

基于预处理的视频镜头边界检测算法

章亦葵^{1*}, 赵 晖²

(1. 天津大学 软件学院, 天津 300072; 2. 天津大学 计算机科学与技术学院, 天津 300072)

(* 通信作者电子邮箱 zhangyk@tju.edu.cn)

摘 要: 针对视频镜头边界检测的高耗时问题, 提出了一种基于视频预处理的视频镜头边界检测(SBD)改进算法。通过使用自适应的阈值选择可能包含镜头边界的候选段, 候选段内首帧与其余各帧进行相似度对比检测出镜头起始帧, 并立即检测切变。若候选段中不包含切变, 则进行渐变检测。调整候选段以保证镜头边界位于同一段内, 段内其余各帧与起始帧进行相似度对比确定镜头结束帧。实验结果表明, 所提算法镜头边界识别准确率能够达到90%以上, 且与倒三角模式匹配方法相比能够节约时间15.6%~30.2%; 与对渐变和切变分别检测的算法相比, 该算法能够在满足识别率的基础上提升检测速度。

关键词: 视频镜头边界检测; 视频分割; 视频检索; 镜头切变; 镜头渐变

中图分类号: TP391.4 **文献标志码:** A

Video shot boundary detection method based on pre-processing

ZHANG Yikui^{1*}, ZHAO Hui²

(1. School of Computer Software, Tianjin University, Tianjin 300072, China;

2. School of Computer Science and Technology, Tianjin University, Tianjin 300072, China)

Abstract: To solve the high consumption problem of fast video Shot Boundary Detection (SBD), an improved Shot Boundary Detection (SBD) method was proposed based on video pre-processing. Adaptive threshold was taken to select the candidate segments which may contain shot boundaries, the beginning frame was detected by comparing the similarity value between the first frame and the rest frames in the candidate segments, and then cut was detected immediately. Gradual transition would be detected if no cut was detected. Candidate segments had to be adjusted to ensure the whole transition was located in the same segment. Ending frame was confirmed by comparing the similarity value between the beginning frame and the rest frames in segment. Experimental results demonstrate that the proposed algorithm achieves an accuracy above 90% and the time cost is reduced 15.6%–30.2% compared with inverted triangle pattern matching method. The proposed algorithm satisfies the need of accuracy and improves detection speed compared with the traditional methods which need detection for both cut and gradual transition.

Key words: Shot Boundary Detection (SBD); video segmentation; video retrieval; shot cut; shot gradual transition

0 引言

随着数字媒体的增长, 视频检索的需求也大幅提升。视频镜头边界检测是视频检索的关键, 视频镜头是指被连续捕获的一系列帧序列^[1], 视频镜头边界有两种主要形式: 切变和渐变。渐变被分为淡入淡出、溶解、消除等多种形式^[2]。根据视频镜头边界的不同, 镜头边界检测方法主要分为两类: 切变检测和渐变检测。早期的研究主要关注于切变, 通过提取相邻帧的特征进行比对, 然后进行决策, 从而判断切变。相邻帧的对应像素差和颜色直方图是最为常见的两种图像特征。如 Hanjalic 等^[3-4]提出利用两帧对应像素差和来解决该问题, 然而镜头内物体的移动或摄像头的晃动以及光照变化均可能在两帧对比时造成干扰, 从而影响实验结果; 而颜色直方图比对应像素差和具有很好的鲁棒性^[5-6], 能够减少外界对实验的影响, 利于检测切变。也有一些采用其他的图像特征来检测视频镜头, 如图像边缘^[7]。近年来也有一些新的方法来解决这一问题, 如采用支持向量机^[8]。渐变由于涉及到

