

一种深度图像帧内编码单元快速划分算法

朱 威^{1,2} 易 瑶¹ 王图强¹ 郑雅羽^{1,2}

(浙江工业大学信息工程学院 杭州 310023)¹ (浙江省嵌入式系统联合重点实验室 杭州 310023)²

摘 要 新一代的三维视频编码标准——3D-HEVC (3D High Efficient Video Coding) 为了显著减少视点个数,增加了包含视频场景几何信息的深度图像,但深度图像编码的计算复杂度非常高,其编码时间是彩色图像的 4 倍左右。为了降低深度图像编码的计算复杂度,文中提出了一种基于纹理特征分析的深度图像帧内编码单元(CU)快速划分算法。首先,对深度图像的编码树单元(CTU)进行初级纹理特征分析,根据深度图像的纹理变化特征,在大津法的基础上对全局灰度进行分级,再通过判断 CTU 内采样点的纹理复杂度以及纹理方向标识来确定当前 CTU 的划分趋势。然后,对纹理复杂度高的 CTU 进行 CU 级别的精细纹理特征分析,利用 CU 内部像素分布的统计特征,自底向上计算不同尺寸的 CU 的纹理划分标识。最后,根据 CTU 的纹理复杂度、纹理方向标识以及 CU 的纹理划分标识预测当前 CTU 的划分深度范围,并判断是否提前终止 CU 划分。实验结果表明,与 3D-HEVC 参考模型中的原始算法相比,所提算法在平均增加 0.8% 左右码率的同时,能够降低 45% 左右的编码时间,同时保持了良好的编码率失真性能;与现有的 3 种快速算法相比,所提算法在整体序列上分别降低了约 12%, 3%, 4% 的编码时间,而在大分辨率序列上则分别降低了 14%, 11%, 10% 的编码时间,并具有相近的编码率失真性能。

关键词 3D-HEVC, 帧内编码, 深度图像, 纹理特征分析

中图法分类号 TP391 文献标识码 A DOI 10.11896/jsjcx.180701337

Fast Coding Unit Partition Algorithm for Depth Maps

ZHU Wei^{1,2} YI Yao¹ WANG Tu-qiang¹ ZHENG Ya-yu^{1,2}

(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)¹

(United Key Laboratory of Embedded System of Zhejiang Province, Hangzhou 310023, China)²

Abstract The 3D high efficient video coding (3D-HEVC) is the new generation of video coding standard, which adopts depth maps to reduce the number of viewpoints significantly. Depth maps contain geometric information which can improve the video compression efficiency, but the depth image encoding has a heavy computation and takes about 4 times as long as the color image encoding. In this paper, in order to reduce the computational complexity of the depth image coding, a coding unit (CU) partition algorithm based on texture analysis was proposed for 3D-HEVC intra coding. Firstly, the rough texture analysis is performed for the depth map, the global grayscale classification based on the OTSU method is calculated through the texture characteristics of the depth map, and the texture complexity and the direction identification of CTU are determined by the sample points in CTU. Then, the fine texture analysis is performed on a CTU with high texture complexity, and the statistical features of the pixel distribution in the CUs are used to compute the textural division flags from bottom to top for different size of CUs. Finally, the texture complexity of CTU, texture direction flags and CU texture division flags are utilized to predict the depth range of current CTU and decide whether to terminate the division of CU. Compared with the original algorithm in 3D-HEVC test model, the proposed algorithm can reduce 45% encoding time on average with only 0.8% increase in Bjontegaard delta bit rate under the all-intra configuration. Compared with three state-of-the-art algorithms, the proposed algorithm reduces the encoding time by about 12%, 3% and 4% respectively for overall sequences, and 14%, 11% and 10% respectively for the large-resolution sequences, with a similar encoding rate distortion performance.

Keywords 3D-HEVC, Intra coding, Depth map, Texture analysis

到稿日期:2018-07-20 返修日期:2018-11-15 本文受浙江省自然科学基金(LY17F010013),国家自然科学基金(61401398)资助。

朱 威(1982—),男,博士,副教授,硕士生导师,主要研究方向为视频编解码和智能视觉处理,E-mail:weizhu@zjut.edu.cn(通信作者);易 瑶(1992—),女,硕士生,主要研究方向为三维视频编码;王图强(1995—),男,硕士生,主要研究方向为三维视频处理;郑雅羽(1978—),男,博士,副研究员,硕士生导师,主要研究方向为视频信息处理。

1 引言

三维视频(Three-Dimensional Video,3DV)是一种新兴的传播媒体,第三维度增加了人眼对物体的深度感知,为传统的2D视频提供了无法比拟的真实视觉感受^[1]。虚拟现实(Virtual Reality,VR)、增强现实(Augment Reality,AR)、混合现实(Mixed Reality,MR)等智能视觉交互技术在2016年得到了井喷式发展,这一系列视觉交互技术的发展也带动了3D视频显示技术的快速发展。同时,传统影视工业在融合三维显示技术后取得了巨大的经济效益,但由于以往应用系统的限制,人们需要通过佩戴眼镜来实现对物体的深度感知。随着人们感知需求的提高,可用于裸眼显示的3D视频也将成为未来发展的趋势。在技术革新、商业价值和现实需求多重驱动下,3D视频成为了近年来的研究热点。

为满足三维视频编码的需求,三维视频编码联合开发组(JCT-3V)于2013年成立,并将运动图像专家组(MPEG)于2007年提出的三维视频编码框架——多视点加深度视频(Multi-view Video plus Depth,MVD)与JCT-VC于2012年提出的新一代视频编码标准(High Efficiency Video Coding,HEVC)相结合,形成了新一代三维视频编码标准——3D-HEVC^[2-4]。3D-HEVC将不同视角的彩色视频序列和对应的深度视频序列按拍摄位置逐一编码^[5],将多路编码信息复用至一个比特流中。该比特流中包含多个子比特流,在解码端,可以利用DIBR技术绘制各编码视点间的任意虚拟视角的3D视频^[5]。3D-HEVC作为HEVC的扩展,其深度图的编码主要沿用了HEVC的编码技术,包括编码树单元(CTU)、编码单元(CU)和变换单元(TU)的四叉树递归划分结构及其相关的编码工具。同时,针对深度图像的特征,在HEVC预测单元(PU)已有的35种帧内预测模式的基础上新增了2种DMM预测模式,分别为楔形分割模式DMM1和轮廓分割模式DMM4,以使其更好地预测深度图像像素变化锐利的边界信息。

深度图像的引入为虚拟视点合成减少了大量冗余,在保证合成视点图像质量的情况下节约了编码比特开销,但深度图像编码的计算复杂度消耗了大量的编码时间,严重阻碍了其推广和应用。为降低3D-HEVC编码的计算复杂度,近年来已有不少学者对相关的快速算法展开了研究^[6-19]。大部分编码优化算法主要针对以下3个方面进行优化:CU划分、PU模式决策以及DMM预测模式的选择。文献[6-10]对CU四叉树划分过程进行了优化,以降低递归划分和率失真优化过程带来的计算复杂度。文献[6-7]将CU划分过程归为分类问题,分别利用决策树模型和多类支持向量基模型训练学习,得到对应的分类器来预测CU的划分深度并判断是否提前终止CU划分;文献[8]通过当前CU的方差及其4个子CU的方差之间的关系来决定是否提前终止当前CU的划分;文献[9]在保证合成虚拟视点质量的情况下,根据PU完成预测过程后的尺寸来确定CU划分的尺寸;文献[10]提出自适应获取图像角点特征,并将其作为提前分配CU划分深度的依据。在PU模式决策方面,文献[8,11-13]分别利用最佳预测模式的分布比例、PU模式的梯度特征、相邻PU的空间位置关系以及最佳预测模式的概率密度函数来减少哈达玛优化的预测模式个数和最佳预测模式候选列表率失真优化过程的计算量。在DMM模式选择方面,文献[7,13-14,16]通过CU划

