

# 帧间差分与码书模型相融合的运动目标检测算法

Detection Algorithm Based on Consecutive Frames Subtraction and Codebook Model

陈燕萍 余臻 (厦门大学信息学院自动化系, 福建 厦门 361005)

## 摘要

针对视频检测对实时性和准确性的要求,提出了将帧间差分与码书模型相融合的运动目标检测算法。首先,选取某个t时间内采集到的视频图像作为训练图像,对图像的每一个像素点建立码书模型。把码书模型中表示前景的码书去除,余下的作为背景模型用于检测。检测运动目标时,先将待检测的相邻帧图像进行帧间差分,得到变化区域和没有变化的区域,将有变化的区域与背景模型进行拟合,区分出前一帧运动区域和目标运动区域。更新背景模型时以不同的更新方法对前一帧运动区域和目标运动区域进行更新。

关键词: 运动目标检测, 码书模型, 帧间差分, 阴影检测

## Abstract

Aimed at the real-time and veracity of current algorithm for detecting moving objects, an algorithm based on two consecutive frames subtraction and codebook model is presented. First, select the video images which are collected in a consecutive time as the training images, establish a codebook for every pixel of the images. From the first two images on, subtract two consecutive frames to find out moving areas and background areas. Compare the moving areas with the background model to decide whether the moving area is the moving objects or not. The background updating algorithm varies in the rate of updating according to the different areas of the image.

Keywords: moving objects detection, codebook model, two consecutive frames subtraction, shadow detection

目前,运动目标的检测算法主要有光流、帧间差分、混合高斯模、非参数模、隐马尔可夫模、码书方、W4等,其中,光流法虽然能得到大量的运动信息,由于其计算量大且复杂,很难满足实时性的要求;帧间差分法具有较强的场景变化适应能力,更新速度快,计算简单等优点,但仅用帧间差分法的话,所得的运动信息少,无法区分变化区域是否是运动区域;混合高斯模型,隐马尔可夫模型,码书模型等都属于背景减除算法,不同在于背景模型的建立与更新算法不一样,由此所得的效果也不同。在目前的运动目标检测中,减背景算法是一种比较受欢迎的检测算法,而其中码书模型由于其计算相对简单、速度快,并且能较好地处理光照变化和阴影等噪声影响,将被选作本文建立背景模型的算法,并融合帧间差分,是一种有效、快速的运动目标检测算法。

## 1 码书背景模型的建立

码书模型算法是利用量化和聚类技术来构建背景模型,我们通过对每个像素进行训练,基于色彩标准和亮度界限聚类得到各个像素的码。图像的每一个像素点都包含一个码书,每个码书里有若干个码字,所有像素的码书组成码书模型。不同像素点的码书所包含的码字个数不一定相同,码字的个数与内容依采样图像的变化而不同。在得到包含各个像素码书的码书模型之后,通过一定的过滤方法将像素里表示前景物体的码字去除,剩下的即是表示背景的码书背景模型。

### 1.1 建立码书模型

假设P代表一个像素训练序列,它包含K个RGB向量: $P = \{p_1, p_2, \dots, p_k\}$ 。B代表这个像素的码书,它包含N个码字 $B = \{b_1, b_2, \dots, b_N\}$ (不同的像素的码书对应的N的取值不同),每个码字 $b_i, i=1, \dots, N$ 包含一个RGB向量 $b_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ 和一个六元组 $para_i = \langle \tilde{H}_i, \hat{H}_i, f_i, a_i, l_i \rangle$ 。这个六元组各个变量具体定义如下:

$\tilde{H}_i, \hat{H}_i$ : 分别表示属于背景的每个像素的亮度变化范围的最小和最大界;

$a_i$ : 表示在训练期间,此码字两次出现之间最长的时间间隔;

$f_i$ : 表示在训练期间,该码字出现的频率;

$a$ : 表示训练期间,该码字第一次出现的时间;

$l$ : 表示训练期间,该码字最后一次出现的时间;

每个像素的码书构建如下:

首先,训练之前,码书为空集,即 $N=0, B=\emptyset$ 。对于每个时刻 $t, t=1, 2, \dots, K$ (对应每个训练像素值 $k=1, 2, \dots, K$ ),令 $p_t = (R, G, B)$ ,计算该时刻像素亮度值 $H, H = \sqrt{R^2 + G^2 + B^2}$ 。现在,我们取 $t=1$ 时刻的像素值,此时码书里没有码字,所以要创造一个码字。令 $N=1, b_1 = (\bar{R}_1, \bar{G}_1, \bar{B}_1)$ ,  $para_1 = \langle \tilde{H}_1, \hat{H}_1, 0, 1, 1, 1 \rangle$ 。从 $t=2$ 时刻到 $t=K$ 时刻,分别将该时刻像素值对应的色彩和亮度值与该时刻之前的码书里的所有码字进行匹配,如果能匹配,则更新匹配的码字,否则,创造一个新的码字。匹配的判断原为:

1)  $colordist(p_t, b_s) < \epsilon$

2)  $brightness(H, \langle \tilde{H}_s, \hat{H}_s \rangle) = true$

其中, $b_s$ 为码书中正在被进行匹配的码字。之所以用这两个公式作为匹配原则是因为通过观察发现,前景物体的影响而使像素值发生的变化主要体现在色彩变形和亮度变化两个方面。我们通过限定像素色彩和亮度与背景模型像素对应项的差距便可判断该像素是否属于变化的区域。具体过程如图1所示。

### 1.2 建立背景模型

由于我们对训练图像没有严格的要求,允许在训练图像中出现前景物体和干扰噪声,因此,通过上面方法所得到的码书,不仅包含了表示背景的码字,同时也包含了表示前景和噪声的部分。因此,在把该模型变为背景模型之前,需要先将把这些不是表示背景的码字去除。

码字的去除主要以码字中的时间间隔元素作为过滤过程的主要参。由于图像里不同物体的像素的出现具有不同的周期性,背景物体重复出现的周期最短,而前景物体和噪声物体的像素的出现具有偶然性,即使有重复出现的物体,其周期也较长,因此,我们可通过对像素的某种状态出现的时间间隔,即表示该状态的码字的时间间隔长短就可以判定该码字是否表示背景。过滤原则为:

$$B_{background} = \{b_s | b_s \in T\}$$

其中,  $B_{background}$  为最终的码书背景模型, 包含所有表示背景的码字。T 是时间域值, 若码字重复出现的最大时间间隔  $s$  比规定的时间域值小, 说明该码字重复出现的周期短, 是背景码字, 反之, 则为前景和噪声码字, 将从码书中删除。域值 T 的确定由实验结果得到, 一般设置为  $\frac{2K}{5}$ , 其中 K 为训练帧数, 若码字在  $\frac{2K}{5}$  的帧中没有出现, 则判断为前景码字, 否则为背景码字。

在过滤了表示前景和

噪声的码字之后, 剩下的码书就是较为纯净的背景模型。有了背景模型, 我们可以进行运动目标的检测了。

### 2 运动目标的检测

运动目标的检测分为两步。首先, 通过帧间差分确定图像的变化区域, 缩小检测范围; 其次, 将检测到的变化区域与背景模型相比较, 去除干扰区域, 确定出运动目标。

#### 2.1 帧间差分

通过对相邻的两帧图像进行差分, 可以确定出图像的变化区域。假设训练图像序列中的一帧图像有 m 行 n 列, 第 k 帧图像第 i 行第 j 列像素的灰度值为  $f(k, i, j)$ , 则第 k 帧图像可以表示为:

$$F(k) = \begin{bmatrix} f(k, 1, 1) & \dots & f(k, 1, n) \\ \vdots & & \vdots \\ f(k, m, 1) & \dots & f(k, m, n) \end{bmatrix}$$

两帧差分图像为:  $D(k) = F(k) - F(k-1)$ , 即

$$D(k) = \begin{bmatrix} f(k, 1, 1) - f(k-1, 1, 1) & \dots & f(k, 1, n) - f(k-1, 1, n) \\ \vdots & & \vdots \\ f(k, m, 1) - f(k-1, m, 1) & \dots & f(k, m, n) - f(k-1, m, n) \end{bmatrix}$$

定义差分图像的每个像素为:

$$d(k, i, j) = \begin{cases} 0 & f(k, i, j) - f(k-1, i, j) \\ f(k, i, j) - f(k-1, i, j) & f(k, i, j) - f(k-1, i, j) > 0 \end{cases}$$

