

基于累积帧的自适应双阈值镜头边界检测算法

邓 丽¹ 金立左² 杨文强¹ 费敏锐¹

(上海大学机电工程与自动化学院 上海 200072)¹ (东南大学自动化学院 南京 230039)²

摘 要 镜头边界检测是基于内容视频检索的重要组成部分。针对渐变的视频序列,提出一种基于累积帧的自适应阈值镜头边界检测方法。考虑用一个累积帧把每一帧之间的差异累积起来,放大渐变的特征变化,利用块匹配算法对运动进行估计和补偿,以减少物体运动对镜头边界检测的影响。通过分析帧亮度和饱和度与阈值的关系,用自适应双阈值提高镜头边界检测的准确性。实验结果表明,该算法可以在较小的计算强度下较好地实现对镜头渐变的检测。

关键词 镜头边界检测,累积帧,运动补偿,自适应双阈值

中图分类号 TN391 文献标识码 A

Shot Boundary Detection Algorithm Based on Self-adaptive Dual Thresholds of Accumulative Frame

DENG Li¹ JIN Li-zuo² YANG Wen-qiang¹ FEI Min-rui¹

(School of Mathematics and Engineering and Automation, Shanghai University, Shanghai 200072, China)¹

(School of Automation, Southeast University, Nanjing 230039, China)²

Abstract Improving the precision of shot boundary detection is very important. This paper presented an algorithm for shot boundary detection based on self-adaptive dual thresholds of accumulative frame. It used an accumulative frame to memorize the accumulative differences among the sequential frames of the video and magnify gradual feature changes. A block-matching algorithm was designed to compensate for the motion of the object to reduce the effects of it. Self-adaptive dual thresholds were adopted to improve the veracity of the shot boundary detection. The proposed method can well detect gradual shot with less calculating strength by experiment on various video clips.

Keywords Shot boundary detection, Accumulative frame, Motion compensation, Self-adaptive dual thresholds

1 引言

镜头是摄像机一次操作所拍摄的内容相似的视频图像,是视频分析和基于内容的视频检索的基础。镜头可以分为切变和渐变两种。切变是指镜头直接切入到下一个镜头,中间没有任何编辑效果,比较容易检测;渐变则是在镜头的变换之间加入一些时间或者空间上的编辑效果,没有明显的视觉跳跃,一般归纳为 4 种基本类型:淡入(fade in)、淡出(fade out)、融化(dissolve)和扫换(wipe)。镜头边界检测就是利用特征差异找到镜头边界,实现视频切分。

对镜头切变的检测,由于切变时帧间距离会有一个明显的波峰,因此用一个门限值作为判断依据,就可得到较满意的结果^[1,2]。而渐变及有物体运动的镜头变化比较复杂,常延续十几或几十帧,两帧图像间特征的相似度差异较小,因此仅仅比较两帧图像来判别渐变镜头比较困难,至今仍未有取得和切变检测效果一样的成果,方法主要有光流法^[3]、模型法^[4,5]、智能算法^[6,7]以及双阈值法^[8-10]等。光流法的原理是镜头渐变切换时没有光流,而镜头运动应适合某种特定的光流类型。模型法是利用视频编辑模型来进行镜头边界检测,需要对每种切换类型建立模型,而且建模过程比较复杂。智能算法虽然检测效果较好,但是算法较为复杂,其实用性无法

保证。双阈值(twin-comparison)是一种能够同时检测突变和渐变的镜头边界检测方法,目前应用最为广泛,由 Zhang 等在文献[8]中最早提出,但其阈值是预先设定的全局固定阈值。文献[9]在文献[8]的基础上提出根据直方图获取的局部自适应双阈值,实现镜头检测,但是因为直方图的求取机制,发生帧数较长的渐变会被漏掉。文献[10]考虑到视频序列中图像帧序列之间存在一定的运动,对视频中图像帧进行了运动补偿,基于滑动窗口获取局部自适应双阈值。但是此方法中的当前帧仅与参考帧相比较,参考帧的选取对算法影响较大。

