

文章编号:1000-582X(2011)12-088-08

动态时域检测窗口的快速关键帧提取方法

唐朝伟¹,王 恒¹,唐 晖^{1,2},赵贞贞¹,周旭²

(1. 重庆大学 通信工程学院,重庆 400044;2. 中国科学院声学研究所 高性能网络实验室,北京 100190)

摘 要:在镜头检测基础上对帧图像进行逐帧处理、分析,从而获取关键帧的方法导致提取速度较慢,无法满足实时性要求。针对上述缺点,提出动态时域检测窗口的快速关键帧提取方法。该方法通过时域检测窗口把视频序列分割成长度不等的单元,并对窗口左右边缘帧以及窗口内的抽样帧进行分析、对比,从而判断视频单元的内容变化。同时,采用窗口混合增长模式和窗口慢检测机制保证关键帧提取的质量和速度。最终,对提取的关键帧序列进行聚类,得到最优的关键帧。实验证明,对于不同类型的视频,该方法都具有较高的综合评价性能。

关键词:帧; 图像处理; 时域检测窗口; 视频单元; 混合增长; 慢检测启动;

中图分类号:TP391.4

文献标志码:A

A novel keyframe extraction method based on dynamic time-domain detection windows

TANG Chao-wei¹, WANG Heng¹, TANG Hui^{1,2}, ZHAO Zhen-zhen¹, ZHOU Xu²

(1. College of Communication Engineering, Chongqing University, Chongqing 400044, P. R. China; 2. High Performance Network Laboratory, Institute of Acoustics Chinese Academy of Sciences, Beijing 100190, P. R. China)

Abstract: Real-time keyframe extraction is significant for the on-line video retrieval and recognition system. It will lead to low speed for keyframe extraction by analyzing the frames one by one on the basis of shots detection. A novel keyframes extraction method based on dynamic time-domain detection windows is presented. The video frame-sequence is divided into video units with different lengths. The frames at the edges of the window and the sampled frames inside the window are analyzed to judge the changes of the unit content. Meanwhile, a hybrid growth mode of windows and slow detection mechanism are proposed to ensure the quality and speed of the keyframe extraction. The extracted keyframes are clustered to obtain the best keyframe. The experiments show that the proposed method has a good comprehensive performance for different kinds of videos.

Key words: frames; image processing; time-domain detection windows; video unit; hybrid growth; startup of slow-detection;

关键帧提取是基于内容的视频检索与识别系统中的关键技术之一,提取关键帧能将动态视频的内容研究转换为静态图像的特征分析,在实时视频检

索和识别系统中,关键帧的提取质量制约着系统检索与识别的准确性,而提取速度直接影响系统的实时性^[1]。

收稿日期:2011-06-18

基金项目:国家科技重大专项资助项目(2010ZX03004-002)

作者简介:唐朝伟(1966-),男,重庆大学教授,主要从事移动互联网与宽带多媒体通信、计算机视觉及多媒体信息处理方向研究,(Tel)15123876603;(E-mail)williamcq@163.com。

现有的关键帧提取方法有很多:基于镜头边界提取关键帧、通过视觉内容匹配提取关键帧、通过光流分析、运动逼近法提取关键帧,通过无监督聚类法提取关键帧等。其中较为简单的是提取镜头的首尾帧作为关键帧^[2],方法快速、易于实现,但简单提取镜头首尾两帧无法代表整个镜头的内容变化,提取效果不理想。文献[3]采用尺度不变特征转换算法(scale-invariant feature transform, SIFT)检测视频的不连续视觉信息,然后采用双门限预置法分割视频镜头,从而提取关键帧。镜头分割的好坏制约着关键帧的提取质量,高复杂度的镜头分割算法也增加了关键帧的提取时间。类模糊 C 均值聚类的关键帧提取算法是通过预置最初聚类中心,采用视频帧序列的时序特征来对 FCM(fuzzy C-means)算法进行限定^[4],但是难以根据视频内容确定聚类数目。文献[5]从压缩视频中提取所有帧图像的直流分量,利用 K-mean 算法进行聚类,从聚类中提取关键帧。算法虽然减少了提取时间,但依然无法满足实时系统的要求。同时,从聚类中获得的关键帧易出现帧序颠倒现象。

鉴于此,引入时域检测窗口对视频帧图像序列进行动态划分,对窗口内部帧图像进行抽样并组成抽样帧序列,分别计算抽样帧序列的帧间差,对于窗口不同宽度、窗口内不同帧间差采用不同的关键帧提取策略,从而加快了提取速度。同时引入窗口增长机制和窗口慢检测机制保证了提取关键帧的准确率。

1 帧间相似度

“颜色”是描述图像内容的重要特征之一^[6]。运动、形状、纹理等信息的获取相比颜色信息的获取较为复杂,在描述图像内容方面,也不如颜色信息直观有效。因此,仅采用颜色特征对帧间相似度进行介绍。