分深度的预测值、DMM模式与Planar模式之间的关系、DMM模式与最佳预测模式的概率分布以及PU模式的平滑度进行DMM模式选择;文献[13,15,17]分别利用欧几里德距离计算方法、混合蛙跳搜索方法以及基于边缘检测的两级搜索方法来降低DMM模式的计算复杂度。

CU划分属于编码划分结构的顶层,在上述3个方面的优化中计算量最大。现有算法中关于CU划分的算法主要利用了CU的方差特性、PU的划分特性以及CU基视点和参考视点的空间位置关系,这些算法都一定程度地降低了3D-HEVC的计算复杂度,但利用深度图像自身的纹理特征对CU划分进行优化的算法还有待深入研究,CU递归划分过程的计算复杂度还有待进一步降低。

2 深度图像及其CU划分的特征分析

深度图像是由带红外光测距功能的相机实时跟踪所在场景获取的相对距离信息而组成的特殊灰度图像。深度图像每个像素点的灰度值反映了相机拍摄位置与所在场景的空间结构关系,这种关系反映到二维平面上,表现为深度图像的纹理特征变化,像素变化锐利的边界区域呈现了当前场景内物体在空间位置上的显著差异,像素变化平缓的区域则说明当前场景内物体在空间位置上的差异较小。图1为测试序列GT_Fly基视角的彩色图像及其同一时空关系下对应的深度图像,该序列的内容为大片荒漠中的村庄,场景内整体分布的空间范围广,因此其图像纹理变化较为平缓,能直观地反映场景的空间位置变化与深度图像灰度值之间的关系。式(1)表示空间场景距离 Z 与原始灰度级 g (范围为0到255,共256级灰度)的线性量化关系, Z_{far} 到 Z_{near} 是深度图像距离的量化极限范围, Z_{far} 表示当前视角下可量化的最远距离, Z_{near} 表示当前视角下可量化的最近距离。

$$Z = Z_{\text{far}} + g \times \frac{Z_{\text{near}} - Z_{\text{far}}}{255} \quad (1)$$

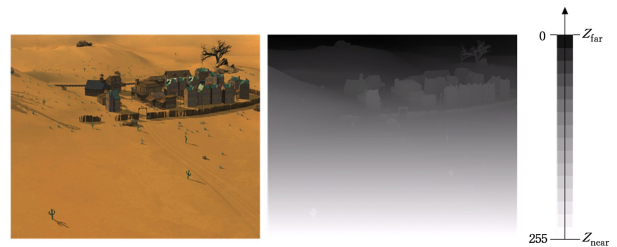


图1 测试序列GT_Fly

Fig.1 Test sequence GT_Fly

为了进一步分析深度图像纹理特征与CU划分特征之间的关系,本文先采用3D-HEVC参考模型HTM16.0对深度图像进行编码,再分析编码后的CU划分结果。选用的标准测试序列为Balloons,内容为杂技演员表演场景,其空间结构层次分明,具有边界区域与平坦区域变化明显的纹理特征。HTM的基本配置如下:各视点间的编码顺序参考相机拍摄位置的顺序,编码方式为全帧内模式(AI)。图2为测试序列Balloons在深度图像量化参数(QP)为34,45时,其基视点第一帧深度图像CU四叉树划分的结果以及部分CTU的放大图。红线为CTU(64×64)的划分结果,蓝线为每个CTU中CU的划分结果, $A_1, A_2, B_1, B_2, C_1, C_2, D_1, D_2$ 表示部分CTU的放大区域。物体内部的大面积平坦区域以及像素变

化缓慢的区域(如 A_1, A_2 区域)多采用较大尺寸 CU 进行编码;凸显物体空间位置差异的边界区域(如 B_1, B_2 区域)多采用较小尺寸 CU 进行编码;对于一个 CTU, 其内既有像素变化平缓的区域, 又有像素变化锐利的边界(如 C_1, C_2, D_1, D_2 区域)。若 CTU 内像素变化的边界具有明显且清晰的方向(如 C_1, C_2 区域), 现有的预测模式可以直接进行较好的预测, 则采用大尺寸 CU 进行编码;若 CTU 内像素变化的边界没有明显的方向性规律(如 D_1, D_2 区域), 现有的预测模式难以直接且较好地预测 CU 的方向性, 则一般采用较大尺寸 CU 与小尺寸 CU 组合编码。经上述对图 2 的直观分析可知, 深度图像 CTU 四叉树划分与其自身纹理特征具有紧密的相关性, 纹理平坦的区域通常采用大尺寸 CU 进行编码, 纹理复杂的区域通常采用小尺寸 CU 进行编码。

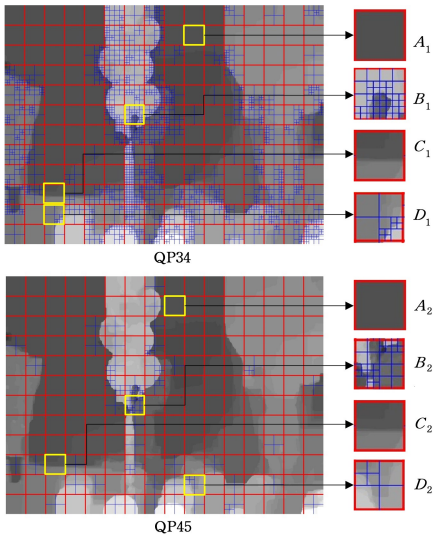


图 2 测试序列 Balloons 的四叉树划分图及部分 CTU 放大图
(电子版为彩色)

Fig. 2 Quadtree partition diagram and CTU enlarged diagram of Balloons sequence view of CTU

为了进一步分析深度图像纹理特征与 CU 划分特征之间的关系, 本文选取了具有不同纹理复杂度和分辨率的测试序列 Balloons, Kendo, GT_Fly, UndoDancer。表 1 列出了在 4 个不同 QP 下对于这 4 个标准测试序列, CU 划分深度的分布比例。将 CU 的划分深度标记为 $Depth$, 其取值范围为 $0 \sim 3$ 。本节的软件实现平台和相关编码配置, 与图 1 所示的配置相同, 每个序列编码 50 帧。从表 1 中可以看出, 不同的 $Depth$ 值所占的分布比例不同, 但分布规律相似, 其中 $Depth$ 为 0 所占的比例高于其他深度值, 且 $Depth$ 为 0 所占的比例随着 QP 值的增加而提高, 在 QP 值为 45 时, 对于 4 个测试序列, 其 $Depth$ 值为 0 的比例高达 $71.65\% \sim 95.22\%$, 其他深度值的比例随着 QP 值的增加而降低。此外, 序列的分辨率不同时, 其 $Depth$ 值为 0 的分布比例趋势也不同, 对于大分辨率序列, 其 $Depth$ 值为 0 的比例高于小分辨率序列。对于纹理复杂度高且分辨率小的测试序列 Balloons, 不同的 QP 值下, $Depth$ 值为 0 的比例为 $29.53\% \sim 73.65\%$, $Depth$ 值为 1, 2, 3 的比例随 QP 值的增加而逐步递减。对于纹理复杂度低且分辨率小的测试序列 Kendo, 其 $Depth$ 值的分布规律与 Balloons 类似, 但其 $Depth$ 值为 0 的比例比纹理复杂的 Balloons

高。对于纹理变化梯度大且分辨率大的测试序列 GT_Fly, 不同的 QP 值下, $Depth$ 值为 0 的比例为 $60.04\% \sim 95.22\%$ 。对于纹理简单且分辨率大的测试序列 UndoDancer, 其 $Depth$ 值的分布规律与 GT_Fly 类似, 但其 $Depth$ 值为 0 的比例比纹理平坦的 GT_Fly 低。上述数据表明: 测试序列划分深度的分布比例与序列自身深度图像的纹理特征相关, 纹理较复杂的序列的 CU 划分深度相对较大, 而纹理较简单的序列的 CU 划分深度相对较小。

