一种基于镜头聚类的视频场景分割方法

王学军 丁红涛 陈贺新
(吉林大学通信工程学院, 长春 130012)

摘 要 为了更好地进行视频信息检索和浏览,提出了一种利用视觉和运动特征来进行场景分割的方法,该方法在把镜头聚类到场景中时,不仅考虑同一场景内镜头的视觉特征相似性,而且还考虑了镜头的运动特征的一致性。此外,为避免过度分割,还提出了一种方法用来合并过度分割出的场景。实验结果表明,这种方法是有效的。

关键词 场景分割 镜头相似度 镜头聚类
中图法分类号: TN919.8 **文献标识码**: A **文章编号**: 1006-8961(2007)12-2127-05

A Shot Clustering Based Approach for Scene Segmentation

WANG Xue-jun, DING Hong-tao, CHEN He-xin
(*School of communications engineering, Jilin University, Changchun* 130022)

Abstract In this paper, a scene segmentation method utilizing both visual and motion features is presented. Not only the visual similarity but also the motion consistency of shots within a scene is considered in clustering shots into scenes. In addition, we present a method to merge the over-segmented scenes. Experimental results show the effectiveness of our algorithm.

Keywords scene segmentation, shot similarity, shot clustering

1 引 言

随着以计算机和通讯为代表的信息技术的迅速发展,多媒体特别是视频越来越成为人们生活和工作中不可或缺的一部分。为了解决视频信息在检索和浏览等方面的困难,人们提出了基于内容的视频检索和分析技术,而视频分割则是基于内容的视频分析和检索技术的基础。视频分割是根据视频数据的低层特征(颜色、运动信息、纹理、人脸等),把视频分成一个个能表达一定语义的逻辑单元,以方便人们检索或浏览。这样按照逻辑语义分割视频所得到的视频单元就是场景(或逻辑故事单元)。根据目的和形式的不同,场景构造可分分类和聚类两种。分类指的是仅考虑镜头间的特征相似性,而不考虑时间上的连续性,例如,根据镜头内容的重复性,可区分为对话型和动作型,其中对话型动作比较固定,但对象重复交替出现,动作型镜头则跟随事件不固定在一个位置,很少有镜头的重复;聚类指的是把属于同一个场景的镜头结合起来,以形成视频层次结构。其不仅要考虑内容上的相似性,还要考虑时间上的连续性。本文的算法讨论的是第 2 种情况。

场景分割的第 1 步通常是先对镜头进行分割,然后是关键帧提取,最后把相似的镜头合并为场景[1]。把视频分割成场景后,每个场景就可以用若干个关键帧来表示这个场景的内容,这样相对于用镜头的关键帧表示镜头内容的方法来得更精练,其所要处理的数据更少。

目前大多数场景分割算法都采用比较镜头相似度的方法把相关的镜头聚类成场景[2]。其中比较有代表性的场景分割算法是时间受限的镜头聚类算法[3]和时间自适应分组法[4]。但时间受限的镜头聚类算法只考虑位于一个固定时间窗口 T(以视频帧为单位)内的镜头的相似性,而位于 T 外的镜头

基金项目:国家自然科学基金项目(60372060);科技部国际合作项目(2005DFA10300)
收稿日期:2005-09-10;**改回日期**:2006-09-20
第一作者简介:王学军(1965 ~ .),男,副教授,博士。主要从事多媒体数据库检索方向研究。E-mail:wangxuejun@email.jlu.edu.cn

的相似性则不予考虑(即相似度为零),其时间相似特性类似于一个矩形函数,因此聚类结果不够完全。为了克服时间受限镜头聚类算法的不足, Yong Rui 等提出了时间自适应分组法[4],即镜头的相似度随着它们之间的时间距离的改变而变化,距离越大,相似度越小。Alan Hanjalic 和 Wallapak Tavanapong 采用了比较相近的聚类算法来构造场景 c[5,6]。Alan Hanjalic 先把镜头的关键帧合并为一个图像,然后对合并后的图像分块,求出镜头关键帧图像之间距离最小(相似度最大)的 B 个块,最后把这些最小距离的平均值作为镜头之间的不相似度,再根据求得的不相似度,应用镜头重叠链接的算法(overlapping links connecting similar shots)对镜头聚类。而 Wallapak Tavanapong 则先把图像分成几个区域,每个区域都从不同角度体现电影场景的特征,然后通过依次比较对应区域之间的相似度来确定镜头的相似度,再应用镜头链算法提取场景。但是以上两种聚类方法都只使用了图像的局部颜色特征,并且后者在提取颜色特征时只利用了 MPEG 视频帧 DC 图像的亮度分量。本文提出并实现了一种利用图像的全局颜色特征和运动特征来分割场景的方法,同时为了减小过度分割的影响,还提出了一种合并过度分割出的场景的方法,从而获得了最佳的视频场景分割效果。