1.1 帧图像的特征向量

RGB 颜色模型的可分辨色差呈非线性,各分量易受亮度影响,不适应人类的视觉特点。HSV 颜色模型从人类视觉系统出发,用色调(hue)、色饱和度(saturation)和亮度(value)来描述色彩,具有线性伸缩性、可感知色差与颜色分量相应样值的欧式距离成正比的特点^[7],在计算机视觉和图像处理中更方便。因此,将帧图像的颜色信息由 RGB 转换到 HSV 空间模型中。

首先,按照人类的颜色感知将 HSV 圆锥空间

模型进行区域划分,并对其进行等间隔量化,将色度 H 分为 8 个量化等级,色饱和度 S 和亮度 V 分为 3 个,圆锥空间被分为 72 个区域^[8],每个区域分配一个编号 $n(0 \leq n \leq 71)$ 。帧图像的颜色信息可以由一个 72 维的列向量 $(a_0, a_1, \dots, a_n, a_{71})^T$ 表示,其中各元素下标分别对应编号所指区域。通过量化,使得 H 、 S 、 V 各分量与特征向量形成一一映射关系。在颜色信息 $(a_0, a_1, \dots, a_n, \dots, a_{71})^T$ 中,每个元素 $a_n(0 \leq n \leq 71)$ 的值表示图像中有 a_n 个像素点的 H 、 S 、 V 分量映射在下标 n 所指向的区域中,其中 $0 \leq a_n \leq M \times N$, $M \times N$ 为帧图像的分辨率。

根据上述颜色信息,定义帧图像的特征向量由一个 73 维的列向量 $\mathbf{S} = (a_0, a_1, \dots, a_{71}, a_{72})^T$ 表示。其中元素 $a_n(0 \leq n \leq 71)$ 为帧图像的颜色信息, $a_{72}(1 \leq a_{72} \leq V_N)$ 表示此帧处于视频中的位置, V_N 为视频 V 中包含帧图像的数量。例如 $a_{72} = 4$, 表示此帧为视频中第 4 帧。帧图像的特征向量中不仅包含了内容信息,同时也包含了位置信息。

1.2 计算帧间相似度

假设 F_i 和 F_j 为选自同一部视频中的不同帧图像,其特征向量分别为 $\mathbf{S}_i = (a_{i0}, a_{i1}, \dots, a_{i71}, a_{i72})^T$ 和 $\mathbf{S}_j = (a_{j0}, a_{j1}, \dots, a_{j71}, a_{j72})^T$ 。为了提高算法的处理效率,直观反映帧图像之间的颜色变化信息,算法选用街区距离(city-block distance)度量两帧之间的帧间差,定义为

$$\text{dif}(F_i, F_j) = \sum_{k=0}^{71} |a_{ik} - a_{jk}|. \quad (1)$$

根据电影拍摄、剪辑原理,帧图像之间的相似性不仅与颜色信息有关,并且与两帧在视频中出现的的时间间隔有关。因此,任意两帧 F_i 和 F_j 之间的帧间相似度可定义为

$$\text{Sim}(F_i, F_j) = \text{Sim}(T_i, T_j) \times \text{Sim}(C_i, C_j), \quad (2)$$

其中 $\text{Sim}(T_i, T_j)$ 和 $\text{Sim}(C_i, C_j)$ 分别为第 i 帧与第 j 帧的时间相似度与内容相似度。

根据式(1),任意两帧 F_i 和 F_j 之间的内容相似度 $\text{Sim}(C_i, C_j)$ 定义为^[8]

$$\text{Sim}(C_i, C_j) = 1 - \text{dif}(F_i, F_j), \quad (3)$$

帧间的时间相似度与帧图像的位置距离成反比,时间间隔越长,帧相似性的概率越小,反之,概率越大。因此,任意两帧 F_i 和 F_j 的时间相似度 $\text{Sim}(T_i, T_j)$ 可以由两帧的特征向量中 a_{72} 元素计算得到

$$\text{Sim}(T_i, T_j) = k / |a_{i72} - a_{j72}|. \quad (4)$$

其中 k 为常数。

2 时域检测窗口

通过分析视频的组织结构可知,在1 s内所包含的24-30帧帧图像中,多数帧图像的颜色信息连续且高度相似,只有在镜头发生突变时才会引起颜色信息的跳变。文献[9-12]均采用逐帧分析的策略,增加了大量无意义的计算,难以突破实时性的要求。为了提高关键帧的提取速度,算法结合视频组织结构的特点,引入时域检测窗口,利用窗口检测算法提取视频关键帧。