表 1 不同 QP 值下 CU 划分深度的分布比例

Table 1 Distribution ratio of partition depth under different QP

Sequences	QP	$Depth$ 值的分布比例/%			
		0	1	2	3
Balloons (1024×768)	34	29.53	31.02	25.85	13.59
	39	40.87	37.69	17.15	4.29
	42	54.77	35.98	7.88	1.41
	45	71.65	25.43	2.49	0.43
Kendo (1024×768)	34	37.02	33.93	20.74	8.30
	39	50.63	36.42	10.90	2.05
	42	68.55	27.03	4.01	0.40
	45	86.42	12.75	0.79	0.04
GT_Fly (1920×1088)	34	60.04	22.7	11.11	6.15
	39	71.21	21.34	5.98	1.47
	42	85.51	12.73	1.62	0.14
	45	95.22	4.53	0.23	0.01
UndoDancer (1920×1088)	34	33.34	34.23	19.30	13.13
	39	55.87	29.06	10.30	4.77
	42	71.72	21.35	5.38	1.55
	45	81.36	16.05	2.30	0.29

在 CU 划分的过程中, 还须根据 CU 尺寸进行 PU 划分, 划分方式包含 $2N \times 2N$ 和 $N \times N$ 。尺寸为 $64 \times 64, 32 \times 32, 16 \times 16$ 的 CU, 只进行 $2N \times 2N$ 的划分; 尺寸为 8×8 的 CU, 可进行 $2N \times 2N$ 和 $N \times N$ 两种划分。深度图像边界区域的 CU 划分结果包含了大量的小尺寸 CU (8×8), 而小尺寸 CU 的 PU 划分及其 PU 模式预测效果又是影响深度图像编码率失真性能的关键。为了分析深度图像 PU 划分的计算复杂度, 表 2 统计了上文所分析的 4 个测试序列在关闭 PU 的 $N \times N$ 划分方式后的编码性能, 测试帧数为 50 帧。统计结果表明, 编码时间变化率 (ΔT) 平均降低了 21.51%, 码流变化率 (BDBR) 增加了 1.3%。分析表 2 可知, 3D-HEVC 虽然仅对 8×8 的 CU 进行了 $N \times N$ 的划分, 但这一过程仍带来了很大的计算量。因此, 在保证深度图像边缘等重要编码区域图像质量的情况下, 可以减少部分平坦区域的 8×8 CU 的 PU 划分, 降低计算复杂度。

表 2 关闭 $N \times N$ 划分方式的编码性能

Table 2 Coding performance of turning of $N \times N$ division
(单位: %)

Sequences	关闭 $N \times N$	
	BDBR	ΔT
Balloons	1.1	20.41
Kendo	0.9	21.24
GT_Fly	1.8	22.79
UndoDancer	1.4	21.59
Average	1.3	21.51

在 3D-HEVC 参考模型 HTM 中, 四叉树划分过程需递归遍历所有深度级别的 CU, 而每个 CTU 在确定最佳划分的过程中最多需要进行 $341(1+4+4^2+4^3+4^4=341)$ 次递归遍历, 并且每次递归都需要计算出所有 PU 预测模式的哈达玛

代价和候选预测模式的率失真代价。这种划分方式虽然可以取得最好的编码率失真性能,但对最终确定为大尺寸的 CU 明显存在较大的计算冗余。如果帧内预测模式能很好地预测平坦区域,那么大尺寸的 CU 反而能节约编码的比特。图像边界是人眼合成深度感知的关键,将影响视频的主观质量。深度图像边界信息的失真将造成合成视点的失真,而平坦区域的失真对合成视点失真度的影响较小。在图像边界区域,若划分为大尺寸 CU,则传统帧内预测模式和 DMM 模式都难以准确预测边缘信息变化的多方向性,需要进一步划分 CU,利用小尺寸 CU 细分边界信息并结合帧内预测模式的方向多样性来提高深度图边缘信息的编码图像质量。根据 CU 划分特征以及影响深度图像编码性能的关键点,利用深度图像纹理变化特征估计不同纹理区域 CU 划分的深度范围并提前终止部分 CU 划分,以获得合理的 CU 划分深度,跳过不必要的率失真代价计算过程,从而降低计算复杂度。这也是本文对深度图像 CU 划分过程的优化方向。

3 深度图像帧内 CU 快速划分算法

本文根据深度图像的纹理变化特征与 CU 划分特征之间的对应关系,提出了基于深度图像纹理特征分析的帧内 CU 快速划分算法。该算法主要由基于全局灰度分级和阵列采样的初级纹理特征分析(RTA)、基于 CU 内部像素分布统计的精细纹理特征分析(FTA)以及 CTU 深度范围预测过程(CDRP) 3 个部分组成。此外,本文利用测试序列纹理特征和 QP 值调整算法中的灰度分级数和 CU 纹理复杂度阈值,

使算法具有更好的序列适应性。

3.1 基于全局灰度分级和阵列采样的初级纹理特征分析

天津法(OTSU)^[18]是一种自适应选取图像前景与背景分割阈值的经典方法。图 3 为基于 OTSU 阈值的深度图像全局灰度分级图。图 3(a)为测试序列 Balloons 基视角第一帧的原始深度图像。图 3(b)是对图 3(a)做 OTSU 处理后的二值化图像,这一过程实际上是按照 OTSU 阈值对原始深度图像进行两级分割。对比图 3(b)与原始图像可以看出:OTSU 的分割阈值很好地保留了深度图像纹理复杂度高的边界区域信息。在图 3(c)中,为了进一步分析纹理复杂的边界区域与 CU 划分之间关系,将图 2 中 QP 值为 34 的 CU 二叉树划分结果绘制在经过 OTSU 处理的同一帧二值化图像上,可以看出阈值分割的边界主要是 CU 划分细密的区域。但由于阈值分割的过程相当于将当前场景的空间位置进行绝对二值量化,该过程会引起的一部分区域边界量化的失真,因此在图 3(c)的同一分级量化区域内仍存在 CU 划分细密的区域。为了减少两级分割引起的部分区域边界失真,进一步细化前景区域与背景区域之间的边界特征,本文根据深度图像所在场景中物体的空间位置线性量化反映的灰度图像纹理特性,在原始灰度分级(256 级)的基础上,对灰度值进行全局线性灰度分级,即将物体空间位置相近的区域归类至一个灰度等级中,同时尽可能地保留并突出纹理复杂度高的边界区域。图 3(d)、图 3(e)和图 3(f)为增加灰度分级数后的效果,从中可以发现:灰度分级数越高,分级后的图像的边界纹理特征越接近原始深度图像。

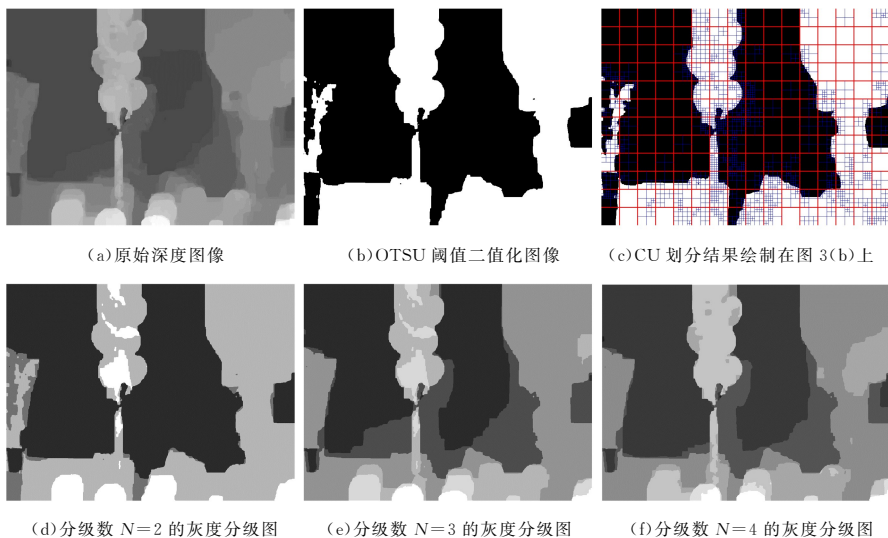


图 3 基于 OTSU 阈值的全局灰度分级图

Fig. 3 Global grayscale based on OTSU threshold

经过上述分析,本文所提算法中的 RTA 通过分析 CTU 级别的纹理特征来快速判别当前 CTU 的划分趋势。首先,对深度图像逐帧进行天津法(OTSU)处理,获得将图像前景区域与背景区域分割的基本阈值 T ,在原始图像 256 级灰度分级的基础上,按式(2)将前景区域和背景区域分别划分为 N 级,以进一步线性量化深度图像前景与背景中物体的空间位置关系。式(2)中 $GV(i)$ 为全局灰度分级后第 i 级的分割阈值,其取值范围仍然为 $0 \sim 255$; g 为深度图像原始灰度值; i 为全局灰度级标号,在背景区域 i 取 $0, 1, 2, 3, \dots, N$, 在前景区域 i 取 $N+1, N+2, \dots, 2N$; N 为全局灰度分级数,分级数

