

基于运动一致性的 3D-HEVC 深度视频帧间编码快速算法

刘 晟, 陈 芬*, 金德富

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘要: 针对深度视频在 3D-HEVC 编码框架中编码复杂度较高, 提出了一种基于运动一致性的深度视频快速帧间编码算法. 首先把深度视频划分为 4 个区域: 彩色和深度视频均运动区域、彩色视频运动但深度视频静止区域、彩色视频静止但深度视频运动区域以及彩色和深度视频均静止区域. 然后, 根据不同分区对最大编码单元设计不同的编码策略. 结果表明, 本算法与原始测试平台 HTM-10.0 相比, 在绘制的虚拟视点质量几乎一致的情况下, 各序列深度视频编码时间平均节省了 63.58%, 总体编码时间节省了 40.13%.

关键词: 深度视频; 3D-HEVC; 帧间编码; 运动相似性; 最大分割深度; 模式决策

中图分类号: TN919

文献标志码: A

文章编号: 1001-5132 (2017) 03-0030-06

多视点彩色加深度视频是主流的三维场景表示方式, 其中深度视频表示场景的几何信息, 被用来实现虚拟视点的绘制^[1-2]. 多视点彩色加深度视频可以采用传统的编码器进行编码, 如高效视频编码(High Efficiency Video Coding, HEVC)^[3], 或采用先进的基于 HEVC 的 3D 扩展 3D-HEVC 编码^[4-6]. 在编码过程中, 一幅深度图像会被 3D-HEVC 的参考软件(Test Model of 3D-HEVC, HTM)分割成多个最大编码单元(Largest Coding Unit, LCU)^[6-7], 每个 LCU 可以被继续分割到 3 个不同的深度级 d , 每个深度级有 4^d 个编码单元(Coding Unit, CU), 且每个 CU 需要遍历 Skip、Merge、Inter-2N×2N、Inter-Others 和 Intra 预测模式, 其中 Inter-Others 包括 Inter-2N×N、Inter-N×2N、Inter-N×N、Inter-2N×nU、Inter-2N×nD、Inter-nL×2N 和 Inter-nR×2N, Intra 包括 Intra-2N×2N 和 Intra-N×N, 然后从中选取一个作为最优预测模式. 在编码过程中, 深度视频 LCU 最优分割深度的确定和 CU 最优预测模式的选择均十分复杂. 因此, 在保证虚拟视点质量不下降的条件下, 降低深度视频编码的复杂度具有重要意义.

目前已有很多学者从统计实验^[8-10]或者深度视频与彩色视频编码信息相关性^[11-13]的角度提出

了降低深度视频编码复杂度的快速算法. Park^[8]对 CU 进行变换得到哈达玛变换系数, 判定 CU 是否存在边缘. 如果 CU 不存在边缘或只存在水平边缘和垂直边缘, 则跳过深度模型模式搜索, 从而减少了帧内模式的复杂度. Shen 等^[9]利用空域相邻 CU、时域相关 CU 和相邻视点对应 CU 最优分割尺寸和预测模式信息, 减少当前 CU 的预测模式数目和遍历的分割深度层数, 从而降低帧间编码的复杂度. Chung 等^[10]利用深度无失真模型, 加速 CU 的模式预测和递归分割过程, 从而降低帧间编码的复杂度. Lei 等^[12]利用彩色视频和深度视频运动和结构相似, 提出了快速 CU 模式决定算法, 降低了深度视频帧间编码复杂度. Mora 等^[11]依据深度视频 LCU 最优分割深度小于对应彩色视频最优分割深度, 提出了一种快速 CU 尺寸预测算法, 并加入到 HTM 中. Pan 等^[13]利用彩色视频 CU 预测模式信息, 加速深度视频 CU 的模式决策, 最终降低了深度视频的编码复杂度.

虽然上述算法在一定程度上降低了深度视频编码复杂度, 但是没有充分利用彩色视频 CU 的编码信息. 此外, 深度视频与对应的彩色视频是同一场景信息的不同表达, 它们具有很高的结构与运

收稿日期: 2016-08-11.

