DOI:10.16136/j.joel.2023.02.0218

一种基于子块纹理差异的 VVC 快速 CU 划分算法

李 亚[1,2],李 强[1,2*],孟 慧[1,2]

(1.重庆邮电大学 通信与信息工程学院,重庆 400065;2.信号与信息处理重庆市重点实验室,重庆 400065)

摘要:针对多功能视频编码(versatile video coding,VVC)由于在四叉树(quadtree,QT)划分的基础上引入多叉树(multi-type tree,MTT)划分结构导致编码复杂度过高的问题,本文提出了一种基于编码单元(coding unit,CU)子块纹理特征差异的 VVC 快速 CU 划分算法。该算法能够有效降低 VVC 的编码复杂度,缩短编码时间。首先通过不同划分方向的子块的纹理复杂度差异提前判断 MTT 划分的方向,跳过不必要的 MTT 划分方向;然后根据当前 CU 的相邻子块间的纹理差异判断是否跳过三叉树(ternary tree,TT)划分,以进一步减少候选列表中划分模式的数量。实验结果表明,在全帧内(all intra,AI)的编码配置下,与官方测试平台 VTM-7.0 相比,本文算法平均能够节省 47.24%的编码时间,*BDBR*(Bjφntegaard delta bit rate)仅增加 1.26%。

关键词:多功能视频编码;编码单元划分;纹理特征;快速算法

中图分类号:TN919.8 **文献标识码**:A **文章编号**:1005-0086(2023)02-0200-08

A fast CU partition algorithm for VVC based on sub-block texture difference

LI Ya[1,2], LI Qiang[1,2*], MENG Hui[1,2]

(1. School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China; 2. Chongqing Key Laboratory of Signal and Information Processing, Chongqing 400065, China)

Abstract: Aiming at the problem of high coding complexity due to the introduction of multi-type tree (MTT) partition structure based on quadtree (QT) partition in versatile video coding (VVC), a fast coding unit (CU) partition algorithm based on the difference of texture features of CU sub-blocks is proposed in this paper. This algorithm can effectively reduce the coding complexity of VVC and shorten the coding time. Firstly, through the text complexity difference of sub-blocks in different partition directions, the direction of MTT partition is judged in advance, and unnecessary MTT partition directions are skipped; Then, according to the texture difference between adjacent sub-blocks of current CU, it is further determined whether to skip the ternary tree (TT) partition, so as to further reduce the number of partition modes in the candidate list. The experimental results show that under the coding configuration of all intra (AI), compared with the official test platform VTM-7.0, the proposed algorithm can save 47.24 % of the coding time on average, and the Bjφntegaard delta bit rate (*BDBR*) only increases by 1.26%.

Key words: versatile video coding (VVC); coding unit (CU) partition; texture features; fast algorithm

0 引 言

随着通信技术和视频技术的发展,人们对更高分辨率视频的需求在不断增长。尽管高效视频编码(high efficiency video coding,HEVC)有着不俗的编码性能,但在面对 4 K、8 K 等高质量视频

* **E-mail**:liqiang@cqupt.edu.cn

收稿日期:2022-03-31 **修订日期**:2022-06-05

基金项目:国家自然科学基金(62176035)和重庆市教委科学技术研究项目(KJZD-K202100606)资助项目

所带来的庞大数据量时,其编码效率也显得捉襟见肘。为了解决这一难题,2015 年,联合视频专家组(joint video exploration team,JVET)便开始了下一代视频编码标准的制定工作。2020 年 7 月,第一版多功能视频编码(versatile video coding,VVC)标准正式发布[1]。VVC 采用了多种新的编码技术,如更加灵活的编码单元(coding unit,CU)划分方式[2],67 种帧内预测模式[3],仿射运动估计等[4]。这些新编码技术的应用,使得 VVC 的压缩效率相比 HEVC 提高 50% 以上[5],然而 VVC 编码时间相比 HEVC 也增加了 10 倍以上[6],如此高的编码复杂度给 VVC 标准的推广和实际应用带来巨大困难。因此如何在保证编码效率的同时,显著降低 VVC 的编码复杂度成为了当前研究的重点。

VVC 采用的四叉树(quadtree,QT)嵌套多叉树(multi-type tree,MTT)的 CU 划分方式,是编码复杂度提升的主要原因之一[7]。因此,优化 CU 划分结构,实现 CU 的快速划分,是当前 VVC 编码加速的重要研究方向。目前 CU 划分快速算法大致可分为如下两类:基于统计的方法和基于学习的方法。