多个帧, 帧间变化不如切变明显, 因此其检测过程与切变的检测相比有很大的不同。Hao 等^[9-10]提出利用窗口计算镜头边界, 对固定窗口大小的视频帧序列进行计算, 并不断向前移动窗口, 在每个窗口中检测镜头边界, 可以得到视频的所有镜头信息; 但是由于窗口的移动仍然会有重叠区域, 从而导致计算的重复。Bescos 等^[11]提出利用不同间距的帧差来匹配波峰模型, 从而确定镜头边界; 但是由于需要的参数较多, 计算过程比较复杂。Grana 等^[12]提出一种改进的线性算法, 利用迭代计算出渐变和切变的中心和长度, 通过三角模型匹配来检测镜头边界; 然而在检测渐变时, 需要迭代计算, 计算量较大。由于上述算法均要对视频的全部帧序列进行考虑, 而镜头边界所涉及的帧只是视频中的一部分。因此 Li 等^[13]提出一种基于预处理的快速检测算法, 其主要思想是预选出可能包含镜头边界的候选段, 边界检测只在这些候选段中进行, 从而减少了对非相关的视频帧段的处理。在此基础上, Lu 等^[14]提出一种倒三角模式匹配方法, 利用渐变的对称性确定镜头边界的中心位置, 进而确定镜头边界。文献[13-14]的

收稿日期: 2014-05-22; 修回日期: 2014-07-20。

作者简介: 章亦葵 (1965-) 男, 湖南长沙人, 副教授, 博士, 主要研究方向: 嵌入式系统、图像处理、数据挖掘、智能交通系统; 赵晖 (1990-) 男, 山西朔州人, 硕士, 主要研究方向: 视频检索、图像处理。

共有缺点是同一个候选段需要分别进行切变和渐变检测,导致时间开销增加。

针对上述问题本文提出了一种改进的视频镜头检测方法,使得候选段只需要检测一次,就可以判定出该段的切变和渐变。

1 镜头边界检测方法

1.1 提取候选段

视频镜头边界处于视频序列的某个位置上,同时也有大量的非边界帧存在于视频帧序列中。提取候选段的主要目的是选取那些可能包含边界的帧段,同时排除一些不可能包含镜头边界的帧段。这样边界检测只需要针对候选段进行计算即可,可以减少计算复杂性。候选帧的提取基于在同一镜头的帧有高度的相似性,因此可以排除首帧和尾帧相似较高的段。本文候选帧的提取主要基于文献[13]中的预处理方法,由于本文后期检测方法的改善,对候选段的提取进行了适当简化,从而减少时间开销。候选帧的提取方法如下所示:

第1步 视频中的帧序列每21帧组成一个段。在第 n 段中,第一帧是 $20n$,最后一帧是 $20(n+1)$ 。这里将每个段的首帧和尾帧的差异定义为段的差异值,并通过计算首帧和尾帧的对应像素的灰度值差的总和来确定这个段的差异值,第 n 段的差异值可用式(1)表示:

$$d(20n, 20(n+1)) = \sum_x \sum_y |F(x, y; 20n) - F(x, y; 20(n+1))| \quad (1)$$

其中 $F(x, y; 20n)$ 是第 k 帧的位于 (x, y) 的像素的灰度值。在之后描述中,用 $d^{21}(n)$ 来表示第 n 段的差异值。

第2步 每10个段被分为一个组,对每个组用式(2)所得阈值来确定候选段:

$$T_L = \mu_L + \alpha(1 + \ln(\mu_G/\mu_L))\sigma_L \quad (2)$$

其中: μ_L 是组内所有段的 $d^{21}(n)$ 均值,而 μ_G 是视频所有段的 $d^{21}(n)$ 均值, σ_L 是组内所有段的 $d^{21}(n)$ 标准差,在文献[13]中,通过实验确定参数 α 取值为0.4,视频帧序列分段和分组情况如图1所示。

