

晃动目标抑制的拟周期背景算法

贺飞越*, 李佳田, 徐 珩, 张 蓝, 徐燕竹, 王红梅

(昆明理工大学 国土资源工程学院, 昆明 650093)

(*通信作者电子邮箱 hovershadow@163.com)

摘 要: 准确的背景模型是目标提取与跟踪的重要基础。针对复杂场景中出现的局部拟周期变化的晃动目标, 在多元高斯背景模型基础上, 提出一种拟周期背景算法(QPBA), 用以抑制晃动目标, 建立准确而稳定的背景模型。具体过程是: 根据多元高斯背景模型建立场景目标分类模型, 分析晃动目标对高斯模型各参数产生的影响; 以颜色分布值为样本建立高斯模型保留晃动所在像元, 并以出现频次、时间间隔为权重因子, 使晃动像元中的晃动模型融入背景模型。将拟周期背景算法与高斯混合模型(GMM)、背景建模算法(ViBe)、CodeBook等典型背景建模算法进行比较, 通过定性、定量与效率三个方面的评估结果表明: 拟周期背景算法对晃动目标抑制作用明显, 误检率小于1%, 可以很好地应对场景中晃动目标干扰; 同时正检个数与其他算法保持一致, 能够完整地保留运动目标; 算法效率高, 解算时间与CodeBook算法近似, 满足实时性的计算要求。

关键词: 晃动目标; 高斯混合模型; 拟周期; 场景模型; 抑制算法

中图分类号: TP391.41 **文献标志码:** A

Quasi-periodicity background algorithm for restraining swing objects

HE Feiyue*, LI Jiatian, XU Heng, ZHANG Lan, XU Yanzhu, WANG Hongmei

(Faculty of Land Resource Engineering, Kunming University of Science and Technology, Kunming Yunnan 650093, China)

Abstract: Accurate background model is the paramount base for object extracting and tracing. In response to swing objects which part quasi-periodically changed in intricate scene, based on multi-Gaussian background model, a new Quasi-Periodic Background Algorithm (QPBA) was proposed to suppress the swing objects and establish an accurate and stable background model. The specific process included: According to multi-Gaussian background model, the object classification in scene was set up, and the effect on Gaussian model's parameters caused by swing objects was analyzed. By using color distribution values as samples to establish Gaussian model to keep swing pixels, the swing model in swing pixels was integrated into background model with weight factors of occurrence frequency and time interval. Comparison among QPBA and the classical background modeling algorithms such as GMM (Gaussian Mixture Model), ViBe (Visual Background extractor) and CodeBook was put forward, and the results were assessed in aspects of quality, quantity and efficiency. It shows that QPBA has a more obvious suppression on swing objects, and its fall-out ratio is less than 1%, so that it can handle the scene with swing objects. At the same time, its correct detection number is consistent with other algorithms, thus the moving objects can be reserved perfectly. In addition, the efficiency of QPBA is high, and its resolving time is approximate to CodeBook, which can satisfy the requirements of real-time computation.

Key words: swing object; Gaussian Mixture Model (GMM); quasi-periodicity; scene model; restraining algorithm

0 引言

运动目标视频分析是计算机视觉领域重要研究内容之一, 其中背景模型的建立是基础关键任务^[1]。背景在监测过程中局部会发生拟周期变化, 这使得在区分背景和运动目标时, 经常会将这些变化的背景误检为运动目标^[2], 将拟周期变化的背景目标称为晃动目标(swing object)。虽然目前已存在多种背景建模算法适应周期性运动以及缓慢变化的场景, 但依然无法较好地抑制晃动目标的干扰。

文献[3–7]针对背景不完全静止的情形, 提出非静态背景建立高斯混合模型(Gaussian Mixture Model, GMM), 利用背

景像素值在样本中出现概率较大的特点, 采用统计方法为每个像元建立多个分布模型作为背景模型。多个概率分布模型能够表示像素点潜在的多种运动状态, 由此达到适应背景动态变化的目的。然而, 由于其权值的增减由样本值是否匹配决定, 而在多个分布模型中, 晃动目标的拟周期匹配特点使得其对应的分布模型的权值无法持续增大而成为背景模型, 因此, 不能很好区分晃动和运动同时存在的情况。

文献[8–10]提出了一种随机训练样本空间的背景建模算法(Visual Background extractor, ViBe), 该算法利用了相近像素点拥有相近的时空分布特性, 随机选取像元周围像素值作为像元背景模型的样本值, 并实时随机更新背景模型中的

收稿日期: 2014-03-17; 修回日期: 2014-05-21。 基金项目: 国家自然科学基金资助项目(41161061, 40901197)。

作者简介: 贺飞越(1989–), 男, 浙江杭州人, 硕士研究生, 主要研究方向: 图像模式识别; 李佳田(1975–), 男, 黑龙江佳木斯人, 副教授, 博士, 主要研究方向: 计算机视觉; 徐珩(1989–), 男, 湖北武汉人, 硕士研究生, 主要研究方向: 图像模式识别; 张蓝(1990–), 女, 河南平顶山人, 硕士研究生, 主要研究方向: 图像处理; 徐燕竹(1989–), 男, 天津人, 硕士研究生, 主要研究方向: 图像处理; 王红梅(1986–), 女, 山东菏泽人, 硕士研究生, 主要研究方向: 图像模式识别。