N 的取值将在 3.4 节详细讨论。

$$GV(i) = \begin{cases} i \times \frac{T}{N}, & \text{if } 0 \leq g \leq T \\ T + (i - N) \times \frac{255 - T}{N}, & \text{if } T < g \leq 255 \end{cases} \quad (2)$$

在进行全局灰度分级后,判断当前 CTU 的原始灰度值是否都在同一级全局灰度级中,将其作为 CTU 的纹理复杂度标识,记为 S_{CTU} 。为了减少灰度分级过程中遍历所有像素点引入新的计算复杂度,本文对尺寸为 64×64 的 CTU 按图 4 中 CTU 内像素点的阵列采样方式,从 CTU 第一行开始

每隔 8 行选取一行,每行中每隔 8 个像素选取一个像素点来判断当前 CTU 的纹理复杂度。首先,通过式(3)确定 CTU 中第 k 个采样点 $pixel_k$ 的全局灰度级 GP_k ,其中 $GV(i)$ 为全局灰度分级后第 i 级的分割阈值, $GV(i+1)$ 为全局灰度分级后第 $i+1$ 级的分割阈值。再按式(4)计算当前 CTU 的纹理复杂度标识 S_{CTU} 。若 64 个采样点的全局灰度级 GP_1-GP_{64} 均在同一个灰度级内,则当前 CTU 属于单灰度级,纹理较为简单,此时 S_{CTU} 取 0;若存在任意一个采样点不在同一灰度级内,则当前 CTU 属于多灰度级,此时 S_{CTU} 取 1,表示当前 CTU 需要进一步分析纹理特征。

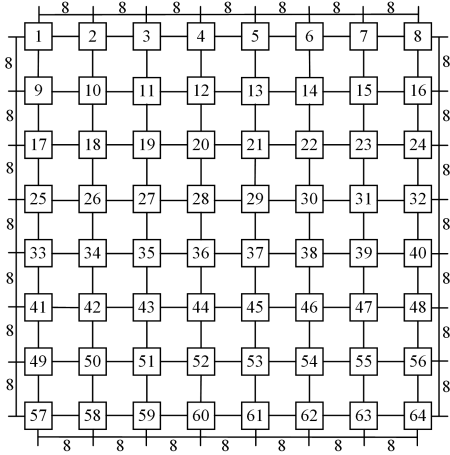


图 4 阵列采样方式

Fig. 4 Array sampling method of pixels

$$GP_k = i, \text{ if } GV(i) \leq pixel_k < GV(i+1) \quad (3)$$

$$S_{CTU} = \begin{cases} 0, & \text{if } \sum_{k=1}^{64} |GP_k - GP_1| = 0 \\ 1, & \text{otherwise} \end{cases} \quad (4)$$

为了减少将部分跨灰度级但其内部像素突变的边界具有清晰方向的多灰度级 CTU 误划分,提高初级纹理分析的准确性,本文通过式(5)计算 CTU 的纹理方向特征。水平方向和竖直方向分别记为 Wp_i 和 Hp_j ; wp_i 和 hp_j 为图 4 所示的 CTU 采样阵列中第 1 列与第 8 列以及第 1 行与第 8 行采样位置原始像素依次相减的差值,其中 i 和 j 分别取 1, 2, ..., 7, 表示采样阵列中对应位置相减后的标号; wp_0 为采样阵列中第 1 列减第 8 列的第 1 个差值,即图 4 中第 1 个像素减第 8 个像素; hp_0 为采样阵列中第 1 行减第 8 行的第 1 个差值,即图 4 中第 1 个像素减第 57 个像素。然后,通过式(6)判断 CTU 的纹理方向标识 S_{WH} ,若小范围波动的 Wp_i 和 Hp_j 的个数大于或等于 6,则当前 CTU 的纹理方向明显,该 CTU 适合直接划分为尺寸为 64×64 的 CU,此时

S_{WH} 取 0,反之 S_{WH} 取 1。

$$\begin{cases} Wp_i = 1, & \text{if } |wp_i - wp_0| \leq 1 \\ Wp_i = 0, & \text{if } |wp_i - wp_0| > 1 \\ Hp_j = 1, & \text{if } |hp_j - hp_0| \leq 1 \\ Hp_j = 0, & \text{if } |hp_j - hp_0| > 1 \end{cases} \quad (5)$$

$$S_{WH} = \begin{cases} 0, & \text{if } (\sum_{i=1}^7 Wp_i \geq 6) \parallel (\sum_{j=1}^7 Hp_j \geq 6) \\ 1, & \text{others} \end{cases} \quad (6)$$

3.2 基于 CU 内部像素分布统计的精细纹理特征分析

深度图像纹理复杂的区域是 CU 划分细密的区域,为了进一步分析上节中多灰度级 CTU 内部 CU 的纹理特征,假设 CU 内的所有像素点为一个离散数列,统计数列中的最大值 (Max)、最小值 (Min) 和均值 (Avg),这 3 个特征值具有 $Max \geq Avg \geq Min$ 的特点,将最大值与均值的差值记为 L_1 ,最小值与均值的差值记为 L_2 ,这两个特征值能衡量当前数列中各元素的分布趋势,而这一分布趋势能反映图像的纹理变化趋势。因此,本节统计了图 2 中放大的 CTU (D_1) 中各个深度级别 CU 内部像素的分布特征,这一区域由不同尺寸类型的 CU 组成,具有一定代表性。图 5 为 D_1 区域不同划分层次 CU 内部像素分布的统计结果,图 5(a) 为 D_1 区域最终的 CU 划分结果,图 5(b) 表示尺寸为 64×64 的 CU ($Depth=0, CU_{64 \times 64}$) 划分为 4 个尺寸为 32×32 的 CU ($CU_{1,32 \times 32}, CU_{2,32 \times 32}, CU_{3,32 \times 32}, CU_{4,32 \times 32}$),这 4 个 CU 中没有继续向下划分的 CU ($CU_{1,32 \times 32}, CU_{2,32 \times 32}, CU_{3,32 \times 32}$) 的特征值 L_1 和 L_2 偏小,纹理变化趋势相对比较平坦;而需要继续向下划分的 CU ($CU_{4,32 \times 32}$) 的特征值 L_1 和 L_2 均偏大,纹理变化剧烈且现有预测模式难以直接准确预测。图 5(c) 为继续向下划分的 CU ($Depth=1, CU_{4,32 \times 32}$) 划分为 4 个尺寸为 16×16 的 CU ($CU_{1,16 \times 16}, CU_{2,16 \times 16}, CU_{3,16 \times 16}, CU_{4,16 \times 16}$),这 4 个子 CU 的特征值 L_1 和 L_2 与 32×32 的 CU 类似, L_1 和 L_2 值偏大的 CU ($CU_{2,16 \times 16}, CU_{3,16 \times 16}$) 纹理复杂,均继续向下划分,而 L_1 和 L_2 值偏小的 $CU_{1,16 \times 16}, CU_{4,16 \times 16}$ 的纹理平坦,均没有继续向下划分。图 5(d) 为继续向下划分的 CU ($Depth=2, CU_{3,16 \times 16}$) 划分为 4 个尺寸为 8×8 的 CU ($CU_{1,8 \times 8}, CU_{2,8 \times 8}, CU_{3,8 \times 8}, CU_{4,8 \times 8}$),达到了 CU 最大划分深度,其特征值 L_1 和 L_2 与自身纹理特征仍然具有类似的相关性, L_1 和 L_2 值偏大的复杂纹理 CU ($CU_{2,8 \times 8}, CU_{4,8 \times 8}$) PU 划分方式为 $N \times N$,而纹理平坦的 CU ($CU_{1,8 \times 8}, CU_{3,8 \times 8}$) PU 划分方式为 $2N \times 2N$,这说明 PU 划分方式由 8×8 CU 的纹理复杂度决定。图 5(e) 为 PU 划分方式为 $N \times N$ 的 CU ($Depth=3, CU_{2,8 \times 8}$) 4 个子 PU 的特征值 L_1 和 L_2 。

