文章编号：1003-0530(2022)10-2180-12

纹理-深度图共同优化的3D-HEVC帧内快速算法

栗晨阳 陈 婧
（华侨大学信息科学与工程学院，厦门市移动多媒体通信重点实验室，福建厦门 361021）

摘 要： 随着立体及3D视频需求的日益增多，针对3D视频编码方法的研究受到越来越多的关注。3D-HEVC编码标准对采用纹理和深度图格式融合的3D视频进行编码，由于加入了深度图编码，因此新增了深度图编码模式、组件间预测和分段直流编码等技术，使其编码复杂度急剧升高。为了减少3D-HEVC的编码时间，本文提出了针对纹理图和深度图的编码单元（Coding Unit，CU）尺寸提前决策快速算法。利用梯度矩阵和作为当前CU和子CU复杂度的判断依据，将CU分为三类：不划分CU（Non-Split Coding Unit，NSCU）、直接划分CU（Split Coding Unit，SCU）以及普通CU。对NSCU，跳过小尺寸的帧内预测过程；对SCU，直接跳过当前CU的帧内预测过程；对普通CU，执行原平台操作。实验结果表明，与原始平台相比，本文算法在合成视点质量基本不变的情况下，平均减少40.92%的编码时间；与最新的联合纹理-深度图优化的3D-HEVC快速算法相比，可以在质量相当的情况下减少更多的编码时间。

关键词： 3D-HEVC；帧内预测；提前CU决策；快速算法
中图分类号： TN919.81 **文献标识码：** A **DOI：** 10.16798/j.issn.1003-0530.2022.10.019

引用格式： 栗晨阳，陈婧. 纹理-深度图共同优化的3D-HEVC帧内快速算法[J]. 信号处理，2022，38(10)：2180-2191. DOI：10.16798/j.issn.1003-0530.2022.10.019.

Reference format：LI Chenyang，CHEN Jing. A fast algorithm for texture-depth co-optimization in 3D-HEVC intra prediction[J]. Journal of Signal Processing，2022，38(10)：2180-2191. DOI: 10.16798/j.issn.1003-0530.2022.10.019.

A Fast Algorithm for Texture-Depth Co-optimization in 3D-HEVC Intra Prediction

LI Chenyang CHEN Jing
（School of Information Science and Engineering，Huaqiao University，Xiamen Key Laboratory of Mobile Multimedia Communications，Xiamen，Fujian 361021，China）

Abstract: With the increasing demand for stereo and 3D video，the research on 3D video coding has attracted more and more attentions. 3D-HEVC standard is for 3D video coding with texture and depth map format. Due to the addition of depth map coding techniques，the coding complexity increases dramatically. Therefore，a fast intra coding algorithm for both texture and depth maps in 3D-HEVC was proposed. First，CU was divided into three categories：Non-split Coding Unit（NSCU），Split Coding Unit（SCU），and ordinary CU，by using the Sum of gradient matrix（SGM）as the complexity judgment basis of current CU and Sub-CU. For NSCU，the small size of the intra-frame prediction process was skipped；for SCU，the intra-frame prediction process of the current CU was skipped directly；and for ordinary CU，the original platform operation was performed. Experimental results show that compared with the original platform，the proposed algorithm reduces the coding time by 40.92% on average when the quality of synthetic viewpoints is basically unchanged.

收稿日期：2022-01-26；修回日期：2022-04-17
基金项目：国家自然科学基金（61871434）；福建省杰出青年科学基金项目（2019J06017）

Compared with the state-of-the-art texture-depth co-optimization fast algorithms for 3D-HEVC, the outperformance of time saving is achieved with relatively equivalent quality.

Key words: 3D-HEVC; intra prediction; early CU size decision; fast algorithm

1　引言

近年来，互联网技术发展迅速，出现了越来越多与视频和多媒体行业相关的应用，由于视频传递的信息更加真实、具体、生动和全面，因此对视频信息的需求不断增加。传统的2D视频已经不能满足人们视觉感受的需求，3D视频、多视点视频（Multi View Video，MVV）[1]能够提供立体视觉感受而受到更多的关注。最初的3D视频采用的是双视点视频格式，为了增强立体视觉感受，越来越多的使用MVV视频格式。由于MVV格式的视频数据量较多，且传输和存储过程变得复杂，为了解决这些问题，提出了多视点加深度图（Multi-view Video Plus Depth，MVD）[2]的视频格式。MVD视频格式中包括来自不同视点的纹理图信息和对应的深度图信息，其余虚拟视点可用深度图渲染（Depth Image Based Rendering，DIBR）技术[3]进行合成。

高效视频编码（High Efficiency Video Coding，HEVC）[4]于2013年4月颁布，其压缩性能在上一个编码标准的基础上提高了一倍。3D-HEVC[5]编码标准在HEVC的基础上提高了立体视频的编码效率。在3D-HEVC中，相较于HEVC新增了一些其他的编码技术，比如组件间预测[6]，深度图特有的预测模式，深度模型模式（Depth Modeling Mode，DMM）和分段直流编码（Segment-wise DC Coding，SDC）等新的技术[7]，这些技术虽然提高了立体视频的编码性能，但同时也提高了编码复杂度。因此，如何在保证重建视频质量基本不变的情况下减少编码时间，是当前需要解决的问题。

在目前的3D-HEVC帧内快速算法中，主要有两种方式减少编码复杂度，一种是提前判断最优编码单元的分割深度。文献[8]提出了一种基于深度图边缘分类卷积神经网络（Depth Edge Classification Convolutional Neural Network，DEC-CNN）的帧内快速算法，通过建立的数据库训练此网络用于CU分类，并将网络嵌入到3D-HEVC测试平台中，针对不同的CU确定其深度范围，并且利用二值化深度图像的像素值对上述分类结果进行了校正，虽然减少了计算复杂度，但是由于数据集较少，导致其合成质量损失较多。文献[9]提出利用决策树判断当前CU是否划分的方法，首先获取不同尺寸CU的特征，包括当前CU的均值、方差、梯度等，利用这些特征生成决策树，并将决策树嵌入原始编码平台来判断当前CU是否需要划分。文献[10]提出一种提前终止CU划分的算法来加速编码过程，它通过计算每个CU的四条边的像素方差之和以及当前CU的率失真代价（Rate Distortion Cost，RDCost），将其分别与设定阈值比较，如果均小于设定阈值，则停止帧内预测过程，即跳过尺寸较小的CU的帧内预测过程；否则，执行原平台操作。文献[9]和[10]所提算法对分辨率较大的视频有更好的效果，对分辨率较小的视频编码性能还需进一步提升。文献[11]利用Otsu's算子计算当前CU的最大类间方差值，将CU划分为平坦CU和复杂CU，对平坦CU终止划分以及减少模式选择数量，本算法虽然对纹理复杂度较低的效果较好，但是对CU划分类型有限。文献[12]提出了一种基于梯度和分段直接分量编码（Sum of Gradient and Segment-wise Direct Component Coding，SOG-SDC）的深度图快速算法，利用Robert算子计算当前CU的梯度，并与设定的阈值比较，判断当前CU是否为平滑CU，如果是平滑CU，则停止CU划分。以上针对CU划分决策的方法主要是用在减少深度图的编码时间上，没有考虑用于同时减少纹理图的编码时间。