其中, 差分图像中像素值等于 0 的像素为背景区域, 像素值等于 1 的像素为图像的变化区域, 包括运动目标区域, 和前一帧图像中运动目标所覆盖的区域。初步的检测出现的错误有两种, 或者将前景区域认为是背景, 或者将背景区域认为是前景。由于被认为是背景的像素在后续步骤中将一直被认为是背景而不再被处理, 因此, 取中间值  $\frac{1}{2}$  的选取应要减少将变化区域误认为背景的出错率, 取其微低于平均值, 定义为:

$$= \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n d(k, i, j) \quad (\text{仅取变化大的像素差分进行求和, 降低了取值})$$

通过上面算法得到图像的变化区域, 即  $d(k, i, j) = f(k, i, j) -$

$f(k-1, i, j)$  的所有像素。在这些变化区域中, 其中一部分为前景物体和噪声, 另一部分为上一帧图像中前景的覆盖区域, 当前景物体移动以后, 这部分又回到原来的像素, 重新属于背景。下面, 通过, 对变化区域与背景模型的拟合, 将区分出运动物体和背景。

#### 2.2 运动目标检测

确定了变化区域, 就可以进行运动目标的检测。对于变化区域的每一个像素, 获取它的色彩值  $p_i = (R, G, B)$ , 计算亮度  $H = \sqrt{R^2 + G^2 + B^2}$ 。在码书背景模型中查找匹配的码字, 匹配原则与码书码书构建时的原则一样, 即

- 1)  $colordist(p_i, s) < \theta$
- 2)  $brightness(H, \langle \tilde{H}_s, \hat{H}_{hs} \rangle) = true$

若找到匹配的码字则该像素属于背景, 否则, 这个像素就是属于前景像素。把所有变化区域的像素都与背景模型的码字相匹配, 没有找到匹配码字的像素集合就是前景区域。

#### 3 背景模型的更新

背景模型的更新, 包括两个部分, 背景区域和上一帧运动物体覆盖区域。

- 1) 属于背景的像素, 更新与其匹配的码字  $b_s$ , 把  $(s = (\bar{R}_s, \bar{G}_s, \bar{B}_s), para_s = (\tilde{H}_{ls}, \hat{H}_{hs}, s, f_s, a_s, l_s))$  更新为  $(s = (\frac{f_s \bar{R}_s + R}{f_s + 1}, \frac{f_s \bar{G}_s + G}{f_s + 1}, \frac{f_s \bar{B}_s + B}{f_s + 1}), para_s = (\min\langle H, \tilde{H}_{ls} \rangle, \max\langle H, \hat{H}_{hs} \rangle, \max\langle t - l_s, s \rangle, f_s + 1, a_s, t))$ ;
- 2) 对于上一帧运动物体覆盖区域, 更新方法将以快速的更新速率将它收入到背景模型中, 把匹配码字  $b_s$  的  $(s = (\bar{R}_s, \bar{G}_s, \bar{B}_s))$  更新为, 把更新为:

$$s = \left( \frac{f_s \bar{R}_s + 2R}{f_s + 2}, \frac{f_s \bar{G}_s + 2G}{f_s + 2}, \frac{f_s \bar{B}_s + 2B}{f_s + 2} \right), \text{把 } para_s = (\tilde{H}_{ls}, \hat{H}_{hs},$$

$s, f_s, a_s, l_s)$  更新为  $para_s = (\min\langle H, \tilde{H}_{ls} \rangle, \max\langle H, \hat{H}_{hs} \rangle, \max\langle t - l_s, s \rangle, f_s + 1, a_s, t)$

#### 4 阴影消除

通常我们所检测出来的前景区域并不仅是运动目标, 还包含着阴影区域, 因为阴影和运动目标具有一些相似的特性, 使得在检测过程容易把阴影误认为是运动目标。因此, 需要一定的阴影检测算法把阴影从前景中分离出去。本文采用文献[9]中基于 HSV 颜色空间的阴影模型。HSV 颜色空间具有较好的颜色感知一致性, 符合人的视觉感受, 比其他颜色空间能更准确检测出阴影。检测阴影的公式为:

$$S(x, y) = \begin{cases} 1 & \frac{I(x, y) \cdot V}{B(x, y) \cdot V} > (I(x, y) \cdot S - B(x, y) \cdot S) \quad \& \quad |I(x, y) \cdot H - B(x, y) \cdot H| > \theta \\ 0 & \text{其他} \end{cases}$$