镜头链聚类算法的特征是在计算一个镜头与其他镜头的相似度时,既考虑当前镜头与其后面若干个镜头的相似度,也考虑当前镜头前面的若干个镜头与当前镜头后面的若干个镜头的相似度。由于镜头的长度是变化的,所以需要计算相似度的镜头之间的最大时间距离(时间窗)也是变化的,从而使镜头相似度的计算是自适应的。

2 基于镜头聚类的视频场景分割方法

2.1 镜头检测和特征提取

镜头检测是视频分析的第 1 步。本文采用基于 MPEG 的宏块类型信息的方法来检测切变镜头[7]。其主要原理是在镜头边界前后,由于镜头的内容不相同,因此运动补偿往往失效。在镜头边界前的 B 帧图像由于与镜头边界后的参考图像不相似,其大部分宏块编码类型将为前向运动补偿类型,镜头边界后的 B 帧图像的大部分宏块编码类型将为后向运动补偿类型,而镜头边界后的 P 帧图像则大部分宏块类型为内部编码类型,因此只要统计出各帧图像中数量占大多数的宏块编码类型,就可以快速而准确地判断出镜头边界来,并且为了计算方便,可选择镜头的首尾两帧作为其关键帧。

镜头运动特征表征了镜头内容的变化程度,或者镜头内图像的不相似度。一般说来,镜头的运动程度越大,镜头内相邻图像之间的不相似度相对地越大;镜头运动程度越小,镜头内图像的不相似度越小。因此,可以根据镜头内图像的不相似度来定义镜头的运动特征。镜头 z 的运动特征 F_z^{m}(下角 m 代表 motion,下同)可表示为

$$F_z^{\mathrm{m}} = \frac{1}{b-a}\sum_{i=a}^{b-1}(1 - S(f_i, f_{i+1})) \tag{1}$$

式中,镜头 $z=\{f_a, f_{a+1}, \cdots, f_b\}$,$f_a$ 和 f_b 为镜头 z 的第 1 帧和最后一帧图像,$S(\boldsymbol{I},\boldsymbol{J})$ 为两幅图像 $\boldsymbol{I}$ 和 $\boldsymbol{J}$ 的颜色相似度:

$$S(\boldsymbol{I},\boldsymbol{J}) = \frac{1}{w\times h}\times\sum_{k=0}^{L-1}\min(H_I(k), H_J(k)) \tag{2}$$

其中,$H_I(k)$ 和 $H_J(k)$ 为图像 $\boldsymbol{I}$ 和 $\boldsymbol{J}$ 的 HSV 彩色直方图,w 和 h 分别为视频帧图像的宽度和高度,L 为直方图的量化台阶数。此处采用 HSV 彩色直方图的原因是因为 HSV 颜色空间与人们感知和解释颜色的方式最为接近,对用户感觉也很直观。在 MPEG 压缩域可以先用快速算法[8]提取出 DC 图像,然后根据 DC 图像的相似度计算镜头的颜色相似度。

镜头相似度表明了镜头之间的关联程度。相似的镜头不仅视觉特征相似,而且运动特征也具有一致性,因此,可以根据视觉和运动这两个特征来定义镜头的相似度。镜头 z_i 和 z_j 的相似度 $S_{\mathrm{shot}}(z_i,z_j)$[4] 定义如下:

$$S_{\mathrm{shot}}(z_i,z_j) = W_{\mathrm{color}}\times S_{\mathrm{v}}(z_i,z_j) + W_{\mathrm{m}}\times S_{\mathrm{m}}(z_i,z_j) \tag{3}$$

式中,W_{color} 和 W_{m} 分别是颜色和运动分量的权重。

视觉相似度 $S_{\mathrm{v}}(z_i,z_j)$(下角 v 代表 visual,下同)定义为

$$S_{\mathrm{v}}(z_i,z_j) = \max[S(b_i,e_i), S(b_i,b_j), S(b_i,e_j), S(e_i,e_j)] \tag{4}$$

式中,b_i 和 e_i 分别为镜头 z_i 的开始帧和结束帧。

镜头运动相似度 $S_{\mathrm{m}}(z_i,z_j)$ 定义为

$$S_{\mathrm{m}}(z_i,z_j) = \frac{2\times\min(F_{z_i}^{\mathrm{m}}, F_{z_j}^{\mathrm{m}})}{F_{z_i}^{\mathrm{m}} + F_{z_j}^{\mathrm{m}}} \tag{5}$$