另一种减少编码复杂度的方法是选择性跳过某些帧内预测模式的计算和判断。文献[13]提出利用决策树判断是否使用DMM模式，首先获取当前CU的特征，包括当前CU的均值、方差，母CU的最佳预测模式以及当前CU的前两个最佳预测模式和对应的RDCost等，然后利用这些特征生成决策树，将决策树嵌入平台，判断当前CU是否需要加入DMM模式，该算法只考虑了帧内模式中消耗时间较多，但使用率较低的DMM模式。文献[14]提出了一种低复杂度帧内模式选择算法，通过计算帧内模

式复杂度(Intra Mode Complexity,IMC),将CU划分为简单CU、普通CU和复杂CU,然后对不用种类的CU选择不同的模式加入帧内预测模式候选列表,即跳过其他模式的帧内预测过程,该算法的IMC参考的CU包含了帧内和帧间的CU,但是最后的模式选择只针对了帧内的模式。文献[15]基于对模式分布的统计分析,提出了一种快速模式帧内决策算法,利用选择特定模式的概率大小,依次计算模式对应的RDCost和设定阈值比较,进而确定最佳帧内预测模式,跳过其余模式的计算,该算法有效的减少了深度图帧内编码复杂度,但是对于全局运动较大的序列质量损失较多。文献[16]提出了一种自适应地减少模式决策过程中原始候选模式数量的方法,首先计算Planar模式、DC模式以及水平和垂直模式的哈达玛变换差值的绝对值和(Sum of Absolute Hadamard Trans-formed Difference,SATD),将SATD值最小的前两个模式加入候选列表,根据模式配对表来加入或删除其余模式。以上文献都只针对深度图进行优化,文献[17-18]则针对纹理图和深度图同时优化,以减少编码复杂度。其中,文献[17]提出了一种基于梯度信息的纹理图和深度图编码的快速模式决策算法,首先利用Sobel算子计算预测单元(Prediction Unit,PU)的梯度,然后利用PU的梯度信息将PU分为三种类型,为PU选择合适的候选模式,跳过其余模式的预测过程。文献[18]则提出了一种基于局部结构张量的边缘检测算法,利用局部边缘方向直方图来识别优势边缘方向,分别对深度图和纹理图确定候选模式的选择范围,以降低3D-HEVC内编码的计算复杂度。

考虑到3D-HEVC编码复杂度较高,且包含纹理图和深度图两种视频序列,因此本文提出了一种纹理图和深度图共同优化的帧内快速算法。首先,分别计算纹理图和深度图当前编码CU的梯度矩阵和SGM(Sum of Gradient Matrix,简称SGM),将其作为复杂度判断因子。利用当前CU和子CU的SGM联合判断当前CU是否需要继续划分,进而将编码CU分为三类:直接划分CU(Split Coding Unit,简称SCU),不划分CU(Non-Split Coding Unit,简称NSCU)以及普通CU,之后对纹理图和深度图不同类型的CU采取不同的CU分割深度决策以降低编码复杂度。

本文余下部分安排如下:第2节介绍3D-HEVC帧内预测过程,包括四叉树结构,帧内预测工具,以及帧内预测过程;第3节具体介绍本文所提算法,包括复杂度判断因子的计算过程,如何利用复杂度判断因子对CU进行分类,分类优化编码的整体算法的框图;第4节是本文算法的实验结果,包括与其他文献的对比结果,主观质量的比较等;第5节是结论。

2 3D-HEVC帧内预测过程

2.1 四叉树结构

3D-HEVC采用了MVD视频格式,并且按照先编码独立视点,再编码其余非独立视点,以及先编码纹理图,再编码对应的深度图的顺序依次编码。编码结构采用如图1所示的四叉树结构。

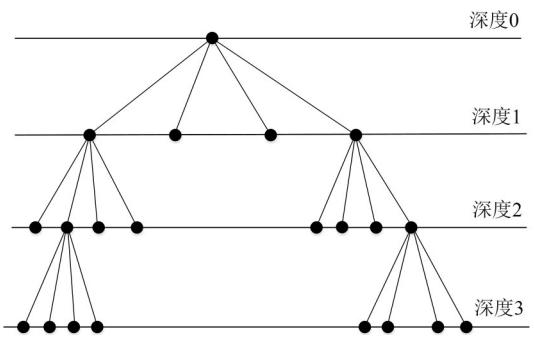


图1 四叉树结构

Fig. 1 Quadtree structure

当前编码的视频帧被划分成若干个编码树单元(Coding Tree Units,CTUs),每个CTU继续递归划分为尺寸大小依次为64×64、32×32、16×16、8×8的CU,其对应划分深度为0、1、2、3。在帧内预测过程中,从64×64到8×8尺寸的CU依次进行帧内预测过程,并且将代价最小的模式作为当前CU的最佳模式,然后从最小尺寸的CU开始,将子节点对应的四个小尺寸CU代价之和与父节点对应的母CU的代价比较,如果前者小于后者,则保留当前树形结构,如果前者大于后者,则移除小尺寸CU对应的节点。以此类推,得到最后的树形结构。

