

一种多视点视频系统中随机访问的 解码路径计算新方法

贺人龙¹⁾ 蒋刚毅^{1,2)} 郁梅^{1,2)} 符冉迪¹⁾

¹⁾(宁波大学信息科学与工程学院,宁波 315211) ²⁾(南京大学软件新技术国家重点实验室,南京 210093)

摘要 在多视点视频系统中,用户端的随机访问性能是表征交互能力的一个重要指标。当用户进行随机访问时,需要快速确定相应的解码路径,以快速有效地获得相应视点与时刻的图像帧。针对 H. 264 多视点视频系统中随机访问效率不高这个情况,分析了传统多视点视频系统随机访问解码路径的计算方法,讨论了其中存在的非最优解码路径问题,提出帧等级矩阵的概念。然后,提出针对 HBP(hierarchical B pictures) 这种具有代表性的多视点视频编码预测结构的解码路径计算快速算法。实验结果表明,所提算法能获得较优解码路径,从而有效地提高随机访问性能,降低解码复杂度。

关键词 多视点视频系统 随机访问 解码路径 帧等级矩阵

中图法分类号:TP391.41 文献标识码:A 文章编号:1006-8961(2009)04-0636-06

A New Method for Decoding Path Computation of Random Access in Multi-view Video System

HE Ren-long¹⁾, JIANG Gang-yi^{1,2)}, YU Mei^{1,2)}, FU Ran-di¹⁾

¹⁾(Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211)

²⁾(National Key Laboratory of Software New Technology, Nanjing University, Nanjing 210093)

Abstract In the multi-view video system, random access performance of client is an important performance of interactivity. When users access one frame of multi-view video sequence randomly, it is necessary for the multi-view video system to know how to find and decode the corresponding frame quickly and efficiently. To improve random access performance of multi-view video system, this paper first analyzes previous decoding path calculation methods of random access, and discusses the problem of non-optimal decoding path. A concept of frame level matrix is presented for decoding path computation, and then a fast decoding path computation algorithm is proposed for the H. 264-based multi-view video coding with the prediction structure of HBP(hierarchical B pictures). Experimental results show that the proposed fast decoding path computation algorithm can obtain the optimal decoding path, so it effectively reduces the decoder complexity and improves random access performance in the multi-view video system.

Keywords Multi-view video system, random access, decoding path, frame level matrix

基金项目:国家自然科学基金项目(60872094,60832003);教育部新世纪优秀人才计划项目(NET-06-0537);国家高技术发展计划项目(2009AA01Z327);宁波市自然科学基金项目(2007A610037)

收稿日期:2007-05-11;改回日期:2007-11-13

第一作者简介:贺人龙(1984~),男。宁波大学信息科学与工程学院通信与信息专业硕士研究生。主要研究方向为多媒体信号处理与通信、多视点视频编码。E-mail:Long19851105@126.com

通讯作者:蒋刚毅,E-mail:jianggangyi@126.com

1 引言

多视点视频序列是由相机阵列拍摄得到的一组视频信号,它能提供拍摄场景不同角度的视频信息。人们利用其中一个或多个视点信息可以合成任意视点的信息,可实现用户交互视点选择与显示的目的^[1-4]。多视点视频系统一般由多视点视频采集、多视点视频编码、网络传输、解码和显示等部分组成,如图 1 所示。随着多视点相机数目的增加,多视点视频的数据量将成倍增加,若将所有码流都传输至用户端,对网络带宽将带来极大挑战。因此,需要研究在多视点系统中发生随机访问时,如何快速有效地确定用户所需码流,即解码路径。从而降低随机切换的响应时间以及用户端的复杂度。