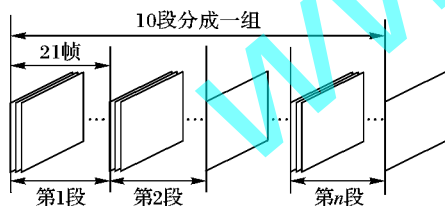


图1 视频帧序列分段分组情况

第3步 对每个组进行计算,对组中的每个段计算其差异值 $d^{21}(n)$,将每个段的差异值与其所在组的阈值 T_L 进行比较,如果该段的差异值比阈值 T_L 大,则认为该段是一个候选段。为了防止误判,选取更多的候选段,对于任何段如果其差异值 $d^{21}(n)$ 远大于相邻段的差异值,则也被认定为候选段。因此如果任何段满足式(3),则被认为是候选段:

$$(d^{21}(n) > 3d^{21}(n-1) \cup d^{21}(n) > 3d^{21}(n+1)) \cap d^{21}(n) > 0.8\mu_G \quad (3)$$

在之前的文献中,候选段需要被分成更小的段来检测镜头边界,但是如果在新的段中没有检测到突变时,需要将这些段重新组合成一个较大的段来检测渐变,从而导致计算的重复。在本文的边界检测过程中,省去了候选段继续划分的过

程,在候选段确定后直接进行对分比较,在保证镜头边界检测充分的前提下节约了计算时间。

镜头边界检测只在候选段中进行,文献[13-14]中,在段中首先进行全面分析检测突变,如果没有突变则需要在段中重新全面分析以检测渐变,这导致了段中的重复计算,为解决这一问题,本文提出了一种新的镜头检测方法,主要基于如下几个观点:

1) 在候选段中的每一个视频镜头边界都有一个起始帧和一个结束帧;

2) 镜头边界的起始帧和段的首帧属于同一镜头,两者的相似度很高;

3) 镜头边界的起始帧和镜头边界结束帧属于不同的镜头,两者的相似度很低。

在本文中利用相似度来对比两帧,采用灰度直方图来求两帧的相似度。首先对每帧提取其灰度直方图,对两帧的直方向量采用余弦来计算相似度,计算方法如式(4)所示:

$$\Phi(i, j) = \cos(f_i, f_j) \quad (4)$$

其中: f_k 是第 k 帧的灰度直方图, $\Phi(i, j)$ 表示第 i 帧和第 j 帧的相似度。从中可以得出相似度的取值区间为 $[0, 1]$,该值越大,两帧相似度越高。

1.2 检测起始帧

从1.1节可知,由于候选段的首帧和镜头边界的起始帧位于同一个镜头,因此候选段首帧和镜头边界起始帧之间的各帧相似度很高。候选段内起始帧之后的各帧与起始帧相似度较低。由于突变只涉及两个相邻的帧,且其结束帧与起始帧相似度较小,故与候选段的首帧相似度也较低。而渐变涉及多个连续的帧,各帧之间的变化较小,且除起始帧外其余各帧与首帧都不属于同一个镜头,在渐变中各帧变化均匀,因而其余各帧与首帧的相似度呈线性递减趋势。

图2所示是一个候选段内各帧与首帧相似度对比变化图,图中展示了切变和渐变两种镜头边界。

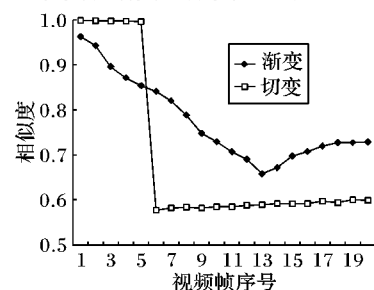


图2 候选段内各帧与首帧的相似度对比

图2中渐变的第3帧是起始帧,切变的第5帧是起始帧。由图2可知,候选段内的各帧和首帧的相似度呈线性递减,而且在渐变中,第3帧之后的各帧与首帧的相似度较小,可以认为之后的各帧与首帧位于不同的镜头;在切变中,第5帧之后的各帧与首帧的相似度较小,与首帧位于不同的镜头。因此镜头边界的起始帧可以被定义为候选段内与首帧相似的最后一帧,可以用一个阈值 T_1 来决定两个帧是否属于同一镜头,满足式(5)的可以认定为起始帧:

$$\Phi(0, k) > T_1 \quad (5)$$

其中第0帧是候选段中的首帧,定义镜头边界的起始帧为 S ,在 S 的检测过程中采用对分比较来减少时间开销。设定 S 处在一个待定区间 $[X_{\text{start}}, Y_{\text{start}}]$ 内,其中 X_{start} 的初始值为0,