本文利用累积帧方法来放大帧间的特征变化,将当前帧与累积帧进行比较,同时结合块匹配算法对运动进行估计和相应的补偿,考虑颜色亮度和饱和度两个方面的影响,得出由颜色亮度和饱和度确定的阈值选取共识,这具有一般性,并且这样选取的阈值对不同镜头根据颜色亮度和饱和度做相应的调整,具有自适应性。

2 算法描述

2.1 累积帧的定义

把一段视频序列 V_i 的第一帧记为 f_1 , 作为一个镜头的开始帧,同时作为第一个累积帧,累积帧 $f_c = f_1$ 。第二帧记为 f_2 , 与 f_1 比较, f_1 与 f_2 相似部分完整地保存在 f_c 中。不

到稿日期:2011-06-17 返修日期:2011-11-10 本文受国家自然科学基金(60774059),上海市科委重点基础项目(10JC1405000)资助。

邓 丽(1978-),女,博士,讲师,主要研究方向为多媒体信息处理、智能学习算法及模式识别,E-mail:dengli002003@126.

相似部分在记忆矩阵 R 中做标记,也保存在 f_c 中。累积帧 f_c 记录 f_1 与 f_2 之间相似和不相似的部分。然后 f_c 与后面的 f_3, f_4, \dots 进行比较,直到不相似部分的累积超过预定的阈值,则认为一个镜头结束,并更换累积帧为下一个镜头的起始帧,继续与后面的帧比较。

2.2 块匹配算法

本文算法利用新三步搜索算法^[11]进行视频帧块的搜索。在实际应用时,累积帧与当前帧对应块比较。如果不相似,则以当前帧的当前块为中心,搜索与累积帧当前块最相似的块。如图1所示,累积帧中的块A与当前帧中的块A*为对应块,经过比较,块A与块A*不相似,则用新三步搜索法,在当前帧块A*周围搜索与累积帧块A相似的块。

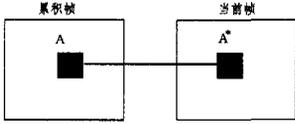


图1 累积帧与当前帧比较

在累积帧与当前帧相应块的块匹配搜索中,引入一个 16×16 的记忆矩阵 R 来有效地记录这些信息,清晰地表明对应块的相似性、移动的方向或不相似性。 R 中的每一元素与帧的每一块相对应, R 的值代表了相应的含义。如果相应块相似,则对应的元素值为0;如果经过搜索没有找到相似块,则标为-1;如果搜索到相似块,则按相似块相对块 (k, l) 的位置在 R 中标记。 R 中元素所代表的含义如表1所列。

表1 记忆矩阵中元素值的含义

R 中元素值	代表含义
0	对应块相似
-1	没有搜索到相似的块
1	移动到左上
2	移动到正上
3	移动到右上
4	移动到正左
5	移动到正右
6	移动到左下
7	移动到正下
8	移动到右下

表1中不相似块的数目是 R 中负数元素数目的总数。当不同块的总数超过全局阈值 T_{block} 时,说明有一个镜头变换。

2.3 自适应双阈值选取

定义两个变量^[12]:当前帧的平均颜色值 local value 和当前饱和度 sat,具体见式(1)和式(2),在 RGB 空间下,可以通过式(3)转换得到饱和度的值。

$$\text{local_value} = \text{mean} \sum_{i=1}^m \sum_{j=1}^n \sum_k^{r, g, b} C_{i,j,k} \quad (1)$$

$$\text{sat} = \text{mean} \sum_{i=1}^m \sum_{j=1}^n S_{i,j} \quad (2)$$

$$S = 1 - \frac{3}{r+g+b} [\min(r, g, b)] \quad (3)$$

式中, m, n 为当前帧的宽度和长度, $i=1, 2, \dots, m, j=1, 2, \dots, n$, $\sum_k^{r, g, b} C_{i,j,k}$ 为点 (i, j) 在 RGB 空间中的颜色值, $S_{i,j}$ 为点 (i, j) 的饱和度值。