样本以及像元周围像元中的背景模型样本。ViBe 算法的优势在于将相邻像元像素值作为样本填充模型样本集,因此,只需要一帧图像即可完成初始化;在背景模型中融入邻近像素值,并在监测过程中随机更替样本值,可以保证样本值的一个指数衰减的平滑生命周期,同时使得背景模型能够很好地适应视频场景的变化,从而达到更好的检测效果。但是其随机更新样本的机制,导致背景模型中的晃动目标的样本不能稳定存在于样本集中,使得当晃动样本被替换时,在前景中出现晃动目标,所以抑制效果不稳定。

文献[11-14]给出了 CodeBook 算法以及多高斯模型改进算法,算法通过视频序列得到各像元的时间序列模型,记录一组特征参数并形成码本(codebook),以描绘背景中感兴趣的状态。算法保持背景的多模特征,将特征参数作为权重因子,根据权重大小保留背景模型和晃动可能模型。但是,算法无法检测是否存在晃动,不能准确找到晃动目标所在位置并进行处理;抑制晃动目标的参数需要根据实际情况进行调试,不能自适应晃动的变化。

本文在多高斯背景模型^[15]基础上,首先建立场景模型,将场景的像元点分为三类,分别对三类像元背景模型中的多个分布模型进行分类^[16]。在此基础上分析晃动目标的特征以及对背景模型产生的影响,得到晃动与运动目标在像元级上的特征参数,以出现频次、时间间隔为特征参数建立高斯分布,能够判断是否存在晃动现象,提取晃动背景的像元,进而对其中的分布模型加以区分,并将晃动模型融入背景模型,从而实现晃动的抑制。对比文献[5,10,12]的方法,本文建立了一个场景目标分类模型,利用特征参数建立二次高斯模型,从本质上区分场景模型中的晃动类;相比文献[5,12]的方法,在背景模型中保留了晃动目标样本,使得在运动目标检测时能够稳定抑制晃动现象。同时提出了拟周期背景算法(Quasi-Periodicity Background Algorithm, QPBA),在大量实验测试序列中,与 GMM、ViBe、CodeBook 算法进行实验对比。结果表明,在晃动抑制方面,本文算法具有更好的抑制效果;同时,算法能够较好地满足实时计算要求。

1 拟周期背景算法

1.1 混合高斯模型

混合高斯模型的基本思想是用 N 个单高斯分布模型来共同描述像元的像素值分布。原理是为图像的每个像素点建立多个高斯分布的组合分布,根据组合中的每一个高斯分布特征,选取其中和背景分布最接近的子高斯分布作为参考分布,依据该参考分布来区分背景像素点和前景像素点。

像素值的概率密度函数(Probability Density Function, PDF)表达式为:

$$\Pr(X_t = \frac{1}{N} \sum_{i=1}^N \omega_{i,t} \times \eta(X_t, \mu_{i,t}, \Sigma_{i,t})) \quad (1)$$

其中: N 是高斯分布的个数, $\omega_{i,t}$ 为第 i 个高斯分布的权值, η 为其中一个高斯概率密度函数。

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = [(2\pi)^{-n/2} |\Sigma_{i,t}|^{-1/2}]^{-1} \exp(-0.5(X_t - \mu_{i,t})^T \Sigma_{i,t}^{-1} (X_t - \mu_{i,t})) \quad (2)$$

其中: μ 为均值; Σ 为协方差矩阵;RGB 颜色空间作为样本空间,各分量独立同分布且有不同方差($\sigma_R, \sigma_G, \sigma_B$)。对新输入的像素值 X_t 按照如下公式匹配:

$$(X_t - \mu_{i,t}) < 2.5\sigma_{i,t} \quad (3)$$

如果满足式(3),则对高斯分布的均值 μ 、方差 σ 和权值

ω 按照如下公式更新:

$$\mu_{i,t} = (1 - \alpha)\mu_{i,t-1} + \alpha X_t \quad (4)$$

$$\sigma_{i,t}^2 = (1 - \alpha)\sigma_{i,t-1}^2 + \alpha(X_t - \mu_{i,t})^T (X_t - \mu_{i,t}) \quad (5)$$

$$\omega_{i,t} = (1 - \varepsilon)\omega_{i,t-1} + \varepsilon(M_{i,t}) \quad (6)$$

如果任一高斯模型与 X_t 均不匹配,构造一个新的高斯模型代替 ω/σ 比值最小的高斯模型。其中, α 是自定义的学习率,且 $0 \leq \alpha \leq 1$, α 的大小决定着背景更新的速度: α 越大更新速度越快;反之,则更新速度越慢。 ε 是参数学习率,与 α 类似决定着权值更新的速度。 $M_{i,t}$ 是判断样本值是否匹配的函数,匹配成功为 1,否则为 0。保留权值较高的高斯模型作为背景模型。检测时,与背景模型不匹配的样本被认为是前景目标。

由式(6)可知,每次仅有一个匹配成功的高斯模型的权值增加,其余高斯模型的权值均衰减。晃动目标产生的高斯模型由于其拟周期匹配的特点,导致权值始终保持在一定范围内,而无法被保留成为背景模型。因此,需要根据拟周期匹配的特点,建立新的权值更新方法,将晃动模型保留在背景模型中。

