一种在MPEG-2系统流上快速搜索I图象的方法*

陈维强 高 文

(哈尔滨工业大学计算机科学与工程系 哈尔滨 150001)

摘要 该文提出了一种在MPEG-2系统流上采用粗跳步与细跳步快速搜索I图象的方法,与一般基于逐字节搜索的方法相比,速度有很大的提高.文中将该快速方法应用于一个MPEG-2演播器的功能算法设计中,实现了包括快进、快退和倒播等算法.该文的搜索方法是基于恒定比特速率码流提出的,并能同样适用于变比特速率码流.

关键词 MPEG-2标准,系统流,I图象,快进,快退,倒播.

中图法分类号 TP391

MPEG(moving picture expert group)即运动图象专家组.继MPEG-1标准[1]之后,MPEG专家组于1994年11月又提出了MPEG-2标准[2]的主体部分(系统、视频和音频).可以说,MPEG-2标准将成为或正在成为包括从宽带通信、广播电视到计算机多媒体与娱乐等应用领域的一项共性技术,其应用范围是非常广泛的.然而在应用过程中,例如在多媒体计算机上进行回放和编辑时,需要解决这样一个问题,即如何在MPEG-2系统流上搜索I图象,严格讲,应是搜索I图象的编码表示,或称存储单元(Access Unit).一般说来,可以采用逐字节搜索方法,即在码流上搜索图象头,并通过图象类型(Picture Type)信息判断是否已搜索到I图象,但是这种方法的搜索速度很慢.在此,本文提出一种采用粗跳步与细跳步的快速搜索方法.该方法的应用对象是恒定比特速率码流,并且适当使用也能用于变比特速率码流.

MPEG-2系统标准定义了节目流(Program Stream)和传输流(Transport Stream)两种系统流.节目流与MPEG-1的系统流[1]类似,但有扩充.它是由1个或多个基本流以PES(packet element stream)分组包的方式形成一个单一流,并且所有的这些基本流有一个共同的基本时基;传输流是由一个或多个节目合成的一个单一流,所有这些节目可以允许有不同的基本时基,并且包长度是固定的(188字节).节目流与传输流都是由系统层与压缩层两层组成.系统层包含有时钟信息等,压缩层包含的是基本流数据,例如,音频或视频数据.另外,节目流与传输流的系统层又分别可以分成两个子层,传输流系统层可分为传输流分组包层(Transport Stream Packet Layer)和PES分组包层(PES Packet Layer),节目流系统层可分为包层(Pack Layer)和PES分组包层(PES Packet Layer).音频和视频等基本流数据就装载于PES分组包中.

本文第1部分是I图象的快速搜索方法;第2部分是该方法的应用,并给出了快进、快退、倒播等算法;第3部分是性能实验;最后是结论.

1 I图象的快速搜索方法

在阐述快速搜索方法之前,首先作以下3个假设:①假设视频流的图象组GOP(group of picture)的长度固定;②假设MPEG系统流为恒定比特速率;③假设系统流包的摆放方式是按输入数据的速率均匀放置.显然,以上3个假设对一般的商用流来说都能成立.由于流的比特速率恒定,且GOP长度固定,因此可以认为,在系统流上两个相邻I图象之间的字节数基本相等,同时考虑到构成系统流的包的层次结构,那么,可以尽可能地跳过非图象数据和非I图象数据,以达到快速搜索I图象的目的.为了叙述方便,这里仅以MPEG-2节目流为例进行阐述,而对MPEG-2传输流和MPEG-1系统流来说,其方法类似.以下是在MPEG-2节目流上正向和反向快速搜索I图象的方法.

---

* 本文研究得到国家863高科技项目基金和国家"九五"科技攻关项目基金资助.作者陈维强,1968年生,在职博士生,讲师,主要研究领域为运动图象压缩技术,系统控制.高文,1956年生,教授,博士,博士导师,主要研究领域为计算机视觉,多媒体技术,智能人机接口技术,虚拟现实,人工智能.

本文通讯联系人:陈维强,哈尔滨 150001,哈尔滨工业大学计算机科学与工程系321信箱

本文1997-03-24收到原稿,1997-06-23收到修改稿

1.1 估　计

在节目流上相邻两个 I 图象之间的字节数记为 $ByteCount_{GOP}$，同时，这里将节目流的比特速率记为 $MuxRate$ (bits/s)，视频流的 GOP 长度记为 N，这些值可从流中获得或测得. 则 $ByteCount_{GOP}$ 可按下式求得.

$$ByteCount_{GOP}=\frac{MuxRate\times N/PictureRate}{8} \tag{1}$$

式中 $PictureRate$ 为图象帧速率(frames/s)，也可从流中获得.