通过一些先验知识以及多次实验,发现全局阈值与颜色平均值成正比关系,与饱和度的平方成正比关系,而局部阈值与颜色平均值、饱和度的均方根成正比。这样就找到一个比较

合适自适应阈值的公式,全局阈值 T_{block} 见式(4),局部阈值 T_{pixel} 见式(5),其中 a, b 为待定系数,实验中选择 $a=10, b=40$ 。

$$T_{block} = a \times \text{local_value} \times \text{sat}^2 \quad (4)$$

$$T_{pixel} = b + 10 \sqrt{\text{local_value} \times \text{sat}} \quad (5)$$

对第一个镜头的开始帧利用式(1)式(3)计算平均颜色 local_value 和饱和度 sat,然后根据式(4)、(5)得到全局阈值和局部阈值。当检测到镜头边界后,对下一个镜头的开始帧按上面的方法重新计算全局阈值和局部阈值。这样对不同的镜头,阈值在不断地更新,直至找到较合适的阈值。

2.4 算法实现

视频的每一个帧都被分割为 16×16 相等大小的块,累积帧 f_c 每一小块与帧 f_i 相应块进行比较。在累积帧与当前帧对应块比较块相似性时,用到一个局部阈值 T_{pixel} 。若找到相似块,则认为是块运动;若没有找到相似块,则认为块不相似,并在记忆矩阵 R 中标记。统计 R 中标记的不相似块个数,块数的多少反映帧的差异程度。当块数超过一个全局阈值 T_{block} 时,说明有一个镜头切换。按照式(5)相应地更新局部阈值和全局阈值。算法具体步骤如下:

步骤1 累积帧 f_c 每一小块与帧 f_i 的对应小块在 RGB 空间进行点对点像素值比较,两帧任意块 (k, l) ($0 \leq k \leq 16, 0 \leq l \leq 16$) 对应点 (x, y) 的差异值记为 $e_{c,i}$:

$$e_{c,i} = \frac{r, g, b}{j} |f_c(x, y, j) - f_i(x, y, j)| / 3 \quad (6)$$

步骤2 $e_{c,i}$ 与局部阈值 T_{pixel} 进行比较,若 $e_{c,i} > T_{pixel}$,则块差异 $e_{(c,i)kl}$ 累积:

$$e_{(c,i)kl} = \begin{cases} e_{(c,i)kl}, & \text{若 } e_{c,i} \leq T_{pixel} \\ e_{(c,i)kl} + 1, & \text{若 } e_{c,i} > T_{pixel} \end{cases} \quad (7)$$

式中, $e_{(c,i)kl}$ 表征了累积帧 f_c 与 f_i 对应块 (k, l) 的差异程度。随后对 $e_{(c,i)kl}$ 进行归一化处理,见式(8),其中 K 是块的长度, L 是块的宽度:

$$e_{(c,i)kl} = \frac{e_{(c,i)kl}}{K \times L} \quad (8)$$

步骤3 $e_{(c,i)kl}$ 与局部阈值 T_{pixel} 进行比较,若 $e_{(c,i)kl} > T_{pixel}$,则认为累积帧 f_c 与帧 f_i 对应块 (k, l) 不相似;

步骤4 在帧 f_i 的块 (k, l) 的周围采用三步搜索的思想进行块匹配,寻找相似块。若找到相似块,则把对应块复制,并在记忆矩阵 R 中做出相应标记。若没有找到相似块,则 $E_{c,i}$ 累积,且记忆矩阵 R 中的对应元素记为-1,见式(9)。这样 $E_{c,i}$ 就记录了累积帧 f_c 与帧 f_i 不相似块的数目:

$$E_{c,i} = \begin{cases} E_{c,i} \text{ 且 } R_{k,l} \text{ 为表1中的对应值,其他} \\ E_{c,i} + 10 \text{ 且 } R_{k,l} = -1, \text{ 若 } e_{(c,i)kl} > T_{pixel} \text{ 且周围没有相似块} \end{cases} \quad (9)$$

步骤5 $E_{c,i}$ 与全局阈值 T_{block} 比较,若小于等于 T_{block} ,则认为属于一个镜头,累积帧 f_c 与下一帧 f_{i+1} 的对应块进行比较,若大于 T_{block} ,则认为检测到一个镜头边界。

3 实验与分析

实验随机选取 CNN 的5段视频(mpeg格式,内容包括切变镜头、渐变镜头以及物体运动),并人工标定每个视频片度所包含的镜头,采用如下定义的查全率和查准率进行量化比较:

查全率=正确检测镜头个数/(正确检出镜头数+漏检镜头数)

查准率=正确检测镜头数/(正确检出镜头数+误检镜头数)

3.1 序列图片切变镜头算法实验

先对5段不同的连续图像帧(每段包含近百帧,一般包含2~3个切变镜头变换)进行初步算法实验。5段连续图像帧的实验结果记录在表2中。实验表明,切变镜头的检测全部准确,本文算法对镜头切变具有很好的效果。

表2 5段连续图像帧切变镜头算法测试结果

视频片断	视频描述	帧数	切变	漏检	误检
CNN 片断 1	2 切变	96	2	0	0
CNN 片断 2	2 切变	112	2	0	0
CNN 片断 3	3 切变	139	3	0	0
ABC 片断 1	3 切变	87	3	0	0
ABC 片断 2	2 切变	103	2	0	0

3.2 序列图片渐变镜头算法实验

对5段包括典型渐变镜头的连续图像帧进行实验。以其中一段连续图片帧 CNN 片段 5(包含3个渐变镜头)为例,用本文算法,对整个累积差异的变化过程作出累积差异曲线,如图2所示。

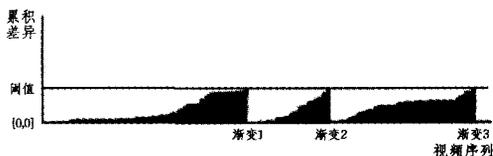


图2 CNN 片段 5 的渐变镜头累积差异曲线

从图2可以看出,渐变镜头帧间变化逐帧积累,累积差异不断增大,逐渐达到阈值。可以看出累积帧放大了渐变镜头帧间的特征变化,使渐变镜头边界的检测变得更容易。

5段包含渐变镜头的连续图像帧(其中有些镜头包含了物体运动)的实验结果记录在表3中。实验表明,渐变镜头的检测基本准确,本文的算法对镜头渐变具有较好的效果。

表3 5段连续图像帧渐变镜头算法测试结果

视频片断	视频描述	帧数	切变	渐变	漏检	误检	运动物体
CNN 片断 4	1 渐变	104	0	1	0	0	1
CNN 片断 5	3 渐变	273	0	3	0	0	0
CNN 片断 6	1 切变 2 渐变	368	1	1	1	0	0
ABC 片断 3	2 切变 1 渐变	247	2	1	0	0	1
ABC 片断 4	2 切变 2 渐变	445	2	2	0	1	1

3.3 固定阈值时算法查全率和查准率实验

对选取 CNN 的3段和 ABC 的2段 mpeg 视频进行实验,实验结果如表4所列。

表4 本文算法检测镜头边界的实验结果

视频片断	视频描述	帧数	切变	渐变	漏检	误检	物体运动	镜头边界检测结果统计	
								查准率	查全率
CNN 片断 7	13 切变	1398	13	0	0	2	0	86.67%	100%
CNN 片断 8	16 切变 1 渐变	2106	16	1	0	4	1	80.95%	100%
CNN 片断 9	8 切变 2 渐变	725	10	2	0	2	0	85.71%	100%
ABC 片断 5	12 切变 2 渐变	1120	12	1	1	3	1	81.25%	92.86%
ABC 片断 6	19 切变 3 切变	2753	19	2	1	5	1	80.77%	95.45%

实验说明本文算法对镜头的检测有一定的效果。对切变镜头边界的检测比较准确,查全率和查准率都比较高,而在渐

变镜头边界检测中存在漏检和误检情况。在这一实验中,阈值是静态的,取经验值,没有根据不同镜头颜色的亮度和饱和度更新阈值。下面将做自适应阈值的算法实验。

3.4 自适应阈值时算法查全率和查准率实验

用2.3小节分析讨论的阈值选取公式,在每个镜头的开始帧计算该镜头的阈值。这样对不同的镜头阈值会有相应的改变,阈值对镜头边界的判断会更加准确。把阈值的更新放在算法中,然后用3.3小节中实验的5段视频做镜头边界检测实验,结果如表5所列。