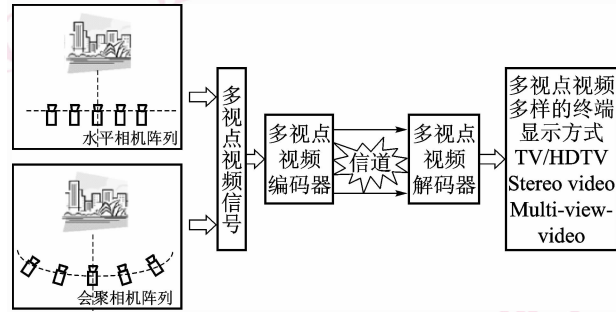


图 1 多视点视频系统
Fig. 1 Multi-view video system

由 HHI (heinch hertz institute) 提出的基于分层 B 帧 (HBP) 的多视点编码 MVC 结构^[5],延续了单视点分层 B 帧的编码优势,已被 ITU-T 的 VCEG 和 ISO/IEC 的 MPEG 的联合视频组 (JVT) 纳入标准草案。HBP 与其他多视点预测编码方案相比,提高了编码效率,但其随机访问性能有待进一步研究。在图 2 的 HBP 预测编码结构^[4]中,不同视点的帧被交叉排列,构成一个单通道序列。其中,水平方向代表时间,垂直方向代表视点。考虑采用传统的 AVC (advanced video coding) 句法,该结构解码所需的帧很大程度上取决于视点的个数和每个视点的 GOP (group of picture) 长度,其解码代价为 $K(G + 1) - 1$, K 代表视点的数目, G 代表 GOP 的长度。对于没有视差估计的情况,随机访问的解码代价仅仅为 G 。在随机切换视点时需要达到实时性,多视点解码器需要比传统的单视点解码器性能好上 K 倍。因此,为了使多视点情况下的用户随机切换及相关过程能

够有效进行, JVT 建议增加 V 帧类型图像^[6-8],同时还对多视点参考帧管理策略做了一定的修改^[9]。但是,多视点随机访问解码路径仍未达到最优,需进一步优化。本文就当前多视点系统中随机切换存在的问题进行研究,对随机切换的解码路径计算提出了一种快速算法,以解决多视点系统中随机切换过慢的问题。

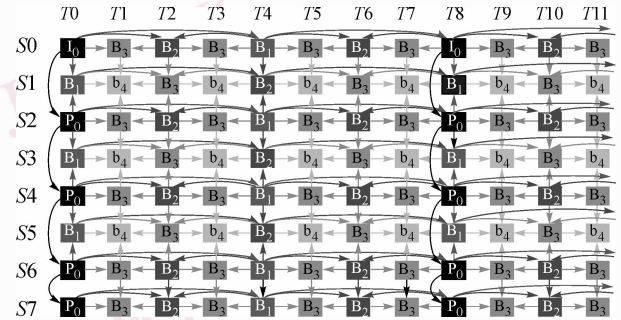


图 2 HBP 预测结构
Fig. 2 HBP prediction structure

2 基于 HBP 的解码路径快速算法

2.1 JVT 对多视点解码与随机访问的相关建议

2.1.1 V 帧定义及其相关分析

为了对视频序列中的任意点进行随机访问,传统解决方案是将 I 帧以一定的周期地放置在序列中^[5]。图 3 说明了多视点视频使用周期 I 帧的相关结构。与图 4 所示结构相比,图 3 方案保证了任意点的随机访问,但同时损失了编码效率。为了逼近图 4 所示结构的编码效率,以及图 3 所示结构的随机访问性能, JVT 建议增加 V 帧类型图像,如图 5 所示^[5], V 帧只与处在同一时刻的其他视点的图像进行视差估计。通过周期地放置 V 帧,能够获得与 I 帧接近的时间随机访问性能,同时较 I 帧又有更高的编码效率。图 3、图 4 及图 5 中的 Access Frame 代表当前需要访问的图像。

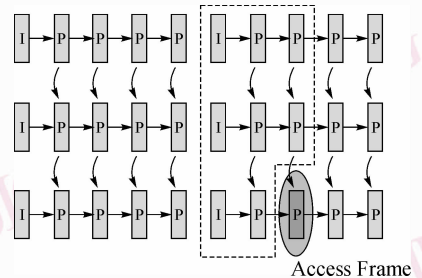


图 3 使用 I 帧的多视点编码结构^[5]
Fig. 3 Multi-view structure using I frame

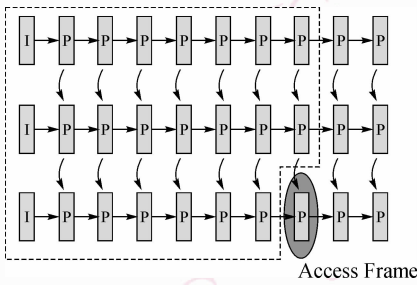


图 4 不使用 I 帧的多视点编码结构^[5]