2.2 帧内预测工具

在3D-HEVC中,纹理图使用的预测模式为HEVC帧内预测模式,帧内预测编码是为了消除空间冗余信息,在这个过程中,参考CU通常是位于当

前CU的上方、左方以及左上方。为了预测结果的准确性，在之前编码标准的基础上，HEVC进行了扩展，增加了更多的预测模式，如图2所示。HEVC采用的帧内预测编码模式一共有35种，包括平面(Planar)预测模式、直流(DC)预测模式以及33个角度预测模式[19]。Planar模式常常用在渐变式的平滑纹理区域，它利用水平方向和垂直方向的线性插值的平均作为当前块像素的预测值；DC模式适用于大面积平坦区域，利用当前CU左侧和上方的参考像素的平均值得到；此外，HEVC规定了33种角度预测模式，细化了预测方向，可以更好地适应不同方向的纹理复杂度。

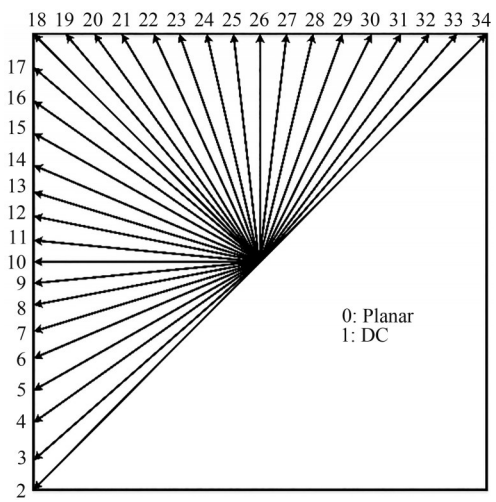
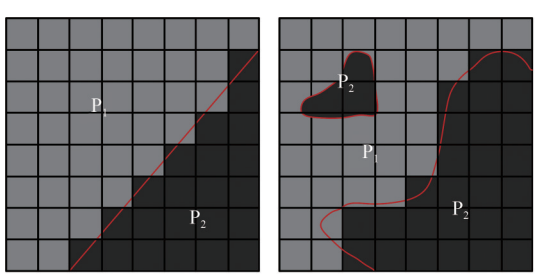


图2　HEVC帧内预测模式

Fig. 2　HEVC intra prediction mode

对于深度图，帧内预测模式加入了两种DMM模式，分别是DMM1楔形模式和DMM4轮廓模式，如图3所示。这种DMM模式将深度图CU分割成两个区域，每个区域内深度近似为常数。DMM1楔形模式是用直线分割，DMM4轮廓模式的边界划分由与该深度块对应的纹理块来决定。



(a) DMM1　　(b) DMM4

图3　深度图DMM模式

Fig. 3　Depth Modeling Mode

2.3　帧内预测过程

3D-HEVC的帧内预测过程如图4所示，首先利用SATD将模式加入候选列表，然后对模式列表里的模式计算其RDCost，最终选择最优的模式和最佳划分方式。

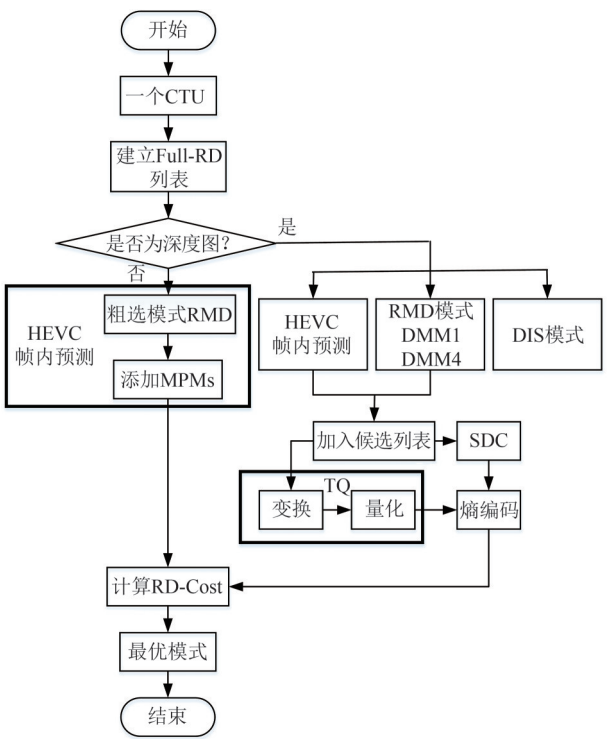


图4　3D-HEVC的帧内预测过程

Fig. 4　The intra prediction process of 3D-HEVC

纹理图帧内预测过程具体如下：

1)首先建立一个用来存储候选模式的空列表。

2)模式粗选过程，利用公式(1)计算HEVC内包含的35种模式的SATD值，然后比较SATD值大小，将较小的SATD值对应的模式加入候选列表中，加入的模式个数与当前CU的尺寸有关[20]。

$$J_{\mathrm{SATD}} = \mathrm{SATD}(\mathrm{Mode}) + \lambda \times R(\mathrm{Mode}) \quad (1)$$

3)加入最可能模式(Most Probable Modes，MPMs)，这个模式根据当前编码块的上方、左方以及左上方已经编码的CU模式，选出3种最可能的模式，加入候选列表。

4)利用公式(2)计算模式候选列表中所有模式的RDCost，选择代价最小的模式为当前CU的最佳模式。

$$J_{\mathrm{RD}} = D_{\mathrm{mode}} + \lambda \times B_{\mathrm{mode}} \quad (2)$$

深度图帧内预测过程如下：

首先对深度图CU进行HEVC帧内预测过程，包括模式粗选过程和MPMs过程，之后将深度图独有的DMMs模式加入候选列表，对列表里所有模式进行变换、量化（Transform and Quantization，TQ）以及SDC，之后对其进行熵编码，对深度帧内跳跃模式（Depth Intra Skip，DIS）则直接进行熵编码。最后通过RDCost的计算和比较，选择代价最小的模式为当前CU的最佳模式[21]。

3 基于复杂度判断因子的提前CU尺寸决策算法

由上一节帧内预测过程可以知道，在3D-HEVC中，无论是纹理图还是深度图，在预测过程中至多遍历85个CU的编码过程，包括1个64×64尺寸的CU，4个32×32尺寸的CU，16个16×16尺寸的CU和64个8×8尺寸的CU。由于每个CU都需要遍历所有模式，因此3D-HEVC的复杂度急剧升高。

