

基于类间距离和准则的镜头分割

郭智强, 黄志华

(福州大学 数学与计算机科学学院, 福建 福州 350002)

[摘要] 镜头切换检测是视频检索研究领域一项基本的、重要的技术. 然而传统的镜头边界检测方法都只考虑了视频相邻两帧的信息, 没有充分考虑到视频序列中前后的连续性, 从而容易导致误检测. 基于此, 提出了一种基于类间距离和准则对镜头边界进行检测的新方法, 该方法能够充分利用视频序列中的连续性信息, 克服传统镜头边界检测方法的不足, 有效地提高了检出率及精确率. 大量实验表明, 该方法是行之有效的.

[关键词] 视频检索, 镜头分割, 类间距离

[中图分类号] TP 391.41 [文献标识码] A [文章编号] 1672-1292(2008)04-0108-04

Shot Detection Base on Maximum-Between-Cluster-Distance

Guo Zh iqiang Huang Zh ihua

(College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350002, China)

Abstract The shot change detection is a basic and important technology in video retrieval. But the traditional shot boundary detection methods only consider the two adjacent video information, and do not take fully into account the previous and consequent video sequences in the continuity, and thus easily lead to error detection. Therefore a new shot boundary detection algorithm based on maximum-between-cluster-distance is proposed. The method can make full use of video sequences in the continuity of information, and overcome the traditional boundary-less than the detection method. Large number of experiments shows that the algorithm has effectively improved the detection rate and precision.

Key words video detection, shot boundary detection, class distance

随着多媒体和 Web 技术的发展, 越来越多的信息以视频形式表示. 如何在海量的视频信息中快速、方便地获得自己需要或感兴趣的内容已经成为广大用户关注的问题. 基于内容的视频检索也成为近年来研究的热点. 目前, 基于内容的视频检索, 一般都是先进行镜头分割^[1], 即在时域上将视频序列按照一定的标准分割为镜头 (Shot), 然后在镜头分割的基础上提取各个镜头的关键帧 (Key Frame) 以供检索. 因此, 如何将视频序列正确地分割为镜头是视频检索首先要解决的问题, 镜头分割的好坏直接影响到视频检索系统性能的优劣^[2]. 传统的镜头边界检测方法都只考虑了视频相邻两帧的信息, 没有综合考虑视频序列中前后的连续性信息, 从而容易导致误检测. 为此, 本文提出基于类间距离和规则的视频分割方法.

1 传统的镜头边界检测算法^[3-7]

模板匹配法是最简单的一种镜头边界检测算法. 利用帧间灰度或者颜色差值计算帧间差别, 判断总的帧差是否大于某个设定的阈值, 如果大于则认为发生了镜头突变. 这种方法原理简单、便于实现, 但是这种方法对于视频中物体、相机的运动十分敏感, 从而容易导致误检测. 直方图描述了一幅图像的灰度和颜色的分布情况, 通过直方图的相似度也可以判断图像之间的相似情况, 并且以此来判断是否有场景切换. 由于直方图不考虑像素的位置信息, 因此它对物体、相机的运动并不敏感, 从而在一定程度上克服了帧差法的缺点. 但是, 直方图是对图像整体情况的描述, 并不能反映出图像的空间信息, 而且对于视频序列中亮度的突然变化, 这种方法也常常产生误判.

收稿日期: 2008-06-18

通讯联系人: 黄志华, 副教授, 研究方向: 数据库系统. E-mail: hzh@fzu.edu.cn

2 基于类间距离的镜头边界检测

上述两种传统的镜头边界检测方法都只考虑了视频相邻两帧的信息, 没有充分考虑到视频序列中前后的连续性, 基于此, 我们引入聚类中计算类间距离的思想, 充分利用视频序列中的前后多帧相关性信息来进行边界检测. 如图 1, 在检测视频序列第 i 帧是否是镜头边界时, 考虑了它的前面和后续的多帧相关信息.