基于统计方法的 CU 快速划分算法主要利用视频编码帧的特征,如空时域相关性、纹理特性等,建立 CU 划分的统计模型,然后利用这些模型提前终止 CU 划分,或跳过不必要的划分方式。文献[8]首先根据自适应标准差阈值对 CU 纹理复杂度进行分类,初步缩减划分模式列表;然后采用 Sobel 梯度算子确定纹理方向,跳过非最优划分模式。文献[9]提出了一种基于梯度的 CU 快速划分算法。该算法首先根据当前 CU 的整体梯度的大小来判断是否可以直接终止划分,如果不能终止划分,就进一步根据 CU 的梯度方向判断水平和垂直划分的可能性,从而跳过不必要的划分方式。文献[10]提出了一种基于纹理方差和梯度特征的 CU 快速划分算法。首先利用 Sobel 算子提取梯度特征来决定当前是进行 MTT 划分还是 QT 划分,如果是 MTT 划分,再通过方差从 MTT 划分方式中选择一种作为最佳划分方式。文献[11]提出了一种基于灰度共生矩阵的 CU 快速划分算法。该算法通过灰度共生矩阵计算 CU 的纹理方向,跳过MTT的水平方向或者垂直方向。

在基于学习的方法中,CU 的最佳划分方式可以从大量的数据中自动学习,从而跳过非最佳划分方式。文献[12]提出了一种基于深度学习的帧内 CU 快速划分算法。该算法首先构建多阶段退出划分机制的神经网络模型,依次对包括不划分在内的最多 6 种模式进行决策,然后通过自适应损失函数,设计了一种多阈值的决策方案以平衡编码复杂度和率失真性能。文献[13]提出了一种基于随机森林的 CU 快速划分算法。首先将 CU 分类为简单、复杂和模糊 3 类,对简单和复杂的 CU 训练一个随机森林模型来预测其最佳划分方式,对模糊的 CU 训练一个随机森林模型来预测其是否需要划分。文献[14]提出了一种基于支持向量机的 VVC 帧内快速 CU 划分算法。该算法将纹理信息输入到支持向量机模型,然后对 CU 的分割方式进行预测来提前终止或跳过冗余分割。文献[15]提出了一种基于密集神经网络的 CU 快速算法。该算法通过利用密集神经网络预测 4×4 编码块的边缘是划分边界的概率,然后跳过不必要的划分方式。

上述的快速 CU 划分算法虽然都能够一定程度上降低 VVC 的编码复杂度,但仍然存在着一些问题。如降低的编码复杂度有限,编码效率损失较多,并且大部分算法对不同的视频的加速效果差异较大。为了解决这些问题,本文提出了一种基于子块纹理差异的 VVC 快速 CU 划分算法。该算法通过 CU 子块间的纹理差异,对 CU 划分方式进行筛选,跳过不必要的划分模式,在编码效率损失几乎可以忽略不计的情况下显著降低了编码复杂度。

1　本文算法

在本节中,首先分析了 VVC 的 CU 划分结构以及其带来的编码复杂度;然后根据不同划分方向获得的子块的复杂度差异,判断出 MTT 的划分方向;最后通过相邻子块间的纹理差异,判断是否可以跳过三叉树(ternary tree,TT)划分。

1.1　CU 划分结构

VVC 在 QT 划分基础上引入了更灵活的 MTT 划分。视频中的每帧图像都被划分为若干个 128×128 的编码树单元(coding tree unit, CTU)。每个 CTU 先进行 QT 划分,然后 QT 的每个叶节点再进行 MTT 或 QT 划分。MTT 包括水平方向划分和垂直方向划分,每个方向又包括二叉树(binary tree,BT)划分和 TT 划分。所以一个 CU 最多有 QT、水平二叉树(horizontal binary tree,HBT)、水平三叉树(horizontal ternary tree,HTT)、垂直二叉树(vertical binary tree, VBT)和垂直三叉树(vertical ternary tree, VTT)共 5 种划分方式。所有可能的划分模式被放在候选列表中,编码器对每一个 CU 都需要遍历候选列表中的所有划分模式,计算其率失真代价

(rate distortion cost,RDC),然后选择 RDC 最小的划分模式作为当前 CU 的最佳划分模式。这种递归遍历计算 RDC 的过程是 VVC 编码复杂度的主要来源。图 1(a)给出了一种递归遍历后的最佳划分结构