Fig. 4 Multi-view structure not using I frame

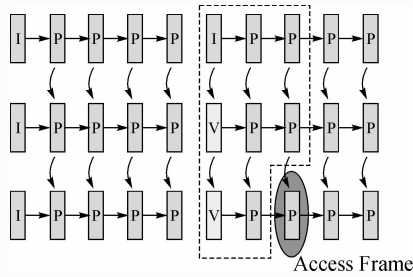


图 5 采用 V 帧的多视点编码结构^[5]

Fig. 5 Multi-view structure using V frame

2.1.2 参考图像管理

在先前的多视点实现中,参考图像管理包括解码顺序标识 (`frame_num`)、播放顺序标识 POC 的编码,是遵循 H. 264/AVC 标准的。其中,POC 被用做传输解码图像输出次序信息,`frame_num` 被用做解码图像次序信息。文献[8]分析了传统的编码方案,指出了其中的不足之处,并做了相应修改,编码方案分别如图 6 所示。图 6 为文献[8]中改进的编码方案,在每一帧的头部增加了一个视点识别信息 (`view_id`)。图 6 中,第 1 个代表 POC,第 2 个代表 `frame_num`,第 3 个代表 `view_id`。如位于图 6 中 ($S1, T0$) 位置的帧,由图可以得出该帧的 POC 编码为 000,`frame_num` 编码为 000,`view_id` 编码为 01。

2.2 解码路径快速算法及其分析

基于图 6 建议的 `frame_num`、POC、`view_id` 编码方案,对解码端随机访问解码路径的计算提出解决方案。为了访问视点 S 时刻 T 对应的帧,解码器需要找到距离随机访问点最近的恢复点,然后从此处一直解码至随机访问点。如果使用了建议句法,本文算法首先会计算解码 (S, T) 帧所需要的帧,然后

	T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12
S0	000/ 000/ 00	001/ 005/ 00	002/ 006/ 00	003/ 003/ 00	004/ 007/ 00	005/ 008/ 00	006/ 002/ 00	007/ 009/ 00	008/ 010/ 00	009/ 004/ 00	010/ 011/ 00	011/ 012/ 00	012/ 001/ 00
S1	000/ 000/ 01	001/ 005/ 01	002/ 006/ 01	003/ 003/ 01	004/ 007/ 01	005/ 008/ 01	006/ 002/ 01	007/ 009/ 01	008/ 010/ 01	009/ 004/ 01	010/ 011/ 01	011/ 012/ 01	012/ 001/ 01
S2	000/ 000/ 02	001/ 005/ 02	002/ 006/ 02	003/ 003/ 02	004/ 007/ 02	005/ 008/ 02	006/ 002/ 02	007/ 009/ 02	008/ 010/ 02	009/ 004/ 02	010/ 011/ 02	011/ 012/ 02	012/ 001/ 00
S3	000/ 000/ 03	001/ 005/ 03	002/ 006/ 03	003/ 003/ 03	004/ 007/ 03	005/ 008/ 03	006/ 002/ 03	007/ 009/ 03	008/ 010/ 03	009/ 004/ 03	010/ 011/ 03	011/ 012/ 03	012/ 001/ 03
S4	000/ 000/ 04	001/ 005/ 04	002/ 006/ 04	003/ 003/ 04	004/ 007/ 04	005/ 008/ 04	006/ 002/ 04	007/ 009/ 04	008/ 010/ 04	009/ 004/ 04	010/ 011/ 04	011/ 012/ 04	012/ 001/ 04
S5	000/ 000/ 05	001/ 005/ 05	002/ 006/ 05	003/ 003/ 05	004/ 007/ 05	005/ 008/ 05	006/ 002/ 05	007/ 009/ 05	008/ 010/ 05	009/ 004/ 05	010/ 011/ 05	011/ 012/ 05	012/ 001/ 05
S6	000/ 000/ 06	001/ 005/ 06	002/ 006/ 06	003/ 003/ 06	004/ 007/ 06	005/ 008/ 06	006/ 002/ 06	007/ 009/ 06	008/ 010/ 06	009/ 004/ 06	010/ 011/ 06	011/ 012/ 06	012/ 001/ 06
S7	000/ 000/ 07	001/ 005/ 07	002/ 006/ 07	003/ 003/ 07	004/ 007/ 07	005/ 008/ 07	006/ 002/ 07	007/ 009/ 07	008/ 010/ 07	009/ 004/ 07	010/ 011/ 07	011/ 012/ 07	012/ 001/ 07