宁波大学学报(理工版)网址: <http://journalg.nbu.edu.cn/>

基金项目: 浙江省自然科学基金(LY16F010002); 宁波市自然科学基金(2015A610127, 2015A610124); 浙江省重中之重学科项目(XKXL1502).

第一作者: 刘晟(1991-), 男, 安徽来安人, 在读硕士研究生, 主要研究方向: 视频图像处理. E-mail: ncxliusheng@126.com

*通信作者: 陈芬(1973-), 女, 四川邻水人, 副教授, 主要研究方向: 视频图像处理. E-mail: chenfen@nbu.edu.cn

动相似性. 在深度视频帧间编码配置条件下, 深度视频静止区域LCU的最优分割深度为0, 且最优预测模式为Skip模式的概率很高; 而深度视频运动区域LCU的最优分割深度为3, 最优预测模式为非Skip模式的概率很高. 当深度视频运动区域所对应的彩色视频区域为静止区域时, 这些区域在一定程度内的失真并不会引起虚拟视点质量的下降. 结合深度视频编码的上述统计规律, 本文提出一种基于运动一致性的深度视频帧间编码快速算法. 该算法把深度视频分成4个不同的区域: 彩色和深度视频均运动区域(Texture Move and Depth Move Region, TMDM)、彩色视频运动但深度视频静止区域(Texture Move and Depth Static Region, TMDS)、彩色视频静止但深度视频运动区域(Texture Static and Depth Move Region, TSDM)以及彩色和深度视频都静止区域(Texture Static and Depth Static Region, TSDS). 如当前LCU属于TMDS和TSDS区域, LCU只需遍历Skip模式, 且最大分割深度强制为0. 如当前LCU属于TSDM区域, LCU的最大分割深度强制为0, 且在Skip模式下的编码残差为0, 则停止模式搜索过程. 如当前LCU属于TMDM区域, LCU的最大分割深度设置为3, 遍历所有的候选预测模式.

1 深度视频区域划分

首先把彩色视频分为运动区域(Texture Move Region, TM)和静止区域(Texture Static Region, TS), 然后把深度视频分为运动区域(Depth Move Region, DM)和静止区域(Depth Static Region, DS), 最后融合彩色视频和深度视频运动信息, 把深度视频分为TMDM、TMDS、TSDM和TSDS四个区域.

1.1 彩色视频运动区域检测

为了准确提取彩色视频TM区域, 并减少提取算法的复杂度, 我们利用彩色视频CU的运动矢量信息, 把彩色视频分为TM和TS区域, 相应规则如下:

$$S_{mv} = \sum_{i=0, j=0}^{i=15, j=15} (|H_{i,j}| + |V_{i,j}|), \quad (1)$$

其中: i, j 表示当前LCU中第 i 行第 j 列的 4×4 块; H 表示 4×4 块的水平方向运动矢量; V 表示 4×4 块的垂直方向运动矢量. S_{mv} 表示LCU中所有 4×4 块的

水平和垂直方向运动矢量绝对值的和. 当 S_{mv} 为0时, LCU属于TS区域, 否则属于TM区域.

1.2 深度视频运动区域检测

为了准确提取深度视频的DM区域, 并使提取方法复杂度低, 我们将当前帧与时域前向和后向参考帧做帧差, 得到2幅差值像素图 D_1 和 D_2 , 对1幅 $W \times H$ 的图像, 其帧差可表达为:

$$\begin{aligned} D_1(i, j) &= |p_{cur}(i, j) - p_{pref}(i, j)|, \\ D_2(i, j) &= |p_{cur}(i, j) - p_{bref}(i, j)|, \\ 0 &\leq i \leq W, 0 \leq j \leq H. \end{aligned} \quad (2)$$