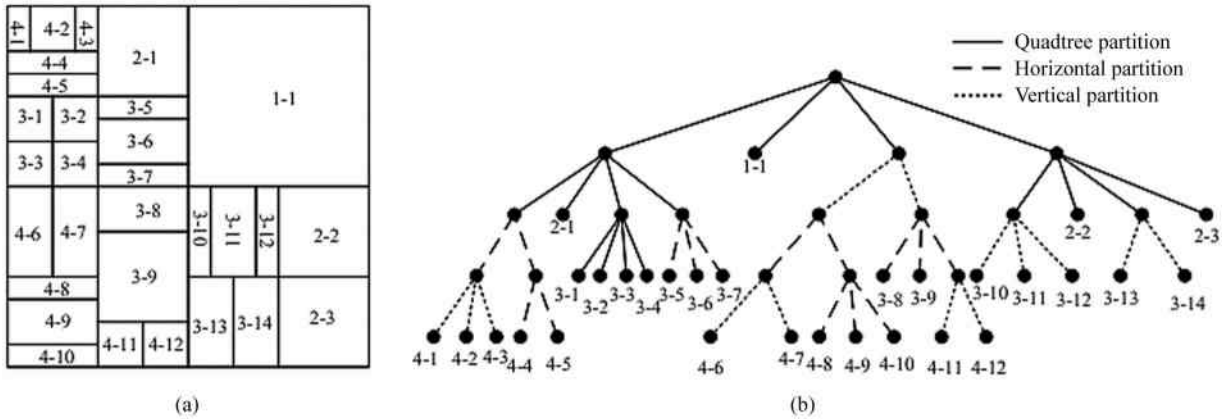

图 1　VVC 中 CU 划分结构示例:(a) CU 划分结构; (b) 对应的树状结构

Fig. 1　Example of CU partition structure in VVC: (a) CU partition structure; (b) Corresponding tree structure

示意图;图 1(b)是对应的树状结构图。

1.2　MTT 划分方向快速决策

图 2 是 VVC 官方测试序列 BasketballPass 的第 1 帧的 CU 划分结果。从图中可以看出 CU 的 MTT 划分方向与 CU 子块的纹理特征有着很大的联系。如果当前 CU 在水平方向存在纹理较为简单的子块,则当前 CU 更倾向于选择水平划分;如果当前 CU 在垂直方向存在纹理较为简单的子块,则当前 CU 倾向于选择垂直划分。图 2 中标注了 4 个 CU,其尺寸均为 $4N\times4N$,其中 A 和 B 选择进行水平划分,A 和 B 在水平方向均存在两个尺寸分别为 $4N\times N$ 和 $4N\times 2N$ 纹理简单的子块;C 和 D 选择垂直划分,C 和 D 在垂直方向均存在纹理简单的子块。纹理复杂度反映了图像纹理简单程度,纹理复杂度越小表明纹理越简单,图像就越平坦。因此可以利用当前 CU 在水平和垂直方向上子块的纹理复杂度差异来判断其最佳划分方向。

为了获取 CU 不同方向上的子块纹理复杂度差异,本文按照图 3 所示的方法,仅使用水平或垂直划分,将当前CU划分为4个子块,然后求每个子块的

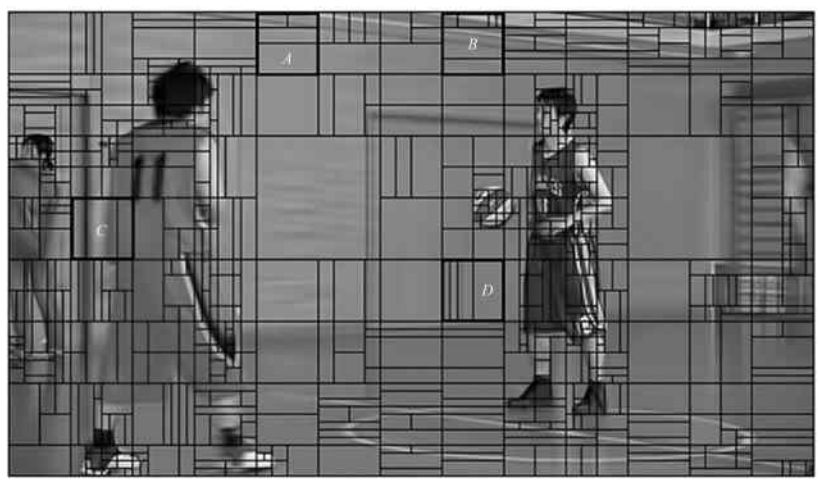

图 2　CU 划分实例

Fig. 2　CU partion example

纹理复杂度。相较于方差和标准差,平均绝对差(mean absolute differences,MAD)的计算复杂度更低,因此本文选用 MAD 来衡量子块的纹理复杂程度。水平方向上4个子块的纹理复杂度计算式

如下:

$$Mad_{h1}=\frac{1}{W\times\frac{H}{4}}\sum_{x=0}^{W-1}\sum_{y=0}^{\frac{H}{4}-1}\left|f(x,y)-mean_{h1}\right|,\quad(1)$$

$$Mad_{h2}=\frac{1}{W\times\frac{H}{4}}\sum_{x=0}^{W-1}\sum_{y=\frac{H}{4}}^{\frac{H}{2}-1}\left|f(x,y)-mean_{h2}\right|,\quad(2)$$

$$Mad_{h3}=\frac{1}{W\times\frac{H}{4}}\sum_{x=0}^{W-1}\sum_{y=\frac{H}{2}}^{\frac{3H}{4}-1}\left|f(x,y)-mean_{h3}\right|,\quad(3)$$

$$Mad_{h4}=\frac{1}{W\times\frac{H}{4}}\sum_{x=0}^{W-1}\sum_{y=\frac{3H}{4}}^{H-1}\left|f(x,y)-mean_{h4}\right|,\quad(4)$$