图 6 文献[8]的 `frame_num`/`POC`/`view_id` 编码方案

Fig. 6 POC/frame_num coding method for multi-view structure in reference[8]

从恢复点开始延着一条合适的路径进行解码。提出的标记算法如下所示,假设当前用户随机访问 (S, T) 帧。

(1) 如果当前图像已经标记过,则终止本次递归过程;否则,继续下面的步骤。

(2) 标记第 T 时刻、第 S 个视点的图像。

(3) 找出时刻在上述图像之前的 IDR (instantaneous decode refresh) 类型或者 V 类型的图像,设置时刻 T' 为此图像的时刻。

(4) 找出时刻在上述图像之后的 IDR 类型或者 V 类型的图像,设置时刻 T'' 为此图像的时刻。

(5) 在 (S, T') 图像所对应的 VPS (view parameter set) 参数信息中,对所有视点起始位置处对应的参考视点 s ,递归执行上述过程。

(6) 在 (S, T'') 图像所对应的参数信息中,对所有视点内部所对应的参考视点,递归执行上述过程。

(7) 定义对应编码结构的帧等级矩阵 $A = [a_{i,j}]$,其中 $a_{i,j}$ 对应编码结构中帧的等级信息,针对 HBP 编码结构,帧等级矩阵如下所示:

$$A = [0, 3, 2, 3, 1, 3, 2, 3] \quad (1)$$

矩阵中的数字代表一个 GOP 中对应位置帧的等级,其中 0 代表非 B 帧,1~4 分别代表 B 帧等级 1 到 4,利用图 6 的编码方案中的 POC 信息,查询得出目标图像的等级信息。预测结构不同时,可以调整帧等级矩阵来适应结构变化。

(8) 在 (S, T') 图像所对应的参数信息中,对所

有视点内部所对应的参考视点 s ,对在区间 $[T, T'' - 1]$ 上的 t ,依次递归调用上述过程。但有如下限制条件:当前图像的等级信息不小于目标图像时,中止当前递归,继续上层递归过程;如果当前图像的等级信息查询等于 1,跳至第 8 步最外层的视点循环中。

(9) 对每一个在 T' 和 T'' 之间的 t ,依次递归调用上述过程。但有如下限制条件:当前图像的等级信息不小于目标图像时,中止当前递归,继续上层递归过程;如果当前图像的等级信息查询等于 1,跳出第 9 步。

(10) 标记 (S, T') 和 (S, T'') ,如果之前没有标记过,则标记。

所提算法流程如图 7 所示,第 1 步是一个在大多数递归过程中都会用到的中止递归条件。第 2、3、4 步计算出当前递归循环中需要标记的帧。第 5、6 步对正确解码第 4 步指定帧所需要的图像进行了标记,上述步骤说明 V 帧在打破序列时间相关性的同时保存了视点间的依赖性。第 7 步针对 HBP 编码结构定义了一个帧等级矩阵,将在第 8、9 步中用到。第 8 步标记出所有其他视点上可能预测当前视点帧。最后,第 9、10 步标记出当前视点中可能需要进行时间预测的所有 I、P、B 帧。与传统多视点视频随机访问过程相比,本文算法主要的不同之处在于能够决定哪些帧需要被解码传输以及哪些帧不需考虑,有效节省了网络带宽。

