, \dots, c_L\}$ 是该像素的码本,每个码字 $c_i (i=1, \dots, L)$ 定义为二元组结构

$$y_i = (\overline{R_i}, \overline{G_i}, \overline{B_i}), u_i = \langle I_i^{\min}, I_i^{\max}, f_i, \lambda_i, p_i, q_i \rangle$$

式中, I^{\min} 和 I^{\max} 记录码字对应像素的最小和最大亮度值; f 表示该码字出现的次数; λ 表示码字在训练中没有再次出现的最大时间间隔; p 和 q 分别表示该码字出现后的第一次和最后一次匹配时间,可简单设为帧序数。

在构建码本的训练期间,将 t 时刻的像素点 x_t 的采样值与当前的码本做比较,如果有码字 $c_m (m$ 是某个码字的索引)与其匹配,则以码字 c_m 作为该采样点的编码近似值,表示该码字出现一次。由于匹配的码字可能有多个,算法根据颜色失真度和亮度范围来决定哪个码字匹配得最好。下面是码本提取算法的详细过程。

步骤 1 首先将每个像素的码本置空, $L=0$ 。

步骤 2 对于训练视频的每个像素的序列值 $\chi = \{x_1, x_2, \dots, x_N\}$, $x_t = (R_t, G_t, B_t)$, $t=1, \dots, N$;

(1) 如果码本为空, $L=0$, 则创建一个码字

$$L = L + 1, I = \sqrt{R_t^2 + G_t^2 + B_t^2} \quad (1)$$

$$y_L = (R_t, G_t, B_t), u_L = \langle I, I, 1, t-1, t, t \rangle \quad (2)$$

(2) 如果码本不为空,则根据下面两个条件从码本中找出与 x 匹配的码字 c_m

$$\text{colordist}(x_t, y_m) \leq \epsilon_1 \quad (3)$$

$$\text{brightness}(I, \langle I_m^{\min}, I_m^{\max} \rangle) = \text{true} \quad (4)$$

①如果找不到,即码本中没有码字满足上述条件,则利用(1)、(2)式为该像素创建一个码字。

②如果有码字 c_m 满足上述条件,内容是

$$y_m = (\overline{R_m}, \overline{G_m}, \overline{B_m}),$$

$$u_m = \langle I_m^{\min}, I_m^{\max}, f_m, \lambda_m, p_m, q_m \rangle$$

那么将该码字做如下更新

$$\begin{aligned} y_m &= ((f_m \overline{R_m} + R_t) / (f_m + 1), \\ & (f_m \overline{G_m} + G_t) / (f_m + 1), \\ & (f_m \overline{B_m} + B_t) / (f_m + 1)) \end{aligned} \quad (5)$$

$$u_m = \langle \min \{I, I_m^{\min}\}, \max \{I, I_m^{\max}\}, f_m + 1, \max \{\lambda_m, t - q_m\}, p_m, t \rangle \quad (6)$$

步骤 3 训练结束后,计算该像素每个码字没有再次出现的最大时间间隔,即对于 $c_i, i=1, \dots, L$

$$\lambda_i = \max \{\lambda_i, (N - q_i + p_i - 1)\} \quad (7)$$

步骤 4 然后利用 λ 消除冗余码字,得到最能代表真实背景的精练的初始码本 $M (k$ 是码字的索引)

$$M = \{c_k \mid c_k \in C, \lambda_k \leq T_M\} \quad (8)$$

其中, ϵ_1 是全局阈值变量,需针对具体应用做合适调整;阈值 T_M 通常取训练帧数的一半,即 $N/2$,表示所有代表背景的码字必须至少在 $N/2$ 帧中出现。

步骤 2(2) 中两个条件的满足情况是 x_t 和 c_m 的颜色非常近似,并且 x_t 的亮度在 c_m 的可接受亮度范围内。这里只要求找到第一个满足这两个条件的码字即可,参见下一节。

之所以引入时间准则 λ 是因为训练过程中得到的码本存在冗余,其中可能有一些表示前景运动目标和噪声的码字,利用公式(8)即可将这些码字从概率意义上分离出去,从而允许初始训练过程中存在运动目标。

