数字视频流的时序结构分析方法

周启龙 朱淼良 王东辉
(浙江大学计算机科学系, 杭州 310027)

摘 要 视频分析方法和技术的研究是有效组织和利用视频资源的基础. 为了有效地组织和利用视频资源, 以实现视频流内容自动分析, 提出了视频流的时序结构分析方法, 即通过对视频流的分解和快速聚类分析, 以便能够重构出视频流的时序结构. 实验结果表明, 用该方法得到的时序结构图反映了视频流中暗藏的故事发展线索, 适合于视频内容的非线性访问.

关键词 视频页 视频流分解 聚类分析 时序结构图
中图法分类号: TP391 **文献标识码**: A **文章编号**: 1006-8961(2000)09-0759-05

An Analysis Method for Building Sequence Structure of Video Stream

ZHOU Qi-long, ZHU Miao-liang, WANG Dong-hui
(*Department of Computer Science, Zhejiang University, Hangzhou* 310027)

Abstract The paper presents a new analysis method for effectively recognizing story structures of video programs. First, video stream are decomposed into sequences of video pages, and key frames are also extracted to represent video pages. Techniques and formulations are then proposed to match and cluster video pages of similar visual contents, taking into account the visual features and temporal arrangement of clustering elements. In addition, we use the Sequence Structure Graph representation to show the story development clue—— story line extracted from video. The proposed analyses lead to automatic segmentation of story units and the building of a compact representation of video contents. Hence we are able to decompose video into compact representations that reflect the flow of stories. This offers an efficient mean for browsing and non-linear access of video. Experimental results demonstrate the effectiveness of the new analysis method.

Keywords Video page, Video stream decomposing, Clustering analysis, Sequence structure graph

0 引 言

多媒体技术和网络技术的发展使得视频已成为与文本、图象等同样重要的信息表达形式, 其原因一方面是如今利用各种计算平台制作数字化视频变得越来越容易和越来越快速; 另一方面, 人们可以通过网络等方便地获取和访问到这些视频资源. 为了有效地组织和利用数字化视频资源, 实现视频流内容的自动分析、层次组织和语义理解等目的, 需要发展各种视频分析方法和技术.

视频流分解是目前视频内容分析的主要方法. 它的思路是沿时间维将视频流分解为若干视频分解单元, 然后将内容相关的连续视频图象划归同一视频分解单元中, 并用视频分解单元中的若干关键帧(key frames)来代表它的内容. 而由所有关键帧组成的关键帧集, 就可以充分表达被分解视频流的内容, 从而起到了浓缩视频流的作用. 相对于对整个视频流进行操作来说, 对关键帧集的操作可以显著地节省运算开销, 也更高效, 而且视频流分解的结果不仅可以直接用于视频流的快速浏览, 也可作为进一步分析的基础. 目前, 基于视频图象的可视特征(例如颜色、形状、纹理、运动)的镜头边界检测(shot boundary detection)是视频流分解方法的主要实现形式[1~3]. 可是, 镜头边界检测的主要问题就是在实际视频制作中常常运用的大量特技效果, 使得镜头的边界难以辨别, 而针对某些特技效果的检测模型又往往不适用于其它效果[2,3], 因此难以实现一个能

基金项目: 国家自然科学基金(69875015); 浙江省自然科学基金(111101N29809)
收稿日期: 1999-10-08; 改回日期: 2000-04-06

检测各种效果镜头的通用模型. 为此, 本文提出了视频页(video page)概念, 以它作为视频流的分解单元, 并利用视频流在视觉内容上的连续性和相似性来进行视频流的分解, 而不去考虑实际的镜头边界.

此外, 由于采用一维的图象数组形式的视频流分解方法, 所以不能充分体现视频分解单元之间的时序结构关系. 例如, 一段视频流能够描述在同一时段内几个事件的并行发展. 因此, 通过对视频流进行时序结构分析, 可获得相应的时序结构和对故事发展的抽象描述. 其中, 重复出现的视频分解单元可以通过视觉相似性来辨别.

文献[4]提出了一种聚类方法, 用于视频分解单元的进一步处理, 并用于视频流的浏览和注解. 该方法虽可以获得视频分解单元间的层次关系, 但未能体现视频分解单元在时间维上的发展关系, 即视频流的时序结构.