表5 自适应阈值算法检测镜头边界的实验结果

视频片断	视频描述	帧数	切变	渐变	漏检	误检	物体运动	镜头边界检测结果统计	
								查准率	查全率
CNN 片断 7	13 切变	1398	13	0	0	2	0	86.67%	100%
CNN 片断 8	16 切变 1 渐变	2106	16	1	0	2	1	89.95%	100%
CNN 片断 9	8 切变 2 渐变	725	10	2	0	1	0	92.31%	100%
ABC 片断 5	12 切变 2 渐变	1120	12	1	0	2	1	86.67%	100%
ABC 片断 6	19 切变 3 切变	2753	19	2	0	3	1	87.50%	100%

综合分析表5可以知道这种情况下算法的查全率为100%,没有漏检现象。但仍然对一些镜头产生误检,对5段视频的实验结果用统计的方法计算其查准率为86.67%。与表4相比,可以看出阈值更新对镜头边界检测有改进,查全率和查准率都有所提高。

本文算法与典型的模板匹配算法以及颜色直方图匹配算法的查准率和查全率比较如表6所列。比较发现本文算法在查全率和查准率方面均优于颜色直方图匹配算法和模板匹配算法。可以看出本文算法对镜头边界检测是有效的。

表6 本文算法与传统算法的比较

	模板匹配算法	颜色直方图匹配算法	本文算法
查准率	78.48%	85.50%	86.67%
查全率	100%	95.16%	100%

结束语 本文对视频镜头边界检测进行了研究,从一个新的角度考虑视频序列的帧间差与镜头变化的关系,提出了一种镜头边界检测的新思想。利用累积帧把相邻若干帧的差异累积起来,体现了镜头变化的过程,对渐变镜头的检测很有帮助;利用块匹配补偿物体运动,减少了物体运动对镜头分割的影响;讨论了阈值与图像颜色亮度和饱和度的关系,并据此在镜头变换时及时对阈值进行更新,使得阈值对不同的镜头都有良好的自适应性,对视频镜头的边界判断更加有效。实验证明此算法对切变镜头和渐变镜头都有较好的效果。

参考文献

- [1] Choubey S K, Raghavan V V. Generic and fully automatic content-based image retrieval using color[J]. Pattern Recognition Letters, 1997, 1(8): 1233-1240
- [2] Theodore V. Cut detection in video sequences using phase correlation [J]. IEEE Signal Processing Letters, 2000, 7(7): 173-175
- [3] 刘振宇,周莉,陈杰.一种基于运动补偿的运动检测和边缘检测算法[J].微电子学与计算机, 2010, 27(11): 10-13
- [4] Chau W S, Chong T S, Chan T W, et al. Efficient Scene Change Detection in MPEG Compressed Video Using Composite Block Averaged Luminance Image Sequence [C]// Proceedings of the Fifth International Conference on Information, Communications and Signal. 2005, Bangkok, Thailand, 2005: 688-691

(下转第296页)

从表 7 可以看出, GPU 双精度算法相比 CPU 双精度算法局部加速比为 134, 整体加速比为 14.3。

4.2 GPU 混合精度性能测试与分析

对程序在 GPU 上使用双精度和混合精度算法做性能测试, 得到表 8 中数据。由表 8 可以看到, 当局部 GPU 并行时, 混合精度与双精度加速比比值为 1.6~1.8, 整体效果为 1.3 左右。

针对程序进行结构优化, 对混合精度 GPU 有限元算法整体性能及精度进行测试。表 8 是我们测得的性能实验结果。

在数值精度方面, 我们测得的结果是有 12 位有效数值, 精确度在 10 到 11 位, 满足精度需要, 与上述精度结果一致。

表 8 GPU 双精度混合精度算法时间

时间	GPU 双精度	GPU 混合精度	GPU 双/GPU 混合
求内力	128	76.7	1.67
求合力	892	662	1.35
总时间	1305	971	1.34

由表 8 可得: 与 CPU 双精度相比, 在网格结点为 112211 时, GPU 双精度算法的总体性能加速比(不含输出)为 14~15, GPU 混合精度算法的总体性能加速比(不含输出)为 19~20。而单就计算内力 F(不含组装)部分, 亦即采用 GPU 计算对应部分(包括 GPU 计算, GPU 传输), 在网格结点数为 112211 时, GPU 双精度加速比为 134 倍, GPU 混合精度加速比高达 224 倍。