1.2 场景目标分类

在对晃动目标抑制之前,首先需要判断该场景是否出现晃动目标干扰,然后找到晃动目标影响的场景区域,最后才能对晃动目标进行抑制。由于多高斯背景建模方法在训练时可保留较多高斯模型,并且能提供较多特征参数用于分析,因此,首先根据多高斯背景模型建立一个场景目标分类模型,在概念上区分背景、运动目标和晃动目标。

假设图像是由 n 个像元组成,场景稳定没有抖动干扰,而且没有晃动目标,则一个理想场景可以定义如下:

$$I = B + M \quad (7)$$

其中: I 为图像像元集; B 为背景像元集,指在场景中所有一直保持静止的目标所对应的像元集合, $B = \{B_0, B_1, \dots, B_i\}$; M 为运动像元集,指在场景中所有受运动目标影响的像元集合, $M = \{M_0, M_1, \dots, M_j\}$ 。

然而,场景中不仅仅只有绝对静止的背景,还包含具有复杂运动的背景,即包含随时间变动的背景单元。将场景进一步地定义如下:

$$I = B + M + S \quad (8)$$

其中: S 为晃动像元集,指受晃动目标影响的像元集合, $S = \{S_0, S_1, \dots, S_k\}$; i, j, k 分别为集合 B, M, S 中像元个数, $n = i + j + k$ 。晃动是在场景中,非固定物体由于某些外力的作用在一定区域内做拟周期运动,如摇曳的树枝,摆动的窗帘等,对运动检测毫无意义并且产生干扰。因此,无论这些目标是否运动,应将其归属于背景。

将像素中颜色分布的高斯模型细分为三类:背景模型 b 、运动模型 m 和晃动模型 s 。则背景像元集 B 的像元中只有背景模型 b ;运动像元集 M 的像元中有两类高斯模型:背景和运动模型 $b \& m$;晃动像元集 S 的像元中包含三类高斯模型:背景、运动目标和晃动模型 $b \& m \& s$ 。

式(7)所描述的是一种理想模型,根据高斯模型个数,能直接区分背景像元集和运动像元集($B \& M$),并根据式(3)从 M 中去除高斯模型 m 。然而当场景为式(8)时,使用高斯背景方法会将晃动像元混入运动像元集,因此,需要从复杂模型中区分晃动像元集和运动像元集($S \& M$),然后在 S 中保留高斯模型 s 同时去除 m 。最终得到晃动抑制后的场景背景模型 I_B 。

1.3 晃动区分与抑制

晃动目标介于背景与运动目标之间:相对于背景的绝对

静止而言,它属运动目标;但相对于检测需要的运动目标而言,它应属于背景才不至于产生干扰。对晃动目标的抑制,如图1所示,可以理解为在背景模型中尽可能保留晃动产生的模型,而去除运动产生的模型,使得晃动目标被识别为背景。因此,晃动抑制算法的基本思想是:根据样本值的分布程度,提取受晃动影响的像元,再根据出现频率区分各模型,保留其中的晃动模型,以扩大背景模型,达到抑制效果。



图1 晃动抑制原理

为了得到 s 模型,需要在 S 集合中区分 b 和 m 模型,而在区分 b 和 m 模型前,首先要从 I 集合中区分 S 集合。

以像元为单位分析,晃动影响的区域映射到图像中,形成一组像元集,集合可以分为被晃动目标遮挡和未被遮挡两部分,每个像元都可能出现晃动目标的像素值或被遮挡的像素值,具体可描述如下:

$$C_{i,k} = \begin{cases} \text{Color_Swing}, & \text{遮挡} \\ \text{Color_Background}, & \text{未遮挡} \end{cases} \quad (9)$$

$C_{i,k}$ 为 t 时刻第 k 个晃动像元的颜色值,如果被晃动目标遮挡,则颜色值为Color_Swing,否则颜色值为背景颜色值Color_Background。

基于拟周期运动的可能,晃动时,目标 S 像素值在原静止像素中出现次数减少,时间间隔增大;在影响区域的出现次数增加,时间间隔减小。运动目标的出现次数很少,时间间隔很长。因此,在混合高斯模型中,晃动导致颜色值混入周围像元的分布模型中,颜色分布范围更宽,可能产生新颜色值(晃动目标静止时被遮挡的颜色值),产生新的高斯分布,使得原本集中于背景颜色值周围的高斯分布改变为集中于背景和晃动目标颜色值周围的多个高斯分布,最终导致描述像元的颜色分布曲线平缓程度的参数 σ_{pixel} 变大。在实验3的表1中可以明显看出这一特点。

基于上述分析,方差可用于描述模型颜色的分布情况,是运动目标和晃动目标所具有的不同特征。图像中各像元产生的高斯模型相互独立,并且方差独立同分布。因此,可以在原高斯模型的基础上,以像元的颜色分布程度值 σ_{pixel} 作为样本建立一个高斯模型,用以区分场景中的各像元 B 和 M 和 S ,从而找到晃动像元点。

多高斯背景训练后,每个像元得到 $1 \sim k$ 个高斯模型,模型包含参数 $(\mu, \sigma, q, f, \lambda)$ 。其中: μ 和 σ 分别为模型的均值和标准差,可由式(4)、(5)计算得到; q 为最后匹配的时刻,用于实时计算 λ ; f 为模型在训练时的匹配次数; λ 为模型最大匹配时间间隔。时间间隔是当前匹配时刻至前一次匹配时刻的时间, λ 为训练过程中时间间隔中的最大者。