1.2 粗　跳

粗跳的步长记为 $Step_A$，是由 $ByteCount_{GOP}$ 来决定的. $Step_A$ 可由式(2)计算.

$$Step_A=(ByteCount_{GOP}/2)\times SpeedUp \tag{2}$$

式中 $SpeedUp$ 是加速因子，可以根据不同的应用需要而设定不同的值. 假设当前 I 图象的第 1 个 PES 分组包在节目流文件中的位置为 $I\_File\_Pointer$. 如果是正向搜索下一个 I 图象，则应先将文件针挪到 $I\_File\_Pointer+Step_A$ 位置上，然后再搜索 I 的起头 PES 分组包；如果是反向搜索上一个 I 图象，则应先将文件指针挪到 $I\_File\_Pointer-Step_A$ 位置上，然后再反向搜索 I 的起头 PES 分组包. 为了清楚理解粗跳，可参见图 1.

图1　粗跳步示意图

1.3 细　跳

当做完粗跳时，还可以做细跳以进一步提高搜索 I 图象的速度. 首先是正向细跳方法：为了搜索下一个 I 图象，则应先找到 I 图象的头一个 PES 分组包. 在搜索过程中，当遇到如下情况时，可做细跳转：①搜索到包头字，则可适当作包头跳转，将头部的一些域信息跳过；②搜索到系统起头字，则可根据其长度作系统头跳转，将系统信息跳过；③搜索到音频等非视频 PES 分组包时，根据分组包长度直接跳过；④搜索到视频 PES 分组包，但非 I 图象起头 PES 分组包，则同样根据分组包长度跳过. 反向细跳的方法是，为了搜索到上一个 I 图象，则也应找到上一个 I 图象的头一个 PES 分组包. 搜索方法是这样的：将做完粗跳后的文件指针位置记录下来，放在某变量如 $File\_Pointer$ 中，然后将文件指针往前移 M(如 $M=2\,048$)个字节，并将 $File\_Pointer$ 减去 M，把当前指针后的 M 个字节读入缓冲区中；然后再在这 M 个字节中搜索 I 图象的头一个 PES 分组包，如果没查到，则 $File\_Pointer$ 再减去 M，并将文件指针省向该位置. 读入 M 个字节，继续在新读入的 M 个字节中搜索 I 图象的头一个 PES 分组包，如此往复，直至搜索到为止. 当在 M 个字节中搜索到视频 PES 分组包的前部分，但不能确定是不是 I 图象的头一个 PES 分组包时，则可将其后的数据读入，进一步确认. 显然，反向的细跳步规则与正向相同，只不过反向的细跳局限于 M 个字节，超出 M 个字节时，则不必往下搜索，接着读入前 M 个字节，继续搜索.

由此可见，该方法是通过粗跳步和细跳步来加速的，并可通过调整加速因子 $SpeedUp$ 来获得不同的加速档次需要. 当流为变比特速率码流时，可以通过在搜索过程中不断更新 $MuxRate$，以获得较准的 $ByteCount_{GOP}$，而不影响对 I 图象的快速准确搜索.

2 方法的应用

本节将上述快速搜索 I 图象方法应用在一个演播器的功能算法设计中，实现了快进、快退、倒播等算法. 对快进、快退，通过设定加速因子 $SpeedUp$，可获得不同的加速档次需要，例如 $SpeedUp=3.0$ 时，可跨过一个 I 图象；对倒播，显然不允许有跨过 I 图象的现象发生，但也可通过谨慎设置 $SpeedUp$ 来获得较高的搜索速度.

2.1 快进与快退算法

2.1.1 快进算法

(a) 采用细跳步正向搜索到 I 图象；
(b) 记录 I 图象的头一个 PES 分组包位置，并存入变量 $I\_File\_Pointer$ 中；
(c) 系统解码出此 I 图象的视频压缩数据流；
(d) 视频解码此 I 图象；
(e) 显示该 I 图象；
(f) 粗跳步：$I\_File\_Pointer+=Step_A$，将文件指针指向 $I\_File\_Pointer$ 处；
(g) 到文件尾吗？如果不是，跳转到步骤(a)；
(h) 返回.

2.1.2 快退算法

(a) 采用细跳步反向搜索I图象;
(b) 记录I图象的头一个PES分组包位置,并存入*I_File_Pointer*中;
(c) 系统解码出此I图象的视频压缩数据;
(d) 视频解码此I图象;
(e) 显示该I图象;
(f) 粗跳步:*I_File_Pointer*−=$Step_A$,将文件指针指向*I_File_Pointer*处;
(g) 文件到头? 如果不是,跳转到步骤(a);
(h) 返回.