Y_{start} 的初始值为 N (假设候选段的长度为 N), 为了防止镜头边界在两个段之间而误判, 对每个段计算时多考虑一帧, 因此该区间比候选段多一帧。检测 S 的步骤如下:

第 1 步 如果 $Y_{\text{start}} - X_{\text{start}} \leq 1$, 则执行第 2 步; 否则计算候选段首帧和 S 待定区间中间帧二者的相似度, 相似度可以表示为 $\Phi(0, s)$, 其中 s 是待定区间中间帧的位置, 计算公式如式(6)所示:

$$s = \lfloor (X_{\text{start}} + Y_{\text{start}}) / 2 \rfloor \quad (6)$$

将相似度 $\Phi(0, s)$ 和阈值 T_1 进行比较, 如果 $\Phi(0, s) \geq T_1$, 说明 S 位于区间 $[s, Y_{\text{start}}]$, 更新 X_{start} 的值为 s , 然后继续执行第 1 步; 否则说明 S 位于区间 $[X_{\text{start}}, s - 1]$, 更新 Y_{start} 的值为 $s - 1$, 然后继续执行第 1 步。

第 2 步 在执行完第 1 步后, 如果 $X_{\text{start}} = Y_{\text{start}}$, 则确定起始帧的位置为 X_{start} ; 否则起始帧的范围被缩减为一个两帧的区间。由于在候选段中的其他帧和首帧的相似度呈线性递减, 因此在两帧的区间内确定起始帧时只需要判定首帧和第 Y_{start} 帧的相似度。如果 $\Phi(0, Y_{\text{start}}) < T_1$, 则第 X_{start} 帧是起始帧; 否则, 第 Y_{start} 帧是起始帧。

起始帧 S 在候选段中的位置确定后, 需要立即检测切变。在候选段中切变只涉及两帧 (起始帧和结束帧)。由于已经确定了起始帧的位置, 为减少时间开销, 可以在 S 确定后即检测切变, 而且切变的检测只需要再执行一次比较即可。

第 3 步 在起始帧 S 确定后, 检测切变时只需要判定起始帧 S 和其下一帧的相似度即可判定切变是否存在。因此如果满足式(7), 即可说明切变检测成功。

$$\Phi(s, s + 1) < T_1 \quad (7)$$

其中: s 是起始帧 S 的位置, $s + 1$ 是 S 的下一帧。如果满足式(7), 说明切变检测成功, 在该候选段内之后的步骤可以忽略; 否则这个候选段内可能存在渐变, 需要继续检测。

1.3 候选段调整

在本文中, 镜头边界检测是基于假设整个边界处于同一个段内, 即镜头边界的起始帧和结束帧都要处于同一个段内。而镜头边界的起始帧和结束帧可能跨越在两个不同的段内, 因此在起始帧确定后需要对整个段进行调整, 以确保起始帧和结束帧处于同一个段内。对于起始帧确定的段, 只需要将段整体向后移动, 同时保持段的长度不变即可, 候选段向后调整之后得到新的候选段, 可以保证结束帧和起始帧都处在这个新的段中。在候选段调整后, 可以进行渐变结束帧的检测。

1.4 检测结束帧

由于在候选段中镜头边界的起始帧和结束帧位于不同的镜头, 因此二者的相似度很小。起始帧 S 的位置在 1.2 节中已经确定, 可以利用 S 的信息来确定结束帧的位置。在候选段调整后, 候选段的第一帧是镜头边界的起始帧, 候选段里渐变的其余各帧与起始帧的相似度呈线性递减趋势。而且由于渐变的结束帧和起始帧处于两个不同的镜头, 因此需要用一个阈值来判断结束帧, 镜头边界的结束帧是候选段里满足式(8)的第一帧。

$$\Phi(s, k) < T_2 \quad (8)$$

其中: T_2 是指在一个镜头边界的起始帧和结束帧的相似度的最小值, s 是起始帧的位置。图 3 是一个关于候选段内渐变各帧与镜头边界起始帧相似度对比的例子, 图中渐变中起始帧后的第 3 帧是结束帧。从图 3 可看出: 候选段内的各帧和起始