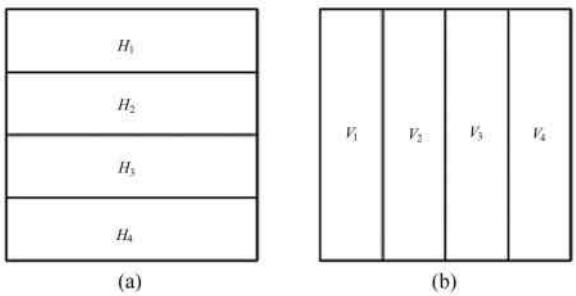

图3　不同划分方向获得的子块:(a)水平方向;(b)垂直方向

Fig. 3　Sub-blocks obtained from different partition directions: (a) Horizontal direction; (b) Vertical direction

式中,Mad_{h1}、Mad_{h2}、Mad_{h3}和 Mad_{h4}分别为当前 CU 在水平方向上 4 个子块 H_1、H_2、H_3 和 H_4 的纹理复杂度,W 和 H 表示当前 CU 的宽度和高度,$f(x,y)$为 CU 在坐标(x,y)处的亮度值,$mean_{h1}$、$mean_{h2}$、$mean_{h3}$和 $mean_{h4}$分别表示 4 个子块的平均亮度值。

同理垂直方向上 CU 的 4 个子块纹理复杂度计算式如下:

$$Mad_{v1}=\frac{1}{H\times\frac{W}{4}}\sum_{y=0}^{H-1}\sum_{x=0}^{\frac{W}{4}-1}\left|f(x,y)-mean_{v1}\right|,\quad(5)$$

$$Mad_{v2}=\frac{1}{H\times\frac{W}{4}}\sum_{y=0}^{H-1}\sum_{x=\frac{W}{4}}^{\frac{W}{2}-1}\left|f(x,y)-mean_{v2}\right|,\quad(6)$$

$$Mad_{v3}=\frac{1}{H\times\frac{W}{4}}\sum_{y=0}^{H-1}\sum_{x=\frac{W}{2}}^{\frac{3W}{4}-1}\left|f(x,y)-mean_{v3}\right|,\quad(7)$$

$$Mad_{v4}=\frac{1}{H\times\frac{W}{4}}\sum_{y=0}^{H-1}\sum_{x=\frac{3W}{4}}^{W-1}\left|f(x,y)-mean_{v4}\right|,\quad(8)$$

式中,Mad_{v1}、Mad_{v2}、Mad_{v3}和 Mad_{v4}分别为当前 CU 在垂直方向上的 4 个子块 V_1、V_2、V_3 和 V_4 的纹理复杂度,$mean_{v1}$、$mean_{v2}$、$mean_{v3}$和 $mean_{v4}$表示每个子块的平均亮度值。

为了更简单地比较出当前 CU 在不同方向上各个子块纹理复杂度的差异,本文仅选取水平方向和垂直方向上的纹理复杂度最小的子块进行比较,计算方式如下:

$$MadH_{\min}=\min(Mad_{h1},Mad_{h2},Mad_{h3},Mad_{h4}),\quad(9)$$

$$MadV_{\min}=\min(Mad_{v1},Mad_{v2},Mad_{v3},Mad_{v4}),\quad(10)$$

式中,$MadH_{\min}$和 $MadV_{\min}$分别为当前 CU 按照图 3 的划分方式,在水平方向和垂直方向获得的子块的纹理复杂度最小值。

利用上述计算方法计算图 2 中标注的 4 个 CU 的 $MadH_{\min}$和 $MadV_{\min}$,结果如图 4 所示。根据图 4 不难发现:A 和 B 两个选择水平方向划分的 CU 均满足 $MadH_{\min}<MadV_{\min}$,C 和 D 两个选择垂直方向划分的 CU 均满足 $MadH_{\min}\geqslant MadV_{\min}$。