2.2 倒播算法的设计

为了获得特殊的视觉效果,往往需要将视频图象序列反序播放出去,即倒播.要想实现视频图象序列的倒播,则必须首先反向搜索到最近的一个I图象,并1次解码1个GOP,然后再反序播放出去.如果解完一个GOP,然后再播放出去,这样做则势必导致播放过程的中间停顿.为了克服停顿,可以先预解一个GOP,然后在解下一个GOP的同时穿插播放上一个GOP的图象.这时,为了方便,可申请一个双图象组缓冲区.下面是算法实现的具体描述.

(a) 反向搜索I图象,记录I图象的头一个PES分组包位置,并存入*I_File_Pointer*中;
(b) 到文件头吗? 如果是,跳转到步骤(q);
(c) 此I帧号大于当前显示帧号吗? 如果是,将文件指针指向*I_File_Pointer*处,跳转到步骤(a);
(d) 系统解码;
(e) 视频解码一帧图象;
(f) 一个GOP解完了吗? 如果不是,跳转到步骤(d);
(g) 将解码图象次序排成倒播次序;
(h) 将文件指针指向*I_File_Pointer*处;
(i) 反向搜索I图象,记录I图象的头一个PES分组包位置,并存入*I_File_Pointer*中;
(j) 到文件头吗? 如果是,跳转到步骤(q);
(k) 反序显示上一个GOP中的一帧图象;
(l) 系统解码;
(m) 视频解码一帧图象;
(n) 一个GOP解码完成了吗? 如果没有完成,跳转到步骤(k);
(o) 将解码图象次序排成倒播次序;
(p) 退出吗? 如果不退出,跳转到步骤(h);
(q) 上一个GOP全显示完了吗? 如果没显示完,将剩余的显示出去;
(r) 返回.

在反向搜索I图象时,算法中可以只采用细跳步.当采用粗跳步时,其粗跳步长要适当设置,以免跨过I图象.

3 性能实验

对于精确的恒定比特速率码流,从理论上讲,当*SpeedUp*=2.0时,采用本文提出的方法可以将搜索I图象的时间开销降低到最小极限.然而,事实上,所谓的恒定比特速率码流也有一定的波动,同时,考虑到性能实验的合理性(充分保证不跨过I图象),这里加速因子*SpeedUp*只取一个较小值,即对以下所有实验*SpeedUp*都取1.0.为了检验本文所提出的快速搜索I图象方法与演播功能算法的性能,本文主要针对恒定比特速率码流设计了两组实验:①本文搜索方法与逐字节搜索方法的速度性能测试;②快进、快退和倒播的速度性能测试.所有这些实验都是在奔腾133,内存为32M的微机上进行.

测试用的MPEG-2节目流和传输流是采用由作者设计的MPEG-2系统复用器,将视频测试流football与一段音乐编码流合成的,其文件名分别记为football.ps与football.ts;MPEG-1系统流采用XingMPEG所带的test.mpg节目,这些节目的一些参数见表1.

表1 测试用码流的一些参数

测试节目	平均码流速率(bits/s) (*MuxRate*)	图象帧速率(frames/s) (*PictureRate*)	GOP长度 (*N*)	I图象总数
football.ps	6 333 751	25	12	8
football.ts	6 468 511	25	12	8
test.mpg	1 241 912	30	15	10

3.1 搜索方法的速度性能比较

表 2 是本文搜索方法与逐字节搜索方法的测试结果，表中给出的是在码流上每秒能正向/反向连续搜索 I 图象的个数(单位：I 图象/s)．测试结果表明：①本文方法的搜索速度是逐字节方法的 2～3 倍；②正向与反向的搜索速度比较接近，这是因为反向搜索增加的开销较小；③对 MPEG-2 传输流的搜索速度比节目流稍慢，这是因为传输流的包结构比节目流复杂一些．显然，实际应用时，在保证不跨过 I 图象的情况下，可将 *SpeedUp* 调大一点，如等于 1.8，则可获得更高的搜索速度．

表 2　本文搜索方法与逐字节搜索方法测试结果(I 图象/s)

测试节目	正向搜索		反向搜索	
	本文搜索方法	逐字节搜索方法	本文搜索方法	逐字节搜索方法
football.ps	26.31	11.22	22.18	10.40
football.ts	24.42	10.23	21.34	9.60
test.mpg	125.00	47.60	100.00	41.67