2.1 窗口的增长机制

时域检测窗口(以下简称“窗口”)是用来检测帧间相似度的视频虚拟窗口,通过窗口的移动和自增长将视频分割成不同宽度的视频单元。窗口具有最小宽度 wnd_min 和最大宽度 wnd_max ,单位为帧。为了适应不同类型的视频,提出窗口混合增长机制。

$$wnd_i = \begin{cases} 2^i \times wnd_min; & wnd_i < wnd_thr; \\ 2^{\lfloor \log_2 \frac{wnd_thr}{wnd_min} \rfloor} + (i - 2^{\lfloor \log_2 \frac{wnd_thr}{wnd_min} \rfloor}) \times wnd_min; & wnd_i \geqslant wnd_thr; \\ wnd_max; & wnd_i \geqslant wnd_max, \end{cases} \quad (5)$$

式中 wnd_i 为第 i 次增长时的窗口宽度,当 $i=0$ 时,窗口为初始化窗口。其中 $\lfloor \cdot \rfloor$ 为取整函数。

窗口混合增长过程如图1所示

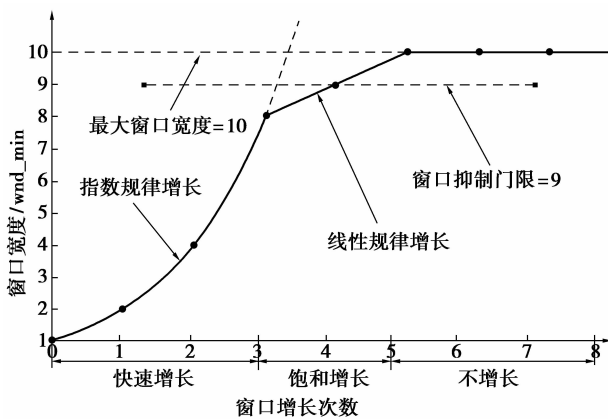


图1 窗口增长过程示意图

图1中窗口抑制门限宽度为 $9(9 \times wnd_min)$,最大窗口宽度为 $10(10 \times wnd_min)$ 。首先窗口以指数规律进行快速增长,当窗口进行第4次增长时,窗口宽度($wnd_wid=16$)大于抑制窗口门限($wnd_thr=9$),因此,窗口将采用饱和增长模式线性增长,即第4次增长后的检测窗口宽度为9。当窗口进行第5次增长时,窗口宽度($wnd_wid=10$)等于最大窗口宽度($wnd_max=10$),窗口停止增长,以最大窗口宽

度进行检测。

窗口混合增长机制包括快速增长和饱和增长两种模式。首先采用最小窗口宽度 wnd_min 对镜头进行检测,假设窗口内各帧都具有较高的帧间相似度,则窗口将以指数形式继续增长,文中将这种增长称为快速增长(如图1所示)。

显而易见,窗口快速增长可以带来帧的快速提取,同时也会增加关键帧的漏检率。因此,定义窗口抑制门限 wnd_thr ,窗口抑制门限可以控制窗口的增长速度,当快速增长的窗口宽度大于等于窗口抑制门限($wnd_wid \geqslant wnd_thr$)时,窗口增长进入饱和增长模式。饱和增长是指当检测窗口宽度大于等于窗口抑制门限时,窗口宽度不再以指数形式增长,而以固定的宽度进行递增,进入线性增长。如图1所示。随着窗口的增长,当宽度大于等于最大宽度时,即 $wnd_wid \geqslant wnd_max$,窗口宽度将保持为 wnd_max ,不再增长。即有

度进行检测。

2.2 慢检测启动机制

视频包含了许多镜头的突变和渐变。一旦检测出窗口内帧图像序列间具有较小的相似度,即窗口内出现了镜头的变化,则采用“宁多勿错”的关键帧提取原则,使用最小宽度的窗口 wnd_min 对本单元位于最左端的变化处进行重新检测,然后窗口进入对应的增长模式,这一过程被称为慢检测启动机制。慢检测并非指速度慢的检测策略,而是算法为了减少重复检测,降低误检率,所采取的一种最优启动机制(如图2所示)。这一过程可由式(6)表示,其中 wnd_wid 为窗口宽度。

$$wnd_wid = \begin{cases} wnd_i; & \text{if}^* \\ wnd_wid = wnd_min; & \text{else} \end{cases} \quad (6)$$

式中 wnd_i 为式(5)取值,*表示窗口内存在镜头变化。

图2中,当算法采用第5次增长的窗口宽度($wnd_wid=10 \times wnd_min$)进行检测时,发现窗口内存在镜头的变化(由图中虚线表示)。算法则采用慢检测启动机制,丢弃原检测窗口,采用宽度为 wnd_min 的窗口进行重新检测。