帧的相似度呈线性递减, 而且在渐变中, 第 4 帧之后的各帧与起始帧的相似度较小, 可以认为之后的各帧与起始帧位于不同的镜头。

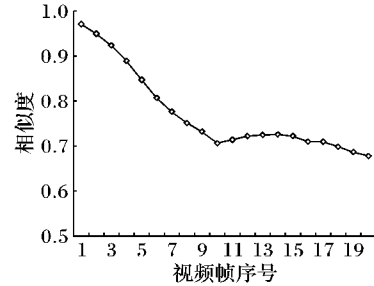


图3 候选段内渐变各帧和起始帧的相似度对比

帧的检测过程和起始帧的检测过程相似。定义镜头边界的结束帧为 T , 假设结束帧 T 处于一个待定区间 $[X_{\text{end}}, Y_{\text{end}}]$, X_{end} 的初始值为 $s + 1$, 即起始帧的下一帧, Y_{end} 的初始值为 N , 即候选段的最后一帧。结束帧的检测步骤如下:

第 1 步 如果 $Y_{\text{end}} - X_{\text{end}} \leq 1$, 则执行第 2 步; 否则计算镜头边界起始帧和 T 待定区间中间帧的相似度。相似度可以表示为 $\Phi(s, t)$, 其中 t 是待定区间中间帧的位置:

$$t = \lfloor (X_{\text{end}} + Y_{\text{end}}) / 2 \rfloor \quad (9)$$

将相似度 $\Phi(s, t)$ 和阈值 T_2 进行比较, 如果 $\Phi(s, t) < T_2$, 说明 T 位于区间 $[t + 1, Y_{\text{end}}]$, 更新 X_{end} 的值为 $t + 1$; 否则说明 T 位于区间 $[X_{\text{end}}, t]$, 更新 Y_{end} 的值为 t 。然后继续执行第 1 步。

第 2 步 在执行完第 1 步后, 如果 $X_{\text{end}} = Y_{\text{end}}$, 则确定起始帧的位置为 X_{end} ; 否则结束帧的待定区间缩减为两帧 (第 X_{end} 帧和第 Y_{end} 帧)。由于在候选段中的其他帧和起始帧的相似度呈线性递减, 因此只需要判定起始帧和第 X_{end} 帧的相似度。如果 $\Phi(s, X_{\text{end}}) < T_2$, 说明第 X_{end} 帧和起始帧位于不同的镜头, X_{end} 是结束帧; 否则, 第 Y_{end} 帧是结束帧。

只有在检测渐变时, 才要求结束帧的位置, 在正常情况下, 渐变的范围就是起始帧和结束帧之间的区间, 为了防止误判, 在检测渐变后需要进行验证。

第 3 步 渐变验证。由于受到光照强度等影响, 一些非渐变区间可能会被误判为渐变, 因此需要对渐变区间进行判定。在整个渐变的过程中, 各帧之间的变化是均匀的, 因此各中心对称的帧与区间中间帧的相似度的差基本一致。定义一个阈值通过判断对称来验证渐变的合理, 利用式(10)验证渐变:

$$|\Phi(s, m) - \Phi(m, t)| < T_3 \quad (10)$$

其中 m 是渐变的镜头边界的中间帧。分别求起始帧、结束帧与中间帧的相似度, 如果相似度的差较小, 满足式(10), 就可以确定镜头渐变区间的正确性。

2 镜头边界检测方法

基于本文讨论, 可以得到一种新的基于预处理的视频镜头检测的方法, 总结该方法其流程如下:

第 1 步 提取视频中可能包含镜头边界的候选段。

视频中的帧序列每 21 帧组成一个段, 利用式(1)对每个段计算其段首帧和尾帧的对应像素的灰度的差的总和, 定义这个总和为段的差异值。对于视频帧段, 每 10 个段组成一个组, 利用式(2) ~ (3) 求出每个段的阈值并确定每个组中的候选段。

