

一种基于模型的扫换检测方法*

金红, 周源华

(上海交通大学 图像通信与信息处理研究所, 上海 200030)

E-mail: jinhong@cdiv.org.cn

http://www.sjtu.edu.cn

摘要: 视频自动分割是实现视频数据库检索必不可少的一个过程, 其基础是镜头边界检测。当前已有的算法能够较准确地检测出镜头突变, 但对于镜头的渐变则常常会漏检, 这是由于镜头渐变时帧间差没有一个明显的峰值, 因而其检测比突变检测要困难得多。扫换是一种常用的视频空间编辑手段, 用于实现多种镜头变化。通过分析各种类型的扫换, 提出了一种新的基于视频空间编辑模型的扫换检测算法, 其性能优于 Alattar 提出的基于统计特征的算法。对用 Adobe Premiere 5.1 生成的各种扫换视频进行检测。实验结果表明, 这种扫换检测算法能够较好地适应不同类型和持续时间的扫换。

关键词: 视频数据库; 视频检索; 镜头边界的检测; 渐变; 扫换

中图法分类号: TP391 **文献标识码:** A

随着多媒体技术的发展和信息高速公路的出现, 数字视频已成为多媒体服务中不可缺少的一部分, 如视频点播、电子购物和电子图书馆等。由于视频数据具有语义复杂和数据量大的特点, 要提供上述服务, 首先必须对视频进行处理, 实现自动索引, 否则大量的视频数据将难以使用和管理。

镜头边界的检测是实现视频自动索引的第 1 步, 它直接影响到视频检索的成败和精度。镜头的转换可分为突变和渐变两种, 突变是从一个镜头直接切换为下一个镜头, 中间没有时间上的延迟, 渐变是为了改善视觉效果, 通过某种编辑手段(可分为空间编辑和色度编辑两类)从上一个镜头逐渐地转换为下一个镜头。目前的镜头边界检测算法对于突变一般都能取得较好的效果, 因为在镜头切换处, 帧间差有一个明显的峰值, 通过设置适当的阈值就能检测出镜头突变。但渐变转换是一个逐渐的过程, 帧间差没有一个明显的峰值, 常常会发生漏检。

近年来, 文献中出现了一些渐变检测算法。Zhang 等人^[1]提出用双阈值法来检测渐变; Zabih 等人^[2]使用进入和消失边缘点所占的比例来检测淡入/淡出; Yeo 和 Liu^[3]通过检测延迟的帧间差是否出现“高原”来判定镜头的渐变; Hampapur^[4]使用了前后帧之间相对亮度的变化来检测淡入/淡出; Alattar^[5]利用扫换过程中视频帧均值和方差的线性变化规律来检测扫换。前 3 种方法都要求仔细地选择阈值, 选择不当会使检测效果大为下降。Hampapur 的方法仅适用于淡入/淡出, 而不能用于慢转换和扫换。Alattar 的方法假设在扫换过程中, 前后两个镜头出现在画面上的区域, 其均值始终不变。这个假设在大多数实际视频中都不能成立, 因为局部与整体的均值一般是不相等的。用自然图像生成的视频进行实验, 文献[5]中介绍的方法基本上会失败, 只有用局部与整体极为相似的图像所生成的视频才能取得成功。

扫换是视频空间编辑方法的统称, 是一类常用的镜头渐变方法。Adobe Premiere 5.1 中提供的

* 收稿日期: 1999-09-27; 修改日期: 1999-12-11

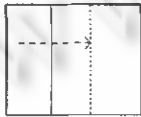
作者简介: 金红(1968—), 女, 浙江杭州人, 高级工程师, 主要研究领域为视频与图像检索; 周源华(1940—), 男, 江苏人, 教授, 博士生导师, 主要研究领域为图像处理与通信, 模式识别, 多媒体技术。

扫换方法达六七十种之多,包括随机扫换(random wipe),滑动扫换(slide),Split,Random blocks等等,可以提供非常丰富的视觉效果.文献中只有 Alattar 对扫换的结构进行了研究,但是他所提出的方法并不适用于用自然图像生成的视频.本文提出了一种新的基于空间编辑模型的扫换检测算法.实验结果表明,对用自然图像生成的视频序列该算法能成功地检测出扫换的起始和结束帧,并且对扫换的类型和扫换的速度都有较强的适应性.

