

一种新的基于运动矢量的 MPEG VIDEO 码流 cut 检出算法

周小明 李凤亭

(清华大学电子工程系, 北京 100084)

摘要 介绍了一种对 MPEG VIDEO 码流中场景切换点进行高速检出的新方法, 在计算各帧相对于预测帧的类似度之后进行了类似度的归一化, 降低了 cut 检出算法的复杂度, 提高了检出的速度。

关键词 cut 运动矢量 宏块 类似度

0 引言

MPEG 作为视频编码的标准, 其应用范围已越来越广泛。大量的 MPEG VIDEO 码流保存在数据库中, 对这些码流的检索、编辑等工作的工作量较大。为了浏览一段 MPEG VIDEO 而把它播放一遍显然消耗的时间太长而且需要较多人工干预。为了减少所需的时间和人力, 我们可以研究图象序列的特点。图象序列所记录的内容是发生在一个个场景中的, 对于每个场景所持续的时间内, 图象序列的各帧总体来看内容变化不大。但不同的场景下, 内容变化较大。因此只要能找出图象序列中每次场景变化后的第一帧, 仅仅播放这些帧就能大概知道该 MPEG VIDEO 的内容, 由此可以节省人工干预的时间。这些帧被称为 cut。

cut 的另一个重要应用就是视频编辑。因为编辑过程中需要对内容作取舍, 一般取舍的最小单位就是一个场景, 因此场景切换点 (cut) 在 MPEG VIDEO 码流中的精确定位, 对于提高视频编辑的速度和效率是十分重要的。

为了检出 MPEG VIDEO 码流中的 cut, 传统的方法是先对码流中的各帧解码, 然后对相邻帧的内容进行差值比较。这种方法要做 IDCT, 因此所花的时间比较长。

MPEG VIDEO 码流中的运动矢量表达了前后帧之间的类似程度, 因此也体现了特定帧是 cut 的可能

性。为了减少 cut 检出的时间, 针对 MPEG VIDEO 码流的特点, 在相关算法的基础上, 本文提出了一种新的基于运动矢量的 MPEG VIDEO 码流的 cut 检出算法。

1 MPEG VIDEO 码流中的运动矢量

MPEG VIDEO 码流中的各帧被划分成多个 16×16 的宏块 (Macroblock), 运动估计和运动补偿就是针对这些宏块进行的。一个宏块的数据包括运动矢量以及 6 个补偿块的 (Block) 的 DCT 系数。

在 I 帧中, 所有的宏块都编码而且都是 Intra-coded Macroblock, I 帧不依赖于其它帧就可以解码。

在 P 帧中, 可能有跳过的宏块 (Skipped Macroblock), 这些宏块没有编码。MPEG 规定, P 帧中 Skipped Macroblock 的运动矢量 (只有前向) 取零, 补偿块数据取全零, 由此规定可以重建 Skipped Macroblock。对于未跳过的宏块, 虽然宏块本身已编码, 但它们的运动矢量却可能没有编码, 因而也成了 Intra-coded Macroblock, 这种情况是因为预测效果不好而放弃了预测, 直接对图象数据作 DCT 的结果。

在 B 帧中, 也可能有 Skipped Macroblock。MPEG 规定, B 帧中 Skipped Macroblock 的运动矢量 (前向及后向) 取上一个宏块的运动矢量 (对应的前向及后向), 补偿块数据取全零, 由此规定亦可重建 Skipped Macroblock。对于那些编码的宏块, 跟 P 帧情况类

似,它们的运动矢量也可能没有编码,或者两个运动矢量中只有一个编码。这种情况的原因同样是两个方向预测效果都不好或者某个方向预测效果不好。

2 基于运动矢量的 MPEG VIDEO 码流 cut 检出算法

为了检出 MPEG VIDEO 码流中的 cut,可以先算出各帧相对于前面若干帧的类似程度。如果该帧为 cut,它跟前面帧之间的类似程度必然较低;如果不是,类似程度相对就较高。因此 cut 帧类似程度在各帧的这种类似程度中就成为一个谷底。利用这种特性可以检出码流中的 cut。