文献[5]提出了一种基于聚类和图分析的视频分析方法. 该方法通过聚类分析所得到的场景变换图(STG), 虽可以描述视频分解单元在时间维上的发展关系, 也能直观显示视频流的时序结构, 但该方法存在的主要问题是: ① 聚类过程的计算量很大, 因而降低了它的实用性; ② 它在采用时间约束的聚类方法时(time-constrained clustering), 是通过设定时间窗口(time window)的长度来限制参加聚类的视频分解单元数目, 然而, 这种约束是违背实际情形的, 因为视频流中, 相关的视频分解单元之间的时间距离是不存在统一的约束尺度的, 而且不同大小的时间窗口约束会导致不同的视频分析效果.

因此, 本文提出了一种新的聚类方法, 用于视频流的时序结构分析. 该聚类方法不仅具有较低的计算复杂度, 而且可以适应相关视频分解单元的时间距离的变化. 由此得到的时序结构图直观地反映了视频流中暗藏的故事发展线索—— 故事线(story line), 从而为视频流的非线性访问提供了可能.

1 视频流的时序结构分析方法描述

视频流的时序结构分析过程分为以下几个步骤:

(1) 分解视频流到视频页;

(2) 从视频页中提出关键帧;

(3) 视频页的聚类分析;

(4) 构造视频流的时序结构图.

为了提高效率, 在分解视频流的同时, 还进行了关键帧的提取.

1.1 视频流分解和关键帧提取

显然, 连续视频流可看作为沿时间维变化的图象序列. 视频流分解, 即是按图象视觉特征的变化, 把视频流分解为若干个子图象序列—— 视频分解单元. 本文把视频页定义为如下的视频分解单元:

定义 1 视频页 VP(video page)是满足下述条件之一的连续图象序列:

(1) 相邻图象间的内容相似;

(2) 相邻图象间的内容连续.

对于任意的相邻视频图象 f_k, f_{k+1}(图象大小为 $S = M \times N$), 则首先应计算对应象素点的色彩差别 $D(c_k^{m,n}, c_{k+1}^{m,n})$, $0 \le m \le M$, $0 \le n \le N$, 设色彩差别小于给定阈值的象素点数目为 P_k;

定义 2 设 S 是图象中总的象素点数目, 如果比值 P_k/S 大于预设阈值 $\epsilon(0 < \epsilon < 1)$, 则判定该相邻视频图象内容是相似的;

对于色彩差别大于给定阈值的象素点则应分别统计其色彩直方图 h_k、h_{k+1}, 并计算色彩直方图的距离 $D(h_k, h_{k+1})$;

定义 3 如果 $D(h_k, h_{k+1})$ 小于预设阈值 ξ $(0 < \xi < 1)$, 则判定该相邻视频图象内容是连续的.

内容的相似性表现为主要视觉对象位置无变化; 内容的连续性表现为主要视觉对象位置虽然有变化, 但仅为帧内变化, 并无主要视觉对象的消失或产生.

根据上述定义, 对于给定的视频流 $V_f = \{f_0, f_1, \cdots, f_L\}$, 其中: $f_k (k \in [0, L])$ 为视频图象序列, 视频流分解和关键帧提取的算法描述如下:

算法 1 视频流分解和关键帧提取算法步骤为:

(1) 设定 $i = 0, k = 0, s = 0$, 视频页 $vp_0 = \{f_k\}$, 其关键帧集为 $kf_0 = \{f'_s\}$ $(f'_s = f_k = f_0)$;

(2) 如果 $k \ge L$, 算法停止; 否则, 按定义 2, 对视频图象 f_k 和 f_{k+1}, 计算 P_k/S 值, 如 P_k/S 大于预设阈值 ϵ, 则 $vp_i = vp_i \cup \{f_{k+1}\}$, 转第(5)步;

(3) 否则按定义 3 计算, 如果 $D(h_k, h_{k+1})$ 小于预设阈值 ξ, 则 $vp_i = vp_i \cup \{f_{k+1}\}$, 转第(5)步; 否则,

(4) $i = i + 1, k = k + 1, vp_i = \{f_k\}, kf_i = \{f'_s\}$; $s = 0, f'_s = f_k$; 转第(2)步;

(5) 计算当前帧 f_{k+1} 与关键帧集 kf_i 之间相似性

$$d(f_{k+1}, kf_i) = \min_{f_s \in kf_i} D_1(f_{k+1}, f'_s) \quad (1)$$

其中, $D_1(.,.)$ 为两视频帧之间的相似性, 若 $d(f_{k+1}, kf_i) <$ 预设阈值 λ, 则 $kf_i = kf_i \cup \{f_{k+1}\}$;

(6) $k=k+1$, 转第(2)步.