其中,  $I(x, y)$  表示当前帧,  $B(x, y)$  表示背景模型,  $H, S, V$  表示在 HSV 颜色空间下的各个组成分量。限制亮度的强弱, 取值越低, 亮度越强;  $0 < \theta < 1$ , 设置  $\theta$  是为了去除噪声的影响。  $s_n$  一般取负值,  $s_n$  区分当前帧与背景模型之间的明显差异。

通过阴影消除, 将阴影和噪声排除, 剩下的前景区域就是运动目标。

#### 5 实验结果

对一组连续时间内获取的图像进行实验, 实验结果如图 2 所示。原图像从网上下载, 共有 8 帧。在图 2 中, 图 2(a) 为前四帧原图像, 将其作为训练图像, 通过对其训练得到背景模型。图 2 (b) 为第五帧原图像, 即用于检测的当前图像, 与它的前一帧图像(即第四帧图像)进行帧间差分的结果。从图中可以看出, 帧间

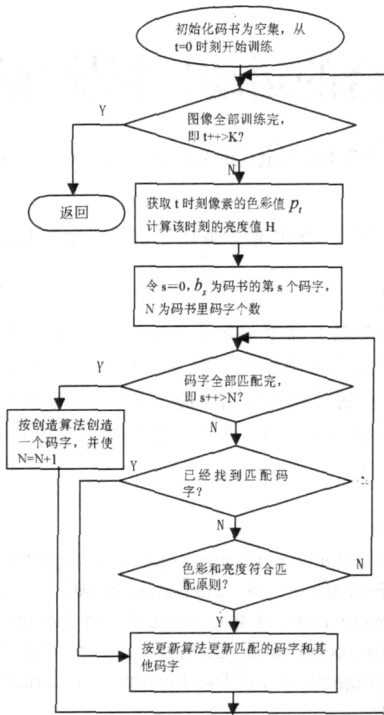


图 1 流程图

差分以后得到的变化区域包含了前景物体，也包含前一帧运动物体覆盖区域和阴影部分。对图 2(b)进行进一步的处理，将变化区域与背景模型进行拟合，得到图 2(c)。图 2(c)去除了前一帧图像运动物体覆盖的区域，留下前景物体和阴影。最后，将阴影从变化区域中移除，得到纯净的运动物体，结果如图 2(d)所示。

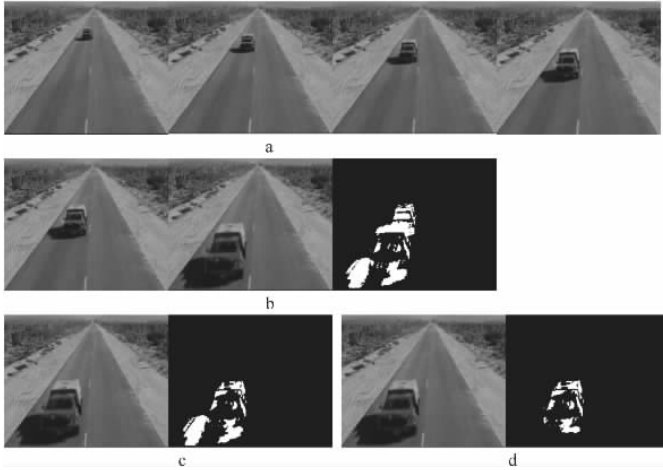


图 2 运动目标图像

实验在 PC 机上运行，原图像大小为 704 × 384 像素，实验每秒能处理 16 帧图像。从实验结果可以看出，该方法能较好地满足实时性要求，实验结果准确、有效。

参考文献

[1] DNG GG, 研究 GUO BL. New Fast Motion Estimation Algorithm Based on Line Search [J], Journal of xi'an JiaoTong University, 2004, 38(2): 36- 39