2.1 各帧相对于预测帧类似度的计算

设帧 F 为要计算类似度的帧及 P 为它的前向类似度, B 为后向类似度, F_p 为它的前向预测帧, F_b 为它的后向预测帧。它们之间的关系如图 1 所示。

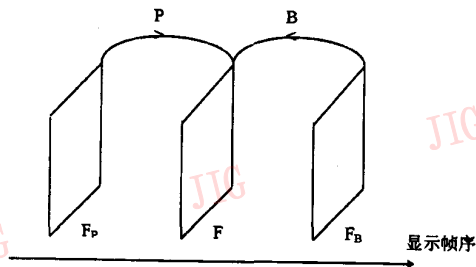


图 1 预测帧和类似度的关系

对于 I 帧,没有 F_p 、 F_b ,故 $B=0$ 、 $P=0$ 。对于 P 帧,没有 F_b ,故 $B=0$ 。不失一般性,对 I 、 P 、 B 帧三种情况下,考虑到帧是由宏块组成,它的类似度可由宏块的类似度得到:

$$P = \frac{1}{N_{MB}} \sum_{i=1}^{N_{MB}} PMB(i)$$

$$B = \frac{1}{N_{MB}} \sum_{i=1}^{N_{MB}} BMB(i)$$

其中 N_{MB} 为帧 F 中宏块总数。 $PMB(i)$ 、 $BMB(i)$ 分别为宏块 i 的前向、后向类似度。

对于宏块 i ,它是由前向预测结果和后向预测结果的平均值来重建的。对于一个方向的预测来说,运动矢量的值大表明该宏块相对于预测帧的

应位置已有了较大偏移,或者仅仅是两个宏块的内容相近,但不是同一物体上的宏块。这种情况下两个宏块的类似度可定义为较低。运动矢量的值小或者为零表明该宏块相对于预测帧的对应位置移动较小,这种情况下两个宏块的类似度可定义为较高。对于 Skipped Macroblock,从前面的论述中可以看到,它们的运动矢量被规定为零(P 帧中)或者跟前一个宏块相同(B 帧中),所以它们跟预测帧的对应位置的类似程度最高。对于运动矢量未编码的宏块,它们只能靠本宏块中的 DCT 系数来重建,没有预测关系,因此类似程度最低。综上所述,可以看到宏块的类似度与它的运动矢量成反比关系。由此,可定义:

$$PMB = \begin{cases} 1 & \text{Skipped Macroblock} \\ 0 & \text{No Motion Vector} \\ \frac{1}{MV_{hp} + MV_{vp}} & \text{else} \end{cases}$$

$$BMB = \begin{cases} 1 & \text{Skipped Macroblock} \\ 0 & \text{No Motion Vector} \\ \frac{1}{MV_{hb} + MV_{vb}} & \text{else} \end{cases}$$

其中 MV_{hp} 、 MV_{vp} 、 MV_{hb} 、 MV_{vb} 分别为前向水平方向、前向垂直方向、后向水平方向、后向垂直方向的运动矢量。

2.2 各帧类似度的计算

以上方法求出了各帧相对于它的预测帧的类似度,但对于 I 、 P 、 B 帧 3 种情况,求得的类似度的期望值是不同的,这样 cut 就不一定处于各帧类似度的谷底。为了使各帧类似度的期望值归一化,还要将以上求得的类似度转化成各帧相对于前面若干帧的类似度,后者用符号 SIM 表示。