3.2 快进、快退与倒播的速度性能

表 3 是本文快进、快退与倒播功能算法的测试结果．对快进、快退，表中给出的是在码流上每秒能连续搜索、解码与显示 I 图象的个数(单位：I 图象/s)；对倒播，表中给出的是在码流上每秒能连续倒播 GOP 的个数(单位：GOP/s)．测试结果表明：①结合表 2 可以看出，搜索 I 图象在快进、快退与倒播中占有的时间开销比例较小，经计算得出，在快进和快退中占 1/3～1/4，在倒播中占 1/50～1/60；②结合表 1 可以求出，对 football.ps 和 football.ts，其快进、快退速度大约只是正常播放速度的 3 倍，对 test.mpg 则接近 15 倍；③结合表 1 可知，对 test.mpg 的倒播接近达到实时(实时播放速度是 2 个 GOP/s)，而对 MPEG-2 无论是节目流，还是传输流都没有达到实时，并有相当一段距离．显然，对 MPEG-1 系统流来讲，采用本文提出的方法，快进、快退与倒播能达到实用要求，而对 MPEG-2 系统流进行快进、快退或倒播速度较慢的原因并不在于搜索，而是解码与显示，或者说还在于实验用的机器．这是作者正在研究的另外一个关键问题，将在今后另文报道．实际应用时，通过调整 *SpeedUp*，使粗跳步能跨过 1 或 2 个 I 图象，如此可获得更高的快进、快退速度．

表 3　快进、快退与倒播的测试结果

测试节目	快进(I 图象/s)	快退(I 图象/s)	倒播(GOP/s)
football.ps	6.33	5.99	0.41
football.ts	5.89	5.78	0.39
test.mpg	30.30	28.57	1.73

需要说明的是，表 2 与表 3 的数据都是在做多次循环实验的基础上求平均值得到的．此外，为了说明本文所提出的快速搜索 I 图象方法对变比特速率(VBR(variable bit rate))码流也有效，作者人为生成了一个变比特速率的 MPEG-2 节目流．其做法是：将 96 帧 football 图象序列分成 8 段，每段 12 帧(1 个 GOP)，分别由 MPEG-2 视频编码器以不同的比特速率编码；然后将其拼接在一起，由系统复用器将该视频编码流与一段音乐编码流合成为一个变比特速率 MPEG-2 节目流，文件名记为 football.vbr．前面已经提到，对变速率码流，本文方法需要在搜索过程中不断更新 *MuxRate*，以自适应调整粗跳步长 $Step_A$．作如此改动后的搜索方法在 football.vbr 上的实验表明，本方法(*SpeedUp*=1.0)不仅能正确搜索到 8 幅 I 图象，而且能获得很好的搜索速度，大约是逐字节方法搜索速度的 2.2 倍．

4 结　论

本文提出了一种在 MPEG-2 系统流上快速搜索 I 图象的方法，即采用粗跳和细跳步来提高搜索速度，该方法从理论上讲可以将搜索 I 图象的时间开销降低到最小极限．同时，文中应用该快速方法实现了包括快进、快退和倒播在内的演播器功能算法，并通过实验得出如下几个主要结论(加速因子 *SpeedUp*=1.0)：(1) 本方法的搜索速度是逐字节方法的 2～3 倍；(2) 本文提出的快进、快退与倒播算法对 MPEG-1 系统流的执行速度能满足实用需要，对 MPEG-2 系统流的执行速度较慢，其主要原因在图象的解码和显示上，而不是在搜索 I 图象上；(3) 本文的搜索方法同样适用于变比特速率码流，并能获得较好的搜索速度，是逐字节方法的 2.2 倍．目前，这种快速搜索 I 图象方法以及上述快进、快退和倒播等算法已被应用于由作者设计的 MPEG-2 演播器中，并取得了很好的效果．显然，本文所提出的方法不仅可应用在 MPEG 码流的回放上，而且还可广泛应用于码流的检索和编辑等方面．

参考文献

1 International Standard ISO/IEC 13818—— Information Technology-Generic Coding of Moving Pictures and Associated Audio. ISO/IEC JTC1/SC29/WG11, 1994

2 杨品,钟玉琢等. MPEG-1 运动图象压缩编码标准(ISO/IEC 11172). 北京:机械工业出版社,1995
(Yang Pin, Zhong Yu-zhuo *et al*. Standard of MPEG-1 Moving Picture Compress and Coding (ISO/IEC 11172). Beijing: Machine Industry Press, 1995)

A Method of Fast Searching I Pictures in MPEG-2 System Stream

CHEN Wei-qiang GAO Wen

(*Department of Computer Science and Engineering Harbin Institute of Technology Harbin* 150001)

Abstract In this paper, a method of fast searching I pictures in MPEG-2 system stream using long step and short step is presented. The method achieves higher speed performance comparing with the byte by byte searching method, and is applied in the design of the functions of MPEG-2 players, such as fast forward, fast backward and play backward. The implementation of these functions is described in detail. Moreover, this method can also be applied in VBR (variable bit rate) MPEG system stream, which is proposed based on CBR (constant bit rate) stream.

Key words MPEG-2 standard, system stream, I picture, fast forward, fast backward, play backward.