[2] 秦莉娟. 基于内容的自动视频监控研究[D]. 杭州: 浙江大学计算机科学与技 术, 2006: 22- 31

[3] M. Proesmans, L. Van Gool, E. Pauwels and A. Oosterlinck. Determination of optical flow and its Discontinuities using non- linear diffusion, in 3rd European Conference on Computer Vision, ECCV'94, Volume 2, pages 295- 304, 1994

[4] Chen Shao Yi, Ma Shyh- Yih, Chen Liang- Gee. Efficient moving object segmentation algorithm using background registration technique [J], IEEE Transactions on Circuits and Systems for Video Technology, 2002, 12 (7): 577- 586

[5] Lee D S, Hull J, Erol B. A Bayesian framework for gaussian mixture background modeling [A], In: Proceedings of European Conference on Computer Vision[C], Dublin, Ireland, 2000, 2: 336- 350

[6] Elgammal A, Duraiswami R, Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance, in Proc IEEE, Vo1 90, No.7, July 2002

[7] Stenger B, Ramesh V, Paradios N, Coetzee F, Bouhman J. Topology free hidden markov models: Application to background modeling. in Proc IEEE International Conference Computer Vision, 2001

[8] Haritaoglu I, Davis Larry S, Harwood D. W4 Who? When? Where? What? A real time system for detecting and tracking people[A]. In: Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition[C]. Nara. Japan, 1998: 222- 227

[9] Prati A, Mikic I, Trivedi M, et al Detecting moving shadows algorithms and evaluation [J], IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(7): 918- 923

[ 收稿日期: 2007.12.10]

(上接第 42 页)

它们的可重用性。为了避免这种情况，系统软件的开发中引入了 Observer 模式。

Observer 模式定义了对象间一对多的依赖关系，当一个对象的状态发生变化时，所有依赖它的对象将得到通知并自动更新。这个模式的关键是目标 (Subject) 和观察者 (Observer)。一个目标可有任意数目的依赖它的观察者，当目标改变时，所有观察者将得到通知。作为对这个通知的响应，每个目标都将查询目标，以使自身状态与目标状态保持同步。实现 Observer 模式的类图如图 3 所示。

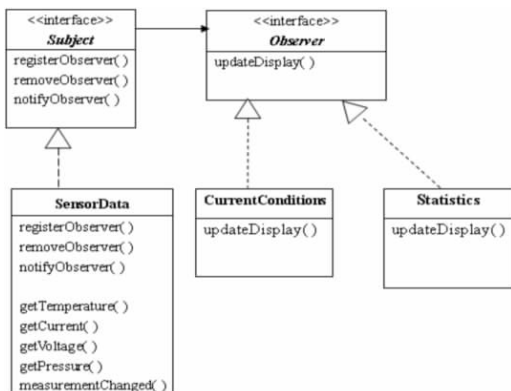


图 3 实现 Observer 模式类图

图 3 中的 SensorData 类实现了 Subject 接口，而 CurrentConditions 类和 Statistics 类都实现了 Observer 接口，Sensor-

Data 对象为目标，而 CurrentConditions 和 Statistic 对象为观察者，SensorData 类中的 registerObserver( ) 方法将给定的观察者对象添加到自己的观察者列表中，也即 Subject 对象注册了它的 Observer 对象。这样当 SensorData 对象的状态发生改变时，SensorData 对象通过 notify( ) 方法遍历自己的 Observer 对象 (这里为 CurrentConditions 和 Statistics 对象) 列表，调用它们的 updateDisplay( ) 方法实现对显示数据的修改，即实现了 Subject 对象向 Observer 对象发出通知的功能。

Observer 模式降低了传感器采集到的数据信息和它们的显示方式之间的耦合度，实现了数据信息和显示方式的分离，当有新的显示需求出现时，只需要添加一个实现 Observer 接口的类就能在不修改原系统的基础上满足需求，提高了系统的可扩展性。

3 结束语

通过在真空炉控制系统软件的开发中引入设计模式，使得软件开发领域的优秀思想和优势得以体现，并且也使控制系统的灵活性，可维护性和可扩展性得到了增强，提高了系统的适应性。

参考文献

[1] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software[M]. 北京: 机械工业出版社, 2002

[2] Eric Freeman & Elisabeth Freeman, Head first Design Patterns(影印版)[M]. 南京: 东南大学出版社, 2005

[ 收稿日期: 2007.12.3]