由于多高斯模型在一个像元中产生多个单高斯模型,使得每个像元中有多个与模型相对应的方差,要描述像元的颜色分布程度就需要综合各模型的方差值。分析可知, b 模型对应的方差较小,而频率较大; m 模型方差较不稳定,频率较小; s 模型的方差相对 b 模型较大,频率小于 b 模型但大于 m 模型。为了消除 m 模型方差较不稳定的影响,根据匹配频率大小作为各模型的权重,根据以下公式计算描述像元中颜色分布曲线的平缓程度:

$$\sigma_{\text{pixel}}^2 = \sum_{i=0}^k (f_i/f_{\text{sum}}) \cdot \sigma_i^2; 0 \leq ci \leq c \quad (10)$$

其中: σ_{pixel} 为描述第 ci 个像元的平缓程度; k 为模型个数; f_i 、 σ_i 分别为像元中各模型训练后的匹配次数和标准差; f_{sum} 为所有模型训练次数, $f_{\text{sum}} = N$ 。

以像元方差为样本,建立一个图像像元平缓度分布,计算模型的期望 μ_{frame} 和方差 σ_{frame} :

$$\mu_{\text{frame}} = \frac{1}{c} \sum_{ci=0}^c \sigma_{\text{pixel}}^2 \quad (11)$$

$$\sigma_{\text{frame}}^2 = \frac{1}{c} \sum_{ci=0}^c (\sigma_{\text{pixel}}^2 - \mu_{\text{frame}})^2 \quad (12)$$

其中 c 为像元个数。如果 σ_{pixel} 分布集中,则 $\sigma_{\text{frame}} \leq \beta$,即没有产生或没有明显产生晃动目标,可以选择常规方法进行更新;如果 σ_{pixel} 分布分散,则 $\sigma_{\text{frame}} > \beta$,即晃动目标相对较多,需要对背景模型进行调整,找出受晃动影响的像元。 β 为判断是否晃动的阈值。

依次判断每个像元,筛选出模型中方差值高的像元放入 S 像元集中,按照如下公式筛选:

$$|\sigma_{\text{pixel}} - \mu_{\text{frame}}| > \gamma \times \sigma_{\text{frame}} \quad (13)$$

其中 γ 为晃动比例参数,可根据晃动影响区域大小进行调整。

选出训练时受到晃动影响的像元,得到 S 像元集后,需要对每个像元中的高斯模型区分,保留 b 、 s 模型,去除 m 模型。在分析中可知, b 和 s 模型 λ 较小,而 f 较大; b 模型相对 s 模型 λ 较小, f 较大。因此,在三种模型中, b 模型 λ 相对最小, f 相对最大; m 模型 λ 相对最大, f 相对最小。

对每个模型按照权重大小进行排序:

$$\omega_i = f_i^{p_1} / \lambda_i^{p_2} \quad (14)$$

其中 p_1 、 p_2 分别表示 f 和 λ 对于模型归属的重要程度。归一化

权值得使得 $\sum_{i=0}^k \omega_i = 1$,保留前 i 个模型,作为背景模型:

$$M_i = \omega_{i-1} + (M_{i-1}); M_i \leq \tau \quad (15)$$

其中 τ 为权重阈值。按式(14)排序后,像元中依次为 b 、 s 、 m 模型,式(15)处理每个 S 像元集中的像元后,去除了位于末尾的 m 模型,得到晃动抑制后的背景模型 I_B 。进入运动检测阶段,当输入像素 I_t 和每个背景分布均值之差的绝对值都大于该分布标准差的 D 倍,则 I_t 被认为是运动前景,否则被判为背景。

1.4 算法过程

如图2所示,首先,在多高斯背景模型基础上加以训练,获得记录参数 $(\mu, \sigma, q, f, \lambda)$ 的多高斯模型;然后,根据已有模型中的参数 μ ,建立一个高斯模型,当判断有晃动影响时,进入晃动抑制过程;其次,根据新建立的高斯模型找到晃动影响的像元点,以不同的过滤标准对模型筛选,提高晃动目标模型在各模型中的权重,去除运动目标模型同时保留晃动目标模型。当 $\sigma_{\text{frame}} < \beta$ 判断不存在晃动影响时,仅舍去超阈值模型,即虚线部分。

算法输入:一组视频帧序列 (X_0, X_1, \dots, X_t) ;

算法输出:融入晃动模型的背景模型融入晃动模型的背景模型 $\{I_B = B \cup (M - m) \cup (S - m)\}$ 。

1) 进行传统多高斯建模,得到各模型的 μ 和 σ ,并记录各个高斯模型的匹配次数 f 、最后匹配时刻 q 和最大时间间隔 λ ,形成数据集 $(\mu, \sigma, f, \lambda, q)$ 。