其中: $p_{cur}(i, j)$ 表示当前帧在位置 (i, j) 处的像素值; $p_{pref}(i, j)$ 和 $p_{bref}(i, j)$ 分别表示当前帧的前向参考帧和后向参考帧在位置 (i, j) 处的像素值. 为取阈值确定运动与静止区域, 对帧差图 D_1 和 D_2 利用大津阈值算法(OTSU), 得到自适应的最优阈值 T_{OTSU1} 和 T_{OTSU2} , 以避免固定阈值引起失真. 提取的运动掩模 D 可表示为:

$$D(i, j) \in \begin{cases} DS, & \text{if } D_1(i, j) < T_{OTSU1} \text{ and } D_2(i, j) < T_{OTSU2}, \\ DM, & \text{其他,} \end{cases} \quad (3)$$

其中: DS表示深度静止区域; DM表示深度运动区域. 用上述方法基本能完成深度视频的运动和静止区域的分割.

1.3 深度视频区域分割

当深度视频LCU区域内所有像素属于DS区域时, 如果对应的彩色视频LCU属于TS区域, 则该LCU属于TSDS区域, 否则属于TMDS区域. 当深度视频LCU区域内有部分像素属于DM区域时, 如果对应的彩色视频LCU属于TM区域, 则该LCU属于TMDM区域, 否则属于TSDM区域.

2 深度视频快速LCU深度划分和模式选择策略

2.1 深度视频编码特征统计分析

图1为Balloons、Kendo、Newspaper、GhostTownFly和Shark等序列中属于TMDM和TSDM区域的LCU的最大分割深度分别强制为0时, 编码性能(Bjontegaard Delta Bit Rate, BDBR)^[14]的变化情况. 从图1可看出这些序列的BDBR波动很剧烈. 其原因是TMDM区域的LCU绝大部分的最优划分深度大于0. 如果这些LCU的划分直接为0时, 会引起该区域的编码失真. 为了不影响最终

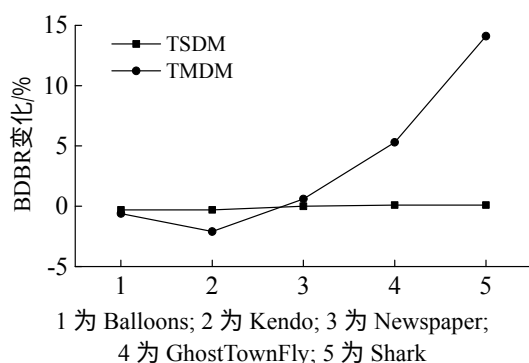


图1 不同序列 TSDM 和 TMDM 区域的 BDBR 影响

的编码性能, 我们把 TMDM 区域的 LCU 最大分割深度设置为 3. 对于 TSDM 区域的 LCU, 如果最大分割深度强制为 0 时, BDBR 波动很小. 因此, 属于 TSDM 区域的 LCU 的最大分割深度可强制为 0, 而不需再继续分割. 这样的编码策略能在维持编码性能的基础上, 极大地降低编码复杂度.

表 1 为 TSDM 区域中 LCU 划分深度限制为 0 时最优预测模式分布. 从表 1 可知, TSDM 区域 LCU 的最优预测模式为 Skip 模式的平均比例为 84.42%, Merge 模式的平均比例为 6.88%, 其他的预测模式平均比例小于 8.70%. 如果可以提前判断出 TSDM 区域 LCU 的最优预测模式为 Skip 模式, 就可以跳过其他的候选预测模式搜索过程, 从而节省编码时间. 研究发现, TSDM 区域的 LCU 在 Skip 模式下的残差系数(Residual Coefficient, RC)为 0 时, 该 LCU 采用 Skip 为最优模式概率很高.

表 1 TSDM 区域中 LCU 划分深度限制为 0 时最优预测模式分布 %

序列	Skip	Merge	Inter-2N×2N	Inter-Others	Intra
Balloons	83.04	8.09	5.88	1.45	1.54
Newspaper	70.72	12.31	10.14	3.39	3.44
Kendo	82.99	5.32	3.27	0.92	7.50
GhostTownFly	92.39	5.83	1.22	0.09	0.47
Shark	92.96	2.86	2.55	0.91	0.72
平均	84.42	6.88	4.61	1.35	2.73

表 2 为 TSDM 区域的 LCU 在 Skip 模式下的残差系数的统计规律. 当 RC 为 0 时, LCU 的最优预测模式为 Skip 模式的平均比例为 95.46%. 因此, 拟通过 RC 是否为 0 对当前编码模式是否为 Skip 模式进行预测. 如果当前 LCU 在 Skip 模式下的 RC 为 0, 可以只做 Skip 模式, 提前终止模式搜索过程,

从而节省编码时间.