1 扫换的模型和检测算法

扫换是一类常用的镜头渐变方法.它采用空间编辑的方式,即上一个镜头画面的一部分逐渐地被下一个镜头的画面所代替,直到全部变成第 2 个镜头的画面.第 2 个镜头在画面上所占的区域随时间单调递增.

分析各种扫换模型,可以总结出以下规律:扫换可大致分为两类,一类是第 2 个镜头在画面上所占区域的面积线性地增加,如 Slide 和 Bardoor 等;另一类是下一个镜头画面的面积以平方递增,如 spin 和 iris round 等.如图 1 所示,我们分别称(a)和(b)为线性扫换和二次扫换.



(a) The area in the left is occupied by the new shot, it increases linearly with time (slide effect)

(a) 左边为新镜头所占的区域,其面积随时间线性地增加(以slide为例)



(b) The area in the circle is occupied by the new shot, it increases quadratically with time (iris round effect)

(b) 圆圈内为新镜头所占的区域,其面积随时间以平方关系增加(以iris round为例)

Fig.1 Illustration of wipe effects

图1 扫换效果示意图

对于数字视频,区域的大小与区域内像素的个数成正比.

设 N 为一帧视频的像素个数,则可以用 N 维向量表示视频帧的内容.令 $F(n)$ 为 $1 \times N$ 向量,表示前一个镜头的视频序列中的第 n 帧; $G(n)$ 为 $1 \times N$ 向量,表示后一个镜头的视频序列中的第 n 帧; $M(n)$ 为一个 $N \times N$ 的对角矩阵,其对角线上 1 的个数为 $C(n)$,0 的个数为 $N - C(n)$, $C(n)$ 是一个单调递增的函数; $H(n)$ 为 $1 \times N$ 向量,表示经过扫换编辑后得到的视频序列的第 n 帧.

设扫换过程从第 K 帧开始,到第 L 帧结束,则 $H(n)$ 可以表示为

$$H(n) = \begin{cases} F(n) & n < k \\ F(n)(I - M(n)) + G(n)M(n) & k \leq n < L, \\ G(n) & n \geq L \end{cases} \quad (1)$$

其中 I 是 $N \times N$ 维的标识矩阵.

$c(n)$ 的定义与不同的扫换类型有关,它表示了扫换过程中第 2 个视频在画面上所占区域中的像素个数.而 $c(n)$ 的一阶导数 $dc(n)/dn$ 则反映了变化区域(即实线与虚线之间)中像素的个数.

我们采用以下方法计算 $dc(n)/dn$.

(1) 计算前后帧之差,得到差值图像 $D(n) = |H(n+1) - H(n)|$,在画面由第 1 个镜头变为第 2 个镜头的区域内,对应的元素差值较大,而在其他区域,差值为 0 或较小.

(2) 通过阈值比较从 $D(n)$ 得到一个二值图像 $D_T(n)$,即

$$D_T(n, i) = \begin{cases} 1 & \text{if } D(n, i) \geq T \\ 0 & \text{otherwise} \end{cases}$$

T 的设置可以去除由大多数镜头内小运动引起的帧间差.

(3) 对 $D_T(n)$ 中的一元进行计数, 得到 $dc(n)/dn = \sum_i D_T(n, i)$. 假设在扫换时, 前后两个镜头中没有运动, 则 $dc(n)/dn$ 可以表示变化区域中的像素个数.

对应于图 1 的两种情况, $dc(n)/dn$ 的变化规律如图 2 所示.

图 2(a) 是以 slide 为代表的线性扫换, $dc(n)/dn$ 在扫换过程中为一个正的常数; 图 2(b) 是以 iris round 为代表的二次变化的扫换, $dc(n)/dn$ 在扫换过程中先近线性地增加, 最后再快速回落.