1.2 确定颜色失真度和亮度范围

为了解决阴影和高亮等光照变化,人们常用规格化的 RGB 颜色模型。这对于阴暗区域的颜色比较效果不好,阴暗区域比亮区域的像素差别具有更高的不稳定性。比如代表阴暗的像素 $\langle 10, 10, 10 \rangle$ 和 $\langle 9, 10, 11 \rangle$,与代表较亮的像素 $\langle 200, 200, 200 \rangle$ 和 $\langle 199, 200, 201 \rangle$,它们的颜色失真度分别为: $(|10-9| + |10-10| + |10-11|) / 30 = 2/30$; $(|200-199| + |200-200| + |200-201|) / 200 = 2/200$ 。可见,在低亮度情况下颜色变化比较大,从而会牺牲高亮度情况的检测敏感度。详细讨论可参见文献[11]。

我们知道,RGB 颜色空间与人眼感知的差异很大,而 HSV (hue, saturation, value) 颜色空间则较好符合人眼感知特性,在基于内容的视频图象检索中有较为广泛的应用。通过 HSV 颜色空间,人眼可以独立感知各颜色分量的变化,可感知的颜色差与颜色分量的欧几里德距离成正比。因此本文将像素从 RGB 转换到 HSV 空间进行颜色失真度计算,以便更加灵敏地判断前景像素与背景之间的变化。

设函数 $RGB2HSV()$ 实现将像素 RGB 值转换为 HSV 值(这里不做赘述), h, s, v 是像素在 HSV 颜色模型中的 3 个分量, $h \in [0, 2\pi], s \in [0, 1], v \in [0, 1]$ 。对于像素 $x_t = (R_t, G_t, B_t)$ 及 $y_m = (\overline{R_m}, \overline{G_m}, \overline{B_m})$, 则

$$\begin{aligned} p_t &= RGB2HSV(x_t), p_m = RGB2HSV(y_m) \\ \text{colordist}(x_t, y_m) &= \\ & (p_t \cdot s^* \sin(p_t \cdot h) - p_m \cdot s^* \sin(p_m \cdot h))^2 + \\ & (p_t \cdot s^* \cos(p_t \cdot h) - p_m \cdot s^* \cos(p_m \cdot h))^2 + \\ & (p_t \cdot v - p_m \cdot v)^2 \end{aligned} \quad (9)$$

为了界定阴暗和高亮区域,运动目标检测中的亮度变化应有个范围,从后面的实验部分可见这一步的作用。对于每个码字,其范围定义为 $[I_{low}, I_{hi}]$

$$I_{low} = \alpha I^{max}, I_{hi} = \min \{ \beta I^{max}, I^{min} / \alpha \} \quad (10)$$

其中 $\alpha < 1, \beta > 1$ 。典型地, $0.4 < \alpha < 0.7$, 越小定义的亮度范围越大; $1.1 < \beta < 1.5$, 这个范围在码本更新过程中比较稳定。1.1 节中的亮度函数是如下定义的

$$brightness(I, \langle I^{min}, I^{max} \rangle) = \begin{cases} true, & I_{low} \leq \|x_t\| \leq I_{hi} \\ false, & \text{其他} \end{cases} \quad (11)$$

1.3 利用码本检测前景运动目标

对于减背景运动目标检测技术, 最直接的方法是将当前帧与背景模型相减。码本模型根据像素采样值与其码本是否匹配来判断。对于运动目标检测过程中的新输入像素 $x=(R, G, B)$ 及其对应的码本 M , 减背景操作 $BGS(x)$ 定义为:

(1) 计算亮度 $I = \sqrt{R^2 + G^2 + B^2}$, 定义布尔变量 $matched=0$, 并给阈值变量 ϵ_2 赋值;

(2) 根据下面式(12)、(13)两个条件从其码本 M 中找出与 x 匹配的码字 c_m , 如果找到则 $matched=1$

$$\omega lordist(x, y_m) \leq \epsilon_2 \quad (12)$$

$$brightness(I, \langle I_m^{min}, I_m^{max} \rangle) = true \quad (13)$$

(3) 判断前景运动目标像素

$$BGS(x) = \begin{cases} x \text{ 为前景像素, } matched = 0 \\ x \text{ 为背景像素, } matched = 1 \end{cases} \quad (14)$$

也就是将满足这样两个条件的像素判断为背景:

- (1) 像素与某个码字的颜色失真度小于检测门限;
- (2) 像素的亮度在该码字的亮度范围内。

否则被认为是前景像素, 无码字与其匹配。这里 ϵ_2 比 ϵ_1 稍大, 需针对具体应用做合适调整。

1.4 目标检测过程的码本更新

为了解决视频监控过程中的光照变化(如开关灯或乌云)和运动目标本身(如停下或驶离的车)引起的背景变化, 目标检测过程中的码本更新采用如下策略。

缓存精练后的背景码本 M 为 M' , 如果一个像素的采样值不与已有码本 M 匹配, 则为其创建一个新的码字到 M' 中, 并且新码字的周期由时限 T_M 过滤。然后, 把重新出现次数超过 T_{add} 的码字增加进 M 中, 把超过时间 T_{del} 都还没被匹配访问的码字从 M 中删除。具体过程是:

(1) 对于新输入像素 $x=(R, G, B)$, 从 M 中找出与之匹配的码字 c_m , 按如下公式更新其向量

$$y_m = ((1-\gamma)f_m \overline{R_m} + \gamma R) / (f_m + 1), ((1-\gamma)f_m \overline{G_m} + \gamma G) / (f_m + 1), ((1-\gamma)f_m \overline{B_m} + \gamma B) / (f_m + 1) \quad (15)$$

(2) 如果在 M 中没有匹配的码字, 则在 M' 中进行查找, 如果找到也按式(15)更新; 如果也找不到, 则创建一个新的码字 c' 添加到 M' 。

(3) 根据 T_M 精练 M' , 即

$$M' = M' - \{c'_k \mid c'_k \in M', \lambda_k > T_M\} \quad (16)$$

(4) 将在 M' 中停留足够时间的码字移到 M 中, 即

$$M = M + \{c'_k \mid c'_k \in M', f_k > T_{add}\} \quad (17)$$

(5) 从 M 中删除超过一定时间未被匹配的码字, 即

$$M = M - \{c_k \mid c_k \in M, \lambda_k > T_{del}\} \quad (18)$$

其中, γ 表示码字向量对像素值变化的学习率, 跟 f 一起用于解决背景扰动和光照变化。

2 实验与结果分析

2.1 多种颜色失真度计算方法及其效果

从上述可见, 像素亮度的计算公式采用的是 $I = \sqrt{R^2 + G^2 + B^2}$ (式 1), 而不是通常的 RGB 颜色空间的亮度计算方法(式 19)。

$$I = 0.299R + 0.587G + 0.114B \quad (19)$$

对于像素 $x_t=(R_t, G_t, B_t)$ 、码字 c_m 及其分量 $y_m=(\overline{R_m}, \overline{G_m}, \overline{B_m})$, 分别以这两种亮度计算方法进行实验。在计算颜色失真度时, 除了本文方法外, 还分别以下述三种方法进行实验。以图 1 所示场景(帧率 25 fps)的目标检测为例, 在未做形态学等后处理的情况下, 分别取得如组图 2 所示的效果。



图 1 颜色失真实验的参考场景

(1) 亮度差值方法, 即

$$\omega lordist(x_t, y_m) = fabs(0.299 * (R_t - \overline{R_m}) + 0.587 * (G_t - \overline{G_m}) + 0.114 * (B_t - \overline{B_m})) \quad (20)$$

(2) 文献[12]给出的方法, 即

$$\omega lordist(x_t, y_m) = sqrt(4 * (R_t - \overline{R_m})^2 + 8 * (G_t - \overline{G_m})^2 + (B_t - \overline{B_m})^2) \quad (21)$$

(3) Kim 等在文献[10]给出的方法, 即

$$\|x_t\|^2 = R_t^2 + G_t^2 + B_t^2 \quad (22)$$

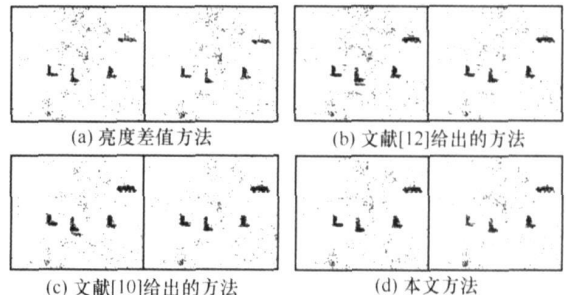
$$\|y_m\|^2 = \overline{R_m}^2 + \overline{G_m}^2 + \overline{B_m}^2 \quad (23)$$