第 2 步 检测镜头边界的起始帧。

在候选段中,镜头边界的起始帧和段的首帧处于同一镜头,因此起始帧 S 是和首帧相似的最后一帧,即满足式(5)的段中的最后一帧,利用对分比较可以快速地确定起始帧的位置。

第 3 步 突变检测。

镜头突变涉及连续的两帧(起始帧和结束帧),当起始帧确定时,需要进行一次比较判定突变,通过比较起始帧和起始帧的下一帧,得到两者的相似度,如果满足式(7),就可以判定这是一个突变。

第 4 步 调整候选段。

为了便于检测结束帧,需要保证镜头边界的起始帧和结束帧处于同一个段中,因此在起始帧 S 的位置确定后,需要调整整个候选段,将整个候选段向后移动,保证调整后的候选段长度不变。

第 5 步 检测结束帧。

如果在第 3 步中检测突变失败,需要对候选段进行渐变的检测,由于起始帧的位置已经确定,而且镜头边界的起始帧和结束帧处于不同的镜头,因此结束帧是和起始帧不相似的第一帧,即满足式(8)的第一帧,利用对分比较可以快速确定结束帧的位置。

第 6 步 渐变验证。

在渐变的各帧中的变化是均匀的,因此起始帧和结束帧相对于中间帧是对称的,如果渐变的起始帧和结束帧与中间帧的相似度基本一致,即如果满足式(10),可以确定渐变的正确性。

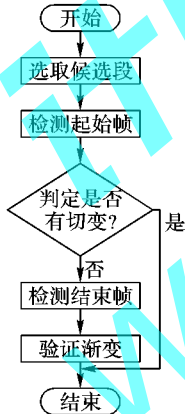


图 4 本文视频镜头边界检测算法流程

3 实验与分析

3.1 衡量标准和实验数据

为验证本文提出的镜头边界检测方法,实验采用了如表 1 所示的 6 段视频数据,其中前四段(V1~V4)视频选自 TRECVID2001,除此之外,本文还选择了另外两段纪录片进行实验。本实验中所有的视频都采用 MPG 格式,所有的视频中共有 150 个渐变和 300 个切变。在相同的实验条件下对这些数据分别用本文方法和文献[14]方法对检测速度和相关参数进行了对比,对比参数如下所示:

查全率 = (正确数) / (正确数 + 丢失数)

准确率 = (正确数) / (正确数 + 错误数)

$F_1 = 2 * \text{准确率} * \text{查全率} / (\text{准确率} + \text{查全率})$

其中 F_1 是对准确率和查全率的一个综合比较参数。

表 1 测试数据视频

视频来源	视频	帧数	切变数	渐变数	镜头边界总数
TRECVID2001 测试数据	V1	11 363	38	26	64
	V2	12 306	32	59	91
	V3	14 186	73	3	76
	V4	16 329	68	6	74
纪录片	D1	11 363	73	21	94
	D2	11 722	26	35	61
合计			310	150	460

3.2 参数选择

如上所述,在本文方法中有三个参数,通过对表 1 中的视频 V1~V3 进行实验选择参数。参数 T_1 在式(5)中出现,用来判定两帧是否处于同一镜头之内,由于帧的相似度是通过帧的灰度直方图的余弦所得,所以相似度的取值区间为 $[0,1]$,如果两帧相似,则其值较大,因此设定 $T_1 = 0.9$ 。参数 T_2 用来检测镜头边界的结束帧,所以镜头边界的起始帧和结束帧处于不同的镜头,因此相似度较低,而且因为 T_2 是检测渐变的参数,起始帧和结束帧之间会有其他帧,所以设定 $T_2 = 0.85$ 。参数 T_3 在式(10)中判断渐变的对称性,如果渐变对称,起始帧和结束帧与中间帧的相似度之差较小,因此设定 $T_3 = 0.2$ 。