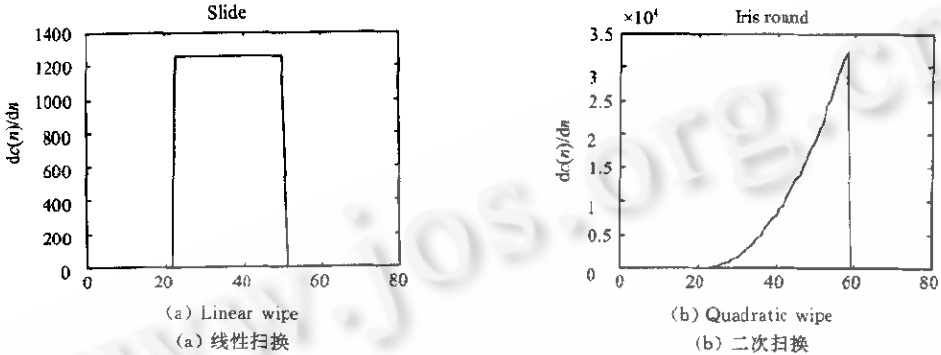


Fig. 2 $dc(n)/dn$

图 2 $dc(n)/dn$

$|d^2c(n)/dn^2|$ 的曲线如图 3 所示. 图 3(a) 表明, $|d^2c(n)/dn^2|$ 在线性扫换的开始和结束处各有一个峰值, 可用来界定扫换的区域. 图 3(b) 表明, $|d^2c(n)/dn^2|$ 在二次扫换过程中为一个正的常数, 并在结束之前出现一个负的大峰值.

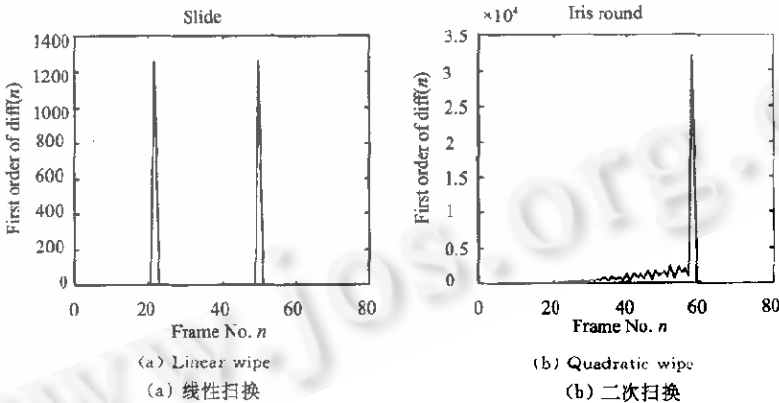


Fig. 3 $d^2c(n)/dn^2$

图 3 $d^2c(n)/dn^2$

对用 Adobe Premiere 5.1 生成的一次扫换(以 band slide 为例)和二次扫换(以 iris square 为例)的实际视频, 按上述算法分别计算 $dc(n)/dn$, 其变化曲线如图 4 所示. 对于同样的视频数据, Alttar 的方法所得结果如图 5 所示, 可见 $dc(n)/dn$ 和 $d^2c(n)/dn^2$ 能够更好地与图 2 和图 3 相吻合.