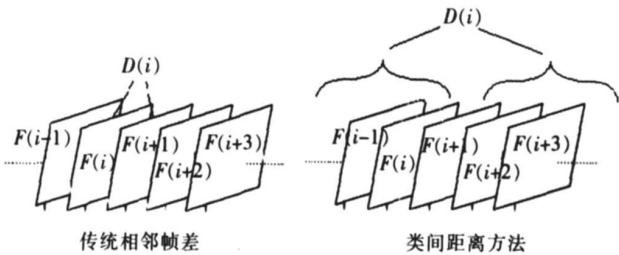


图 1 视频序列的帧信息
Fig.1 Frames of video sequence

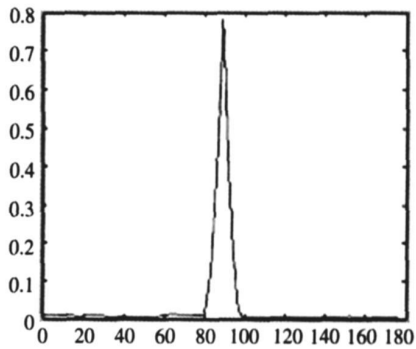


图 2 视频片段的 DFCD 值
Fig.2 DFCD values in a video clip

2.1 相关定义

聚类中的类间距离^[8]可用于描述聚类结果的类间距离分布状态. 两类问题 w_1/w_2 类间距离常用下式计算: $J_b = (m_1 - m_2)^T (m_1 - m_2)$, 式中, m_j 为 w_j 类型的样本均值向量; $m_j = \frac{1}{n_j} \sum_{j=1}^{n_j} x_j$. 类间距离和准则描述不同类型之间的分离程度, 所以 J_b 的值越大, 表示各类之间的分离性越好, 相似性越差.

本文采用计算两类的类间距离的思想, 把镜头分割处的前后两个序列当作两类, 镜头分割处前 n_1 帧为第一类, 后 n_2 帧为另一类. 对于视频序列中相邻的两个镜头, 由于同一镜头内的内容一般都是相似的, 而不同镜头之间的内容差异比较大, 因此, 对视频序列中每帧都进行类间距离的运算, 其分离程度 J_b 越高, 表示其发生镜头转移的可能性越大, 而在两镜头的边界处 J_b 值会达到最大, 表示在此处把视频分为两类效果最好. 为了更好的阐述本文的观点, 下面介绍一些相关的定义.

定义 1 帧分类距离: 所谓帧分类距离是指以某个视频帧为界, 把这个帧前面的若干个帧当作一个类, 把这个帧后面的若干个帧当作另一类, 然后按照以下的式 (1) 计算这两个类所得到的距离. 下面用 J_i 表示帧分类距离. 有:

$$J_i = \left[\sum_{j=0}^{n_h} \left(\frac{1}{n_1} \sum_{i=1}^{n_1} h(i, j) - \frac{1}{n_2} \sum_{i=n_1+1}^{n_1+n_2} h(i, j) \right) \right]^2, \tag{1}$$

其中, n_j 为第 j 类 $(j = 1, 2)$ 的帧个数, $h(i, j)$ 表示第 i 帧的第 j 个直方图向量.

定义 2 归一化: 所谓的归一化是把归纳样本值使其统一分布在一个小区间里.

定义 3 归一化后的帧分类距离: 所谓归一化后的帧分类距离是指对定义 1 的帧分类距离进行归一化后的结果. 归一化后, 就可以比较方便我们选取阈值. 归一化后的帧分类距离可用以下的公式表示.

$$f_i = \frac{1}{(MN)^2} \left[\sum_{j=0}^{n_h} \left(\frac{1}{n_1} \sum_{i=1}^{n_1} h(i, j) - \frac{1}{n_2} \sum_{i=n_1+1}^{n_1+n_2} h(i, j) \right) \right]^2, \tag{2}$$

式中, $M \times N$ 表示帧的像素数; n_h 表示每帧的直方图特征向量个数.

2.2 算法思想

提出算法 DFCD (detection by frame classification distance). 算法说明: 以一个大小为 $n_1 + n_2$ 的视频序列滑动窗口在整个视频序列上滑动. 这个滑动窗口在时间序列中分成两类. 取前面 n_1 帧表示视频序列类 w_1 , 取后面的 n_2 帧表示类 w_2 . 随着这个窗口在时间序列中的滑动, 如果有一个镜头边界帧进入, 则利用公式 2 计算这个帧的帧分类距离 f_i . 相对应的边界值的 f_i 会越来越大, 当这个帧处于窗口的中间时, f_i 达到最大, 然后再随着这个边界帧退出窗口, f_i 值又逐渐变小. 类间距离显示出来的是一个向上凸的一个波浪型曲线, 如图 2 所示. 图 2 中的视频序列在 89 帧处的帧分类距离取得最大值, 所以 89 帧是一个镜头分割点.