2) 根据式(10)计算像元平缓度 σ_{pixel} 。

3) 根据式(11)、(12),以 σ_{pixel} 为样本建立高斯模型,得到均值 μ_{frame} 和方差 σ_{frame} 。

4) 判断新的高斯模型方差 σ_{frame} 是否小于 β , 如果不满足条件, 转至步骤5); 如果满足条件, 转至步骤6)。

5) 依次输入各像元的 σ_{pixel} , 判断不等式(13), 如果不满足条件, 则转至步骤6); 如果有满足条件的像元, 则转至步骤7); 如果完成像元的遍历, 则转至步骤8)。

6) 舍去 λ 大于阈值的模型, 返回步骤4), 继续匹配下一个像元。

7) 对该像元中的模型, 根据式(14) 计算权重并进行排序, 根据式(15) 舍去累计权值大于阈值 τ 后的模型。返回步骤4), 继续匹配下一个像元。

8) 将得到的 I_B 作为背景模型, 对新输入的图片进行背景差运算, 检测运动目标。

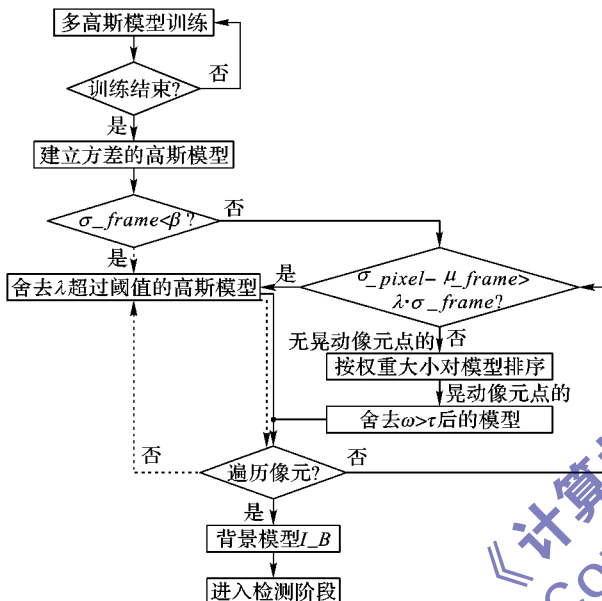


图2 算法流程

2 实验与分析

将本文提出的 QPBA 与文献[5, 11, 15]中的三个经典算法进行比较, 通过 5 组实验从定性、定量和效率三方面进行对比分析, 根据运动检测结果, 进一步解释晃动抑制算法, 分析各算法对于存在晃动干扰的场景应用所存在的不足。其中, 定性分析以远景和近景两组视频序列为例分析各算法对于晃动抑制在理论上的不足; 定量分析主要通过分析特征参数的变化特点、误检数和正检数来进一步说明晃动抑制算法, 区分方法的可靠性及其优势; 效率分析则主要根据计算速度和误检数在训练样本上的收敛速度进行评估。

实验环境: CPU 为 Intel Core 2 Duo CPU P8700 2.53 GHz, 4 GB 内存; 显卡为 NVIDIA GeForce G 105M; 开发软件采用 VS 2008 和 Windows XP 系统; 软件开发库为 OpenCV 2.4; 所采用的视频序列大小均为 640×480 。

2.1 定性评估

实验 1 远景晃动。实验测试序列 1 为校园道路视频序列, 其中存在树枝的晃动干扰, 晃动目标为远景, 距离镜头较远, 产生的误检区域小, 影响的像元个数较少, 但若不对它处理, 依然会干扰运动目标检测的最终结果。图 3 为视频序列 1 在第 312 帧图像上各算法对远景有晃动目标影响的运动目标检测结果。

图 3(a) 为原始图像, 场景中存在两种晃动目标: 场景上半部分, 由于风的影响导致树来回晃动; 场景左下方, 广告牌

以一定周期循环显示多个广告。这些拟周期目标产生的晃动影响导致检测结果不准确。图 3(b) 为 GMM 算法结果, 在晃动影响的区域能明显被检测为前景运动目标, 且 GMM 算法训练时间长, 运算量大。图 3(c) 为 ViBe 算法结果, 可以看到晃动目标对 ViBe 算法的结果影响较小, 可使用形态的开运算处理检测结果, 然而, 使用开运算处理后, 抑制晃动目标的同时也腐蚀了一部分运动目标。因此, 晃动目标在 ViBe 算法中间接地影响了运动目标检测结果。图 3(d) 为 CodeBook 算法结果, 虽然运动目标显示完整, 优于 ViBe 算法, 但晃动干扰明显存在, 无法使用形态学运算去除, 使得背景模型无法用于运动检测。图 3(e) 为 QPBA 结果, 使用 QPBA 后, 能较好地排除晃动目标的干扰, 同时保证运动目标的完整性, 从而能够较好地跟踪运动目标。

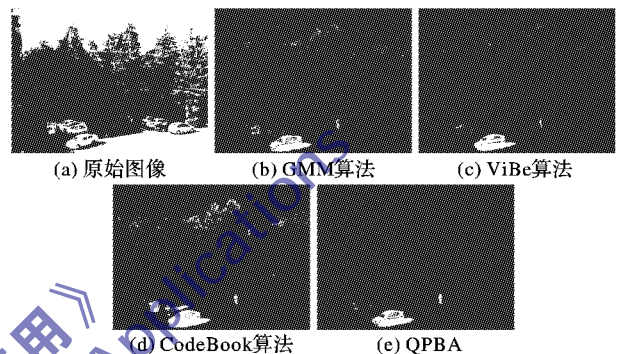


图3 远景晃动干扰的检测结果

实验 2 近景晃动。当晃动目标靠近于镜头时, 运动目标容易被晃动目标遮挡, 晃动目标将占据场景较大区域且晃动幅度较大。实验测试的场景为存在近景晃动的校园路口, 图 4 为视频序列 2 在第 228 帧图像上各算法对近景有晃动目标影响的运动目标检测结果。