由于视频流相关性很高, 对于大多数视频帧, 其内容相似, 算法 1 无须计算第(3)步, 因而减少了计算量. 第(5)步为关键帧提取步骤, 根据视频页内容的变化情况, 一个视频页可以提取多个关键帧. 显然, 算法 1 只需一遍"扫描"即可同时实现视频流分解和关键帧提取.

经算法 1 处理, 视频流可表示为 $V_p=\{vp_0, vp_1, \cdots, vp_D\}$, 其中: $vp_i, i\in[0, D]$ 为视频流分解后的视频页; 对于任一视频页 vp_i, 均有相应的关键帧集 kf_i.

1.2　视频页的聚类分析

视频页虽然是顺序排列的, 但在其序列中存在着各种结构. 通过观察可以发现, 在某些故事情节中, 摄像机镜头可能在几个不同拍摄角度或在不同拍摄对象(如演员、物品等)之间来回切换. 典型的例子是"对话"情节, 摄像机镜头在几个演员之间来回切换, 以追踪说话演员的表情和动作. 这样造成内容相似的视频页在时间维上的重复出现. 通过聚类分析, 找出这些相似视频页, 就可以重构出视频流的时序结构和描述故事发展的线索.

定义 4　对于给定的两个视频页 vp_i, vp_j, 相应的关键帧集为 kf_i, kf_j, 计算

$$d(vp_i, vp_j)=\min_{f'_{s_i}\in kf_i, f'_{s_j}\in kf_j} D_2(f'_{s_i}, f'_{s_j}) \qquad (2)$$

其中, $D_2(.,.)$ 为两关键帧之间的相似性. 若 $d(vp_i, vp_j)<$ 预设阈值 δ, 则视频页 vp_i 和 vp_j 的内容相似.

把符合上述定义的一组视频页称为视频类 VC (video cluster). 给定某个已包含 N 个视频页的视频类 $vc_i=\{vp'_0, vp'_1, \cdots, vp'_{N-1}\}$, 对于新的视频页 vp_n, 可以按定义 4 计算它与 N 个视频页的相似性度量

$$d(vp_n, vc_i)=\max_{vp_i\in vc_i} d(vp_n, vp_i) \qquad (3)$$

若 $d(vp_n, vc_i)<$ 预设阈值 δ, 则视频页 vp_n 归并入视频类 vc_i.

对于分解后的视频流 $V_p=\{vp_0, vp_1, \cdots, vp_D\}$, 给出下列视频页聚类算法:

算法 2　视频页聚类算法步骤如下:

(1) 设定 $i=0$;

(2) 如果视频流 $V_p=\{$空集$\}$, 即所有视频页都已归并入某视频类, 则算法停止;

(3) 否则, 取视频流 V_p 中的第一个视频页 vp_{head}, 令视频类 $vc_i=\{vp_{\text{head}}\}$, $V_p=V_p\backslash\{vp_{\text{head}}\}$;

(4) 对视频类 V_p 中的各 vp_n, 按式(3)逐次计算, 其中: $vp_n\in V_p$,

若 $d(vp_n, vc_i)<$ 预设阈值 δ,

则 $vc_i=vc_i\cup\{vp_n\}$, $V_p=V_p\backslash\{vp_n\}$;

(5) 否则, $i=i+1$, 转第(2)步.

显然, 上述算法停止时的 i 值即为生成的视频类数目. 在最坏的情况下(每个视频页为一个视频类), 所需的视频页间的计算次数为

$$D+(D-1)+(D-2)+\cdots+2+1=D\times(D+1)/2$$

对于包含数百个(甚至上千个)视频页的现代动作影片, 上述聚类算法的计算量是相当可观的. 为了降低聚类算法的计算量, 提高算法效率, 对算法 2 作进一步的优化, 提出了快速聚类算法.