算法输入: MPEG 视频文件 MPEG; 算法输出: 镜头边界帧的位置.

算法过程:

- (1) 读取前 $2n$ 帧的直方图特征向量, 计算前 n 帧的样本均值, 计算后 n 帧的样本均值;
- (2) 计算式 (1) 并保存;
- (3) 去掉滑动窗口中的第 1 帧特征向量, 读入窗口下一帧特征向量 (即滑动窗口在视频序列中向后移动一位);
- (4) 计算前 n 帧的样本均值, 计算后 n 帧的样本均值;
- (5) 计算式 1, 并保存;
- (6) 判断读入视频帧是否完了, 如果没有读完, 则转步骤 3 否则执行下一步;
- (7) 通过 (2) 式进行归一化, 选定一个阈值, 读入步骤 7 的结果数据, 若读入的数据小于这个阈值则继续读下一个数据, 若读入的数据大于阈值, 继续读下一个数据并且和之前的数据比较, 如果大于之前的数据则再读数据并比较, 直到读到的数据小于之前的数据, 则输出之前读入的数据. 这个步骤主要是找出那些大于阈值 曲线的峰值, 并标记这些峰值数据在视频序列中的所在位置 $shotframe$ 为镜头边界处.

3 实验结果及分析

3.1 样本集的处理

本算法是在 VC++ 6.0, windows XP 操作系统的环境下进行的实验. 在实验中, 我们选取各帧的 HSV 颜色直方图为各帧的特征向量, 每帧取 64×3 个特征向量. 滑动窗口的大小取值为 20 即在式 (4) 中, $n_1 = n_2 = 10$ 实验中我们使用了包含 CNN 和 CCTV 新闻, 以及含有大量广告的 CCTV4 故事片作为视频测试数据.

3.2 实验度量标准

实验过程中对镜头边界检测采用了通用的评价标准: 查全率 (recall) 和查准率 (precision),

查全率 = 正确检测数 / (正确检测数 + 漏检数),

查准率 = 正确检测数 / (正确检测数 + 误检数).

3.3 DFCD 算法和其它边界检测方法的效果比较

我们分别用同 3 种方法对一个新闻视频序列片段进行实验, 这个视频文件包括一些闪光和大物体移动的镜头帧, 其中发生镜头突变的地方分别是在第 33、137、197、258 帧, 而在第 9、30、47、87、121、317、393、411、570 帧有闪光出现, 如图 3 所示. 通过对比发现, 模板匹配 (左) 和直方图 (中) 的方法对视频文件的闪光处检测时, 很容易造成误检, 检测效果较差; 而利用类间距离 (右) 的算法只要取合适的判定阈值, 就能够有效地检测出镜头边界.

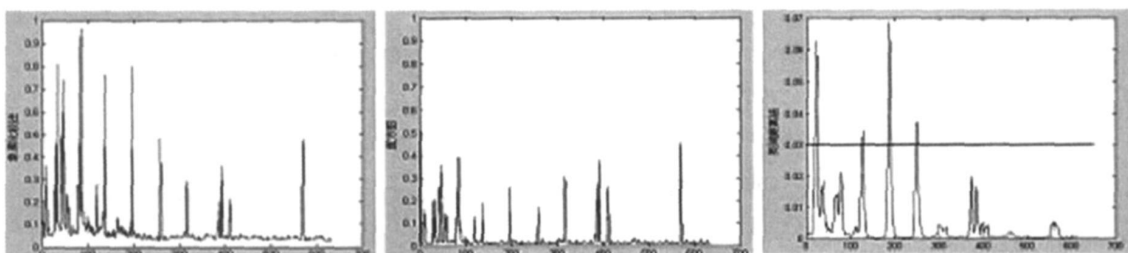


图 3 3 种方法的比较

Fig.3 Comparison of three methods

3.4 DFCD 算法实验结果

以文献 [9] 的直方图算法和本文的 DFCD 算法相比较, 图 4 为两种算法的查全率, 图 5 为查准率. 从实验结果我们可以看出, 利用基于类间距离的聚类方法在检查镜头分割的效果上比传统的颜色直方图方法能显著地提高. 因为类间距离准则的特性, 它对闪光和大物体运动也具有相当好的鲁棒性. 对于一个视频序列中发生闪光的帧, 如果仅用两帧之间的帧差来计算其相似度值, 由于两帧图像的颜色亮度信息差异较大, 其不相似度也会增大, 很容易造成误判. 而如果用类间距离的方法, 由式 (1) 可得它会把发生闪光处的

帧的相似度的值均衡,使其值变小,所以通常不会造成误判.