颜色直方图相似度和运动特征相似度形成了总体的镜头相似度。由于颜色特征和运动特征属于不

同的物理空间,通过高斯归一化的过程可把不同值域的实体转换到相同的范围,因此在场景构造之前,可先通过归一化的方法来得到颜色直方图和运动特征的均值和平方差(μ_m,σ_m,μ_v,σ_v)。

$$\mu_v = \frac{1}{\sum_{k=1}^{F}(N-k)}\sum_{k=1}^{F}\sum_{i=0}^{N-k-1}S_v(z_i,z_{i+k}) \tag{6}$$

$$\sigma_v = \sqrt{\frac{1}{\sum_{k=1}^{F}(N-k)}\sum_{k=1}^{F}\sum_{i=0}^{N-k-1}(S_v(z_i,z_{i+k})-\mu_v)^2} \tag{7}$$

式中,N 为镜头的个数,F 为前向搜索范围。类似地,可以计算出镜头运动相似度 S_m 的均值 μ_m 和方差 σ_m。

高斯归一化以后,颜色直方图和运动特征的相似值都落在同样的取值空间,并且结合在一起用来表示相似度。为了反映运动特征的重要性大小,可为运动特征设定不同的权重。

$$W_v = \frac{\sigma_v}{\sigma_v+\sigma_m} \tag{8}$$

$$W_m = \frac{\sigma_m}{\sigma_v+\sigma_m} \tag{9}$$

这样镜头 z_i 和 z_j 的相似度的计算公式就变成

$$S_{shot}(z_i,z_j) = \frac{\sigma_v}{\sigma_v+\sigma_m}\times\frac{S_v(z_i,z_j)-\mu_v}{\sigma_v} + \frac{\sigma_m}{\sigma_v+\sigma_m}\times\frac{S_m(z_i,z_j)-\mu_m}{\sigma_m} \tag{10}$$

2.2　重叠镜头链算法步骤

根据上一节讨论的镜头之间相似度的表达公式,采用镜头迂回聚类算法来进行场景分割的具体步骤如下:

输入:F 为前向搜索范围;E 为后向搜索范围($F \geqslant E$);

输出:场景边界;

符号说明:$N_{preshot}$ 为当前镜头之前的镜头数目;$N_{futureshot}$ 为当前镜头之后直至视频最后一个镜头的镜头数目;

开始:将视频段的第 1 个镜头作为当前镜头;

(1) 前向比较:$r=\min(E,N_{preshot})$,$d=\min(F,N_{futureshot})$;在当前镜头之后 d 个镜头内查找与当前镜头最匹配的镜头;如果发现最匹配镜头,则所发现镜头与当前镜头之间的所有镜头都属于同一场景,并且匹配镜头变为当前镜头,跳至步步骤(1)否则执行步骤(2);

(2) 后向比较:z_i 为当前镜头;在镜头 z_{i+1},z_{i+2},…,z_{i-1+f}中寻找与镜头 z_{i-1} 最匹配的镜头;如果匹配镜头找到,则该镜头与镜头 z_{i-1} 之间的所有镜头都属于同一场景,并且匹配镜头变为当前镜头,跳至步骤(1);否则 $z_i=z_{i-1}$;重复步骤(2),直至当前镜头前的第 r 个镜头变为当前镜头;

(3) 宣布一个场景分割完毕,并且下一个镜头变为当前镜头;

重复以上步骤,直到所有镜头聚类完毕;场景分割结束。

下面举例来说明一下上述算法:设 F 为前向搜索范围,E 为后向搜索范围,并且 $F \geqslant E$,则重叠镜头链算法如下(分 3 种情况,如图 1 所示):

(1) 若镜头 z_{k_1} 与 z_{k_1+p}($p \leqslant F$)相似,则镜头 z_{k_1} 和 z_{k_1+p} 属于同一个场景 C_m,并且所有位于它们之间的所有镜头也属于场景 C_m。

(2) 在第 k_2 个镜头 z_{k_2} 后面 F 个镜头范围内没有镜头与 z_{k_2} 相似,但 z_{k_2} 前面的镜头 z_{k_2-t}($t \leqslant E$)与 z_{k_2} 后面的镜头 z_{k_2+q}($q \leqslant F-t$)相似,则镜头 z_{k_2-t} 和 z_{k_2+q} 以及它们之间的所有镜头(包括 z_{k_2})都属于场景 C_m。