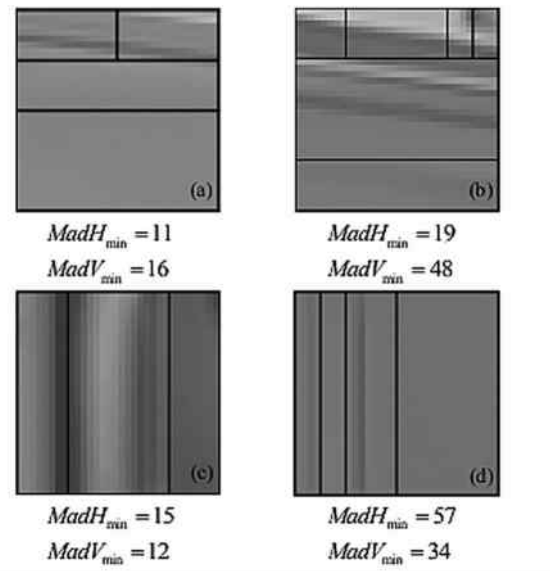

图 4　划分实例中 4 个 CU 的 $MadH_{\min}$和 $MadV_{\min}$:(a) *A*; (b) *B*; (c) *C*; (d) *D*

Fig. 4　$MadH_{\min}$ and $MadV_{\min}$ of CU in partition instance: (a) *A*; (b) *B*; (c) *C*; (d) *D*

为了进一步探究 $MadH_{\min}$和 $MadV_{\min}$的大小关系与 CU 的 MTT 划分方向之间的联系,本文在 VVC 参考软件 VTM-7.0 平台上进行统计实验。实验方法符合 JVET 推荐的通用测试条件(common test condition,CTC)。从 VVC 官方测试序列中,选择 6 个不同分辨率、不同运动特性的视频,采用全帧内(all intra,AI)的编码方式对每个视频的前 5 帧进行编码,量化参数(quantitative parameters,QP)分

别为 22、27、32 和 37。本文统计出了如表 1 所示的 $MadH_{\min}$ 和 $MadV_{\min}$ 的大小关系与对应的不同 CU 划分方式的占比。表中的 horizontal 和 vertical 分别代表 MTT 水平划分、垂直划分的比例，others 表示 QT 划分和不划分的比例。R 为 $MadH_{\min}$ 和 $MadV_{\min}$ 的比值，其计算式如下：

$$R=\frac{MadH_{\min}}{MadV_{\min}}, \tag{11}$$

R 反映出 $MadH_{\min}$ 和 $MadV_{\min}$ 的大小关系，即按图 3 所示的方式对 CU 进行划分后，CU 在水平方向上子块的最小复杂度与垂直方向上子块的最小复杂度的大小关系。

从表 1 可看出：当 $R<1$ 时，垂直划分模式的平均比例仅为 19%，并且水平划分模式平均比例为垂直划分模式平均比例的两倍以上；当 $R\geqslant 1$ 时，水平划分模式的平均比例仅为 17%，并且垂直划分模式平均比例为水平划分模式平均比例的两倍以上。

表 1　CU 在 $R<1$ 和 $R\geqslant 1$ 时的划分结果

Tab. 1　Partition result of CU when $R<1$ and $R\geqslant 1$

Sequence	R	Horizontal /%	Vertical /%	Others /%
Campfire	$R<1$	37	23	40
	$R\geqslant 1$	17	42	41
CatRobot1	$R<1$	33	19	48
	$R\geqslant 1$	13	44	43
BasketballDrive	$R<1$	67	6	27
	$R\geqslant 1$	23	31	46
BasketballDrill	$R<1$	36	26	38
	$R\geqslant 1$	24	38	38
BasketballPass	$R<1$	45	21	34
	$R\geqslant 1$	12	48	40
FourPeople	$R<1$	42	20	38
	$R\geqslant 1$	14	43	43
Average	$R<1$	43	19	38
	$R\geqslant 1$	17	41	42

基于以上统计数据，本文按照以下规则设计 MTT 在划分方向上的快速决策方案。当 $R<1$ 时，表示着当前 CU 在水平方向存在纹理复杂度较低的子块，此时可以提前跳过垂直方向上的 MTT 划分。相反，当 $R\geqslant 1$ 时，表示当前 CU 在垂直方向存在纹理复杂度较低的子块，此时可以提前跳过水平方向上的 MTT 划分。此外，为了平衡编码效率与算法加速效果，本文只考虑对尺寸为 32×32 和 16×16 的亮度 CU 执行快速划分决策。

1.3　TT 划分的跳过

在进行 MTT 划分方向决策之后，CU 仍然需要遍历 QT、同一方向的 BT 和 TT 3 种划分模式。为了进一步降低 VVC 的编码复杂度，需要进一步减少候选列表中划分模式的数量。

本文统计了 6 个视频测试序列中 BT 和 TT 的占比，统计结果见图 5。图中 CE、CT、BE、BL、BS 和 FE 分别表示视频序列 Campfire、CatRobot1、BasketballDrive、BasketballDrill、BasketballPass 和 FourPeople。在 6 个视频序列中，TT 划分的比例均未超过 20%，并且都要明显低于 BT 划分的比例。因此，对于大多数 CU 而言，TT 划分是不必要的，所以本文考虑根据 CU 子块的纹理差异，分析 TT 划分的必要性，进一步判断当前 CU 是否可跳过 TT 划分。