2.3 窗口的抽样检测

对于宽度较小的窗口,通过计算窗口边缘左右两帧的相似度即可反映出窗口内帧图像的变化。但

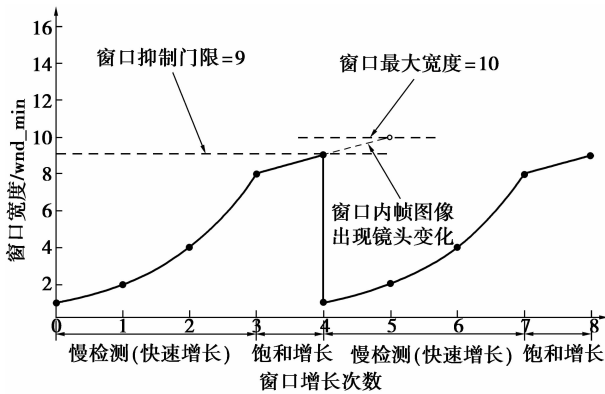


图 2 慢检测启动示意图

随着窗口的增长,仅采用这样的方法无法达到预期的效果,同时也导致了关键帧漏检率的增加。因此,算法将对窗口内帧图像进行抽样提取,从而准确判断窗口内是否存在镜头的变化。

当窗口宽度小于等于最小窗口宽度时,即 $wnd_wid \leq wnd_min$,算法不采用抽样检测。

当窗口宽度大于最小窗口宽度时,即 $wnd_wid > wnd_min$,算法以最小窗口宽度为间隔,对窗口内视频帧进行抽样,抽样得到的帧图像与窗口左右边缘帧按照时间顺序组成抽样帧序列,并且计算帧序列中相邻帧的帧间差,通过帧间差判断窗口内是否存在帧图像信息的“跳变”。若存在,则采用慢检测启动机制,从位于最左端的跃变点处进行重新检测。同时从窗口左边缘与跃变点之间的帧序列中提取关键帧。反之,则直接提取本序列的关键帧。窗口依照增长机制继续增长,如图 3 所示。

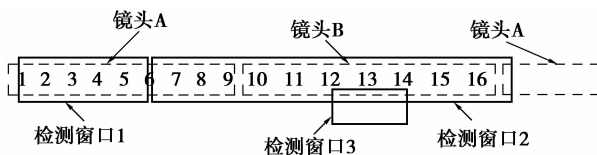


图 3 窗口抽样检测示意图

例如帧图像序列 1 至 19,其中 1-9、17-19 为 Shot A 内容片段,10-16 为 Shot B 内容片段,如图中虚线框标出。假设最小窗口宽度为 4, Detected Window 1 宽度为 6,对窗口内帧图像进行抽样检测,由于 1-6 帧图像均来自 Shot A,抽样帧序列中不存在图像颜色信息的“跳变”。随后窗口进行增长,宽度增长为 12,对 Detected Window 2 进行抽样检测,即抽样帧序列由第 6、9、12、15、17 帧组成,通过计算抽样帧序列的帧间差,发现在第 9、12 帧之间出现了帧图像颜色信息的跃变。根据慢检测机制,

窗口以最小检测窗口宽度从“跃变点”处(第 12 帧)重新开始检测,如图中 Detected Window 3 所示。同时,提取第 6-9 帧间的关键帧。

3 基于时域检测窗口的关键帧提取算法

根据不同的窗口宽度,采取不同的关键帧提取策略。基于时域检测窗口的关键帧提取算法步骤如下,其中 T 为帧间相似度阈值:

1) 首先,使用初始窗口(宽度 wnd_min)对视频帧进行检测,由式(2)计算得出检测窗口左右边缘帧的帧间相似度 $\text{Sim}(F_{left_wnd}, F_{right_wnd})$,其中 F_{left_wnd} 和 F_{right_wnd} 分别代表窗口的左边缘帧和右边缘帧;

2) 判断 $\text{Sim}(F_{left_wnd}, F_{right_wnd}) > T$ 是否成立;

①若 $\text{Sim}(F_{left_wnd}, F_{right_wnd}) \leq T$,则将窗口内包含的所有帧图像作为此窗口的关键帧,返回步骤(1)。

②若 $\text{Sim}(F_{left_wnd}, F_{right_wnd}) > T$,判断窗口内的帧图像具有较高相似度,则提取右边缘帧 F_{right_wnd} 作为此窗口的关键帧,进入步骤(3);

3) 根据窗口增长机制,进入对应的增长模式,得到新窗口宽度 $wnd_wid (wnd_wid > wnd_min)$ 。再次计算窗口边缘帧的相似度 $\text{Sim}(F_{left_wnd}, F_{right_wnd})$;

4) 判断 $\text{Sim}(F_{left_wnd}, F_{right_wnd}) > T$ 是否成立;

①若 $\text{Sim}(F_{left_wnd}, F_{right_wnd}) \leq T$,则窗口内至少存在一处帧图像颜色信息的跃变点。算法将不对此窗口进行关键帧提取,直接返回至步骤 1)。