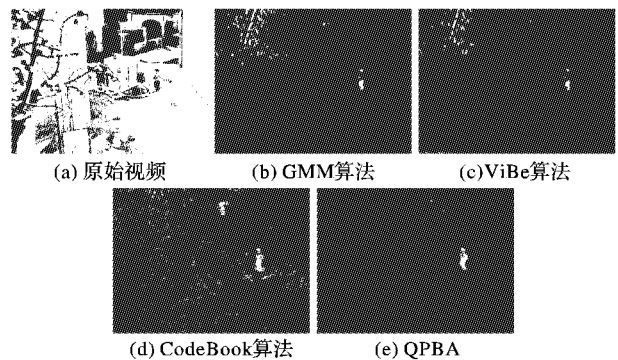


图4 近景晃动干扰的检测结果

图 4(a) 为原始图像, 场景的变化主要来自左半部分, 窗台上靠近镜头的树枝保持拟周期性的摆动, 场景右边是道路, 有运动目标经过。图 4(b) 显示的 GMM 算法结果, 既无法去除晃动影响, 又无法保证运动目标的完整, 表现最差。图 4(c) 为 ViBe 算法结果, 由于近景晃动影响较大区域且变化快速, 使得该算法无法由邻近点的样本值代替描述像元点在时间上的分布特性, 导致在图像左边枝叶摆动的位置存在大面积晃动干扰。图 4(d) 为 CodeBook 算法结果, 根据匹配次数和最大时间间隔过滤模型后, 晃动抑制结果优于 GMM 和 ViBe 算法, 但是该算法在过滤模型前并没有筛选晃动像元, 而是直接对全局像元中的模型进行过滤, 以致于晃动抑制后影响了整个场景的检测效果。通过多次实验表明, 该算法无法通过形态学运算、延长训练时间或调整检测参数的方法改善其结果。而在图 4(e) 中可以看到, 本文算法与前三算法

相比,图像左侧没有被检测为运动目标的白色点,对晃动产生的干扰有明显的抑制效果,并且运动目标检测完整。

2.2 定量评估

实验3 参数变化分析。为了更好地解释晃动导致的各参数变化情况,在实验场景3中建立一个理想的静止和晃动对比模型,并监视各参数的变化情况。如图5所示,场景中有3个目标,分别是书本、杯子和笔杆,没有明显光照变化,图(a)和(b)分别为原始帧和晃动抑制结果。视频包含两阶段:从0到800帧,所有目标都处于静止;从801到1600帧,场景中的笔杆上下晃动,杯子略有移动,书本仍处于静止。选取在笔杆位置的像元,记录从第1帧到第1600帧像元产生的模型方差、频率、最大时间间隔;输出第800帧和第1600帧各模型记录的参数,如表1所示。

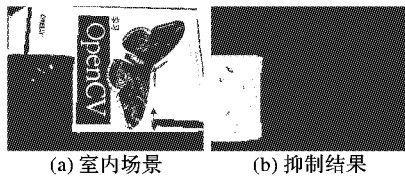


图5 室内晃动实验

表1 静止与晃动后,方差与频率的变化

像元 p		σ	f	λ
静止,	模型 b_1	0.2	800	1
$p \in B$	模型 s_1	无	无	无
晃动,	模型 b_2	3.5409	607	2
$p \in S$	模型 s_2	4.0317	194	6

由表1可知,像元没有受晃动影响前,由于样本保持不变或在极小范围内变化,则样本只在颜色空间中聚类为一个类,产生一个稳定的背景模型 b ,其方差收敛,并随着样本数的增加而趋近于0,匹配次数等于训练次数;晃动影响后,样本随着晃动样本的加入而开始变化,假设晃动样本具有相同的颜色值,则经过样本在颜色空间中的聚类,最终将形成两个类,产生背景模型 b 和晃动模型 s ,由于晃动样本的影响,使得颜色分布不再集中于一点,描述颜色分布情况的方差变大,且 s 模型的方差较大,晃动在像元上停留的时间较短,这就决定了晃动模型的匹配次数 f 较少,最大时间间隔 λ 较长。实验3没有选取运动目标经过像元,仅考虑晃动现象,原因在于晃动目标的颜色值稳定不变,而运动目标的颜色值多变,这将使得颜色空间形成多个样本聚类,而无法凸显背景模型和晃动模型的特点。但当有运动样本和晃动样本同时加入时,在理论上可推知:

1) 颜色分布将更加分散,使得方差值更大;

2) 运动样本的个数虽然可能多于晃动样本,但运动样本形成聚类较多,各运动模型 m 的匹配次数不及个数较少的 s 模型;

3) 运动样本在样本中随机出现,且有多个聚类,因此其中一个聚类模型的样本最大匹配时间间隔 λ ,必定长于拟周期匹配的最大时间间隔。

实验4 算法准确率分析。通过本文算法和以上算法的定性评估,初步显示了本文算法的优势。实验继续以视频序列2为例,从200帧开始到350帧结束,根据统计得到各算法对于晃动目标的误检和正检情况,对各算法作进一步的定量评估。