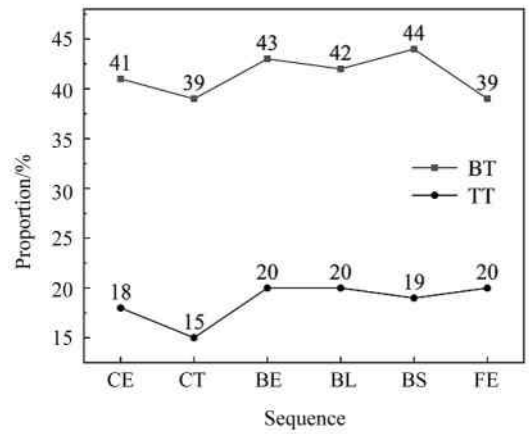

图 5　BT 和 TT 在 6 个序列中的占比

Fig. 5　Proportion of BT and TT in six sequences

CU 最终划分结果倾向于将纹理相似的区域划分在同一个子 CU 中，而将纹理差异较大的区域划分为两个或多个独立的子 CU。如图 3 所示，如果 CU 选择 HBT 划分，H_1 和 H_2 被划分在同一区域，H_3 和 H_4 被划分在同一区域，而 H_2 和 H_3 被划分在不同的区域，此时说明相邻子块之间 H_2 和 H_3 的纹理差异较大；如果 CU 选择 HTT 划分，H_1 和 H_2 被划分在不同的区域，H_3 和 H_4 被划分在不同的区域，而 H_2 和 H_3 被划分在同一区域，此时说明相邻子块之间 H_2 和 H_3 的纹理差异较小。因此可以计算 CU 相邻子块之间的纹理差异，如果中间相邻子块的纹理差异较大，就可以跳过 TT 划分。

水平方向上的相邻子块间的纹理差异的计算公式如下：

$$DiffH_{12}=\left|(Mad_{h1}-Mad_{h2})\times(mean_{h1}-mean_{h2})\right|, \tag{12}$$

$$DiffH_{23}=\left|(Mad_{h2}-Mad_{h3})\times(mean_{h2}-mean_{h3})\right|, \tag{13}$$

$$DiffH_{34}=\left|(Mad_{h3}-Mad_{h4})\times(mean_{h3}-mean_{h4})\right|, \tag{14}$$

式中,$DiffH_{12}$、$DiffH_{23}$ 和 $DiffH_{34}$ 分别表示当前 CU 水平方向上的相邻子块 H_1 与 H_2、H_2 与 H_3、H_3 与 H_4 之间的纹理差异。式(12)-(14)的计算方式兼顾了相邻 CU 间的平均亮度差异性以及亮度值离散程度的差异性,更好地反映出了子块之间的纹理差异。

同理,CU 垂直方向上的相邻子块间的纹理差异计算式如下:

$$DiffV_{12} = |(Mad_{v1} - Mad_{v2}) \times (mean_{v1} - mean_{v2})|, \quad (15)$$

$$DiffV_{23} = |(Mad_{v2} - Mad_{v3}) \times (mean_{v2} - mean_{v3})|, \quad (16)$$

$$DiffV_{34} = |(Mad_{v3} - Mad_{v4}) \times (mean_{v3} - mean_{v4})|, \quad (17)$$

式中,$DiffV_{12}$、$DiffV_{23}$ 和 $DiffV_{34}$ 分别表示当前 CU 垂直方向上相邻子块 V_1 与 V_2、V_2 与 V_3、V_3 与 V_4 之间的纹理差异。

为了比较中间相邻子块间的差异和两端相邻子块之间的差异的大小关系,本文定义了下式:

$$D_h = \frac{DiffH_{23}}{\min(DiffH_{12}, DiffH_{34})}, \quad (18)$$

$$D_v = \frac{DiffV_{23}}{\min(DiffV_{12}, DiffV_{34})}, \quad (19)$$

式中,$D_h>1$ 时,表示 H_2 和 H_3 的纹理差异较大;$D_v>1$ 时,表示 V_2 和 V_3 的纹理差异较大。

根据 6 个视频测试序列 CU 的划分结果,本文统计两种比例数据:当 $D_h>1$ 时,选择 HTT 的 CU 个数占比;当 $D_v>1$ 时,选择 VTT 的 CU 个数占比。统计结果如图 6 所示,CE、CT、BE、BL、BS 和 FE 表示的视频序列同图 5。当 $D_h>1$ 时,6 个测试序列中选择HTT划分的CU的比例很低,最高的是序列 FourPeople,占比仅为 8%;当 $D_v>1$ 时,选择 VTT 划分得到 CU 也很少,最高的是 BasketballDrive,仅为 8%。统计结果表明,当 $D_h>1$ 时,绝大多数的 CU 可跳过 HTT 的 RDC 计算过程;而当 $D_v>1$ 时,大多数 CU 不必要计算 VTT 的 RDC。