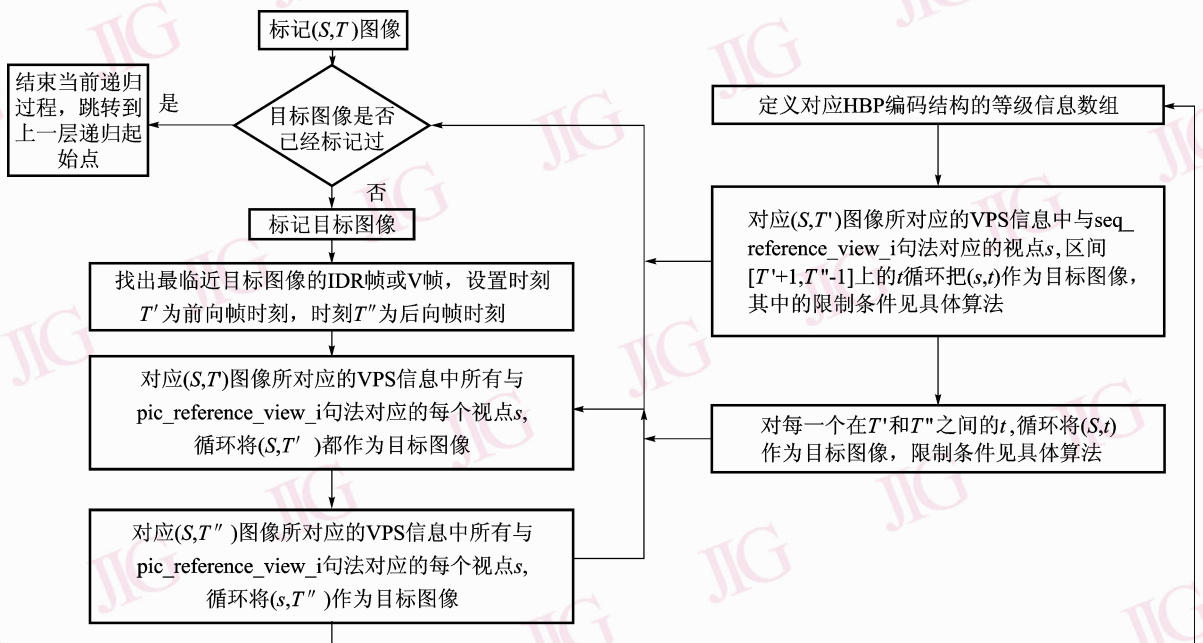


图 7 随机访问解码路径计算算法流程

Fig. 7 Process for computing random access decoding path

3 实验结果与分析

为了验证本文算法的有效性,将本文方案和 Martinian 等人的方案^[6]、传统的解决方案在所需解码帧数、随机访问平均解码代价、响应速度提升率 3 个方面进行了实验分析、比较。同时,为了验证本文算法的容错能力,还对解码路径计算过程进行误差代价分析。

实验采用 HBP 多视点压缩编码方案,对 8×8 的 GOP 结构进行了实验分析,本文方案与使用的多视点测试序列具体内容无关。实验结果如下表 1、表 2、表 3 所示,假设用户随机访问(7,1)位置对应的帧,即第 7 视点、第 1 时刻的图像。表中,横向代表 GOP 中的时间方向,纵向代表 GOP 中的视点方向,粗体所标位置为用户随机访问点。

3.1 解码帧数比较

表 1 对应传统随机访问解码路径计算解决方案的结果,其中数字 1 至 69 对应算法中对所需传输帧标记的顺序,表示用户端当前的随机切换操作需要服务端传输 69 帧。

表 1 传统解决方案中随机切换需要传输的帧

Tab. 1 Traditional method's result for random access

V/T	0	1	2	3	4	5	6	7	8	9	10	11
0	1	13	11	14	10	15	12	16	9	0	0	0
1	3	29	27	30	26	31	28	32	25	0	0	0
2	2	21	19	22	18	23	20	24	17	0	0	0
3	5	45	43	46	42	47	44	48	41	0	0	0
4	4	37	35	38	34	39	36	40	33	0	0	0
5	7	61	59	62	58	63	60	64	57	0	0	0
6	6	53	51	54	50	55	52	56	49	0	0	0
7	8	69	67	70	66	71	68	72	65	0	0	0