②若 $\text{Sim}(F_{left_wnd}, F_{right_wnd}) > T$,对窗口内进行抽样检测。由帧图像的特征向量表达式可知,任意长度的抽样帧序列都可以表示为: $L = (S_1, S_2, \dots, S_{N_L}, \bar{L})$,其中 S_i 为第 i 帧的特征向量。 \bar{L} 为抽样序列 L 的平均直方图: $\bar{L} = \left(\sum_{k=1}^{N_L} S_k \right) / N_L$ 。其中 N_L 表示抽样帧序列中包含的帧图像数量。同样, \bar{L} 能够由 73 维的列向量 $S_L = (\overline{a_0}, \overline{a_1}, \dots, \overline{a_{71}}, \overline{a_{72}})^T$ 表示。

a) 若存在帧间差差值的跃变,将从窗口左边缘与跃变点之间的帧序列中提取关键帧,并启动慢检测机制,返回步骤 1)。

b) 若不存在帧间差差值的跃变,便将 S_L 作为抽样帧序列的“信息中心”,由 F_L 表示。由式(2)、(3)、(4),分别计算各帧与 F_L 的帧间相似度 ($\text{Sim}(F_1, F_L), \dots, \text{Sim}(F_k, F_L), \dots, \text{Sim}(F_L, F_{N_L})$),从中选取相似度最小和最大的两帧作为此抽样帧序列的关键帧,窗口继续增长,返回步骤 3)。

5) 以上算法步骤循环执行,直到视频结束为止;

6) 关键帧聚类。算法将采用聚类技术^[11]对时间接近、颜色信息类似的帧图像进行聚类,以便以最优的关键帧表达视频内容。得到的关键帧序列可以由 (S_1, S_2, \dots, S_N) 表示,其中 S_i 对应第 i 关键帧的特征向量, N 表示提取的关键帧总数。首先设置聚类门限 T' ,并将 S_1 作为第一个聚类集合的质心,继续后一关键帧图像与质心的帧间差。若帧间差 $dif(S_1, S_2)$ 小于 T' 时,重新计算聚类集合的新质心 $(S_1 + S_2)/2$;若帧间差大于 T' 时,将此关键帧作为新聚类集合的质心进行聚类。以此类推,对所有视频关键帧进行分类处理。最后提取距离质心最近和最远的帧图像作为子聚类集合的关键帧。

算法流程图如图 4 所示

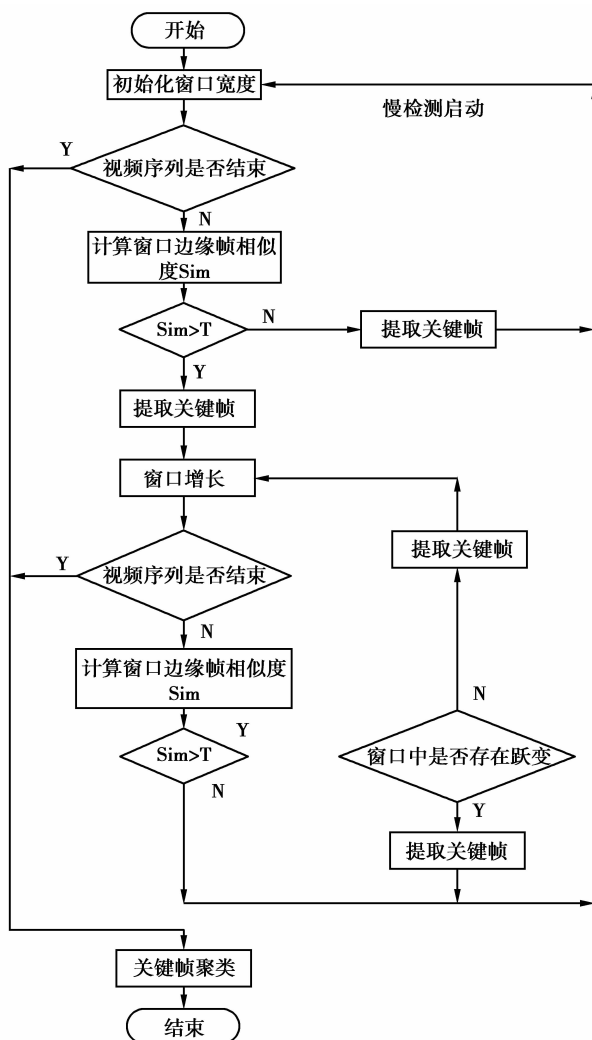


图 4 基于时域检测窗口的关键帧提取算法流程图

4 算法实现与性能分析

基于静态视频数据库的关键帧提取方法将准确

率作为算法的主要评价标准,而对提取速度并没有很高的要求。然而,很多实际应用表明,在某些实时系统中,算法的时效性同样重要。实验将准确率与时效性作为算法的评价标准。

4.1 实验参数的选择

首先选取最小窗口宽度、最大窗口宽度、窗口抑制门限等实验参数。

根据视觉停留理论,人眼在观察景物时,光信号传入大脑神经需要一段短暂的时间,视神经的反映速度为 $1/24 \text{ s}$ ^[13],人类接受图像信息的时间大于视神经的反映速度。例如,在帧率为 24 帧/s 的视频镜头中,若存在单独一帧帧图像突变,虽然视觉可以感觉到这张图像的存在,但却无法获得此图像的信息。在实际视频中,这样的情况是不可能出现的。因此,算法将时域检测窗口的最小宽度定义为 3 帧,即 $wnd_min=3$ 。若最小窗口宽度选择过大,则会导致漏检率的上升;反之,又无法提高提取速度。