表 1 实验结果

Table 1 The result of the experiment					
算法	视频	帧数	镜头边界	漏检	误检
直方图算法	CNN新闻	112087帧	243	58	30
DFCD 算法				41	22
直方图算法	CCTV1新闻	50950帧	361	80	62
DFCD 算法				63	40
直方图算法	CCTV4故事片	104866帧	709	205	198
DFCD 算法				125	84

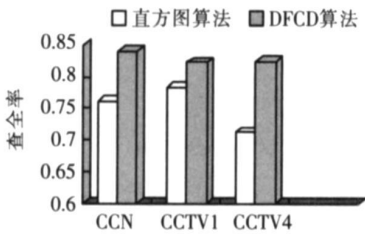


图 4 查全率结果
Fig.4 Comparison of recall

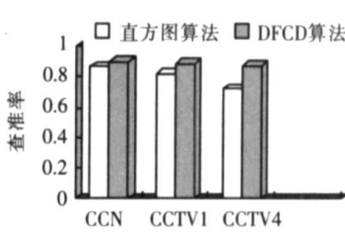


图 5 查准率结果
Fig.5 Comparison of precision

实验中,发现本文算法能够检查得到镜头突变和渐变,但是检测出来的渐变镜头,只是渐变镜头中的中心位置,并没有完整地显示渐变的起始帧和结束帧,这也是我们下一步要解决的问题.

4 结语

本文提出的基于类间距离的判断镜头分割方法,相比于传统的分割算法其性能有显著提高,能有效地检查出镜头的突变,并且对闪光和大物体运动具有良好的鲁棒性,在视频镜头分割中,具有一定的实用性.

[参考文献] (References)

[1] Ngo C W, Pong T C, Chin R T. Video partitioning through temporal slices analysis[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2001, 11(8): 941-953

[2] 林通, 张宏江, 封举富, 等. 镜头内容分析及其在视频检索中的应用[J]. 软件学报, 2002, 13(8): 1 577-1 585.
Lin Tong, Zhang Hongjiang, Feng Jufu, et al. Shot content analysis for video retrieval applications[J]. Journal of Software, 2002, 13(8): 1 577-1 585. (in Chinese)

[3] Boon-Lock Yeo, Bede Liu. Rapid scene analysis on compressed video[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1995, 5(6): 533-544

[4] Idris F, Panchanathan S. Review of image and video indexing techniques[J]. Journal of Visual Communication and Image Representation, 1997, 8(2): 146-166

[5] Courtney J D. Automatic video indexing via object motion analysis[J]. Pattern Recognition, 1997, 30(4): 607-625

[6] Yasuyuki Nakajima. A video browsing using fast scene cut detection for an efficient networked video database access[J]. IEEE Transactions on Information and Systems, 1994, E77- D(9), 1 355-1 364
for video database. In: Proc ICASSP 96, Atlanta GA, 1996

[7] Chang C W, Lee S Y. Video content representation, indexing and matching in video information system[J]. Journal of Visual Communication and Image Representation, 1997, 8(2): 107-120

[8] 边肇祺, 张学工. 模式识别[M]. 2版. 北京: 清华大学出版社, 2000
Bian Z Q, Zhang X G. Pattern Recognition[M]. 2nd ed. Beijing: Tsinghua University Press, 2000. (in Chinese)

[9] 丁洪丽, 陈怀新. 基于累积直方图的视频镜头边界检测方法[J]. 电讯技术, 2008, 48(3): 65-69.
Ding Hongli, Chen Huaxin. Video shot boundary detection method based on cumulative histogram[J]. Telecommunication Engineering, 2008, 48(3): 65-69. (in Chinese)

[责任编辑: 刘 健]