表 2 RC 为 0 时, TSDM 区域 LCU 选取 Skip 模式的比例

序列	比例/%
Balloons	94.83
Newspaper	88.83
Kendo	97.22
GhostTownFly	98.97
Shark	97.43
平均	95.46

表 3 为深度视频 DS 区域 LCU 的最优分割深度分布. 从表 3 可知, DS 区域 LCU 的最优分割深度为 0、1、2、3 的平均比例分别为 95.80%、1.48%、0.84%和 1.88%. 当 LCU 属于 DS 区域时, 其最大分割深度可以强制为 0, 提前终止 CU 递归分割过程, 这种编码策略极大地节省了编码时间, 相应的平均准确率高达 95.80%, 对最终编码性能的影响几乎可忽略不计.

表 3 深度视频 DS 区域 LCU 的最优分割深度分布 %

序列	LCU 最优分割深度			
	0	1	2	3
Balloons	93.90	1.92	1.29	2.89
Newspaper	93.38	1.61	1.18	3.82
Kendo	97.45	1.23	0.58	0.74
GhostTownFly	99.66	0.21	0.07	0.05
Shark	94.59	2.44	1.07	1.90
平均	95.80	1.48	0.84	1.88

表 4 为深度视频 DS 区域 LCU 的最优预测模式分布. 从表 4 可看出, DS 区域 LCU 的最优预测模式为 Skip 模式的平均比例为 96.39%, Merge 模式的平均比例为 1.48%, Inter-2N×2N 模式的平均比例为 0.56%, 其他预测模式平均比例为 1.58%. 因此,

表 4 深度视频 DS 区域 LCU 的最优预测模式分布 %

序列	Skip	Merge	Inter-2N×2N	Inter-Others	Intra
Balloons	96.88	1.00	0.67	0.07	1.38
Newspaper	98.52	0.42	0.65	0.06	0.35
Kendo	94.66	0.92	0.56	0.10	3.77
GhostTownFly	97.42	2.25	0.03	0.00	0.30
Shark	94.47	2.79	0.91	0.10	1.74
平均	96.39	1.48	0.56	0.07	1.51

当LCU属于DS区域时,可以只做Skip模式,提前终止模式搜索过程,从而降低编码复杂度。

2.2 深度视频各区域快速编码策略

本文整体算法如图2所示,包括区域划分和快速算法两个部分。如果当前LCU属于TMDM,则采用原始的编码方法,即LCU的深度搜索范围为0到3,并采用全搜的模式选择策略。如果当前LCU属于TSDM,只需要搜索深度0,并根据RC的值确定是否提前终止模式搜索。如果RC为0,则提前终止模式选择过程;否则进行模式全搜索。如果当前LCU属于TMDS和TSDS区域,则LCU的最大分割深度设置为0,且只要搜索Skip模式。

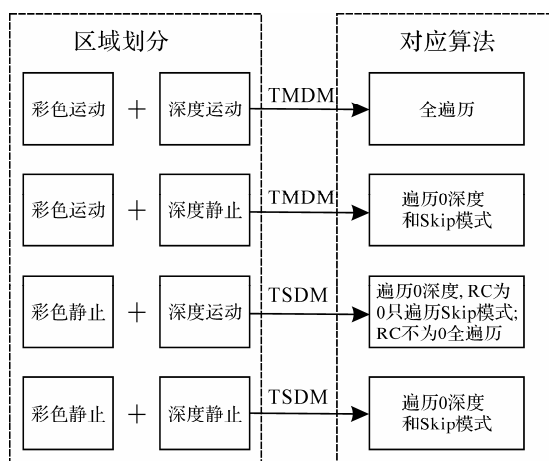


图2 本文算法框图

3 实验结果

为验证其有效性,本算法在3D-HEVC参考软件HTM-10.0上进行测试,实验参数配置采用HTM-10.0的标准测试环境^[15],使用帧间编码配置文件。实验测试条件为:编码结构采用分层B帧,GOP长度为8,I帧编码周期为24,运动估计和视差估计范围为64,LCU的分割深度为0~3,LCU的大小为64×64。实验测试序列、编码左视点、编码右视点和绘制虚拟视点选择见表5。