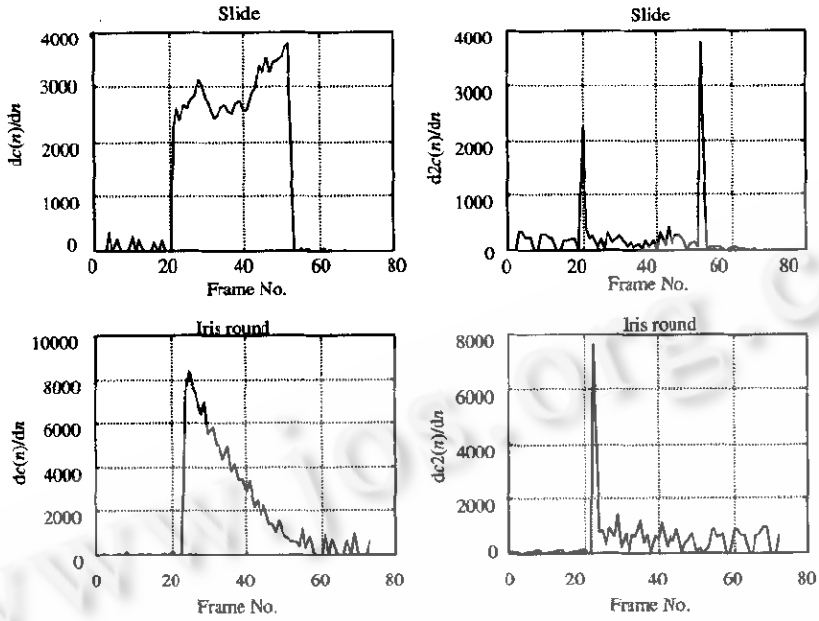


Fig. 4 $dc(n)/dn$ and $d2c(n)/dn$

图 4 $dc(n)/dn$ 及 $d2c(n)/dn$

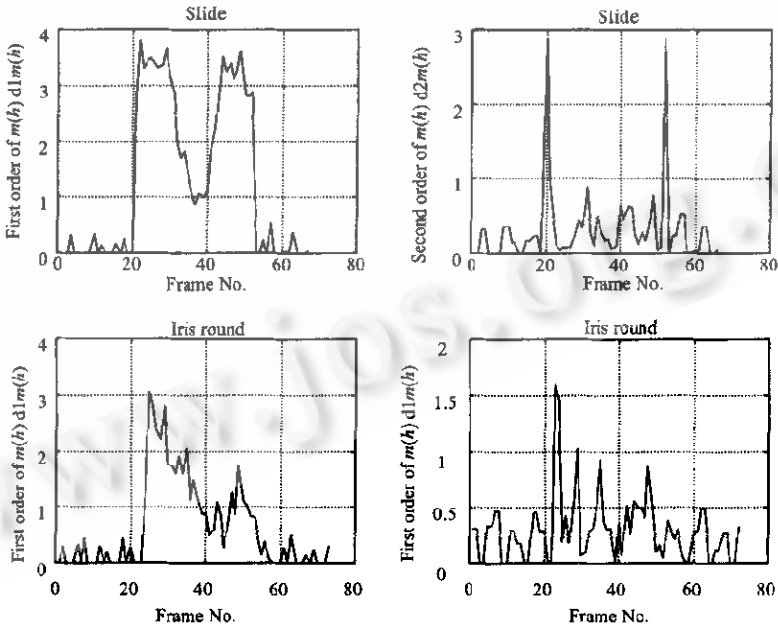


Fig. 5 $d1m(n)/dn$ and $d2m(n)/dn$

图 5 $d1m(n)/dn$ 和 $d2m(n)/dn$

2 扫换的检测算法

根据以上分析,我们设计的扫换检测算法如下:

(1) 计算各帧中 8×8 小块的均值, 以此作为一个像素点值, 得到 $N/64$ 维的向量 $H'(n)$, 这样做的目的是为了减少噪声和运动的影响。

(2) 计算各帧的 $dc(n)/dn$ 和均值 $m(n)$ 。

(3) 计算 $dc(n)/dn$ 和 $m(n)$ 的一阶方差, 即 $d2c(n)/dn$ 和 $dm(n)/dn$ 。

(4) 检测出 $d2c(n)/dn$ 中所有大于阈值 $T2$ 的峰值, 并把相应的帧号记录到数组 $S(m)$ 中。

(5) 对于 $S(m)$ 中任意两个相邻的值 $S(m0)$ 与 $S(m0+1)$, 计算两个峰值之间各帧 $dc(n)/dn$ 的均值, 如果大于阈值 $T3$, 则认为 $S(m0)$ 与 $S(m0+1)$ 之间各帧有一次线性扫换, 修改 $m0=m0+2$ 。否则, 检测 $S(m0)$ 的两边, 如果任意一边在一个可调节的窗口范围内, $d2c(n)/dn$ 的均值大于阈值 $T3$, 则认为有一次二次扫换。窗口的大小通过 $dc(n)/dn$ 大于阈值 $T4$ 来确定, $m0=m0+1$ 。如果不是上述两种情况, 则去掉 $S(m0)$, 因为这很可能是镜头内物体的运动, $m0=m0+1$ 。

(6) 去掉长度小于 3 帧的扫换, 因为这很可能是误检测。

3 实验结果

Table 1 Wipe detection result for different types of wipe

表 1 不同类型扫换的检测结果

Video sequence name/wipe type ^①	Actual wipe range ^②	Detected wipe range ^③
Band slide	23~45	23~45
Band wipe	23~45	23~45
Barn doors	23~45	23~45
Center merge	23~45	23~42
Center peel	23~45	25~41
Center split	23~45	23~41
Checker wipe	23~45	23~45
Clock wipe	23~45	23~25 27~29... 43~45 *
Clock wipe	23~45	23~46
Cross stretch	23~45	23~45
Cube spin	23~45	23~45
Curtain	23~45	23~45
Fold up	23~45	22~46
Inset	23~45	27~48
Iris round	23~45	27~41
Iris square	23~45	27~46
Page peel	23~45	26~42
Page turn	23~45	25~43
Pinwheel	23~45	43~45 *
Push	23~45	23~45
Radial wipe	23~45	23~45
Random wipe	23~45	22~45
Spin	23~45	22~46
Split	23~45	23~34 36~45 *
Swap	23~45	23~45
Swirl	23~45	26~47
Venetian Blinds	23~45	missed ^④
Wipe	23~45	23~46
Zoom	23~45	27~48
Zoom Boxes	23~45	27~47