1.3　视频页的快速聚类算法

通过仔细观察, 得出下列结论和推论:

结论 1　若 vp_i, vp_j 为两个相邻视频页, 则它们不应聚入同一视频类.

既然 vp_i 和 vp_j 被分解为两个相邻视频页, 则说明它们相邻的视频帧内容相差很大; 由视频页的定义可知, vp_i 和 vp_j 各自所包含的视频帧的内容不相似, 因此 vp_i 和 vp_j 分别描述了不同的内容, 不应聚入同一视频类.

推论 1　在视频页的聚类算法中, 对于某个已包含 N 个视频页的视频类 $vc_i=\{vp'_0, vp'_1, \cdots, vp'_{N-1}\}$, 对于新的视频页 vp_n, 若 vp_n 与 vc_i 中的任一视频页相邻, 则可放弃 vp_n.

推论 2　对于视频流 $V_p=\{vp_0, vp_1, \cdots, vp_D\}$, 其中存在内容相似的两视频页 vp_i, vp_j, $i, j\in[0, D]$, 若 $|i-j|>$ 给定的常数 c, 则 vp_i, vp_j 应聚入不同的视频类.

显然, 常数 c 限制了相似视频页的重复率, 这与实际的故事发展是相一致的. 如果代表某个故事内容的视频页在经过若干个视频页切换之后仍未出现, 则可认为那个视频页所描述的故事片段已暂告一段落. 而与之内容相似的视频页再次出现时, 则可认为是在描述另一段故事内容.

由上述两个推论可给出以下快速聚类算法:

算法 3　视频页快速聚类算法步骤如下:

(1) 设定 $i=0$, $Temp=\{$集$\}$;

(2) 如果视频流 $V_p=\{$空集$\}$, 即所有视频页都已归并入某视频类, 则算法停止;

(3) 否则, 取视频流 V_p 中的第一个视频页 vp_{head}, 令视频类 $vc_i=\{vp_{\text{head}}\}$, $V_p=V_p\backslash\{vp_{\text{head}}\}$;

(4) 选取 V_p 中的某视频页 vp_n, 若 vp_n 满足: vp_n

与 vc_i 中的所有视频页不相邻,且 vp_n 与 vc_i 中的任一视频页在原始视频页序列中间隔不超过 c 个位置,则按式(3)计算 $d(vp_n, vc_i)$;否则,转第(6)步;

(5) 如 $d(vp_n, vc_i) < \delta$,则 $vc_i = vc_i \cup \{vp_n\}$, $V_p = V_p \setminus \{vp_n\}$,转第(4)步;否则,让 $Temp = Temp \cup \{vp_n\}$, $V_p = V_p \setminus \{vp_n\}$,转第(4)步;

(6) $V_p = V_p \cup Temp$, $Temp =$ {空集}, $i = i + 1$,转第(2)步.

由于优化了聚类算法,同样在最坏的情况下(每个视频页为一个视频类),所需的计算次数为

$$(c-1)+(c-1)+\cdots+(c-1)+(c-2)+\cdots+1$$
$$=(D-(c-2))\times(c-1)+(c-2)\times(c-1)/2$$

当 $D = 1\,000$, $c = 5$ 时,算法 2 约需 0.5M 次计算,而算法 3 仅需 4 000 次计算. 其计算量的比值约为 $D/(c-1)$.

1.4 视频流的时序结构图

一段视频流总是包含一些故事情节. 我们把这些故事情节称为故事单元(story unit). 故事单元的内容可以由若干个侧面或故事线索来表达. 一个视频类表达了某个故事情节中的一个侧面,即代表了一条故事线索,而多个视频类则代表了多个故事线索. 当多个视频类在时间维上相互交叠、衔接时,通过沿视频流来追踪视频页在多个视频类(故事线索)间的"跳动"顺序,就可以重构出视频流的时序结构.

为了能直观地显示出视频流的时序结构,我们定义了视频流的时序结构图.