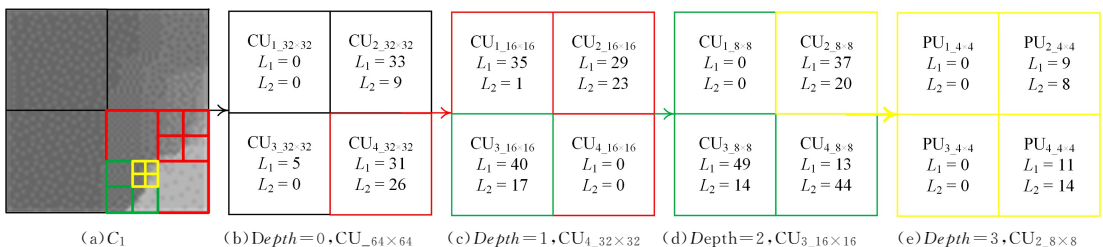


图 5 CU 内部像素分布特征

Fig. 5 Pixel distribution characteristics of CU

经过上述对 CU 内部像素分布统计特征与 CU 自身纹理特征之间相关性的分析,本文对纹理复杂度高和纹理方向不清晰的 CTU 进行了 CU 纹理特征精细分析(FTA),并以特征值 L_1 和 L_2 衡量 CU 的纹理复杂度。考虑到小尺寸 CU 能以自底向上的方式通过纹理特征合并得到更大尺寸 CU 的纹理特征,FTA 先对 CTU 中 64 个 8×8 的 CU 进行纹理特征分析。在分析过程中,首先把当前 CU ($n \times n$) 平均分成 4 个相同尺寸 ($m \times m$) 的子块,并按式(7)计算每一个子块的纹理特征复杂度。式中 k 取 $0 \sim 3$, TF_k 为第 k 个子块的纹理特征,包含 L_1 和 L_2 ,两个特征值都偏小则说明该子块的纹理变化平缓,否则说明该子块的纹理变化明显;对于 8×8 CU, n 值为 8, m 值为 4。完成 4 个子块 TF_k 的计算后,通过式(8)计算当前 CU 的纹理划分标识 $S_{n \times n}$,用于判断当前尺寸为 $n \times n$ 的 CU 是否需要下一个划分深度的纹理特征分析。 TF_i 为当前 CU 中第 i 个子块的特征,若 4 个子块中的最大特征值小于阈值 TH ,则当前 CU 的 4 个子块的特征值 L_1 和 L_2 均小于 TH ,即 4 个子块均为简单纹理块,此时 $S_{n \times n}$ 取 0,表示当前 CU 不需要进行下一个深度级别的划分;否则 $S_{n \times n}$ 取 1,表示当前 CU 需要继续划分, TH 的取值将在 3.4 节详细说明。

$$TF_k = \{(L_1)_{m \times m}, (L_2)_{m \times m}\} \quad (7)$$

$$S_{n \times n} = \begin{cases} 0, & \text{if } \arg \max_{i \in [0,3]} TF_i \leq TH \\ 1, & \text{otherwise} \end{cases} \quad (8)$$

在得到 64 个 8×8 CU 的 $S_{8 \times 8}$ 后,自底向上分别计算当前 CTU 中 16 个 16×16 CU、4 个 32×32 CU 和 1 个 64×64 CU 的纹理划分标识。对于每个 16×16 CU,若其包含的 4 个 8×8 CU 中存在一个 $S_{8 \times 8}$ 值为 1,则直接将当前 16×16 CU 的 $S_{16 \times 16}$ 设为 1,即表示当前 CU 需要进行纹理划分;否则采用与 8×8 CU 相同的纹理特征分析方法,按式(7)和式(8)计算其纹理划分标识 $S_{16 \times 16}$,式中 n 值为 16, m 值为 8。对 CTU 中 32×32 CU 和 64×64 CU 的纹理特征进行分析的方式与 16×16 CU 一致,均按式(7)和式(8)计算其纹理划分标识 $S_{n \times n}$ 。对于 32×32 CU,式中 n 值为 32, m 值为 16;而对于 64×64 CU,式中 n 值为 64, m 值为 32。

3.3 CTU 深度范围预测及 CU 提前终止的划分判定

在完成上文的初级和精细纹理特征分析后,按式(9)和式(10)计算当前 CTU 的深度范围 R ,记为 $[D_{\min}, D_{\max}]$,其中 D_{\min} 和 D_{\max} 分别表示当前 CTU 的最小深度值和最大深度值。 D_{\min} 按式(9)计算,首先对当前 CTU 的纹理复杂度标识、纹理方向特征以及 64×64 CU 的纹理划分标识进行判断。若 CTU 纹理简单,或者纹理方向清晰,又或者 64×64 CU 不需要进行纹理划分,就将 D_{\min} 设为 0;否则将 D_{\min} 设为 1。 D_{\max} 按式(10)计算,若当前 CTU 的纹理简单或 64×64 CU 不需要进行纹理划分,就将 D_{\max} 设为 0;否则对 4 个 32×32 CU 的纹理划分标识进行判断。若所有 32×32 CU 都不需要进行纹理划分,就将 D_{\max} 设为 1。若上述 2 个条件均不满足,则继续判断 16 个 16×16 CU 的纹理复杂度标识,如果所有 16×16 CU 都不需要进行纹理划分,则将 D_{\max} 设为 2;若上述 3 个条件均不满足,则至少有一个 16×16 CU 需要进行纹理划分,将 D_{\max} 设为 3。

$$D_{\min} = \begin{cases} 0, & \text{if } S_{CTU} = 0 \parallel S_{WH} = 0 \parallel S_{64 \times 64} = 0 \\ 1, & \text{otherwise} \end{cases} \quad (9)$$

$$D_{\max} = \begin{cases} 0, & \text{if } S_{CTU} = 0 \parallel S_{64 \times 64} = 0 \\ 1, & \text{if } S_{CTU} = 1 \& \& \sum_{l=1}^4 S_{32 \times 32}(l) = 0 \\ 2, & \text{if } S_{CTU} = 1 \& \& \sum_{l=1}^{16} S_{16 \times 16}(l) = 0 \\ 3, & \text{others} \end{cases} \quad (10)$$

为了进一步加快编码速度,CDRP 利用当前 CTU 的纹理复杂度标识 S_{CTU} 以及不同尺寸 CU 的纹理划分标识 $S_{n \times n}$,通过式(11)判定是否提前终止当前 CU 的编码划分。式中 $T_{n \times n}$ 为 CU 终止编码划分的标志,若当前 CTU 的 S_{CTU} 为 0 或者当前尺寸为 $n \times n$ 的 CU 纹理划分标识为 0,则 $T_{n \times n}$ 值取 1,表示提前终止当前 CU 的编码划分;否则 $T_{n \times n}$ 取 0,表示当前 CU 需要进行四叉树递归划分。

$$T_{n \times n} = \begin{cases} 1, & \text{if } S_{CTU} = 0 \parallel (S_{CTU} = 1 \& \& S_{n \times n} = 0) \\ 0, & \text{others} \end{cases} \quad (11)$$