①视频序列名/扫换类型, ②实际扫换范围, ③检测到的扫换范围, ④没有检测到。

表 1 是利用扫换检测算法对用 Adobe Premier 5.1 生成的 30 种扫换视频片段进行测试的结

果. 对于大多数的线性扫换, 该算法都能取得很好的效果; 对于二次扫换, 也能成功地检测, 只是在扫换的开始, $dc(n)/dn$ 很小, 因此 $T4$ 不能太大, 但是如果 $T4$ 太小, 又会与镜头内运动混淆. 另外, 对于 Checkboarder, Split, Venetian Blinds 这 3 种扫换, 由于在扫换过程中每间隔两三帧就出现停顿, $dc(n)/dn$ 接近 0, $dc(2n)/dn$ 中频繁地出现峰值, 从而导致检测失败.

利用上述扫换检测算法对用 Adobe Premier 5.1 生成的不同持续时间的扫换进行测试, 结果见表 2, 说明该算法能适应不同的扫换速度.

Table 2 Wipe detection result for wipes with different duration
表 2 不同持续时间的扫换检测结果

Video sequence name ^①	Actual wipe range ^②	Detected wipe range ^③
Band slide 1	9~13	9~13
Band slide 2	9~20	21~30
Band slide 3	9~59	9~39
Band slide 4	9~94	8~94
Band slide 5	9~139	9~140
iris square 1	9~13	9~13
iris square 2	9~20	11~20
iris square 3	9~59	16~60
iris square 4	9~94	18~96
iris square 5	9~139	26~143

①视频序列名, ②实际扫换范围, ③检测到的扫换范围.

4 结论与讨论

本文分析了扫换的过程, 提出一种新的扫换检测算法. 实验结果表明, 它能适应大多数类型的扫换和不同持续时间的扫换. 但是对于在扫换过程中出现停顿的情况, 算法检测失败. 另外, 当扫换过程中有较大的运动时, 算法的性能也会下降. 上述算法可以和突变检测算法及淡入、淡出算法集成使用, 以提高镜头边界检测的性能.

References:

- [1] Zhang, H. J., Kankanhalli, A., Smoliar S. Automatic partitioning of full-motion video. *Multimedia Systems*, 1993, 1(1): 10~28.
- [2] Zabih, R., Miller, J., Mai K. A feature-based algorithm for detection and classifying scene breaks. In: Zellweger, P., ed. *Proceedings of the 4th ACM Conference on Multimedia*. San Francisco, CA: ACM Press, 1995. 189~200.
- [3] Yeo, B. L., Liu, B. Rapid scene analysis on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 1995, 5(6): 533~544.
- [4] Hampapur, A., Jain, R., Weymouth, T. E. Production model based digital video segmentation. *Multimedia Tools and Applications*, 1995, 1(1): 9~46.
- [5] Alattar, Adnan M. Wipe scene change detector for use with video compression algorithm and MPEG-7. *IEEE Transactions on Consumer Electronics*, 1998, 44(1): 43~51.

A Model-Based Wipe Scene Change Detection Method*

JIN Hong, ZHOU Yuan-hua

(Institute of Image Communications and Information Processing, Shanghai Jiaotong University, Shanghai 200030, China)

E-mail: jinhong@cdtv.org.cn

http://www.sjtu.edu.cn

Abstract: Automatic video segmentation, which is based on SBD (shot boundary detection), is one necessary

step in establishing video databases. Existing SBD algorithms can detect abrupt scene changes quite accurately but may fail for gradual scene changes. This is due to the fact that there are no apparent peak values of frame difference during gradual scene change periods and hence it is more difficult to detect them. Wipes are frequently used to edit video in spatial domain to generate various kinds of scene changes. Through analysis of various wipes, a new algorithm is proposed in this paper based on a spatial editing model for wipe scene change detection. This method is better than the statistic-based method proposed by Alattar. Test results for video sequences generated by Adobe Premiere 5.1 containing wipe scene changes show that the algorithm is well suited for various types and durations of wipes.

Key words: video database; video retrieval; shot boundary detection; gradual change; wipe scene change

* Received September 27, 1999; accepted December 11, 1999