表6列出了原始测试平台和Mora等^[11]提出的CU分割深度快速选择算法与本文算法在不同量化参数QP(彩色视频QP分别为25、30、35和40,对应的深度视频QP为34、39、42和45)^[15]下的编码时间对比。 DT_x 表示 x 算法与原始测试平台相比总体编码时间(彩色视频编码时间和深度视频编码时间)节省百分比。 Dt_x 表示 x 算法与原始测试平台相

表5 测试序列

序列	图像分辨率	左视点	右视点	虚拟视点
1	1024×768	3	5	4
2	1024×768	3	5	4
3	1024×768	4	6	5
4	1920×1088	1	9	5
5	1920×1088	1	9	5

注:1为Balloons序列;2为Kendo序列;3为Newspaper序列;4为GhostTownFly序列;5为Shark序列。

比深度视频编码时间节省百分比。计算公式为:

$$DT_x = 100\%(T_x - T_{\text{HTM-10.0}}) / T_{\text{HTM-10.0}},$$

$$x \in \{\text{Mora, Pro, Final}\},$$

$$Dt_x = 100\%(t_x - t_{\text{HTM-10.0}}) / t_{\text{HTM-10.0}},$$

$$x \in \{\text{Mora, Pro, Final}\},$$

(4)

其中: DT_{Final} 和 Dt_{Final} 分别表示本文算法和Mora等提出算法结合后与原始测试平台相比总体编码时间和深度视频编码时间节省百分比。

从表6可以看出,本算法节省了40.13%的总

表6 本算法与对比文献和原始平台相比

		编码时间节省情况						%
序列	QP	DT_{Mora}	Dt_{Mora}	DT_{pro}	Dt_{pro}	DT_{Final}	Dt_{Final}	
1	25	-32.14	-52.64	-33.89	-55.69	-8.35	-21.71	
	30	-36.19	-57.74	-38.59	-61.62	-7.61	-20.47	
	35	-39.69	-60.98	-42.50	-65.54	-7.90	-20.23	
	40	-42.49	-63.25	-45.78	-68.52	-7.99	-20.24	
2	25	-32.86	-53.92	-31.68	-52.39	-6.37	-17.34	
	30	-35.05	-57.18	-35.44	-58.27	-6.80	-19.15	
	35	-37.42	-59.87	-38.93	-62.55	-7.04	-20.16	
	40	-39.74	-62.09	-42.35	-66.48	-7.32	-20.54	
3	25	-36.79	-57.40	-39.72	-62.23	-8.71	-22.37	
	30	-39.82	-60.42	-42.91	-65.26	-7.87	-20.53	
	35	-42.33	-62.48	-45.55	-67.31	-7.67	-19.34	
	40	-44.33	-64.21	-47.95	-69.64	-7.70	-19.30	
4	25	-34.40	-56.52	-33.75	-55.12	-7.29	-20.39	
	30	-39.10	-61.72	-38.52	-61.52	-7.45	-20.13	
	35	-43.74	-64.88	-43.66	-65.85	-6.90	-19.19	
	40	-45.78	-66.43	-47.55	-69.74	-7.89	-19.29	
5	25	-25.26	-46.22	-32.03	-58.89	-9.06	-23.55	
	30	-30.18	-53.90	-35.95	-65.03	-8.25	-24.15	
	35	-36.07	-59.80	-40.47	-68.67	-6.88	-22.08	
	40	-40.22	-63.27	-45.32	-71.34	-8.53	-21.96	
平均		-37.68	-59.25	-40.13	-63.58	-7.68	-20.61	