$$\langle x_t, y_m \rangle^2 = (\overline{R_m} R_t + \overline{G_m} G_t + \overline{B_m} B_t)^2 \quad (24)$$

$$\eta^2 = \|x_t\|^2 \cos^2 \theta = \frac{\langle x_t, y_m \rangle^2}{\|y_m\|^2} \quad (25)$$

$$\delta = \|x_t\|^2 - \eta^2 \quad (26)$$

$$\omega lordist(x_t, y_m) = \sqrt{\delta} \quad (27)$$



注: 每组图的左边以式(1)计算亮度, 右边以式(19)计算亮度

图 2 不同颜色失真计算方法的目标检测效果

实验时, ϵ_1 和 ϵ_2 的取值和模型在建模与检测时的处理速度分别如表 1 和表 2(单位为 fps , 即每秒处理的帧数), 模型的其他参数设置如表 3 所示。

表 1 以式(1)计算亮度时, 码本模型的 ϵ_1 和 ϵ_2 取值及其建模与检测速度

	ϵ_1	ϵ_2	训练	检测
(a)	25	30	15	22.8
(b)	50	90	18	22.7
(c)	6	20	16	22.6
本文方法	0.15	0.2	13	16.8

表 2 以式(19)计算亮度时, 码本模型的 ϵ_1 和 ϵ_2 取值及其建模与检测速度

	ϵ_1	ϵ_2	训练	检测
(a)	25	30	18	22.8
(b)	50	90	18	22.7
(c)	12	20	18	22.8
本文方法	0.15	0.2	13	15.2

表 3 码本模型的其他参数设置

α	β	γ	N
0.7	1.1	0.2	50

任何一个参数取值有变化, 算法取得的实验效果都是不一样的。参数 ϵ_1 和 ϵ_2 是通过将函数 $colordist(x_i, y_m)$ 的部分值写成文本文件, 然后用 Matlab 绘图, 通过查看数值分布, 经多次实验评估调整到合适值的。例如图 3 所示的函数值分布图, 则将 ϵ_1 和 ϵ_2 分别取为 0.15 和 0.2, 会得到比较好的目标检测效果。

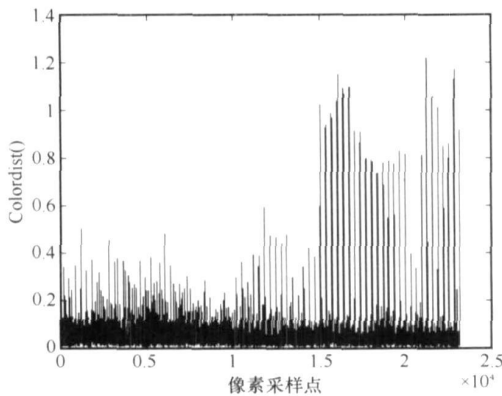


图 3 像素颜色失真分布图

文献[12]给出的方法主要是解决 RGB 颜色系统在视觉上的不均匀性, 系数 4、8、1 是其经典值, 而文献[10]的方法则是将码本向量规格化为输入像素的亮度值, 从而使算法在整个亮度范围内保持恒定的错检率。从图 2 可见, 总体上, 以式(19)计算亮度(图 2 中的右半图)要比以式(1)计算亮度(图 2 中的左半图)的目标检测效果更精确些, 后者容易把非运动目标检测出来(参见图 2(b)和 2(c)的左半图), 亮度差值方法效果最差, 检测出来的目标体积最小且杂物(摇动的树叶)最多, 图 2(b)就比图 2(a)体现的前景目标较充实, 但比图 2(c)和图 2(d)也体现了更多的杂物。

从图 2(d)可见, 由于在 HSV 颜色空间中计算的颜色

相异度/失真度较符合人眼感知特性, 能够很好区分前景运动目标像素与码本像素之间的相似性, 本文所述的基于 HSV 颜色空间的失真度计算方法建立的码本背景最准确, 确实可以取得比其他方法更为真实和更为充实的目标检测效果。图 2(d)中的运动目标就比图 2(c)的更充实, 目标边界更清晰、锯齿更少。