为了分析3D-HEVC最佳CU尺寸分布，表1统计了四个序列纹理图和深度图的CU深度级别（CU Depth Level，CUDL），分别是分辨率为1024 × 768的Balloons和Newspaper序列、分辨率为1920 × 1088的Shark和Undo_Dancer序列，编码平台为HTM-16.0，采用全帧内配置，量化参数（Quantification Parameter，QP）设置为：(25,34)，(30,39)，(35,42)，(40,45)。从表1可以看出，纹理图选择大尺寸CU的概率较小，而深度图选择大尺寸CU的概率较大，因此，如果能够利用复杂度判断因子提前决定纹理图和深度图CU是否需要划分，则可以减少编码复杂度。

表1 纹理图和深度图CU划分深度分布

Tab. 1 The distribution of CU partition depth of texture and depth map

T/D	序列	CUDL=0	CUDL=1	CUDL=2	CUDL=3
纹理图	Balloons	16.76%	35.23%	30.23%	17.78%
	Newspaper1	8.46%	37.88%	32.23%	21.43%
	Shark	32.59%	29.68%	20.93%	16.80%
	Undo_Dancer	20.18%	32.85%	24.01%	22.96%
深度图	Balloons	45.90%	29.59%	15.72%	8.79%
	Newspaper1	36.73%	34.74%	18.84%	9.69%
	Shark	56.03%	28.37%	10.37%	5.24%
	Undo_Dancer	67.40%	22.27%	7.82%	2.51%

3.1 复杂度判断因子计算

本文利用梯度矩阵和（SGM）计算CU的复杂度判断因子。图5为一个3×3纹理单元，利用四个方向上的像素差值的绝对值之和计算中心像素点的梯度，四个方向分别是水平、对角以及垂直方向，其计算公式如下：

$I(x-1,y-1)$	$I(x,y-1)$	$I(x+1,y-1)$
$I(x-1,y)$	$I(x,y)$	$I(x+1,y)$
$I(x-1,y+1)$	$I(x,y+1)$	$I(x+1,y+1)$

图5 3×3纹理单元

Fig. 5 3×3 texture unit

$$P_1(x,y)=\left|I(x-1,y)-I(x+1,y)\right| \tag{3}$$

$$P_2(x,y)=\left|I(x,y-1)-I(x,y+1)\right| \tag{4}$$

$$P_3(x,y)=\left|I(x+1,y-1)-I(x-1,y+1)\right| \tag{5}$$

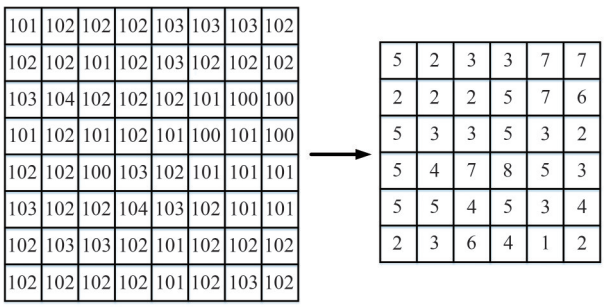
$$P_4(x,y)=\left|I(x-1,y-1)-I(x+1,y+1)\right| \tag{6}$$

其中$P_1(x,y)$，$P_2(x,y)$，$P_3(x,y)$和$P_4(x,y)$分别为中心像素点$I(x,y)$四个方向（0°，90°，45°，145°）的梯度绝对值，四个方向梯度和表示如下：

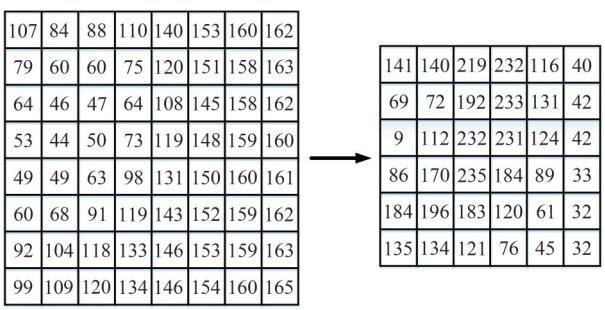
$$P(m,n)=\sum_{i=1}^{4}P_i(m,n) \tag{7}$$

式中，$P(m,n)$是四个方向的像素差值的绝对值之和，即为当前像素点的梯度和。其中(m,n)为纹理单元中心像素点的位置，$m,n\in(1,N-2)$，因为当前CU的差分矩阵需要计算四个方向的像素值差，则无法获取CU外层像素点的梯度信息，因此$N\times N$的CU对应的差分矩阵大小为$(N-2)\times(N-2)$，如图6和图7所示，8 × 8大小的CU，对应的差分矩阵大小为6 × 6。

图6（a）、（b）为纹理图简单CU和复杂CU及其对应的差分矩阵，图7（a）、（b）为深度图简单CU和复杂CU及其对应的差分矩阵。可以看出，差分矩



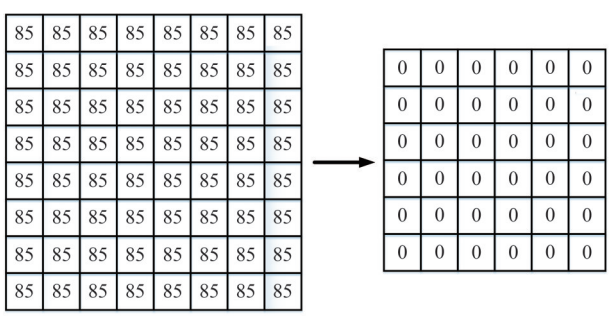
(a) 纹理图简单CU及其差分矩阵
(a) Texture simple CU and its difference matrix



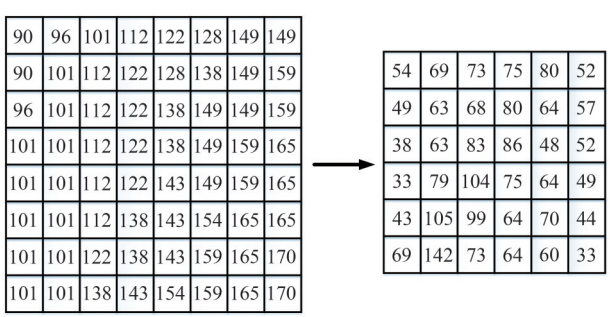
(b) 纹理图复杂CU及其差分矩阵
(b) Texture complex CU and its difference matrix