(3) 如果对于镜头 z_{k_3} 来说,前两种情况都不成立,并且 z_{k_2} 属于第 m 个场景 C_m,则 z_{k_2} 为场景 C_m 的最后一个镜头。

(4) 开始下一个的循环,提取第 $m+1$ 个场景(如图 1 所示,图中虚线箭头所示的为自动包含的镜头)。

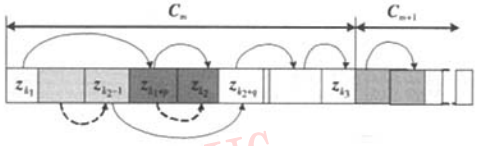

图 1　重叠镜头链算法示意图($F=3$,$E=2$)

Fig. 1　Overlapping links diagram($F=3$,$E=2$)

2.3　场景分割后续处理

F、E 和镜头相似度阈值的取值越大,则所分割得到的场景就越多,越容易造成过度分割;若取值越小,则分割得到的场景就越少,且将导致真实的场景检测不出来。一般而言,过度分割所造成的危害较小一些,因为被过度分割的场景可以通过某种方法恢复出来,而没有检测出的场景却是很难再恢复出

的[3]。为此,实验中采取了先让这些参数取比较大的数值,然后再对过度分割出的场景进行合并的方法。在合并场景时,基于这样的一个假定,即一个真实的场景所包含的镜头应该不少于 3 个,对于任何所包含的镜头个数小于 3 的场景都认为是误检,应进行合并。通过对电影中真实场景的研究,这个假定在大多数情况下是成立的。场景合并过程如下,首先找出镜头个数小于 3 的场景 C,和 C 最前面和最后面所包含镜头个数不小于 3 的场景 C_F 和 C_B,计算出场景 C 中镜头与 C_F 和 C_B 中镜头各自的最大相似度 S_F^{shot} 和 S_B^{shot},如果 $S_F^{shot} \geq S_B^{shot}$,则场景 C 合并到 C_F 中,否则把 C 合并到 C_B 中,这样重复下去,直至所有场景所包含的镜头个数都不小于 3 为止。

3 实验结果分析

为了验证本文所提出的算法的有效性,本文从两部电影“Sleepless in Seattle(西雅图不眠夜)”和“A Beautiful Mind(美丽心灵)”抽取了两个片段(MPEG-1 格式)作为测试视频。这两个视频片段的有关数据如表 1 所示。实验结果如表 2 和表 3 所示,分割的结果如图 2、图 3 所示。其中包括了非压缩域和压缩域处理的结果。在实验中,前向搜索范围 F 取值为 3,后向搜索范围 E 取值为 2。查全率(recall)和查准率(precision)定义如下:

$$R_{recall} = \frac{N_c}{N_c + N_d} \times 100\%$$
$$R_{precision} = \frac{N_c}{N_c + N_f} \times 100\% \quad (11)$$

其中,N_c 为检测到真实场景的个数,N_d 为没有检测到的场景个数,N_f 为检测到错误场景的个数。

表 1 测试视频的有关数据

Tab.1 Test videos

电影片段名称	“西雅图不眠夜”	“美丽心灵”
片长(s)	1951	2263
场景个数	15	19
镜头个数	217	449
切变镜头个数	217	431
渐变镜头个数	0	18

从实验数据可以看出,相似度阈值越大,检测到的真实场景越多,查全率增大,但错误检测到的场景

表 2 场景分割结果(“西雅图不眠夜”)

Tab.2 Scene segmentation results(“Sleepless in Seattle”)

阈值	N_c	N_d	N_f	查准率(%)	查全率(%)	
0.3	11	4	8	57.9	73.3	DC 图像
0.3	13	2	4	76.5	86.7	非压缩域
0.2	12	3	3	80.0	80.0	DC 图像
0.2	13	2	2	86.7	86.7	非压缩域

表 3 场景分割结果(“美丽心灵”)

Tab.3 Scene segmentation results(“A Beautiful Mind”)

阈值	N_c	N_d	N_f	查准率(%)	查全率(%)	
0.3	13	6	20	39.4	68.4	DC 图像
0.3	16	3	16	50.0	84.2	非压缩域
0.2	14	5	16	46.7	73.7	DC 图像
0.2	14	5	12	53.8	73.7	非压缩域

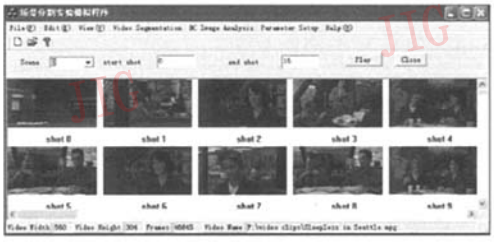

图 2 场景分割结果(“西雅图不眠夜”)