定义 5 视频流的时序结构图 VG 是一个三元组 (VC, E, W),其中: VC 是视频流中所有视频类的集合, E 是 VC 中元素之间关系的集合, W 是权值的集合; E 中每个元素对应一条有向边 $e_{ij} = (vc_i, vc_j)$,它表示存在两个相邻视频页 vp_k, vp_{k+1}, $vp_k \in vc_i$, $vp_{k+1} \in vc_j$, $vc_i \in VC$, $vc_j \in VC$,且 $i \neq j$; W 中每个元素 w_{ij} 表示有向边 e_{ij} 的重复次数.

例如存在分解后的视频流 $V_1 = \{vp_0, vp_1, vp_2, vp_3, vp_4, vp_5, vp_6, vp_7, vp_8, vp_9\}$:

聚类后得到 5 个视频类:

$vc_1 = \{vp_0, vp_2, vp_4\}$

$vc_2 = \{vp_1, vp_3\}$

$vc_3 = \{vp_5, vp_8\}$

$vc_4 = \{vp_6\}$

$vc_5 = \{vp_7, vp_9\}$

则其时序结构图 $VG_1 = (VC_1, E_1, W_1)$,其中:

$VC_1 = \{vc_1, vc_2, vc_3, vc_4, vc_5\}$

$E_1 = \{e_{12}, e_{21}, e_{12}, e_{21}, e_{13}, e_{34}, e_{45}, e_{53}, e_{35}\}$

$W_1 = \{w_{12} = 2, w_{21} = 2, w_{13} = 1, w_{34} = 1, w_{45} = 1, w_{53} = 1, w_{35} = 1\}$

作出的时序结构图 VG_1 如图 1 所示:

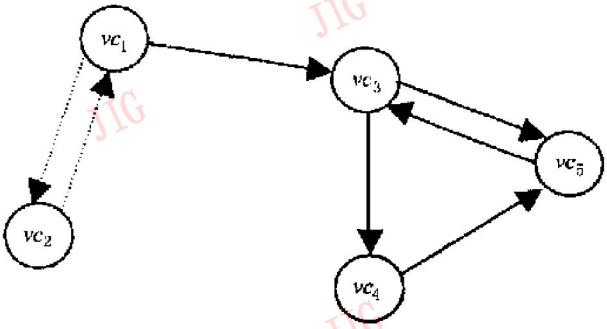

图 1 时序结构图 VG_1($w_{ij} > 1$ 时以虚线表示)

2 实验结果

为了检验本文中提出的时序结构分析方法对于视频浏览的效果,选取了几个视频片段作为实验对象.

首先,采用视频流分解算法对视频段进行分解,结果如表 1 所示.

表 1 视频流分解结果

视频流	长度(帧)	分割结果(视频页)
《红楼梦》片段 1	35 151	60
《红楼梦》片段 2	98 204	161
《The Fifth Element》片段	23 330	220

由于表 1 中的前两个视频流——电影《红楼梦》片段 1、2 的视觉内容变化较平缓,因而分解后得到较少的视频页;而后两个视频流都属于动作影片,其镜头切换频繁,视觉内容变化较剧烈,因而分解后得到较多的视频页.

再运用本文中的视频页快速聚类算法对表 1 的视频流分解结果进行聚类处理,结果如表 2 所示.

表 2 视频页快速聚类结果

视频流	视频页数目	视频类数目
《红楼梦》片段 1	60	51
《红楼梦》片段 2	161	129
《The Fifth Element》片段	220	126

由于电影《红楼梦》片段 1、2 属于叙事性的视频流,故事的发展是以逐个顺序展开的,很少有多条故事线索相互交叠,因而分解后得到较少的视频类;而后两个视频流恰恰相反,多条故事线索相互交叠的情况时有发生,因而分解后得到较多的视频类. 我们可以通过观察《The Fifth Element》片段的部分聚类

结果数据来进一步了解其细节. 其部分聚类结果数据如表 3 所示.