图6　纹理图CU及其差分矩阵

Fig. 6　Texture CU and its difference matrix



(a) 深度图简单CU及其差分矩阵
(a) Depth simple CU and its difference matrix



(b) 深度图复杂CU及其差分矩阵
(b) Depth complex CU and its difference matrix

图7　深度图CU及其差分矩阵

Fig. 7　Depth CU and its difference matrix

阵可以很好的反应CU的复杂程度。因此，SGM可作为CU的复杂度判断因子，计算公式如下：

$$C_0 = \sum_{n=1}^{N-2}\sum_{m=1}^{N-2} P(m,n) \tag{8}$$

其中$P(m,n)$为差分矩阵内的元素，N为当前CU宽度。同时也可以根据子CU的复杂度判断因子联合判断当前CU的复杂度，更好地判断当前CU是否需要划分。公式(9)为当前CU进行四叉树划分后第一个子CU的复杂度判断因子的计算方法，其余子CU的复杂度判断因子C_2、C_3、C_4同理可得。

$$C_1 = \sum_{n=1}^{(N-2)/2}\sum_{m=1}^{(N-2)/2} P(m,n) \tag{9}$$

3.2　基于复杂度判断因子的CU分割深度提前决策判断

针对纹理图和深度图，本文利用当前CU和子CU的SGM联合判断其复杂度，利用复杂度判断因子的值所在范围将CU划分为NSCU、SCU以及普通CU。NSCU代表纹理不复杂的CU，即跳过小尺寸CU的帧内预测过程；SCU表示纹理复杂的CU，即跳过当前CU的帧内预测过程，直接划分，进行小尺寸CU的判断；其余则是普通CU，执行原平台操作。

3.2.1　NSCU提前决策

对于纹理图和深度图，如果当前CU复杂度判断因子的值较小，则直接判断为NSCU，如果当前CU的复杂度判断因子的值较大，但子CU的复杂度判断因子均小于母CU的复杂度判断因子的一半，依然可以判定当前CU为NSCU，具体算法如下：

当前CU复杂度判断因子判断：

$$0 \leqslant C_0 < \mathrm{TH}_{N_i} \tag{10}$$

利用子CU复杂度判断因子联合判断：

$$\begin{cases} \mathrm{TH}_{N_i} < C_0 < \mathrm{ave}_{N_i} \\ (C_1 \& C_2 \& C_3 \& C_4) < 0.5 \times C_0 \end{cases} \tag{11}$$

其中，C_0、C_1、C_2、C_3、C_4为当前CU和其子CU的复杂度判断因子，TH_{N_i}为NSCU的阈值，i表示CU深度，ave_{N_i}为训练帧中被判断为NSCU的复杂度判断因子的均值，训练帧为每个序列的第1帧，对于训练帧则直接使用原平台算法，考虑到单个视频序列的相关性，每30帧更新一次阈值，即每编码30帧后重新计算阈值。

$$TH_{N_i} = \alpha \times ave_{N_i} \tag{12}$$

由公式(12)可知,阈值的设定与复杂度判断因子均值及系数α相关,均值每30帧更新一次,因此确定α的值也很重要,如果α较大,则阈值较大,则判断为NSCU的可能性变大。由表1可知,纹理图选择大尺寸CU的概率小于深度图选择大尺寸CU的概率,因此判断纹理图的NSCU范围要小于判断深度图的NSCU范围,因此纹理图α的取值小于深度图的α。对于纹理图,α分别选取0.4、0.6、0.8进行实验,实验结果如图8所示,其中,S/T(Synth PSNR/Total Bitrate)表示合成视点质量的变化,其值越小代表损失越少,从图中可以看出,随着α的增大,时间减少增多,但合成视点质量损失有所改变。考虑到时间和质量的平衡,对于纹理图,选择α = 0.6。对于深度图,设α = 0.8[22]。

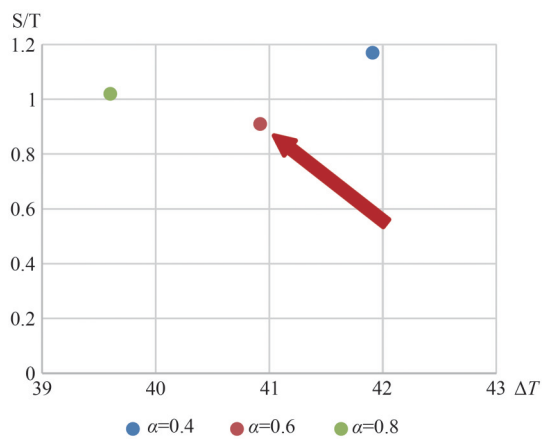


图8 纹理图中选取参数α的实验结果

Fig. 8 Select α experimental result in texture map

3.2.2 SCU提前决策

对于纹理图和深度图,如果当前CU复杂度判断因子的值较大,则直接判断为SCU,如果当前CU的复杂度判断因子的值在某一个范围内,且某个子CU的复杂度判断因子大于母CU的复杂度判断因子的一半,依然可以判定当前CU为SCU,具体算法如下:

当前CU复杂度判断因子判断:

$$C_0 \geqslant TH_{S1_i} \tag{13}$$

利用子CU复杂度判断因子联合判断:

$$\begin{cases} TH_{S1_i} \geqslant C_0 \geqslant TH_{S2_i} \\ (C_1 \| C_2 \| C_3 \| C_4) > 0.5 \times C_0 \end{cases} \tag{14}$$

其中TH_{S1_i}和TH_{S2_i}是SCU的两个阈值,定义如下:

$$TH_{Sn_i} = \beta \times ave_{S_i} \tag{15}$$

其中ave_{S_i}为训练帧中判断为SCU的复杂度判断因子均值。考虑到准确率以及质量,经过实验,TH_{S1_i}应尽可能大,在纹理图和深度图编码时,β均设为2.0,而TH_{S2_i}应尽可能小,同时也要保证复杂度判断因子的覆盖范围,因此,纹理图和深度图中β = 1.5。

3.2.3 深度图DMM模式跳过

由于DMM模式针对的是复杂度较高的CU,且DMM模式计算复杂度较高,因此,对于深度图CU,如果判断其为NSCU,则不仅停止CU的划分,在当前CU的帧内预测过程中选择跳过DMM模式的判断,可以进一步减少编码复杂度。

3.2.4 算法流程图及实现

本算法的总体流程图如图9所示,具体步骤如下:

1)首先初始化纹理图和深度图的CU深度。

2)判断是否为训练帧,如果为训练帧,则计算NSCU和SCU的阈值,否则执行步骤3)。

3)利用公式(3)~(9)计算纹理图和深度图当前CU及其子CU的SGM,作为其复杂度判断因子。

4)利用公式(10)~(11)判断是否为NSCU,如果是NSCU则跳过小尺寸CU的帧内预测过程,并且执行步骤8),否则到步骤5);如果是深度图的NSCU,则还需跳过DMM模式的检测。

5)利用公式(13)~(14)判断是否为SCU,是则执行步骤7),跳过当前CU的帧内预测过程;否则为普通CU,执行步骤6)。

6)对当前CU进行原平台的帧内预测过程。

7)如果当前CU深度为3,则执行步骤8);否则,CU分割深度加一。

8)选择最佳模式以及CU划分。

4 实验结果及分析

4.1 实验配置

本次实验采用HTM-16.0编码平台,全帧内配置,输入视点个数为3,量化参数QP设置为:(25,34),(30,39),(35,42),(40,45),测试序列如表2所示,测试环境为英特尔Core i9-9900K@3.60GHz CPU、16G内存,Window10(64位)操作系统。

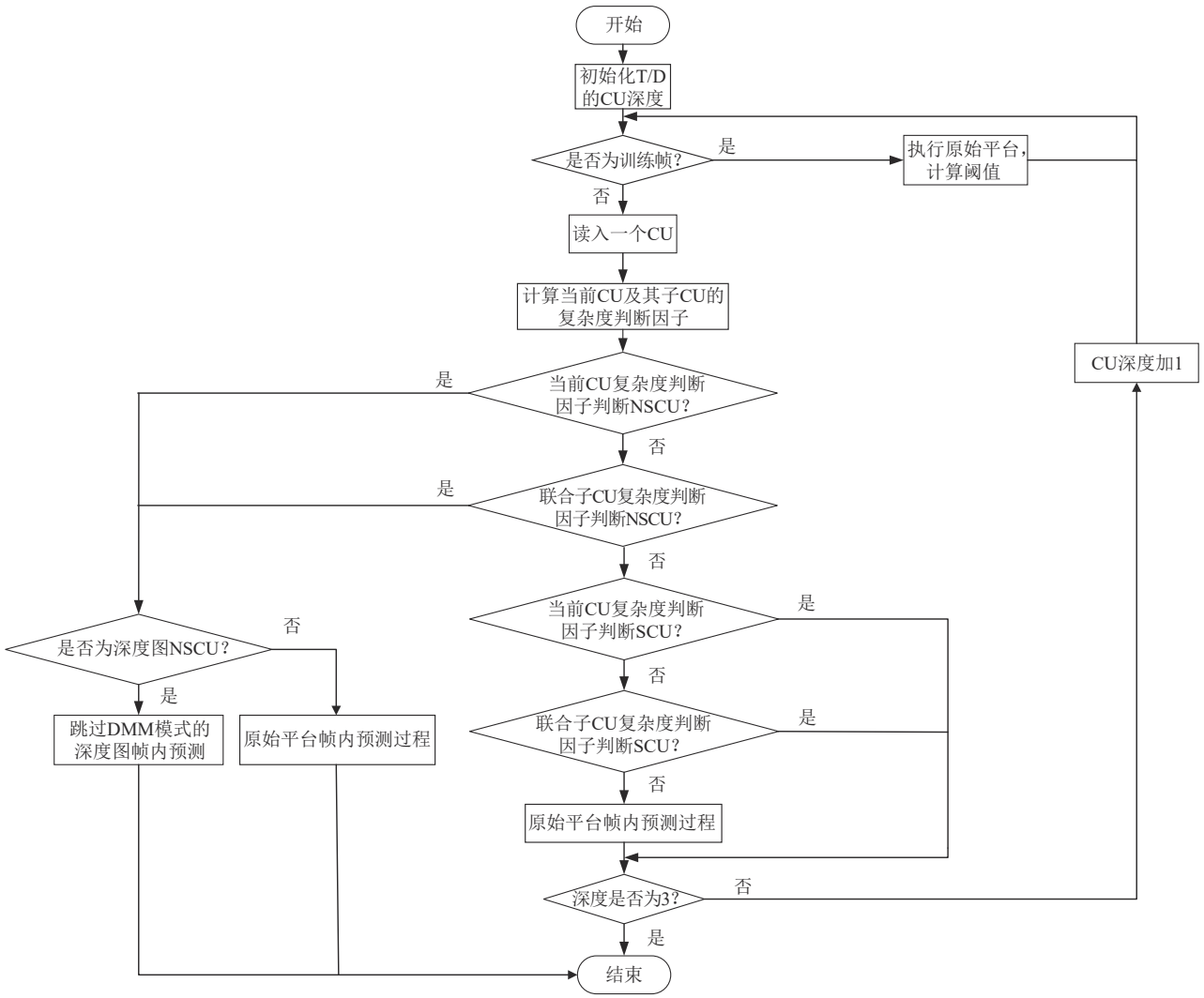


图9　所提算法流程图

Fig. 9　Flowchart of the proposed algorithm

表2　测试序列

Tab. 2　Test sequences

测试序列	分辨率	帧数	视点
Balloons	1024×768	300	1-3-5
Kendo	1024×768	300	1-3-5
Newspaper1	1024×768	300	2-4-6
GT-Fly	1920×1088	250	9-5-1
Poznan_Hall2	1920×1088	250	7-6-5
Poznan_Street	1920×1088	250	5-4-3
Shark	1920×1088	300	1-5-9
Undo_Dancer	1920×1088	250	1-5-9

4.2　实验结果比较

表3显示了本文所提算法与原始平台HTM-16.0编码性能以及编码时间的比较。其中，V/T

表3　本算法与实验平台HTM-16.0实验结果比较

Tab. 3　Comparison between the proposed and the standard platform HTM-16.0

测试序列	V/T(%)	S/T(%)	ΔT(%)
Balloons	0.5	0.66	33
Kendo	0.54	0.8	38.72
Newspaper1	0.55	1.22	35.4
1024×768	0.53	0.89	35.71
GT-Fly	0.21	0.37	48.15
Poznan_Hall2	0.54	1.89	48.84
Poznan_Street	0.7	0.72	35
Shark	0.37	0.62	45.85
Undo_Dancer	0.85	0.98	42.41
1920×1088	0.54	0.92	44.05
Average	**0.53**	**0.91**	**40.92**