文献[13]指出镜头的渐变和突变为时域随机过程,其中渐变的长度多在 2 到 60 帧。因此,定义窗口最大宽度为 60 帧,即 $wnd_max=60$ 。以最小窗口宽度($wnd_min=3$)开始增长,当窗口进行第 3 次增长时,宽度已经达到 24 帧,即帧率为 24 帧/s 视频中的 1 s 时长。若窗口继续以指数形式增长,当进行第 4 次增长时,窗口宽度达到 48 帧,即帧率为 24 帧/s 视频中的 2 s 时长。通过对动作类视频片段《唐山大地震》采用不同的窗口抑制门限进行测试,见表 1。其中,视频时长为 $43 \text{ 分 } 28 \text{ s}$,实验选取最小窗口宽度和最大窗口宽度分别为 $wnd_min=3, wnd_max=60$ 。

表 1 不同抑制门限下的算法性能对比

抑制门限宽度	准确率/%	提取时间/m:s
$wnd_thr=12$	93.6	16:22
$wnd_thr=24$	94	13:07
$wnd_thr=48$	94.3	17:43

采用不同的窗口抑制门限对于算法的准确率影响不大,这是因为算法对于提取得到的关键帧序列进行聚类的结果,但在提取时间上有所差别。当 $wnd_thr=12$ 时,窗口增长速度的减缓降低了提取效率,当 $wnd_thr=48$ 时,虽然提高了窗口的增长速度,但是,对于 2 s 的时域检测窗口而言,增加了窗口内抽样帧检测的数量,从而导致多次窗口的重检测,造成提取时间增加。当 $wnd_thr=24$ 时,得到了理想的测试结果。这个结论在大量的实际视频测试

中也得到了证实。

4.2 算法性能分析

为了验证算法的有效性,对测试视频分别采用基于镜头边界的关键帧提取算法^[2](keyframe extraction Based-on the boundary of the shots,以下简称“KEBS”)、基于非监督聚类的关键帧提取算法^[14](keyframe extraction Based-on the unsupervised clustering,以下简称“KEUC”)以及本文提出的基于时域检测窗口的关键帧提取算法(keyframe extraction based-on Time-domain detecting windows,以下简称“KETW”)进行测试,并对结果进行分析比较。为便于量化分析,引入耗时因子 λ 、时效因子 η 以及提取准确率因子 ρ 对各关键帧提取的质量和速度进行比较。其中,耗时因子 λ 可以定义为

$$\lambda = \frac{t}{L}, \quad (11)$$

式中: t 为关键帧提取时间; L 为视频总时长,单位均为s。耗时因子 λ 反映关键帧提取耗时多少, λ 值越小表示提取视频关键帧耗时越小,反之,越多。根据耗时因子,定义时效因子 η 为

$$\eta = 1/\lambda. \quad (12)$$

根据公式(12)得到的时效因子为耗时因子值的

倒数,反映了关键帧提取的快慢, η 值越大表明提取视频关键帧速度时效越高,反之,越低。

根据目测关键帧数,把由各算法得到的关键帧数与目测关键帧数之比定义为提取准确率因子 ρ ,并由公式(13)得出

$$\rho = T_{kf}/E_{kf}, \quad (13)$$

其中: T_{kf} 是通过各算法得到的关键帧数; E_{kf} 为目测关键帧数。 ρ 越高表明算法的准确率越高,反之,越低。

由式(12)(13),得到算法的综合评价因子CI

$$CI_i = \sum \eta_i \rho_i, \quad (14)$$

其中 i 为下标,代表第 i 个算法对应的综合评价因子、时效因子以及提取准确因子。 CI 越大,表明算法综合性能越好。

实验选取了大量的实际视频进行测试,其中测试视频包含动作类、体育类、新闻类、风景类、MV等,其中分别选取动作类视频片段 Video 1、体育类视频 Video 2、新闻类视频片段 Video 3、风景类视频 Video 4、MV类 Video 5作为各类的代表,如表2所示。

图5中(a)、(b)、(c)分别比较3种算法的提取准确率因子 ρ 、时效因子 η 、以及综合评价因子CI。

表2 3种不同算法的测试结果与性能指标对比

参数	视频播放时长(m:s)	KEBS			KEUC			KETW		
		提取时间(m:s)	$\rho/\%$	η	提取时间(m:s)	$\rho/\%$	η	提取时间(m:s)	$\rho/\%$	η
Video 1	43:28	27:11	83.3	1.59	35:51	90.1	1.21	13:7	89.7	3.31
Video 2	13:12	8:32	83.9	1.57	9:26	91.4	1.41	6:13	91.7	2.14
Video 3	15:00	9:1	81	1.64	13:28	89.3	1.13	4:23	90.1	3.54
Video 4	23:47	15:12	80.3	1.55	18:4	91.5	1.30	9:57	92	2.45
Video 5	3:30	1:57	80.9	2.10	3:1	93.2	1.09	1:23	94	2.68