结束语 本文提出了一种混合精度显格式有限元算法, 用于材料强非线性多尺度模拟。混合精度显格式有限元程序实现了 90% 以上的有限元计算通过单精度完成, 其计算结果与全部使用双精度一致。

参 考 文 献

[1] Strzodka R, Göddeke D. Mixed precision methods for convergent iterative schemes[C]//Proceedings of the 2006 Workshop on Edge Computing Using New Commodity Architectures. May 2006; 59-60

[2] Langou Ju-lie, Langou Ju-lien, Luszczyk P, et al. Exploiting the performance of 32 bit floating point arithmetic in obtaining 64

(上接第 260 页)

[5] Yoo H W, Ryoo H J, Jang D S. Gradual shot detection using localized edge blocks [J]. Multimedia Tools and Application, 2006, 28(3): 283-300

[6] 曹建荣, 蔡安妮. 压缩域中基于支持向量机的镜头边界检查算法 [J]. 电子学报, 2008, 36(1): 203-208

[7] Zhang N, Xiao G Q, Jiang J M, et al. BPNN Algorithm towards Shot Boundary Detection [J]. Journal of Computer Application, 2009, 29(5): 1369-1372

[8] Zhang H J, Kan kan halli A, Smoliar S W. Automatic partitioning of full motion video [J]. Multimedia System, 1993, 1(1): 10-28

bit accuracy (revisiting iterative refinement for linear systems) [C]//Proceedings of the 2006 ACM/IEEE conference on Supercomputing. 2006

[3] Kurzak J, Dongarra J J. Implementation of mixed precision in solving systems of linear equations on the CELL processor[M]. Concurrency Computat. Pract. Exper. to appear

[4] Göddeke D, Wobker H, Strzodka R, et al. Co-processor acceleration of an unmodified parallel solid mechanics code with FEASTGPU[M]. Accepted for publication in the International Journal of Computational Science and Engineering, 2008

[5] Göddeke D, Strzodka R. Performance and accuracy of hardware-oriented native-, emulated- and mixed-precision solvers in FEM simulations (part 2: Double precision GPUs)[R]. Technical University Dortmund, 2008

[6] Göddeke D, Strzodka R, Turek S. Performance and accuracy of hardware-oriented native-, emulated- and mixed-precision solvers in FEM simulations[J]. International Journal of Parallel, Emergent and Distributed Systems, Special Issue: Applied Parallel Computing, 2007, 22(4): 221-256

[7] Strzodka R, Göddeke D. Pipelined mixed precision algorithms on FPGAs for fast and accurate PDE solvers from low precision components[C]//Proceedings of the 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'06). 2006; 259-270

[8] Li X S, Demmel J W, Bailey D H, et al. Design, implementation and testing of extended and mixed precision BLAS[J]. ACM Transactions on Mathematical Software (TOMS), 2002, 28(2)

[9] Cecka C, Lew A J, Darve E. Assembly of Finite Element Methods on Graphics Processors[M]. Int. J. Numer. Meth. Engng., 2000; 1-6

[10] Kurzak J, Dongarra J. Implementation of mixed precision in solving systems of linear equations on the Cell processor[J]. Concurrency and Computation: Practice and Experience, 2007, 19(10): 1371-1385

[11] Taiji M, Narumi T, Ohno Y, et al. Protein Explorer: A Petaflops Special-Purpose Computer System for Molecular Dynamics Simulations[C]//Proc. Supercomputing. 2003

[9] Wang J Y, Luo W. A Self-adapt in g Dual-threshold Method for Video Shot Transition Detection[C]//IEEE International Conference on Networking, Sensing and Control. 2008, 4: 704-707

[10] 张玉珍, 杨明, 王建宇, 等. 基于运动补偿和自适应双阈值的镜头分割[J]. 计算机科学, 2010, 37(9): 282-286

[11] 薛亮, 于敏, 张正炳. 一种改进的运动计算法——新三步搜索法 [J]. 电子技术, 2004, 4: 22-27

[12] Zheng J, Zou F M, Shi M. An Efficient Algorithm for Video Shot Boundary Detection[C]//IEEE Proceeding of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing. Hong Kong, 2004; 266-269