如图 2 所示, $P_1, P_2, \dots, P_n, B_1, B_2, \dots, B_n$ 为用 2.1 的方法求出的相对于预测帧的类似度。考察帧 F_m ,从帧 F_{m+1} 到帧 F_n 的前向类似度 P_{m+1} 到 P_n 体现了这些帧与帧 F_0 的类似程度,这些类似度大说明它们是 cut 的可能性小。但 F_m 处于这些帧和预测帧之间,所以也能说明帧 F_m 是 cut 的可能性小。对于后向类似度也有相似的结论。为此,定义:

$$SIM_m = \sum_{j=m}^n P_j \alpha(0, j) + \sum_{j=1}^{m-1} B_j \beta(j, n)$$

$$m = 1, 2, \dots, n$$

其中 α 、 β 为加权系数。在实际中,为了简化计算,可直接取 $\alpha(m, n) = \beta(m, n) = |m - n|$ 。由此求出的类似度 SIM_m 是多帧类似度加权相加的结果,它跟

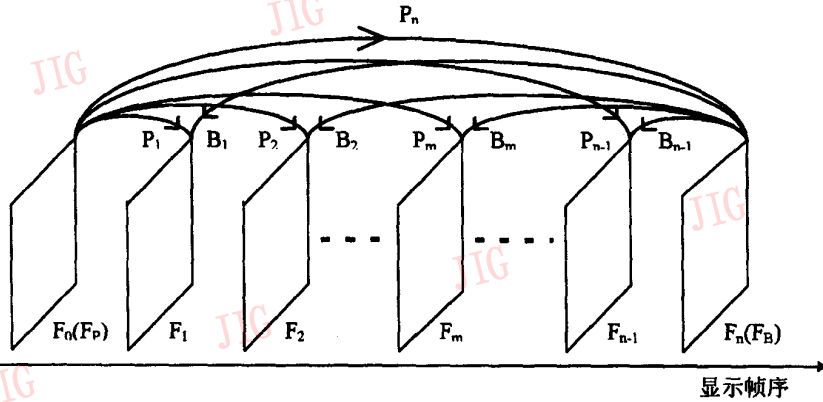


图2 类似度期望值归一化

F_P 到 F_B 之间的帧数,即 n 有关。另外,当 F_B 是 I 帧时, $P_n = 0$ 。为了最后归一化各帧的类似度,可以这样得到帧 F_m 的类似度:

$$SIM_m = \frac{SIM'_m}{\lambda}$$

$$\lambda = \begin{cases} n & F_B \text{ is } P \text{ frame} \\ n - 1 & F_B \text{ is } I \text{ frame} \end{cases}$$

以前的算法一般都是在求出各帧相对于预测帧的类似度之后就进行 cut 检出^[1],没有进行本文中所给出的期望值归一化。此时的类似度是相对于预测帧的,即使是一个没有任何运动的序列,其 IPB 帧因为编码方式不同也可能得到不同的类似度。因此以前的算法都在 cut 检出步骤中使用了比较复杂的滤波器,以期能在这种期望值不一的类似度序列中找出 cut 帧,算法实现的时间比较长。本文算法在进行了类似度归一化之后, cut 检出就可以使用比较简单的门限判断了,减少了整个算法的实现时间。

2.3 cut 检出

从 2.2 得到各帧相对于前面若干帧的类似度,这样就构成了一个类似度序列,序列号即帧号。考察序列 $SIM_0, SIM_1, \dots, SIM_n$, 对于帧 $F_m (0 < m < n)$, 取

$$V_r = SIM_{m-1} - SIM_m$$

$$V_l = SIM_{m+1} - SIM_m$$

且设置条件:

- ① $V_r > T_1$ 且 $V_l > T_2$
- ② $V_l > T_1$ 且 $V_r > T_2$
- ③ $V_r + V_l > T_3$

当 V_r, V_l 满足③且满足①、②之一时,判断帧 F_m 为 cut, 其中 3 个门限满足 $0 < T_1 < T_3, 0 < T_2 < T_3$ 。