从表 1 和表 2 记录的建模与检测速度可见, 以式(1)计算亮度的方法比以式(19)计算亮度的方法要快 2 帧左右的处理速度, 说明浮点运算在视频处理中的速度影响比较大, 也因此本文采用式(1)计算亮度。同时, 虽然本文方法的效果比其他的好, 但其处理速度最慢, 涉及浮点运算是一个原因, 受阈值的影响会生成很多码字从而耗费匹配时间和内存则是其另一个主要原因。后者跟难以找到参数 ϵ_1 和 ϵ_2 、 α 和 β 的最佳值有关, 因为本文所述方法得到的颜色失真度值的范围比较小, 而其他方法的颜色失真度值比之大 10 倍甚至几十倍, 颜色失真度的范围也比较大, 因而阈值更容易调整。

2.2 亮度范围实验及其效果

本文还改变亮度范围的计算方法进行实验, 以观察目标检测的效果。即在比较亮度范围时不采用式(11), 而直接以码本记录的最大和最小亮度值为范围(式 28)做判断。基于本文所述的 HSV 颜色失真度计算方法, 针对图 4 所示的监控场景, 效果差异如图 5 所示。从中可见检测效果大为下降, 采用式(11)获得的目标检测效果较好(图 5 的右半图), 说明参数 α 和 β 对目标检测效果的影响也非常的大。

$$brightness(I, \langle I^{\min}, I^{\max} \rangle) = \begin{cases} true, & I^{\min} \leq \|x\| \leq I^{\max} \\ false, & \text{其他} \end{cases} \quad (28)$$

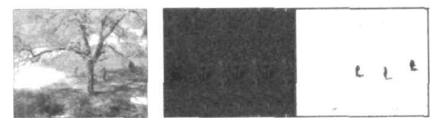


图 4 亮度范围实验的参考场景

图 5 改变亮度范围确定方法的效果差异

2.3 码本模型与混合高斯模型的性能对比实验

为了观察码本模型的性能特点, 基于本文所述的 HSV 颜色空间的失真度计算方法和 Kim 等提出的颜色失真度计算方法, 将其与典型的多模态背景建模技术混合高斯模型^[5]做比较。实验分别包括一组室外监控视频和一组室内监控视频, 在仔细调整码本模型中的参数 ϵ_1 和 ϵ_2 、 α 和 β 后, 经过简单的形态学处理结果如图 6 所示。图 6(a)和图 6(b)分别是室外小目标和室内大目标检测的对比实验结果。可以看出, 不管目标大小和速度, 本文算法都取得了比之更好的效果。

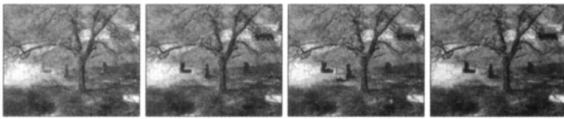
这两个运动目标检测算法的共同点是: 在背景建模过程中都允许运动目标存在, 都能处理彩色视频, 并且都不存储先前的像素数值, 所需内存比较小。同时这两个模型的

参数都比较难调。而相比之下, 码本模型具有以下优点:

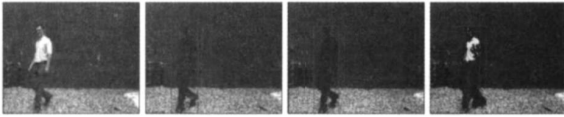
(1) 与混合高斯模型不同, 码本模型并没有假定背景是多模态高斯分布的。如果背景是多模态高斯分布的, 混合高斯模型能取得准确的参数和性能。但实际上这个假设并不一定成立, 背景分布不一定就是正态的。

(2) 码本模型能够处理多个背景的情况, 背景的个数不受限制, 能把一些准周期出现的稀疏像素当作背景。而理论上混合高斯模型能处理的背景情况受模型个数限制。

(3) 混合高斯模型直接利用 RGB 值, 没有分别对亮度和颜色建模, 对大而慢的目标检测效果不好(如图(b)所示), 特别是会将纹理少或对比度低的目标当作背景^[13]。码本模型同时计算亮度差和颜色差, 能有效处理阴暗和高亮区域。



(a) 依次为原始画面、本文算法、Kim方法和混合高斯模型的目标检测效果(原始帧率25 fps, 处理速度分别为11.2, 21.4, 17.1 fps)



(b) 依次为原始画面、本文算法、Kim方法和混合高斯模型的目标检测效果(原始帧率15 fps, 处理速度分别为9.5, 14.8, 14.2 fps)