比较图5(a)中各算法的提取准确率因子 ρ ,KEBS准确率因子普遍较低,均在82%左右,这是因为简单提取镜头首尾帧作为关键帧无法准确代表镜头中的内容变化,在动作类(Video 1)、MV类等视频中(Video 5),存在了大量的突变、渐变镜头,尤其是在体育类视频中(Video 2),有大量的突变镜头存在,这些镜头内容变化较大,导致KEBS算法准确率因子较低;而在风景类视频(Video 4)中,大量渐变镜头的存在也导致了算法无法准确有效提取关键

帧。KEUC与KETW提取准确率因子相对较高,大多数视频准确率均在90%以上。KEUC通过聚类算法得到关键帧,通过较高的计算复杂度提高了提取准确率因子 ρ ,KEUC对于各种类型的视频进行提取,都具有理想的提取准确率因子 ρ 。KETW利用时域检测窗口对各类视频进行关键帧提取,采用抽样检测机制对窗口内不同视频的内容进行检测,并且采用关键帧的聚类对提取得到的关键帧集合进行优化,从而保证了较高的提取准确率。通过

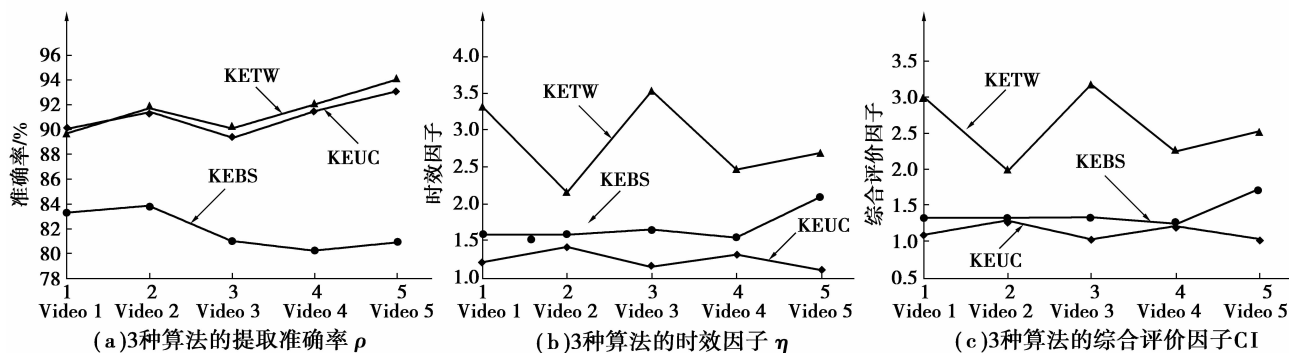


图5 3种算法的性能比较分析

对比, KEUC 与 KETW 具有较高的准确率, 与 KEBS 相比, 准确率约提高了 10%。

比较图 5(b) 中各算法的时效因子 η , KEUC 算法虽然保证了理想的提取准确率, 但严重牺牲了提取速度, 时效因子较低, 约为 1.2。甚至少数视频的时效因子接近于 1, 提取时间接近视频播放时长。KEBS 计算复杂度虽然较低, 但是算法对视频中帧图像进行逐帧分析、提取, 在一定程度上降低了算法的时效性, 时效因子约为 1.6。KETW 算法利用时域检测窗口, 减少了视频帧的处理量, 从而得到理想的提取时间。体育类视频的检测中存在大量的突变镜头, KETW 与另外两种算法的时效因子相差不多, 由图 5(b) 可知, 在体育类视频的检测中, KETW 与 KEBS、KEUC 的时效因子分别相差为 0.57、0.63。这是因为大量的镜头突变限制了时域检测窗口的增长速度, 导致耗时增多, 时效因子下降。在新闻类视频中 (Video 3), 视频固定镜头时间较长、数量较多, 较长的镜头给予窗口增长足够的空间, 使得提取速度较快。相比另外两种算法, KETW 具有更强的时效性。在新闻类视频检测中, KETW 的时效因子均高于 KEBS、KEUC, 相差分别为 1.9 与 2.41。在实际视频测试中 KETW 具有较好的实时性, 与 KEBS 相比, KETW 时效因子高于 KEBS 时效因子约为 1.1 左右, 与 KEUC 相比, 则高出 2.1 左右。

比较图 5(c) 中各算法的综合评价因子 CI, 显而易见, KETW 算法对于各类视频都具有较高的综合评价因子。KEUC 与 KEBS 基于镜头分割, 错误的镜头会对关键帧提取带来的较大影响, 然而前者计算复杂度较低, 具有较高的时效因子, 但是提取准确率低, 而后者提高了准确度却降低了提取速度。因此, 综合评价因子均不理想, KEBS 的综合评价因子约为 1.4, KEUC 的综合评价因子约为 1.2。KETW