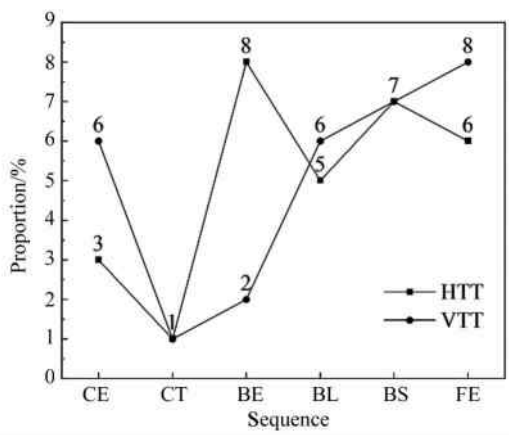

图 6　$D_h>1$ 时 HTT 的比例和 $D_v>1$ 时 VTT 的比例

Fig. 6　Proportion of HTT when $D_h>1$ and proportion of VTT when $D_v>1$

基于以上分析,本文设计了一种跳过 TT 划分的 CU 快速划分方案:如果 CU 选择进行水平 MTT 划分,当 $D_h>1$ 时,表示水平方向上 H_2 和 H_3 两个中间相邻子块的纹理差异较大,此时选择跳过 HTT 划分;如果 CU 选择进行垂直 MTT 划分,当 $D_v>1$ 时,表示垂直方向上 V_2 和 V_3 两个中间相邻子块的纹理差异较大,此时选择跳过 VTT 划分。

1.4　本文算法流程

本文利用 CU 子块之间的纹理差异性,跳过不必要的划分方式,实现一种 CU 快速划分算法。该算法只针对尺寸为 32×32 和 16×16 的亮度 CU 进行快速划分。如果满足 $R<1$,则跳过垂直方向的 MTT 划分;否则跳过水平方向的 MTT 划分。在确定了 CU 的 MTT 划分方向之后,进一步判断是否可以跳过 TT 划分。如果 CU 选择进行水平 MTT 划分,当 $D_h>1$ 时,跳过 HTT 划分;如果 CU 选择进行垂直 MTT 划分,当 $D_v>1$ 时,跳过 VTT 划分。图 7 为本文提出的 CU 划分快速算法的流程图。

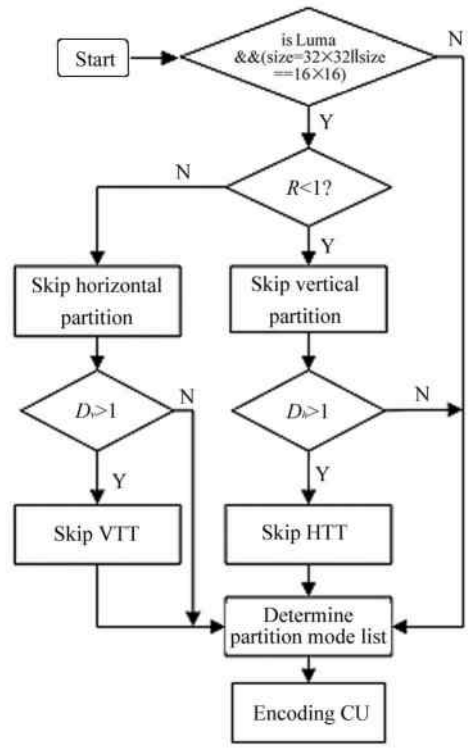

图 7　本文算法流程图

Fig. 7　Algorithm flow chart of this paper

2 实验结果

本文算法在 VVC 测试平台 VTM-7.0 上进行实现和性能测试。测试遵循 CTC，采用 AI 编码模式，*QP* 分别设置为{22，27，32，37}，测试序列为 A1(3 840×2 160)、A2(3 840×2 160)、B(1 920×1 080)、C(830×480)、D(416×240)和 E(1 280×720)6 个类别的 JEVT 推荐测试序列。

本文用来评估算法性能的指标分别为 *BDBR*(Bjφntegaard delta bit rate)和 *TS*。*BDBR* 表示在同样的客观质量下码率的增加情况，*BDBR* 为正表示编码效率下降，*BDBR* 为负表示编码效率提升；*TS* 为编码时间节省，其计算式为：

$$TS=\frac{1}{4}\sum_{i=1}^{4}\frac{T_{org}(QP_i)-T_{pro}(QP_i)}{T_{org}(QP_i)}, \tag{20}$$