表 2 Martinian 方案中随机切换需要传输的帧

Tab. 2 Martinian method's result for random access

V/T	0	1	2	3	4	5	6	7	8	9	10
0	5	0	0	0	0	0	0	0	9	0	0
1	0	0	0	0	0	0	0	0	0	0	0
2	4	0	0	0	0	0	0	0	8	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	3	0	0	0	0	0	0	0	7	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	2	10	11	12	13	14	15	16	6	0	0
7	17	1	18	0	19	0	20	0	21	0	0

表 3 本文方案中随机切换需要传输的帧

Tab. 3 Proposed method's result for random access

V/T	0	1	2	3	4	5	6	7	8	9	10
0	5	0	0	0	0	0	0	0	9	0	0
1	0	0	0	0	0	0	0	0	0	0	0
2	4	0	0	0	0	0	0	0	8	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	3	0	0	0	0	0	0	0	7	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	2	10	11	0	12	0	0	0	6	0	0
7	13	1	14	0	15	0	0	0	16	0	0

表 2 对应 Martinian 等人的方案,POC、frame_num 编码方法与传统方案相同,其中数字 1 至 21 意义同表 1,表示用户端当前的随机切换操作需要服务端传输 21 帧。表 3 对应本文所提算法,采用图 6 的 frame_num、POC、view_id 编码方案,其中数字 1 至 16 为本文解码路径算法计算出的结果,表示用户端当前的随机切换操作仅仅需要服务端传输 16 帧。从上面的结果可以看出,当用户随机访问(7,1)位置对应的图像时,传统的解决方案为了保证用户端正确解码,需要传输表 1 中标记的 69 帧;相比而言, Martinian 等人的方案需要 21 帧;而本文方案有较明显优势,仅需要传输 16 帧。所需传输帧数的减少可以有效降低对网络带宽的需求,从而有效提升视频传输的稳健性,保证视频的 QOS (quality of service) 等特性。

3.2 随机访问平均解码代价比较分析

定义随机访问平均解码代价 W 为 GOP 中任意访问一帧需要预先传输的平均帧数。由于一个 GOP 时间很短,假设用户在一个 GOP 内只做一次随机切换。并且,用于编码的 n 个视点、每个视点 m 帧的多视点视频帧总数 $s = n \times m$ 是有限的。令 X_i 表示随机切换至第 i 帧需要预先传输的帧数, P_i 为用户随机的访问第 i 帧的概率,则随机访问平均解码代价为

$$W = \sum_{i=1}^{n \times m} X_i P_i \quad (2)$$

随机访问平均解码代价在一定程度上反映了对随机访问的支持程度,这个代价越高,说明系统对随机访问的支持能力越差,为支持随机访问而消耗的资源就越多。依据式(2),得出传统方案下,8 视点 HBP 编码结构的随机访问平均解码代价为 9.31, Martinian 等人方案^[6]的随机访问平均解码代价为 3.44,本文方案的随机访问平均解码代价为 2.76,

如表4所示。显然,本文方案好于 Martinian 方案,明显好于传统方案。

表4 3种方案的平均解码代价对比

Tab.4 Average decoding cost for three methods

	传统方案	Martinian 方案	本文方案
平均解码代价	9.31	3.44	2.76

3.3 响应速度提升率 K

定义随机切换时间 T 为用户发出随机切换请求至获得所需码流的总时间,响应速度提升率 K 为两种算法的随机切换时间差的比值,用以描述两种算法随机切换速度提升的情况,如下所示:

$$K = (W_1 - W_2) \times 100 / W_1 \quad (3)$$

响应速度提升率 K 在一定程度上反映一种切换算法的随机访问响应时间,值越小说明系统的随机切换速度越快,随机访问性能越好。依据式(3),针对8视点 HBP 编码结构,得出相对于传统随机切换解决方案, Martinian 等人方案的响应速度提升率为 63.05%,本文方案为 70.35%,如表5所示。从表5可以看出,在实验条件下,本文方案的随机切换速度优于 Martinian 等人方案。