在运动检测中误检的数量将会直接影响检测准确率,所以,晃动目标误检数是评估晃动抑制效果的重要指标。如图

6(a)所示,曲线表示各算法误检数的变化情况。由曲线可知,CodeBook算法和GMM算法的误检个数在500~3000,实验结果不理想,ViBe算法的误检个数在60~140,而本文算法的误检个数在20以下,晃动抑制效果更明显。在减少误检数的同时,运动目标检测的完整性也是一项重要指标,因此算法需要保证足够的正检数量。如图6(b)所示,曲线表示各算法正检数的变化情况。ViBe算法的正检数相对较少,而其他三种算法的变化情况基本相同。通过误检和正检的定量评估表明QPBA的晃动抑制效果优于其他三种算法且不影响运动目标检测。

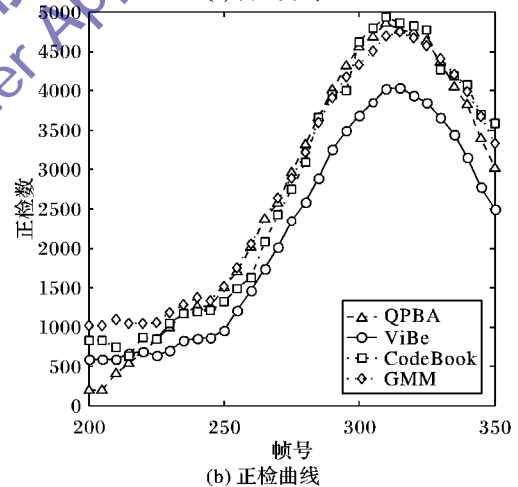
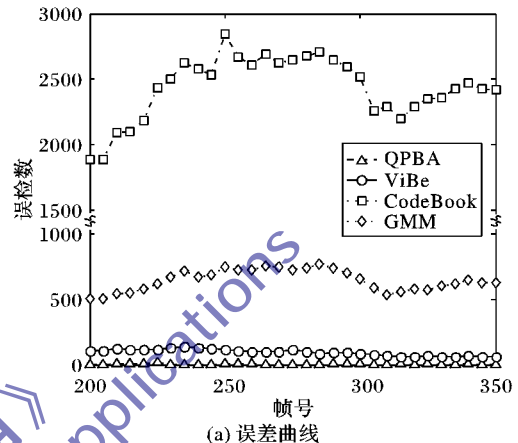


图6 误检和正检比较

2.3 效率评估

为了满足检测的实时性,算法需要在计算时效方面达到实际要求。分析本文算法对近景和远景视频序列的抑制时效,以及场景不发生晃动时算法的检测效果。

使用avi格式大小为640×480的多组视频序列,分别记录前200帧训练时间,训练速度=训练帧数/训练时间,各算法的平均训练速度如表2所示。

表2 训练速度对比

算法	平均训练速度/(frame·s ⁻¹)	算法	平均训练速度/(frame·s ⁻¹)
GMM	0.2	QPBA	4.5
CodeBook	3.0	ViBe	8.0

其中:GMM算法速度无法满足实时性要求;ViBe算法是以第1帧为样本初始化后立即开始检测,因此速度最快;QPBA速度虽不如ViBe算法,但优于CodeBook算法,能够满足实时性要求。

当场景中没有晃动目标时, QPBA 将进入 1.4 节算法的步骤 6), 即图 2 中的虚线部分, 不进行晃动抑制。图 7 为不存在晃动的场景检测结果, 其中图 (a)、(c) 为原始图像, 图 (b)、(d) 为检测结果。可以看到 QPBA 在一般情况下依然能够正常检测。

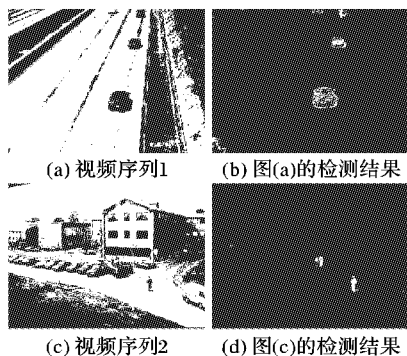


图7 不发生晃动时的检测效果

由于 QPBA 需要经过训练以适应场景的变化, 所以, 需要评估不同晃动场景的训练收敛速度, 以判断算法是否能够适应场景全局的变化速度。

图 8 为 QPBA 经过不同训练次数后对于晃动抑制的效果。实验序列为实验 1 和实验 2 中的近景和远景视频序列, 根据统计得到使用不同帧数训练后的误检个数。可以看到, 对于不同情况的晃动目标, 算法经过 200 帧训练后误检数基本收敛, 场景中的晃动误检数量在 50 以下, 在 600×800 的图像中产生的影响基本为 0。以 4.5 frame/s 速度训练 200 帧大约需要 45 s, 小于实际场景全局的变化速度, 因此在实时更新中, QPBA 能够满足场景变化的实时要求。