引入时域检测窗口, 并非对所有帧进行处理、分析, 同时采用窗口增长机制提高算法的时效性, 采用窗口内的抽样检测和关键帧聚类保证了提取的准确性, 大大提高了综合性能。

实验结果表明, 基于时域窗口检测的关键帧提取算法对于不同类型的视频, 都可以保证较快的提取速度和较高的质量。算法已实际运用于“互联网内容识别及版权保护”平台中, 并取得了理想的效果。

5 结 语

根据视频组织结构的特点, 采用动态的时域窗口方法提高了关键帧的提取速度, 同时采用帧图像抽样检测、关键帧聚类保证了提取的准确性。通过大量实际视频检测, 算法都能够保证优异的提取速度和质量, 从而为在线视频检索和识别奠定了检源基础。今后的研究工作将主要集中在以关键帧为基础的视频场景理解方面。

参考文献:

- [1] CHATZIGIORGAKI M, SKODRAS A N. Real-time keyframe extraction towards video content identification [C]// Proceedings of 16th International Conference on Digital Signal Processing, July 5-7, 2009, Santorini, Greece. [S. l.]: IEEE, 2009: 1-6.
- [2] FANG H, JIANG J M. A fuzzy logic approach for detection of video shot boundaries [J]. IEEE Transactions on Pattern Recognition, 2006, 39 (9): 2092-2100.
- [3] LIU G, WEN X M, ZHENG W, et al. Shot boundary detection and keyframe extraction based on scale invariant feature transform [C]// Proceedings of the Eighth International Conference on Computer and

- Information Science, June 1-3 2009. Shanghai, China; IEEE, 2009; 1126-1130.
- [4] 张亚迪,李俊山,胡双演.类模糊 C 均值聚类的关键帧提取算法[J].微电子学与计算机,2009, 26(2):89-92.
ZHANG YA-DI, LI JUN-SHAN, HU SHUANG-YAN. Keyframes extraction based on simu-FCM clustering[J]. Microelectronics & Computer, 2009, 26(2):89-92.
- [5] SHI F X, GUO X J. Keyframe extraction based on kmeans results to adjacent DC images similarity[C]// Proceedings of 2010 2nd International Conference on Signal Processing Systems (ICSPS), July 5-7 2010. Dalian, China; IEEE, 2010; 611-613.
- [6] VANDESANDE K, GEVERS T, SNOEK C. Evaluation of color descriptors for object and scene recognition[C]// Proceedings of IEEE Conference on 2008 Computer Vision and Pattern Recognition (CVPR 2008). Anchorage, AK, USA; IEEE, 2008, 6; 978-985.
- [7] 田玉敏,王忠,朱虹.基于均值散度直方图的镜头分割算法[J].华中科技大学学报:自然科学版,2009, 37(4):43-46.
TIAN YU-MIN, WANG ZHONG, ZHU HONG. Video segment algorithm using average divergence histogram [J]. Journal of Huazhong University of Science and Technology: Nature Science Edition, 2009, 37(4):43-46.
- [8] 程文刚,须德.一种层次视频摘要生成方法[J].中国图象图形学报,2004, 9(1):118-123.
CHENG WEN-GANG, XU DE. A novel approach of generating hierarchical video abstraction[J]. Journal of Image and Graphics, 2004, 9(1):118-123.
- [9] LIU L J, FAN G L. Combined key-frame extraction and object-based video segmentation [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2005, 15(7):869-884.
- [10] 刘云根,刘金刚.重建误差最优化的运动捕获数据关键帧提取[J].计算机辅助设计与图形学学报,2010, 22(4):670-675.
LIU YUN-GEN, LIU JIN-GANG. Keyframe extraction from motion capture data by optimal reconstruction error [J]. Journal of Computer-Aided Design & Computer Graphic, 2010, 22(4):670-675.
- [11] YIN L, LEE S H, YEH C H, et al. Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques [J]. IEEE Transactions on Signal Processing Magazine, 2006, 23(2):79-89.
- [12] LI G, ZHU L, KATSAGGELOS A. An efficient video indexing and retrieval algorithm using the luminance field trajectory modeling [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2009, 19(10):1566-1570.
- [13] 丁绪星,杨凌云,冯友宏.一种基于视觉注意力的快速运动估计算法[J].仪器仪表学报,2009, 30(12):2539-2543.
DING XU-XING, YANG LING-YUN, FENG YOU-HONG. Fast motion estimation algorithm based on visual attention [J]. Chinese Journal of Scientific Instrument, 2009, 30(12):2539-2543.
- [14] LI Y N, LU Z M. Video abstraction via attention model and on-line clustering[C]// Proceedings of 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC). Kaohsiung, China; IEEE, 2009; 627-630.

(编辑 侯 湘)