由于 8×8 CU 的 PU 划分方式选择在编码过程中存在一定的计算复杂度,且 8×8 CU 的 PU 划分方式与其纹理特征的关系仍然满足 CU 内部像素分布的统计特征,因此直接利用上文 8×8 CU 的纹理划分标识 $S_{8 \times 8}$ 并通过式(11)判断是否提前终止当前 PU 的进一步划分,即当 $T_{8 \times 8}$ 为 1 时,PU 不需要进行 $N \times N$ 划分,否则需要继续划分。

3.4 灰度分级数和阈值的选择

本文算法中的 RTA 和 FTA 分别涉及灰度分级数 N 和阈值 TH 的选择。灰度分级数和阈值都会影响算法的编码性能,不同的视频序列之间深度图像的纹理变化特征差异较大,需要用固定值来平衡编码率失真性能和编码效率的关系。本文综合考虑纹理特征和 QP 参数对深度图像编码的影响,选择具有不同纹理复杂度和分辨率的测试序列(Balloons, Kendo, GT_Fly, UndoDancer)的前 50 帧作为样本序列,在不同的 QP 值下利用测试模型 HTM16.0 对样本序列进行编码,获取原始编码过程中的相关数据,分别对分级数和阈值进行两阶段建模,首先实现深度图像纹理自适应,然后再实现 QP 自适应。

对于全局灰度分级数的选择, N 值将影响全局灰度等级内的像素范围,若同一灰度等级内像素范围跨度越大,则判断为单灰度级 CTU 的数量越多而多灰度级 CTU 的数量越少,此时 RTA 方法的效率就越高。由于多灰度级 CTU 集中了深度图像的边界信息,而边界信息又是影响编码图像质量的关键,因此,最优分级数在保证 RTA 方法效率的情况下,应尽可能合理且准确地保留多灰度级 CTU 区域。为了平衡 RTA 方法的效率和深度图像的编码质量,本文计算具有不同纹理特征样本序列在 4 个典型 QP 值下的两级分割 OTSU 阈值,并将阈值除以样本序列在原始编码划分过程中多灰度级 CTU 的纹理特征值,以衡量样本在某一 QP 值下的最佳灰度分级数,记为 N_{QP} 。 N_{QP} 具体按式(12)计算,式中 $\lfloor \cdot \rfloor$ 为向下取整函数, T_{Avg} 为样本序列前 50 帧 OTSU 阈值 T 的均值, I_{QP} 为样本序列在当前 QP 值下各个视点多灰度级 CTU 的总数, L_{1i} 和 L_{2i} 为第 i 个多灰度级 CTU 的纹理特征值。表 3 为样本序列在 4 个不同的 QP 值下 N_{QP} 的取值结果及其可行范围(Range)。不同的 QP 值下,各个样本序列的 N_{QP} 的变化趋势相似,都随着 QP 值的增大而减小;同一 QP 值下不同样本序列的 N_{QP} 在小范围内波动,但仍具有一定的规律,小分辨率序

列较大分辨率序列的 N_{QP} 值会偏小一些, 纹理复杂度高的序列的 N_{QP} 较纹理简单的序列的 N_{QP} 会偏大一些。通过上述对 N_{QP} 的变化规律的描述可以说明式(12)的建模方法能衡量序列纹理特征与最佳灰度分级数之间的关系, 具有一定的纹理自适应。因此, 本文根据不同的 QP 值下 N_{QP} 的变化规律和同一 QP 值下大分辨率及复杂纹理序列 N_{QP} 对编码图像质量的影响, 按 QP 值分段计算灰度分级数, 如式(13)所示:

$$N_{QP} = \left\lfloor \frac{T_{Avg}}{2I_{QP} \sum_{i=1}^I (L_{1i} + L_{2i})} \right\rfloor \quad (12)$$

$$N = \begin{cases} 10, & \text{if } QP \leq 34 \\ 7, & \text{else if } QP \leq 39 \\ 5, & \text{else if } QP \leq 42 \\ 4, & \text{others} \end{cases} \quad (13)$$

表3 不同的 QP 值下 N_{QP} 的计算结果

Table 3 Calculation results of N_{QP} under different QP

Sequences	N_{QP}			
	$QP=34$	$QP=39$	$QP=42$	$QP=45$
Balloons	5	4	3	3
Kendo	5	4	4	3
GT_Fly	13	12	11	9
UndoDancer	8	5	4	3
Range	[4,12]	[4,12]	[3,11]	[3,9]

对于阈值 TH 的选择, 首先, 本文直接利用具有不同纹理特征的样本序列在原始编码划分过程中不同的 QP 值下不需要继续划分 CU 的纹理复杂度特征值 L_1 和 L_2 来实现阈值选择的纹理自适应。不同的 QP 值下的阈值记为 TH_{QP} , 由式(14)计算, 其中 I 是某一 QP 值下各个视点不需要继续划分的 CU 总数, L_{1i} 和 L_{2i} 为第 i 个不需要继续划分的 CU 对应的纹理特征值。表4给出了样本序列在4个不同的 QP 值下 TH_{QP} 的取值结果。同一 QP 值下不同的样本序列的 TH_{QP} 与各个样本 TH_{QP} 的均值相接近, 且不同的 QP 值下各个样本序列的 TH_{QP} 的变化趋势相似, 都随 QP 值的增大而增大, 这说明以原始编码过程中不分割 CU 的纹理复杂度特征值作为阈值具有较好的纹理自适应能力。最后, 本文利用样本序列在不同的 QP 值下 TH_{QP} 的均值, 根据 QP 值分段计算 TH , 如式(15)所示:

$$TH_{QP} = \left\lfloor \frac{1}{2I_{QP} \sum_{i=1}^I (L_{1i} + L_{2i})} \right\rfloor \quad (14)$$

$$TH = \begin{cases} 3, & \text{if } QP \leq 34 \\ 4, & \text{else if } QP \leq 39 \\ 5, & \text{else if } QP \leq 42 \\ 6, & \text{others} \end{cases} \quad (15)$$

表4 不同 QP 值下 TH_{QP} 的计算结果

Table 4 Calculation results of TH_{QP} under different QP

Sequences	N_{QP}			
	$QP=34$	$QP=39$	$QP=42$	$QP=45$
Balloons	3	4	6	7
Kendo	3	4	5	6
GT_Fly	4	4	4	4
UndoDancer	3	4	5	6
Average	3	4	5	6