表5 响应速度提升率 K

Tab.5 Improved response speed K

	Martinian 等人方案	本文方案
随机切换速度提升率 K	63.05%	70.35%

3.4 解码路径误差代价分析

由于本文算法思想是在一个最大帧集合中去除不需要的部分,得到用户需要的最少帧集合。因此,解码路径算法执行时,某步计算出现误差只会导致得到的不是最少帧集合,即增加了对网络带宽的需求。针对 HBP 编码结构,访问(7,1)位置时,最大帧集合为表1所示的1至69帧,经过本文算法优化,得到用户需要的最小帧集合为表3所示的1至16帧。假设计算过程出现误差,那么用户需求的解码帧数将在区间[16~69]变动;随机访问平均解码代价将在区间[2.76~9.31]之间变动;相对于传统解决方案,本文算法响应速度提升率将在区间[0~70.35%]之间变动。

4 结论

针对多视点系统中随机访问帧的解码路径不易

计算的情况,通过借鉴一些当前研究的最新成果,提出了一种在多视点情况下的计算随机访问解码路径的快速算法,节省了编码端所需传输的码流,提高了多视点解码器的解码速度以及降低了解码复杂度。本文方案优点在于:(1)相比传统方案,由于使用了V帧类型图像,可以减少冗余帧的传输。(2)相比传统方案以及 Martinian 等人方案,进一步减少了冗余帧的传输。(3)定义了帧等级矩阵,可用于减少冗余帧传输。同时,当 HBP 的视点个数、GOP 长度变化或者采用不同的结构时,通过调整矩阵信息,本文算法仍然有效,有一定的适应性。

网络环境复杂多变,因此实际使用中,编码端需根据环境自适应调整 GOP 结构,例如时而采用 HBP 结构,时而采用 M-Picture 结构,新算法针对 HBP 这种编码结构做了相应的优化工作。进一步将开展以下两方面工作:(1)采用多模式思想的架构,针对不同的典型编码结构,研究出相应快速算法,然后实现算法与 GOP 编码结构自适应同步。(2)随机访问解码路径计算的工作主要集中在了编码端,在某些情况下,这是不合适的,作者计划开发针对解码端的路径计算方案,以匹配不同的实际情况。

参考文献 (References)

- 1 ISO/IEC JTC1/SC29/WG11 N6909, Survey of Algorithms Used for MVC[S]. Hong Kong, 2005.
- 2 Kalva H, Christodoulou L, Mayron L, et al. Challenges and opportunities in video coding for 3D TV [A]. In: Proceedings of IEEE International Conference on Multimedia and Expo [C], Toronto, Canada, 2006:1689-1692.
- 3 Kimata H, Kitahara M, Kamikura K, et al. Low-delay multiview video coding for free-viewpoint video communication [J]. Systems and Computers in Japan, 2007, 38(5):14-28.
- 4 Zhang Y, Jiang G Y, Yu M, et al. An approach to multi-modal multiview video coding [A]. In: Proceedings of International Conference on Signal Processing [C], Guilin, China, 2006:1405-1408.
- 5 Mueller K, Merkle P, Schwarz H, et al. Multi-view video coding based on H.264/MPEG4-AVC using hierarchical B pictures [A]. In: Proceedings of Picture Coding Symposium [C], Beijing, China, 2006.
- 6 ISO/IEC JTC1/SC29/WG11, M12828: Multiview Video Compression Using V Frames[S]. Bangkok, Thailand, 2006-01.
- 7 ISO/IEC JTC1/SC29/WG11, M13121, V-Picture Syntax for Random Access in Multi-view Video Compression[S]. Montreux, Switzerland, 2006-04.
- 8 ISO/IEC JTC1/SC29/WG11, M13715, MVC High-level Syntax for Random Access[S]. Klagenfurt, Austria, 2006-07.
- 9 ISO/IEC JTC1/SC29/WG11, M13718, On MVC High-Level Syntax for Picture Management [S]. Klagenfurt, Austria, 2006-07.