Fig.2 Scene segmentation results 3(“Sleepless in Seattle”)

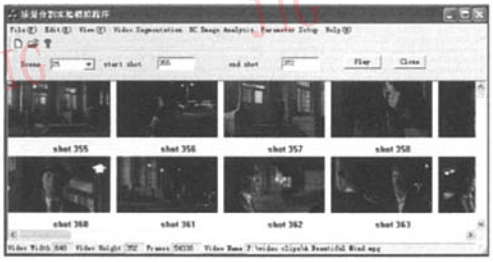

图 3 场景分割结果(“美丽心灵”)

Fig.3 Scene segmentation results 4(“A Beautiful Mind”)

也随之增多,从而造成查准率值减小。“A Beautiful Mind”的场景分割结果不如“Sleepless in Seattle”好,其中的一个原因是在“A Beautiful Mind”中有许多渐变镜头,而且有的场景边界就位于镜头渐变处,而本实验在镜头检测中,只检测了切变镜头,并没有检测渐变镜头。此外,非压缩域的场景分割结果明显优于压缩域(DC 图像序列),其主要原因是由于在求 P 和 B 帧 DC 图像时,使用了近似方法,从而引入

了误差噪声,使得镜头之间的区分度减小。但使用 DC 图像时,分割速度明显快于充分解码时的分割速度,这是它的一个突出优点。

作为对比,表 4 和表 5 给出了文献[6]中的测试视频数据和场景分割实验结果(MPEG 压缩域 DC 图像)。

表 4　测试视频数据

Tab. 4　Test videos

电影片段名称	Home Alone	Far and Away
片长(s)	6 224	7 965
场景个数	62	58
镜头个数	1 349	1 493
切变镜头个数	1 340	1 490
渐变镜头个数	9	3

表 5　场景分割结果

Tab. 5　Scene segmentation results

电影名称	N_c	N_d	N_f	查准率(%)	查全率(%)
Home Alone	32	30	173	51.6	15.6
Far and Away	36	22	213	62.1	14.4

在文献[6]中,F 和 E 的取值同样也为 3 和 2。为了与文献[6]的实验结果相比较,实验中应采取同一组测试序列,因素材不好获取,本文未能采用文献[6]中的两段视频,采用本文算法分割场景的查全率和查准率都明显较高,且分割效果更好。其原因是本文提出的算法对一次分割后的场景进行了后续处理,即对镜头个数小于 3 的场景进行了合并,这样就降低了漏检和误检的场景个数。

4　结　论

本文提出了一种利用镜头的视觉特征和运动全局特征进行场景分割的新方法。该方法在度量镜头的相似度时,除了考虑镜头的视觉特征外,还考虑了镜头的运动特征。此外,本文还对场景过度分割的情形提出了有效的处理方法。实验结果表明,本文提出的算法可以有效地分割视频场景。

参考文献(References)

1　Ngo Chong-Wah, Zhang Hong-jiang, Pong Ting-chuen. Recent advances in content based video analysis[J]. International Journal of Image and Graphic, 2001, **1**(3): 445 ~ 469.

2　Vendrig Jeroen, Worring Marcel. Systematic evaluation of logical story unit segmentation [J]. IEEE Transactions on Multimedia, 2002, **4**(4): 492 ~ 499.

3　Yeung Minerva, Yeo Boon-lock, Liu Bede. Segmentation of video by clustering and graph analysis [J]. Computer Vision and Image Understanding, 1998, **71**(1): 94 ~ 109.

4　Rui Yong, Huang Thomas S, Mehrotra Sharad. Constructing table-of-content for videos[J]. Multimedia Systems, 1999, **7**(5): 359 ~ 368.

5　Alan Hanjalic, Lagendijk Reginald L, Biemond Jan. Automatic high-level movie segmentation for advanced video-retrieval systems[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1999, **9**(4): 580 ~ 588.

6　Wallapak Tavanapong, Zhou Jun-yu. Shot clustering techniques for story browsing[J]. IEEE Transactions on Multimedia, 2004, **6**(4): 517 ~ 527.

7　Pei Soo-chang, Chou Yu-zuong. Efficient MPEG compressed video analysis using macroblock type information[J]. IEEE Transactions on Multimedia, 1999, **1**(4): 312 ~ 333.

8　Yeo Boon-lock, Liu Bede. On the extraction of DC sequence from MPEG compressed video [A]. In: Proceedings of International Conference on Image Processing ICIP[C], Washington, DC, USA, 1995: 260 ~ 263.