3 实验结果

在我们的实验中,为了提高运算速度,对于宏块类似度的计算采用了比较简单的公式,即用运动矢量的编码长度代替了运动矢量的值大小。从 MPEG 标准中运动矢量的码表可以看出,码长与运动矢量的大小是单调函数。这种替代对 cut 检出结果的影响经实验证明是可以忽略的。

我们针对几个序列做了实验,结果如表 1 所示。

表 1 实验结果

	实际 cut 数	检出 cut 数	误检数	漏检数	执行速度 (帧/秒)
序列 1	36	36	2	2	47.8
序列 2	8	9	1	0	49.1
序列 3	38	40	5	3	47.4

注:表中所示时间为 Pentium-75 机器上的结果,包括文件读取、句法分析和 cut 检出等所有步骤。

可以看出,运算的速度已达到实时且 cut 的定位比较准确。对于运动剧烈的 VIDEO,误检率将增大,可以提高门限来降低误检率,但同时也增大了漏检率。表中的误检多是发生在 VIDEO 中运动剧烈的片断。对于黑暗背景和用图象渐变或特技镜头实现场景切换的 VIDEO,漏检率将增大,可以降低门限来降低漏检率,但同时也增大了误检率。由此,对于误检率和

漏检率,只能根据实际需要采取相应的折衷措施。

4 结束语

基于运动矢量的 MPEG VIDEO 码流 cut 检出算法充分利用了 MPEG VIDEO 码流中的运动矢量信息,避免了做 IDCT 的时间开销,因而提高了 cut 检出的速度。较以前的类似算法^[1,2],本方法在用运动矢量求出各帧的类似度 P 、 B 之后,进一步在各帧之间进行了这种类似度的加权平均和归一化,使得最终获得的类似度 SIM_n 更能体现该帧是场景切换点(cut)的可能性,简化了随后的类似度检出。所采用的加权系数虽然比较简单,但经实验证明效果是很好的。比起在类似度检出中对各帧类似度 P 、 B 进行复杂滤波的算法^[1],节省了计算时间,提高了

cut 检出的速度。但该方法还有很多有待改进的地方。例如对于 Closed GOP 和用图象渐变或特技镜头实现场景切换的情况,在现有的算法中只是作了简单的处理,还有待更深入的研究才能提出更好的解决方案。对于其它类型的 VIDEO 码流,如 H.263、MPEG-2 等,它们采用了跟 MPEG 类似的运动补偿机制,因此也可以用类似的方法对这些码流进行 cut 的高速检出。

参考文献

- 1 堀修等. 画像メディア編集技術の中の見る技術. 画像の认识理解シンポジウム(MIR'96), 1996, 7.
- 2 Zhang H J, Kankanhalli A, Smoliar S W. Automatic Partitioning of Full-Motion Video. *Multimedia System*, 1993:10~28.
- 3 ISO CD 11172-2. Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5Mbit/s, International Standard.

周小明 1995年在清华大学电子工程系获学士学位,1998年获清华大学电子工程系信息与信号处理专业硕士学位。主要研究方向为多媒体信息处理,现从事 MPEG-2 视频和音频编解码的研究工作。



李凤亭 清华大学电子工程系图象信息教研室副教授。1970年于该系毕业留校从事通信及图象系统研究工作。多次作为主要成员或负责人参加国家重大工程和重点科技攻关项目,获电子部,国家教委和国家科技进步奖。现重点从事视频编码、处理及网络多媒体通信技术与系统方面的研究工作。



A New MPEG Video Stream Cut Detect Algorithm Based on Motion Vector

Zhou Xiaoming, Li Fengting

(Department of Electronics Engineering, Tsinghua University, Beijing 100084)

Abstract A new fast MPEG video scene cut detection algorithm is proposed in this paper. The similarity of a frame to its predictive frames was calculated based on corresponding motion vector in its MPEG stream and was further normalized. Then, a scene cut can be detected according to the normalized similarity. Test results have shown that this method greatly increased the speed and accuracy of MPEG video scene cut detection.

Keywords Cut, Macroblock, Motion vector, Similarity