注:1为Balloons序列;2为Kendo序列;3为Newspaper序列;4为GhostTownFly序列;5为Shark序列。

体编码时间和 63.58% 的深度视频编码时间. 文献[11]提出的算法实现了 37.68% 的总体编码时间节省和 59.25% 的深度视频编码时间节省. 文献[11]提出的算法主要是优化 CU 的分割深度决策过程, 而本算法主要是提前确定 LCU 的最大分割深度和最优预测模式, 所以编码时间节省优于文献[11]算法. 本算法在文献[11]的算法基础上, 依然能够节省 20.61% 的深度视频编码时间和 7.68% 的总体编码时间, 说明了本算法的有效性.

表 7 是 DPSNR_x 算法与原始平台算法绘制得到的虚拟视点的 PSNR 变化情况. 从表 7 可看出, 文献[11]提出的算法的虚拟视点 PSNR 变化范围为 -0.23~-0.01 dB, 平均下降 0.05 dB. 本算法虚拟视点 PSNR 变化范围为 -0.16~0 dB, 平均下降 0.03 dB. 在虚拟视点质量上, 本算法优于文献[11]提出的算法, 原因是本算法考虑了深度视频在不同区域的编码失真对虚拟视点质量的不同影响. 将文献[11]提出的算法与本算法结合后, 虚拟视点 PSNR 变化范围为 -0.17~0.01 dB, 平均下降 0.03 dB, 几乎可忽略不计, 说明本算法具有较好的鲁棒性.

表 7 虚拟视点质量比较 dB

算法	QP	序列					平均
		1	2	3	4	5	
A	25	-0.03	-0.08	-0.03	-0.11	-0.23	-0.05
	30	-0.02	-0.06	-0.01	-0.02	-0.12	
	35	-0.04	-0.06	-0.02	-0.02	-0.06	
	40	-0.04	-0.03	-0.02	-0.01	-0.01	
B	25	-0.01	-0.05	0.00	-0.09	-0.16	-0.03
	30	-0.02	-0.05	0.00	-0.01	-0.09	
	35	0.00	-0.04	0.00	0.00	-0.06	
	40	-0.03	-0.01	-0.01	0.00	-0.03	
C	25	-0.01	-0.01	0.01	-0.02	-0.17	-0.03
	30	-0.03	-0.02	-0.02	-0.02	-0.13	
	35	0.00	0.00	0.01	-0.02	-0.08	
	40	0.01	0.00	0.01	-0.01	-0.07	

注: 1 为 Balloons 序列; 2 为 Kendo 序列; 3 为 Newspaper 序列; 4 为 GhostTownFly 序列; 5 为 Shark 序列. A 为 DPSNR_{Mora} 算法; B 为 DPSNR_{Pro} 算法; C 为 DPSNR_{Final} 算法.

BDBR_x 表示 x 算法与原始测试平台相比的编码性能(表 8). 从表 8 可知, 本算法的 BDBR 变化范围为 -8.30%~1.80%, 平均提高 2.78%. 文献[11]提出的算法 BDBR 变化范围为 -10.30%~0.40%, 平均提

高了 4.10%. 本算法在文献[11]算法基础上, BDBR 变化范围为 -1.40%~2.90%, 平均下降 0.26%. 在 BDBR 方面, 本算法稍劣于文献[11]算法, 但在编码时间节省和虚拟视点质量上均优于文献[11]算法. 综上说明, 本算法具有较强的有效性和鲁棒性.