3.5 总体算法流程

本文提出的算法主要包括 RTA, FTA 和 CDRP, 总体的

算法流程如图6所示。

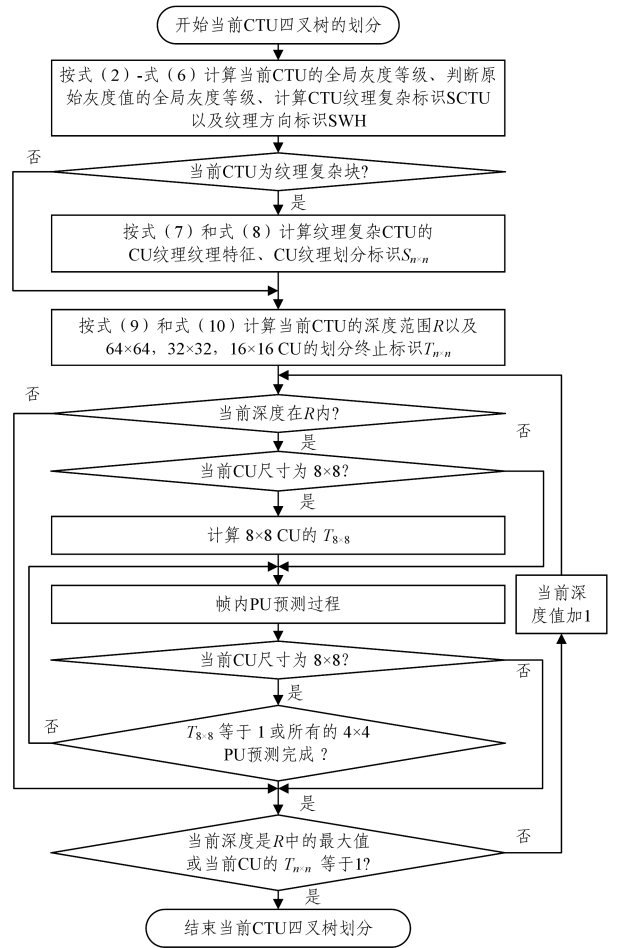


图6 总体算法流程

Fig. 6 Overall algorithm flowchart

4 实验结果与分析

本文算法的软件实验平台为 3D-HEVC 参考模型 HTM16.0, 主要编码配置如下: 各视点间的编码顺序参考相机采集位置顺序, 采用 AI 编码模式, 彩色图像和深度图像的量化参数 QP 按组设为 (25, 34), (30, 39), (35, 42), (40, 45), 其他配置参数按文献[19]设置。实验选取了 JCT-3V 官方推荐的 7 个标准测试序列, 每个序列包含 3 个视点, 每个视点为一组彩色图像和其对应的深度图像, 编码完整帧数, 这些序列包含不同的视频场景、分辨率和纹理复杂度。硬件实验平台的处理器为 Intel Xeon E5-2407, 主频为 2.40GHz, 内存为 32.0 GB。本文采用编码时间变化率 ΔT 来度量计算复杂度的下降, 通过式(16)计算, 其中 T_{Org} 为 HTM 原始编码时间, T_{Pro} 为本文算法优化后的编码时间; 采用 4 个 QP 编码后生成的合成视点的码率变化率 $BDBR$ ^[20] 来度量编码率失真性能的损失。BDBR 由彩色图像和深度图像的总比特率以及合成视点的峰值信噪比 $PSNR$ 计算得到, 其中 $PSNR$ 按式(17)计算, 其中 $PSNR_Y$ 为亮度分量信噪比, $PSNR_U$ 和 $PSNR_V$ 为色度分量信噪比。

$$\Delta T = \frac{(T_{Org} - T_{Pro})}{T_{Org}} \times 100\% \quad (16)$$

$$PSNR = \frac{(PSNR_V \times 6 + PSNR_U + PSNR_V)}{8} \quad (17)$$

本文根据 JCT-3V 制定的三维视频质量客观评价标准(文献[19])及标准文件中测试结果的统计样例,计算得到在优化算法下 6 个合成视点的编码性能,具体结果如表 5 所列。从表中可以看出,与 HTM16.0 原始算法相比,本文算法在不同序列下的 ΔT 的取值范围为 25.79%~64.37%,平均降低了 44.51% 的编码时间;不同序列下同一合成视点的平均 $BDBR$ 的范围为 0.6%~1.0%;同一序列内各个合成视点的平均 $BDBR$ 的范围为 0.4%~1.1%,且同一序列内每个合成视点间的 $BDBR$ 浮动较小,各合成视点的图像质量相对稳定。这些数据表明本文算法显著降低了整体编码时间,同时每个测试序列的各个合成视点的率失真性能损失均较小,这主要是因为本文算法在编码过程中较好地保留了影响合成视点质量的深度图像边界信息。

表 5 各合成视点的编码性能

Table 5 Coding performance of each synthesis view
(单位:%)

Sequences	$BDBR$							ΔT
	$V_2/$	$V_3/$	$V_4/$	$V_6/$	$V_7/$	$V_8/$	All	
Balloons	0.4/0.5	0.4/0.3	0.4/0.3	0.4/0.3	0.4/0.3	0.4/0.3	0.4	34.55
Kendo	0.8/0.8	0.6/0.6	0.6/0.8	0.7/0.7				36.90
Newspaper	0.9/0.8	0.4/0.9	1.3/0.6	0.8				25.79
GT_Fly	0.9/1.4	0.9/0.9	1.4/0.9	1.1				58.93
PoznanHall	0.5/1.1	0.9/0.9	1.5/1.1	1/0.5				50.11
PoznanStreet	0.5/0.7	0.5/0.4	0.6/0.5	0.5				40.89
UndoDancer	0.8/0.9	0.7/0.9	1.1/0.8	0.9				64.37
Average	0.7/0.9	0.6/0.7	1.0/0.7	0.8				44.51

对于小分辨率(1024×768)测试序列 Balloons, Kendo, Newspaper 而言,本文降低了 25.79%~36.90% 的编码时间, $BDBR$ 平均增加了 0.4%~0.8%;对于大分辨率(1920×1088)测试序列 GT_Fly, PoznanHall, PoznanStreet, UndoDancer,本文降低了 40.89%~64.37% 的编码时间,它们的总体 $BDBR$ 增加了 0.5%~1.1%。从数据中可以看出,小分辨率序列相比大分辨率序列,其降低的编码时间相对少一些,但其率失真性能更好。这主要是因为小分辨率序列中 $Depth$ 值为 0 的比例相比大分辨率序列较低,且它们在同一灰度分级内所包含的边界信息也更多,本文算法中 RTA 方法判定 CTU 为单灰度级以及纹理方向清晰的比例降低,FTA 方法对多灰度级 CTU 进行纹理特征分析的比例增加,从而导致计算性能有所下降,这也从侧面反映本文算法的编码性能间接受分辨率的影响。对于小分辨率的测试序列,在率失真性能差不多的情况下,纹理简单的 Kendo 序列降低了 36.90% 的编码时间,纹理复杂的 Balloons, Newspaper 序列分别降低了 25.79%, 34.55% 的编码时间;对于大分辨率的测试序列,纹理简单的 UndoDancer 序列降低了 64.37% 的编码时间且率失真损失较小,纹理复杂度相对较高的 PoznanStreet, PoznanHall 序列分别降低了 40.89%, 50.11% 的编码时间且率失真损失也较小。对于大分辨率且纹理变化趋势平坦的 GT_Fly 序列,本文算法降低了 58.93% 的编码时间,但其编码率失真损失较其他序列偏大,这是因为该序列纹理的变化梯度大,RTA 方法将 CTU 判断为单灰度级的准确率会相对

低一些,但该序列的 $BDBR$ 也仅增加了 1.1%。总的来说,用本文算法对纹理简单的序列进行编码的时间比对纹理复杂的序列进行编码的时间少,这里主要是因为纹理复杂的序列中 RTA 方法将 CTU 判断为单灰度级的次数相对较少,需要通过 FTA 方法进行 CU 级别纹理分析的多灰度级 CTU 更多。

表 6 本文算法与其他方法的性能对比

Table 6 Performance comparison between proposed algorithm and other algorithms

Sequences	本文算法	文献[8]	文献[10]	文献[13]
	$BDBR/\Delta T$	$BDBR/\Delta T$	$BDBR/\Delta T$	$BDBR/\Delta T$
Balloons	0.4/34.55	-0.1/24.40	0.3/39.00	0.7/35.49
Kendo	0.7/36.90	0.1/22.70	0.3/41.00	0.4/38.23
Newspaper	0.8/25.79	0.2/21.10	-0.2/36.00	0.9/33.85
Average Small	0.6/32.41	0.1/22.73	0.1/38.67	0.7/35.86
GT_Fly	1.1/58.93	0.2/40.20	0.2/45.00	0.2/43.14
PoznanHall	1.0/50.11	0.6/39.90	0.3/48.00	0.8/48.33
PoznanStreet	0.5/40.89	0.8/32.40	1.2/40.00	0.3/39.67
UndoDancer	0.9/64.37	0.4/44.10	1.0/39.00	0.5/41.57
Average Large	0.9/53.58	0.5/39.15	0.7/43.00	0.4/43.18
Average All	0.8/44.51	0.3/32.11	0.4/41.14	0.5/40.04