注: 码本模型参数设置: $\alpha=0.7$, $\beta=1.1$, 室内训练帧数 $N=30$, 室外为50。本文算法的 $\epsilon_1=0.15$, $\epsilon_2=0.2$, Kim方法的 $\epsilon_1=6$, $\epsilon_2=20$; 混合高斯法的参数设置: 模型个数3, 初始权重0.02, 权重学习率0.005, 初始偏差12。

图6 码本模型与混合高斯模型的对比实验结果

3 结束语

码本模型能够通过简单训练得出有效的背景模型。多个视频的实验显示, 对于25帧每秒的室外视频, 每个像素平均只需要6个码字就能获取到背景模型, 而室内视频更少, 一两个背景值就够了。实验同时表明, 算法只需要训练一次, 重复训练不会提高检测能力, α 和 β 也会影响算法的速度, 利用 λ 能区分表示真实背景的码字和表示前景运动的码字, 但同时考虑 f 和 λ 与只考虑 λ 得到的性能差不多。

除了可限定内存使用的大小, 码本模型具有以下特点:

(1) 能处理局部或全局的光线变化;

(2) 自适应更新并压缩背景模型, 能捕获长时间的、包含运动的或多种变化的背景;

(3) 能对静止和运动背景建模, 无须单独的训练, 在初始化过程中允许存在前景运动目标。

本文将像素从 RGB 空间转换到 HSV 空间进行颜色失

真比较, 取得了比 Kim 等采用的方法更为准确的目标检测效果, 但由于引入了浮点运算, 使得算法的处理速度却比之要低, 我们将继续研究码本大小的限制策略和码字的排序策略, 以便减少码本占用的内存和匹配时间, 从而进一步提高处理速度。同时, 在自动视频监控系统中, 目标检测算法的自动参数选择也是一个很重要的目标^[14]。本算法中的 ϵ_1 和 ϵ_2 取决于单个背景分布的变化, 对码本大小和错检率的影响较大, 我们还将下一步工作中研究这两个参数的自动选择问题。

参考文献:

- [1] 代科学, 李国辉, 涂丹. 监控视频运动目标检测减背景技术的研究现状和展望[J]. 中国图像图形学报, 2006, 11(7): 919-927.
- [2] Lipton A, Fujiyoshi H, Patil R. Moving target classification and tracking from real time video [C] // *IEEE Workshop on Applications of Computer Vision, Princeton, NJ*, 1998: 8-14.
- [3] Cutler R, Davis L. View-based detection [C] // *Int. Conf. Pattern Recognition, Brisbane, Australia*, 1998: 495-500.
- [4] Wren C R, Azarbayejani A, Darrell T, et al. Pfinder: Real-time tracking of the human body [J]. *IEEE TPAMI*, 1997, 19(7): 780-785.
- [5] Horprasert T, Harwood D, Davis L S. A statistical approach for real-time robust background subtraction and shadow detection [C] // *IEEE Frame-Rate Applications Workshop, Kerkyra, Greece*, 1999.
- [6] Stauffer C, Grimson W. Adaptive background mixture models for real-time tracking [C] // *Int. Conf. CVPR*, 1999, 2: 246-252.
- [7] Elgammal A, Harwood D, Davis L S. Non-parametric model for background subtraction [C] // *IEEE ECCV*, 2000, 2: 751-767.
- [8] Toyama K, Krumm J, Brumitt B, et al. Wallflower: Principles and practice of background maintenance [C] // *IEEE ICCV*, 1999: 255-261.
- [9] Matsuyama T, Ohya T, Habe H. Background subtraction for nonstationary scenes [C] // *ACCV*, 2000: 662-667.
- [10] Kim K, Chalidabhongse T H, Harwood D, et al. Background modeling and subtraction by codebook construction [C] // *Proc. Int. Conf. Image Processing, Singapore*, 2004: 3061-3064.
- [11] Greiffenhagen M, Ramesh V, Comaniciu D, et al. Statistical modeling and performance characterization of a realtime dual camera surveillance system [C] // *IEEE CVPR*, 2000: 335-42.
- [12] 荆其诚. 色度学[M]. 北京: 科学出版社, 1979.
- [13] Stauffer C, Grimson W. Learning patterns of activity using real-time tracking [C] // *IEEE TPAMI*, 2000: 47-57.
- [14] Scotti G, Marcenaro L, Regazzoni C. A SOM based algorithm for video surveillance system parameter optimal selection [C] // *IEEE Conf. Advanced Video and Signal Based Surveillance*, 2003.