(Video PSNR/Total Bitrate)表示全部视频的比特率占比变化,越接近0则编码性能越好;S/T表示合成视点质量的变化,越小越好;ΔT为所提算法相比于原始平台算法节省时间,节省的编码时间计算方法如下:

$$\Delta T = \frac{T_{D\text{-ori}} - T_{D\text{-pro}}}{T_{D\text{-ori}}} \times 100\% \tag{16}$$

其中$T_{D\text{-ori}}$为原始平台所用编码时间,$T_{D\text{-pro}}$为所提算法所用编码时间。

由表3可以看出,相较于原始平台所需编码时间,本文所提算法在合成视点质量平均减少0.91%的情况下,平均减少了40.92%的编码时间。从表中可以看出,对于局部运动或者全局运动频繁的视频序列,本文算法效果较好,如序列"GT-Fly"和"Shark"分别只产生了0.37%和0.62%的质量损失,但其编码时间平均减少了48.15%和45.85%;对于全局运动较少或者具有大量平坦区域的视频序列,如"Poznan_Hall2"序列,合成视点质量平均虽减少了1.89%,但该序列的编码时间平均减少了48.84%。另外,从表3可以看出,本文算法对于分辨率较大的视频序列效果较好,可以减少较多编码时间,而对于分辨率较小的视频质量损失较少。表4展示了不同QP对编码时间的影响。从表4中可以看出,对于纹理图和深度图,不同的编码QP结果不同,QP越大,节省时间越多,主要是因为QP越大,在编码过程中,细节部分编码较少,训练帧中被判断为NSCU的概率较大,使得编码帧中判断为简单CU的阈值变大,故而判断为NSCU的概率较大,进而可以减少较多编码时间。表5显示了本文算法分别针对纹理图和深度图的实验结果比较,从表5可以看出,本文算法平均减少43.7%的深度图编码时间以及45.1%的纹理图编码时间,对于合成视点质量,本文算法针对深度图的质量效果要好于纹理图的效果。

表4 本文算法在不同QP情况下的时间减少比较

Tab. 4 Time reduction comparison of the proposed algorithm under different QPs

测试序列 \ QP	ΔT(%)			
	25	30	35	40
Balloons	26.83	32.57	35.53	36.64
Kendo	35.34	32.66	41.18	44.94
Newspaper1	30.61	34.45	36.44	39.78
GT-Fly	31.16	49.23	54.11	54.96
Poznan_Hall2	41.46	46.19	52.18	54.51
Poznan_Street	27.12	34.28	36.06	41.72
Shark	32.22	41.71	45.58	48.86
Undo_Dancer	37.55	37.98	48.4	56.97
Average	**32.79**	**38.63**	**43.69**	**47.29**

表5 纹理图和深度图编码时间与实验平台HTM-16.0的比较结果

Tab. 5 Comparison of time consuming between the proposed algorithm and the HTM-16.0

测试序列	纹理图			深度图		
	V/T (%)	S/T (%)	ΔT (%)	V/T (%)	S/T (%)	ΔT (%)
Balloons	0.85	0.79	44.3	−0.04	0.22	34.0
Kendo	1.97	1.91	50.2	0.07	0.43	39.6
Newspaper1	0.80	0.93	42.0	0.03	0.68	36.2
1024×768	1.20	1.21	45.5	0.00	0.45	36.6
GT-Fly	0.91	1.03	48.0	0.05	0.17	53.7
Poznan_Hall2	1.08	0.90	57.0	0.07	1.40	51.7
Poznan_Street	1.32	1.25	42.6	0.09	0.29	37.4
Shark	1.60	1.68	35.4	0.01	0.27	51.2
Undo_Dancer	1.41	1.52	41.2	0.07	0.18	45.7
1920×1088	1.26	1.28	44.9	−0.01	0.46	47.9
Average	**1.24**	**1.25**	**45.1**	**0.00**	**0.46**	**43.7**

为了观察本文算法对3D-HEVC编码性能的主观质量影响,选取"Kendo"序列合成视点1的第10帧图像和"Shark"序列合成视点4的第20帧图像作为主观实验结果比较,分别如图10和图11所示。比较了两处细节相对丰富的区域,图10(a)和图11(a)是从原始算法编码后的合成视点中选取,图10(b)和图11(b)是从本算法编码后的合成视点中选取。主观对比结果显示,本算法的合成视点主观质量可以获得和HTM-16.0平台相当的重建效果。

选取同是针对纹理图和深度图联合优化的快速算法[17]和[18]与本算法进行性能比较,结果如表6所列。从表6可以看出,本文所提算法相较于文献[17]、[18]在虚拟合成视点质量损失0.25%和提升0.08%的情况下分别减少了10.32%和9.01%的编码时间。

(a) 原HTM-16.0平台
(a) The original HTM-16.0 platform

(b) 所提算法
(b) The proposed algorithm

图10　所提算法与HTM-16.0平台主观结果比较（Kendo视频序列，合成视点1，第10帧）

Fig. 10　Subjective comparison between the proposed algorithm and HTM-16.0 (Kendo video sequence, synthetic viewpoint 1, 10th frame)

(a) 原HTM-16.0平台
(a) The original HTM-16.0 platform

(b) 所提算法
(b) The proposed algorithm

图11　所提算法与HTM-16.0平台主观结果比较（Shark视频序列，合成视点4，第20帧）

Fig. 11　Subjective comparison between the proposed algorithm and HTM-16.0 (Shark video sequence, synthetic viewpoint 4, 20th frame)

表6　所提算法与文献[17]、[18]的实验结果比较

Tab. 6　Experimental results comparison between the proposed algorithm and [17]、[18]

测试序列	[17]		[18]		本文算法	
	S/T(%)	Δ*T*(%)	S/T(%)	Δ*T*(%)	S/T(%)	Δ*T*(%)
Balloons	0.55	30.80	0.08	30.80	0.66	33.00
Kendo	0.82	33.20	0.06	34.53	0.80	38.72
Newspaper1	0.51	23.70	0.45	23.69	1.22	35.40
1024×768	0.63	29.20	0.20	29.68	0.89	35.71
GT-Fly	—	—	1.50	35.00	0.37	48.15
Poznan_Hall2	0.86	37.80	2.27	40.42	1.89	48.84
Poznan_Street	0.73	23.20	1.44	24.23	0.72	35.00
Shark	0.42	35.60	1.56	31.12	0.62	45.85
Undo_Dancer	0.70	30.10	0.58	35.46	0.98	42.41
1920×1088	0.68	31.70	1.47	33.24	0.92	44.05
Average	0.66	30.60	0.99	31.91	**0.91**	**40.92**