3.3 检测速度对比

将文献[14]方法和本文方法在相同的实验条件下对测试视频进行了实验,比较结果如表 2 所示。从表 2 可看出:采用本文方法,可以节约计算时间 15.6%~30.2%,主要原因是文献[14]算法需要对候选段分别进行切变和渐变的检测,会产生较大的时间开销。本文算法简化了候选段的提取过程,在候选段中只需要一次检测过程就可以确定镜头边界,而且在检测起始帧和结束帧的检测时采用了对分比较的策略,因此检测速度有了较大提升。对于包含较多渐变的视频,如表 2 中的视频 V1 和 V2,不需要对候选段进行全面检测切变,因此处理速度提升较多。对于包含大量切变的视频,如 V3 和 V4 只有少数几个渐变,在检测镜头边界时,采用对分对比的方法,也可以减少一定的时间开销。

表 2 检测速度对比结果

视频	检测时间/s		检测时间 减少比例/%
	文献[14]算法	本文算法	
V1	113	82	27.4
V2	139	97	30.2
V3	128	108	15.6
V4	151	121	19.8
D2	83	63	24.1
D3	89	65	26.9

3.4 指标

文献[14]算法和本文算法在准确率、查全率和 F_1 的对比结果如表 3 所示。从表 3 可看出:两种方法的各项参数结果相似,能够保证满意,因为两种方法都是基于候选段的提取,候选段在选择的时候考虑较充分,切变和渐变的各项参数仍有提升,因为文献[14]算法在检测切变时,会对段进行分割处理,可能会导致切变的丢失。而在检测渐变时,由于文献[14]算法认定渐变的长度为候选段的长度,可能会导致渐变的误检。

表 3 查全率、准确率和 F_1 对比结果

%

视频	文献[14]算法						本文算法					
	切变			渐变			切变			渐变		
	查全率	准确率	F_1	查全率	准确率	F_1	查全率	准确率	F_1	查全率	准确率	F_1
V1	85	85	85	76	100	84	92	90	91	88	96	92
V2	72	89	80	68	80	74	84	90	87	75	85	80
V3	93	99	96	100	75	86	96	91	93	100	75	86
V4	91	89	90	67	100	80	99	100	99	83	100	91
D1	87	98	92	50	89	64	93	97	95	76	89	82
D2	97	96	96	43	100	60	100	97	98	89	91	90

4 结语

本文提出一种改进的基于预处理的视频镜头边界检测的方法,对可能包含镜头边界的候选段进行分析计算。首先检测起始帧,然后判断是否存在切变,若存在则终止检测;否则检测结束帧,从而确定镜头边界。实验结果表明,与传统方法比较,本文算法能够较好地解决视频镜头识别的瓶颈问题,提升镜头检测速度。

参考文献:

- [1] COTSACES C, NIKOLAIDIS N, PITAS I. Video shot detection and condensed representation: a review [J]. *IEEE Signal Processing Magazine*, 2006, 23(2): 28–37.
- [2] YUAN J H, WANG H Y, XIAO L, *et al.* A formal study of shot boundary detection [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2007, 17(2): 168–186.
- [3] HANJALIC A. Shot boundary detection: unraveled and resolved? [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2002, 12(2): 90–105.
- [4] CAI C, LAM K M, TAN Z. A unified shot boundary detection method based on linear prediction with Bayesian cost functions [C]// *Proceedings of the 2005 IEEE International Workshop on VLSI Design and Video Technology*. Piscataway: IEEE Press, 2005: 101–104.
- [5] JANWE N J, BHOYAR K K. Video shot boundary detection based on JND color histogram [C]// *Proceedings of the 2013 IEEE the Second International Conference on Image Information Processing*. Piscataway: IEEE Press, 2013: 476–480.
- [6] FU Q, ZHANG Y C, XU L Y, *et al.* A method of shot-boundary detection based HSV space [C]// *Proceedings of the 2013 9th International Conference on Computational Intelligence and Security*. Piscataway: IEEE Press, 2013: 219–223.
- [7] HUO Y, ZHANG P Z, WANG Y F. Adaptive threshold based video shot boundary detection framework [C]// *Proceedings of the 2012 International Conference on Image Analysis and Signal Processing*. Piscataway: IEEE Press, 2012: 1–5.
- [8] SUN X M, LV X Y, ZHANG M W. Novel shot boundary detection method based on support vector machine [C]// *Proceedings of the 2010 International Conference on Computer and Information Application*. Piscataway: IEEE Press, 2010: 56–59.
- [9] HAO X, TONG R F. An improved moving-window method for shot boundary detection [C]// *Proceedings of the 2009 International Conference on Information Engineering and Computer Science*. Piscataway: IEEE Press, 2009: 1–5.
- [10] QIN T F, GU J Y, CHEN H T. A fast shot boundary detection based on k -step slipped window [C]// *Proceedings of the 2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*. Piscataway: IEEE Press, 2010: 190–195.
- [11] BESCOS J, CISNEROS G, MARTINEZ J M, *et al.* A unified model for techniques on video-shot transition detection [J]. *IEEE Transactions on Multimedia*, 2005, 7(2): 293–307.
- [12] GRANA C, CUCCHIARA R. Linear transition detection as a unified shot detection approach [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2007, 17(4): 483–489.
- [13] LI Y-N, LU Z-M, NIU X-M. Fast video shot boundary detection framework employing pre-processing techniques [J]. *IET Image Processing*, 2009, 3(3): 121–134.
- [14] LU Z-M, YONG S. Fast video shot boundary detection based on SVD and pattern matching [J]. *IEEE Transactions on Image Processing*, 2013, 22(12): 5136–5146.
- [3] PFENNIG S, KIRCHNER M. Spectral methods to determine the exact scaling factor of resampled digital images [C]// *Proceedings of the Fifth International Symposium on Communications Control and Signal Processing*. Piscataway: IEEE Press, 2012: 1–6.
- [4] RYU S J, LEE H K. Estimation of linear transformation by analyzing the periodicity of interpolation [J]. *Pattern Recognition Letters*, 2014, 36: 89–99.
- [5] BRAVO S S, NANDI A K. Automated detection and localization of duplicated regions affected by reflection, rotation and scaling in image forensics [J]. *Signal Processing*, 2011, 91(8): 1759–1770.
- [6] KELLER Y, AVERBUCH A, ISRAELI M. Pseudo-polar based estimation of large translations rotations and scalings in images [C]// *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision*. Piscataway: IEEE Press, 2005, 1: 201–206.
- [7] WU Q, WANG S, ZHANG X. Log-polar based scheme for revealing duplicated regions in digital images [J]. *Signal Processing Letters*, 2011, 18(10): 559–562.
- [8] VAZQUEZ P D, PEREZ G F. Prefilter design for forensic resampling estimation [C]// *Proceedings of the 2011 IEEE International Workshop on Information Forensics and Security*. Piscataway: IEEE Press, 2011: 1–6.
- [9] FENG X, COX I J, DOERR G. Normalized energy density-based forensic detection of resampled images [J]. *IEEE Transactions on Multimedia*, 2012, 14(3): 536–545.
- [10] RYU S J, KIRCHNER M, LEE M J, *et al.* Rotation invariant localization of duplicated image regions based on Zernike moments [J]. *IEEE Transactions on Information Forensics and Security*, 2013, 8(8): 1355–1370.
- [11] HSU H C, WANG M S. Detection of copy-move forgery image using Gabor descriptor [C]// *Proceedings of the 2012 International Conference on Anti-Counterfeiting, Security and Identification*. Piscataway: IEEE Press, 2012: 1–4.
- [12] AVERBUCH A, COIFMAN R R, DONOHO D L, *et al.* Fast and accurate polar Fourier transform [J]. *Applied and Computational Harmonic Analysis*, 2006, 21(2): 145–167.
- [13] LIU Q, SUNG A H, RIBEIRO B, *et al.* Image complexity and feature mining for steganalysis of least significant bit matching steganography [J]. *Information Sciences*, 2008, 178(1): 21–36.

(上接第 3326 页)