表 8 BDBR 编码性能比较 %

序列	BDBR _{Mora}	BDBR _{Pro}	BDBR _{Final}
Balloons	-8.00	-6.60	-0.70
Kendo	-2.70	-1.30	-0.20
Newspaper	-10.30	-8.30	-1.40
GhostTownFly	0.40	0.50	0.70
Shark	0.10	1.80	2.90
平均	-4.10	-2.78	0.26

4 结语

提出了一种基于运动一致性的深度视频帧间编码快速算法, 将深度视频分为 4 个区域, 根据不同区域编码失真对绘制失真的不同影响, 在不同的区域设置了不同的编码策略以加速 CU 尺寸预测和模式选择过程. 实验结果表明, 本算法与原始测试平台 HTM-10.0 相比, 在绘制的虚拟视点质量几乎一致的情况下, 各序列深度视频编码时间平均节省了 63.58%, 总体编码时间节省了 40.13%.

参考文献:

- [1] MORI Y, FUKUSHIMA N, YENDO T, et al. View generation with 3D warping using depth information for FTV[J]. Signal Processing: Image Communication, 2009, 24(1):65-72.
- [2] YANG L, TEHRANI M P, FUJII T, et al. High-quality virtual view synthesis in 3DTV and FTV[J]. 3D Research, 2011, 2(4):1-13.
- [3] SULLIVAN G J, OHM J R, HAN W J, et al. Overview of the high efficiency video coding (HEVC) standard[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12):1649-1668.
- [4] MULLER K R, SCHWARZ H, MARPE D, et al. 3D high-efficiency video coding for multi-view video and depth data[J]. IEEE Transactions on Image Processing, 2013, 22(9):3366-3378.
- [5] TECH G, CHEN Y, MÜLLER K, et al. Overview of the multi-view and 3D extensions of high efficiency video coding[J]. IEEE Transactions on Circuits and Systems for

- Video Technology, 2016, 26(1):35-49.
- [6] JCT-3V. J1003-2014. Test model 10 of 3D-HEVC and MV-HEVC[S].
- [7] JCT-VC. N1002-2013. High efficiency video coding (HEVC) test model 12 (HM 12) encoder description[S].
- [8] PARK C S. Edge-based intra mode selection for depth-map coding in 3D-HEVC[J]. IEEE Transactions on Image Processing, 2015, 24(1):155-162.
- [9] SHEN L, AN P, ZHANG Z, et al. A 3D-HEVC fast mode decision algorithm for real-time applications[J]. ACM Transactions on Multimedia Computing Communications & Applications, 2015, 11(3):1-23.
- [10] CHUNG K L, HUANG Y H, LIN C H, et al. Novel bitrate saving and fast coding for depth videos in 3D-HEVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 26(10):1859-1869.
- [11] MORA E G, JUNG J, CAGNAZZO M, et al. Initialization, limitation, and predictive coding of the depth and texture quadtree in 3D-HEVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2014, 24(9):1554-1565.
- [12] LEI J, LI S, ZHU C, et al. Depth coding based on depth-texture motion and structure similarities[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2015, 25(2):275-286.
- [13] PAN Z, ZHANG Y, KWONG S. Fast mode decision based on texture-depth correlation and motion prediction for multiview depth video coding[J]. Journal of Real-Time Image Processing, 2016, 11(1):27-36.
- [14] BJONTEGAARD G. Calculation of average PSNR differences between RD-curves[EB/OL]. [2016-07-10]. http://wftp3.itu.int/av-arch/video-site/0104_Aus/.
- [15] JCT-3V, C1100-2013. Common test conditions of 3DV core experiments[S].

Motion consistency-based fast algorithm for 3D-HEVC depth video inter-coding

LIU Sheng, CHEN Fen*, JIN De-fu

(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

Abstract: Aiming at the high complexity problem in 3D-HEVC depth video inter-coding, a fast depth video inter coding algorithm is proposed using motion consistency. Firstly, each frame of depth video is divided into four regions, which are texture move and depth move region, texture move and depth static region, texture static and depth move region, and finally texture static and depth static region. Based on these four divided regions, different partition depth levels and prediction modes of LCU are designed for encoding each of the regions. Compared to HTM-10.0, our experimental results show that the proposed algorithm can save 40.13% entire time and 63.58% depth time over various test sequences with insignificant negative impact on the quality of virtual views as a whole.

Key words: depth video; 3D-HEVC; inter coding; motion consistency; largest partition depth; mode decision

(责任编辑 史小丽)