5 结论

为了减少3D-HEVC的编码时间,本文提出了联合纹理-深度的提前CU分割深度决策的快速算法。通过计算纹理图和深度图编码CU的复杂度判断因子SGM将其划分为NSCU和SCU以及普通CU,对NSCU跳过小尺寸的CU帧内预测过程,对SCU则直接划分,跳过当前CU的帧内预测过程。实验结果表明,与原始平台相比,本算法在合成视点质量基本不变的情况下平均减少40.92%的编码时间,与最新的联合纹理-深度3D-HEVC帧内快速算法比较,本算法在节省更多编码时间的情况下更好地保证了合成视点质量。

参考文献

[1] AHMAD I. Multi-view video: Get ready for next-generation television[J]. IEEE Distributed Systems Online, 2007, 8(3): 6.

[2] MERKLE P, SMOLIC A, MULLER K, et al. Multi-view video plus depth representation and coding[C]//2007 IEEE International Conference on Image Processing. San Antonio, TX, USA. IEEE, 2007: I-201.

[3] BOSC E, PEPION R, LE CALLET P, et al. Towards a new quality metric for 3-D synthesized view assessment[J]. IEEE Journal of Selected Topics in Signal Processing, 2011, 5(7): 1332-1343.

[4] 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.

[5] TECH G, CHEN Ying, MÜLLER K, et al. Overview of the multiview and 3D extensions of high efficiency video coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 26(1): 35-49.

[6] YAN Ye, LI Houqiang, HANNUKSELA M M. Multiview-video-plus-depth coding and inter-component prediction in high-level-syntax extension of H.265/HEVC[C]//2013 Picture Coding Symposium (PCS). San Jose, CA, USA. IEEE, 2013: 406-409.

[7] MERKLE P, MÜLLER K, WIEGAND T. Coding of depth signals for 3D video using wedgelet block segmentation with residual adaptation[C]//2013 IEEE International Conference on Multimedia and Expo. San Jose, CA, USA. IEEE, 2013: 1-6.

[8] LIU Chang, JIA Kebin, LIU Pengyu. Fast depth intra coding based on depth edge classification network in 3D-HEVC[J]. IEEE Transactions on Broadcasting, 2021, 68(1): 97-109.

[9] SALDANHA M, SANCHEZ G, MARCON C, et al. Fast 3D-HEVC depth map encoding using machine learning[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(3): 850-861.

[10] HSU Y C, LIN Jieru, CHEN Meijuan, et al. Acceleration of depth intra coding for 3D-HEVC by efficient early termination algorithm[C]// 2018 IEEE Asia Pacific Conference on Circuits and Systems. Chengdu, China. IEEE, 2018: 127-130.

[11] 韩雪, 冯桂, 曹海燕. 3D-HEVC深度图帧内编码快速算法[J]. 信号处理, 2018, 34(6): 680-687.
HAN Xue, FENG Gui, CAO Haiyan. Efficient fast algorithm for depth map in 3D-HEVC[J]. Journal of Signal Processing, 2018, 34(6): 680-687. (in Chinese)

[12] NIAN Chunmei, CHEN Jing, ZENG Huanqiang, et al. A fast intra depth map algorithm based on sum-of-gradient and segment-wise direct component coding[C]// 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS). Xiamen, China. IEEE, 2017: 191-195.

[13] MOURA C, SALDANHA M, SANCHEZ G, et al. Fast intra mode decision for 3D-HEVC depth map coding using decision trees[C]//2020 27th IEEE International Conference on Electronics, Circuits and Systems. Glasgow, UK. IEEE, 2020: 1-4.

[14] SHEN Liquan, LI Kai, FENG Guorui, et al. Efficient intra mode selection for depth-map coding utilizing spatio-temporal, inter-component and inter-view correlations in 3D-HEVC[J]. IEEE Transactions on Image Processing: a Publication of the IEEE Signal Processing Society, 2018, 27(9): 4195-4206.

[15] CHIANG J C, PENG Kuankai, WU Chaochun, et al. Fast intra mode decision and fast CU size decision for depth video coding in 3D-HEVC[J]. Signal Processing: Image Communication, 2019, 71: 13-23.

[16] LEE J Y. Fast depth intra mode decision based on mode analysis in 3D video coding[J]. Electronics, 2019, 8

(4)：430.

[17] ZHANG Qian, JING Ruihai, WANG Bin, et al. Fast mode decision based on gradient information in 3D-HEVC [J]. IEEE Access, 2019, 7: 135448-135456.

[18] HAMOUT H, ELYOUSFI A. An efficient edge detection algorithm for fast intra-coding for 3D video extension of HEVC [J]. Journal of Real-Time Image Processing, 2019, 16(6): 2093-2105.

[19] LAINEMA J, BOSSEN F, HAN W J, et al. Intra coding of the HEVC standard[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1792-1801.

[20] 朱树明, 王凤随, 程海鹰. HEVC压缩域的视频摘要关键帧提取方法[J]. 信号处理, 2019, 35(3): 481-489.
ZHU Shuming, WANG Fengsui, CHENG Haiying. Video summarization key frame extraction method for HEVC compressed domain [J]. Journal of Signal Processing, 2019, 35(3): 481-489. (in Chinese)

[21] SANCHEZ G, SILVEIRA J, AGOSTINI L V, et al. Performance analysis of depth intra-coding in 3D-HEVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 29(8): 2509-2520.

[22] ZUO Jiabao, CHEN Jing, ZENG Huanqiang, et al. Bi-layer texture discriminant fast depth intra coding for 3D-HEVC[J]. IEEE Access, 2019, 7: 34265-34274.

作者简介

栗晨阳　女，1996年生，河南濮阳人。华侨大学信息科学与工程学院硕士研究生，主要研究方向为图像和视频处理。
E-mail：764179832@qq. com

陈　婧　女，1980年生，福建厦门人。华侨大学信息科学与工程学院副教授，博士，主要研究方向为图像和视频处理。
E-mail：chengjing8005@hqu. edu. cn