在实际运行过程中, QPBA 各参数设定如下: $N = 200$, $\lambda = 100$, $\alpha = 0.05$, $\beta = 0.5$, $\gamma = 2.5$, $p_1 = 1$, $p_2 = 1$ 。其中 β 是检测是否发生晃动的参数, 其值越小, 对晃动越敏感; γ 参数决定了晃动影响的像元在场景中所占比例。

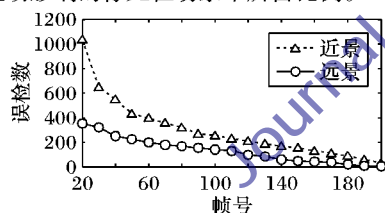


图8 QPBA 训练速度

3 结语

本文提出了一种基于多高斯模型的晃动目标抑制算法。实验结果说明, 本文算法能够自动检测是否存在晃动目标干扰, 准确定位晃动目标影响的区域, 在去除晃动目标的同时能完整保留运动目标。本文抑制方法只对多高斯建模和 CodeBook 方法适用, 需要有晃动训练, 未对其他方法在晃动抑制上进行深入研究, 在未来工作中将研究能普遍适用其他背景建模方法的实时晃动抑制方法, 为运动目标检测提供具有更好鲁棒性的背景模型。

参考文献:

[1] DAI K, LI G, TU D, *et al.* Prospects and current studies on background subtraction techniques for moving objects detection from surveillance video [J]. *Journal of Image and Graphics*, 2006, 11(7): 919–927. (代科学, 李国辉, 涂丹, 等. 监控视频运动目标检测减背景技术的研究现状和展望[J]. *中国图象图形学报*, 2006, 11(7): 919–927.)

[2] LI L, HUANG W, GU I Y H, *et al.* Foreground object detection from videos containing complex background [C]// *MULTIMEDIA '03: Proceedings of the eleventh ACM International Conference on Multimedia*. New York: ACM, 2003: 2–10.

[3] STAUFFER C, GRIMSON W E L. Adaptive background mixture models for real-time tracking [C]// *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society, 1999, 2: 246–252.

[4] CHANG X, ZHANG W, DONG W. Mixture of Gaussian background modeling method based on multi-category visual features [J]. *Journal of Image and Graphics*, 2011, 16(5): 829–834. (常晓夫, 张文生, 董维山. 基于多种类视觉特征的混合高斯背景模型[J]. *中国图象图形学报*, 2011, 16(5): 829–834.)

[5] LEE D-S. Effective Gaussian mixture learning for video background subtraction [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, 27(5): 827–832.

[6] ZIVKOVIC Z. Improved adaptive Gaussian mixture model for background subtraction [C]// *ICPR 2004: Proceedings of the 17th International Conference on Pattern Recognition*. Piscataway: IEEE, 2004, 2: 28–31.

[7] ZIVKOVIC Z, van der HEIJDEN F. Efficient adaptive density estimation per image pixel for the task of background subtraction [J]. *Pattern Recognition Letters*, 2006, 27(7): 773–780.

[8] SUN S, QIN Y, MA X, *et al.* ViBe foreground detection algorithm and its improvement with morphology post-processing for outdoor scene [J]. *Computer Engineering and Applications*, 2013, 49(10): 159–162. (孙水发, 覃音诗, 马先兵, 等. 室外视频前景检测中的形态学改进 ViBe 算法[J]. *计算机工程与应用*, 2013, 49(10): 159–162.)

[9] BARNICH O, van DROOGENBROECK M. ViBe: a powerful random technique to estimate the background in video sequences [C]// *ICASSP 2009: Proceedings of the 2009 International Conference on Acoustics, Speech and Signal Processing*. Piscataway: IEEE, 2009: 945–948.

[10] BARNICH O, van DROOGENBROECK M. ViBe: a universal background subtraction algorithm for video sequences [J]. *IEEE Transactions on Image Processing*, 2011, 20(6): 1709–1724.

[11] KIM K, CHALIDABHONGSE T H, HARWOOD D, *et al.* Background modeling and subtraction by codebook construction [C]// *ICIP 2004: Proceedings of the 2004 International Conference on Image Processing*. Piscataway: IEEE, 2004, 5: 3061–3064.

[12] KIM K, CHALIDABHONGSE T H, HARWOOD D, *et al.* Real-time foreground-background segmentation using codebook model [J]. *Real-time Imaging*, 2005, 11(3): 172–185.

[13] ILYAS A, SCUTURICI M, MIGUET S. Real time foreground-background segmentation using a modified codebook model [C]// *AVSS 2009: Proceedings of the Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. Piscataway: IEEE, 2009: 454–459.

[14] LI Y, CHEN F, XU W, *et al.* Gaussian-based codebook model for video background subtraction [C]// *ICNC 2006: Proceedings of the Second International Conference on Advances in Natural Computation*, LNCS 4222. Berlin: Springer-Verlag, 2006: 762–765.

[15] ZHANG Y, SONG S, ZHANG R, *et al.* Study on adaptive ability of Gaussian mixture background model [J]. *Journal of Computer Applications*, 2011, 31(3): 706–709. (张运楚, 宋世军, 张汝敏, 等. 高斯混合背景模型的适应能力研究[J]. *计算机应用*, 2011, 31(3): 706–709.)

[16] CAI J, WALKER R. Robust video stabilization algorithm using feature point selection and delta optical flow [J]. *IET Computer Vision*, 2009, 3(4): 176–188.