为了进一步评估本文算法的编码性能,表 6 给出了本文算法与同处于 CU 优化级别的文献[8,10]的方法以及处于 PU 优化级别的文献[13]的方法进行编码性能比较的结果。与文献[8]相比,本文所有测试序列的 ΔT 均较高,平均提高了 12.39%,平均 $BDBR$ 仅增加了 0.5%,其中大分辨率序列的平均 ΔT 提高了 14.43%,平均 $BDBR$ 仅增加了 0.4%。与文献[10]比较,本文测试序列整体的平均 ΔT 提高了 3.36%,平均 $BDBR$ 仅增加了 0.3%,小分辨率序列的平均 ΔT 略小于文献[10],但大分辨率序列的平均 ΔT 提高了 10.58%,平均 $BDBR$ 仅增加了 0.2%。对于纹理变化梯度大的 GT_Fly 序列,虽然 $BDBR$ 增加了 0.9%,但平均 ΔT 提升了 13.93%;对于纹理复杂的 PoznanHall 和 PoznanStreet 序列,PoznanHall 序列的 ΔT 提高了 2.11%, $BDBR$ 增加 0.7%,PoznanStreet 序列的 ΔT 基本保持持平,但 $BDBR$ 降低了 0.7%;对于纹理简单的 UndoDancer 序列,其 ΔT 的提升高达 25.37%,且 $BDBR$ 降低了 0.1%。此外,为了更全面地评估本文算法的编码性能,本文选取了包括 PU 模式决策和 DMM 模式两个优化方法的文献[13]的方法进行比较。相比文献[13]的方法,本文方法测试序列的 ΔT 平均提高了 4.47%,平均 $BDBR$ 仅增加了 0.2%;小分辨率序列的平均 ΔT 略小,其中纹理复杂的 Balloons 和 Newspaper 序列的 $BDBR$ 分别提升了 0.3% 和 0.1%;大分辨率序列的平均 ΔT 提高了 10.4%,而平均 $BDBR$ 仅增加了 0.4%。上述比较说明:本文算法充分考虑了深度图像本身的纹理特征以及影响合成视点图像质量的关键因素,通过 CTU 级别的初级纹理分析适应深度图像大面积平坦区域的纹理,通过 CU 级别的精细纹理分析适应边界区域像素变化锐利的纹理,使得本文算法不仅具有较好的纹理自适应性,而且在保持编码质量与现有优秀算法一致的同时,大幅降低了大分辨率序列的编码时间。

结束语 为了降低 3D-HEVC 深度图像编码的计算复杂度,本文提出了一种基于纹理划分特征的深度图像帧内编码单元快速划分算法,包括 CTU 级别的初级纹理特征分析、CU

级别的精细纹理特征分析以及利用纹理特征预测 CTU 划分深度范围及提前终止 CU 划分。实验结果表明,与 3D-HEVC 的参考模型中的原始算法相比,本文算法降低了 45%左右的编码时间,同时保持了良好的编码率失真性能。与文献[8]、文献[10]和文献[13]中的方法相比,本文算法进一步降低了 12%、3%和 4%的编码时间,特别是针对大分辨率序列,分别降低了 14%、11%和 10%的编码时间,并具有相近的编码率失真性能。

参 考 文 献

- [1] UREY H, CHELLAPPAN U H, ERDEN E, et al. State of the Art in Stereoscopic and Autostereoscopic Displays[J]. *Proceedings of the IEEE*, 2011, 99(4): 540-555.
- [2] MÜLLER K, SCHWAR 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.
- [3] MÜLLER K, SCHWAR H, MARPE D, et al. Overview of the Multiview and 3D Extensions of High Efficiency Video Coding [J]. *IEEE Transactions on Image Processing*, 2016, 26(1): 35-49.
- [4] SULLIWAN 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.
- [5] KAUFF P, ATZPADIN N, FEHN C, et al. Depth Map Creation and Image-based Rendering for Advanced 3DTV Services Providing Interoperability and Scalability [J]. *Signal Processing: Image Communication*, 2007, 22(2): 217-234.
- [6] CEN Y F, WANG W L, YAO X W, et al. Decision Tree Based Coding Unit Splitting Algorithm for HEVC[J]. *Computer Science*, 2016, 43(4): 308-312. (in Chinese)
岑跃峰, 王万良, 姚信威, 等. 基于决策树的 HEVC 编码单元划分算法[J]. *计算机科学*, 2016, 43(4): 308-312.
- [7] LIU S, PENG Z J, CHEN J L, et al. Multi-class Support Vector Machine-based Fast Algorithm for 3D-HEVC Depth Video Intra Coding[J]. *Journal on Communications*, 2016, 37(11): 181-188. (in Chinese)
刘晟, 彭宗举, 陈嘉丽, 等. 基于多类支持向量机的 3D-HEVC 深度视频帧内编码快速算法[J]. *通信学报*, 2016, 37(11): 181-188.
- [8] PENG K K, CHIANG J C, LIE W N. Low Complexity Depth Intra Coding Combining Fast Intra Mode and Fast CU Size Decision in 3D-HEVC[C]//*IEEE International Conference on Image Processing (ICIP)*. Phoenix: IEEE Press, 2016: 1126-1130.
- [9] 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.
- [10] ZHANG H B, CHAN Y L, FU C H, et al. Quadtree Decision for Depth Intra Coding in 3D-HEVC by Good Feature[C]//*IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Shanghai: IEEE Press, 2016: 1481-1485.
- [11] SILVA T L, AGOSTINI L V, CRUZ L A S. Fast Intra Prediction Algorithm Based on Texture Analysis for 3D-HEVC Encoders[J]. *Journal of Real-Time Image Processing*, 2016, 12(2): 357-368.
- [12] TOHIDYPOUR H R, POURAZAD M T, NASIOPOULOS P. Online-Learning-Based Complexity Reduction Scheme for 3D-HEVC[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016, 26(10): 1870-1883.
- [13] ZHANG H B, FU C H, CHAN Y L, et al. Probability-Based Depth Intra-Mode Skipping Strategy and Novel VSO Metric for DMM Decision in 3D-HEVC[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, 28(2): 513-527.
- [14] GUO R X HE G, LI Y S, et al. Fast Algorithm for Prediction Unit and Mode Decisions of Intra Depth Coding in 3D-HEVC [C]//*IEEE International Conference on Image Processing (ICIP)*. Phoenix: IEEE Press, 2016: 1121-1125.
- [15] FU C H, ZHAO Y W, ZHANG H B, et al. Depth Modelling Mode Decision for Depth Intra Coding via Good Feature[C]//*IEEE International Conference on Image Processing (ICIP)*. Beijing: IEEE Press, 2017: 4018-4022.
- [16] JABALLAH S, LARABI M C, TAHAR J B. Heuristic Inspired Search Method for Fast Wedgelet Pattern Decision in 3D-HEVC [C]//*European Workshop on Visual Information Processing (EUVIP)*. Marseille: IEEE Press, 2016: 1-6.
- [17] LEI J J, SUN Z Y, GU Z Y, et al. Simplified Search Algorithm for Explicit Wedgelet Signalization Mode in 3D-HEVC [C]//*IEEE International Conference on Multimedia and Expo (ICME)*. Dhaka: IEEE Press, 2017: 805-810.
- [18] OHTSU N. A Threshold Selection Method from Gray-Level Histograms[J]. *IEEE Transactions on systems, Man, and Cybernetics*, 1979, 9(1): 62-66.
- [19] MÜLLER K, VETRO A. Common Test Conditions of 3DV Core Experiments[C]//*ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 11th Meeting*. Geneva, 2014: 1-7.
- [20] BJONTEGAARD G. Calculation of Average PSNR Differences between RD Curves [C]//*Proceedings of the VCEG-M33*. Austin, 2001: 1-4.