表 3　《The Fifth Element》片段的部分聚类结果

视频页	视频类	首关键帧序号	视频页	视频类	首关键帧序号
1	1	0	26	16	2 504
2	2	30	27	17	2 713
3	3	77	28	18	2 847
4	4	121	29	17	2 909
5	5	172	30	19	3 745
6	4	240	31	17	3 826
7	5	272	32	20	3 941
8	4	617	33	17	4 002
9	5	718	34	21	4 081
10	4	827	35	22	4 187
11	6	896	36	23	4 291
12	7	982	37	22	4 400
13	4	1 044	38	23	4 455
14	6	1 208	39	24	4 516
15	8	1 380	40	23	4 703
16	9	1 447	41	24	4 770
17	4	1 528	42	23	4 820
18	10	1 573	43	24	4 869
19	11	1 599	44	23	4 900
20	12	1 994	45	24	4 937
21	11	2 020	46	23	5 016
22	4	2 171	47	24	5 071
23	13	2 242	48	23	5 139
24	14	2 280	49	24	5 170
25	15	2 340	50	23	5 255

从表 3 中可以看出, 第 4~ 10 和第 38~ 50 的视频页序列为明显的两线索交叠片段. 与表 3 对应的时序结构图 VG_2 如图 2 所示:

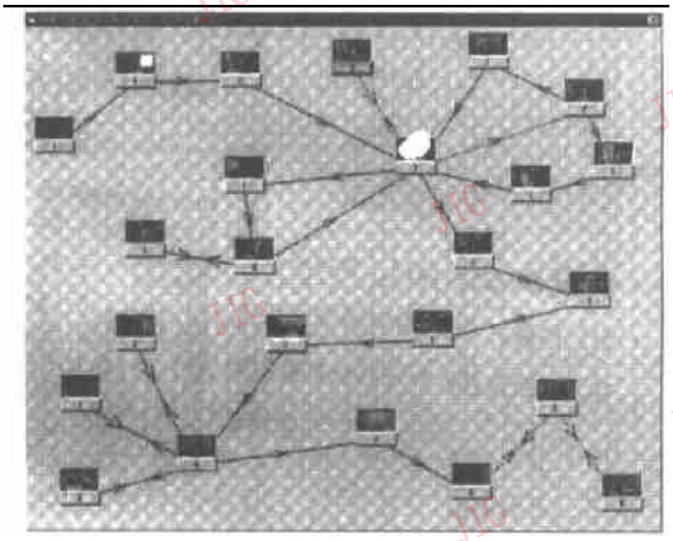

图 2　时序结构图 VG_2($w_{ij}>1$ 时以虚线表示)

通过对比, 我们发现上述时序结构图与实际故事的发展线索接近, 能够体现视频流的时序结构和能反映出故事的发展过程.

3　结　论

从上述实验结果可以看出, 本文提出的视频流时序结构分析方法是可行的, 并且是有效的. 用户可以利用视频流的时序结构图, 快速地浏览视频流, 以形成对视频流中故事发展线索的清晰印象. 同时, 时序结构图可作为视频流的一种有效的非线性索引方式用于视频内容检索等应用.

我们此后的研究将包括对时序结构图的优化、改进和进一步的应用.

参 考 文 献

1　Zhang H J, A Kankanhalli, S W Smoliar. Automatic partitioning of full-motion video. ACM Multimedia System, 1993, 1(1): 10~ 28.

2　Wolf W. Key frame selection by motion analysis. In Proceedings, International Conference on Image Processing, Atlanta, Georgia, press, Atlanta, Georgia, IEEE Press, 1996: 1228~ 1231.

3　Zhang H J *et al*. An integrated system for content-based video retrieval and browsing. Pattern Recognition, 1997, 30(4): 643~ 658.

4　Zhong D, Zhang H J, Chang S F. Clustering methods for browsing and annotation. In: Storage and Retrieval for Still Image and Video Databases. SPIE 2670, San Jose, CA, 1996: 239~ 246.

5　Yeung M M *et al*. Video browsing using clustering and scene transitions on compressed sequences. In: Multimedia Computing and Networking, SPIE 2417, San Jose, CA, 1995: 399~ 413.

6　Minerva Y, Boon-Lock Y, Bede L. Segmentation of video by clustering and graph analysis. Computer Vision and Image Understanding, 1998, 71(1): 94~ 109.

周启龙　1978 年生, 现为浙江大学计算机系研究生. 主要研究领域为视频数据库, 多媒体技术.

朱淼良　1946 年生, 教授, 博士生导师. 主要研究领域为计算机视觉, 人工智能, 多媒体技术.

王东辉　1970 年生, 现为浙江大学计算机系博士生. 主要研究领域为图象处理, 视频数据库, 多媒体技术.