式中，$T_{org}(QP_i)$表示在 *QP* 为 QP_i 时原始 VTM-7.0 编码测试序列的时间；$T_{pro}(QP_i)$表示采用本文快速算法在 *QP* 为 QP_i 时编码测试序列的时间。

表 2 显示了本文算法与原始 VTM-7.0 平台对比的实验结果。本文算法针对 6 类 22 个视频序列的编码均有很好的加速效果，平均能够降低 47.24%的编码时间，而 *BDBR* 仅增加 1.26%。其中编码时间降低最多的序列为 FourPeople，达到了 52.84%。对于 4K 超高清分辨率的序列，如 A1 和 A2 类序列，本文算法也有很好的加速效果，在保证了 VVC 对超高清视频高编码效率的同时，显著降低了编码的复杂度。

表 3 为本文算法与文献[10]算法性能的对比结果，文献[10]的实验同样基于 VTM-7.0。本文选取了与文献[10]相同的测试序列进行对比。由测试数据可知，本文算法不仅节省的编码时间更多，且 *BDBR* 明显更低，说明本文加速算法的性能要高于文献[10]算法。此外，文献[10]算法在 D 类和 E 类视频中的编码加速效果差异较大，而本文算法在各类视频中均有着良好的表现。

为了评估所提出的快速算法的率失真性能，本文给出了如图 8 所示的测试序列 BasketballDrill 的率失真曲线。从图可以看出，对于 *BDBR* 最大的测试序列 BasketballDrill，本文所提出的快速算法的率失真曲线与原始 VTM-7.0 平台的率失真曲线也非常接近。这表明本文算法在编码效率损失很小的同时，能显著降低编码复杂度。

表 2　本文算法与 VTM-7.0 的编码性能对比

Tab. 2　Performance comparison of proposed algorithm and VTM-7.0

Class	Sequence	*BDBR*/%	*TS*/%
A1	Campfire	1.27	45.64
	FoodMarket4	0.52	31.81
	Tango2	1.19	43.32
A2	CatRobot1	1.32	44.74
	DaylightRoad2	1.41	51.21
	ParkRunning3	0.56	36.17
B	BasketballDrive	1.28	48.88
	BQTerrace	1.25	49.18
	Cactu	1.35	49.43
	MarketPlace	1.09	50.07
	RitualDance	1.76	48.65
C	BasketballDrill	2.40	50.77
	BQMall	1.29	50.62
	PartyScene	0.76	50.33
	RaceHorses	0.96	48.62
D	BasketballPass	1.36	46.70
	BlowingBubbles	1.11	47.72
	BQSquare	0.75	49.34
	RaceHorses	1.04	45.33
E	FourPeople	1.90	52.84
	Johnny	1.76	48.55
	KristenAndSara	1.43	49.26
	Average	**1.26**	**47.24**

表 3　本文算法与文献[10]对比

Tab. 3　Proposed algorithm is compared with Ref. [10]

Class	Sequence	Ref. [10] *BDBR*/%	Ref. [10] *TS*/%	Proposed *BDBR*/%	Proposed *TS*/%
B	BasketballDrive	3.28	59.35	1.28	48.88
	BQTerrace	1.08	45.30	1.25	49.18
	Cactu	1.84	52.44	1.35	49.43
	Kimono	1.93	59.51	0.81	49.04
	ParkScene	1.26	51.84	1.21	52.59
C	BasketballDrill	1.82	48.48	2.40	50.77
	BQMall	1.87	52.47	1.29	50.62
	PartyScene	0.26	38.62	0.76	50.33
	RaceHorses	0.88	49.05	0.96	48.62
D	BasketballPass	1.95	47.70	1.36	46.70
	BlowingBubbles	0.47	40.35	1.11	47.72
	BQSquare	0.19	31.95	0.75	49.34
	RaceHorses	0.54	41.69	1.04	45.33
E	FourPeople	2.70	57.57	1.90	52.84
	Johnny	3.22	56.88	1.76	48.55
	KristenAndSara	2.78	55.11	1.43	49.26
	Average	**1.63**	**49.27**	**1.29**	**49.33**

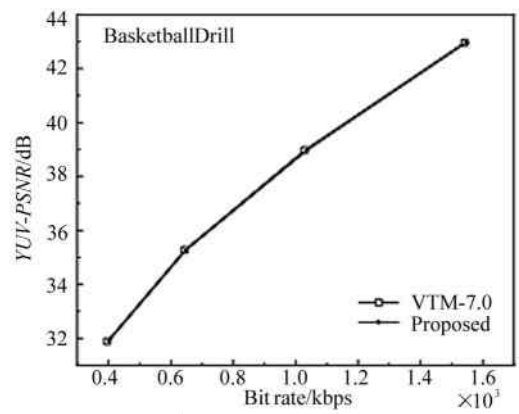

图 8　本文算法与 VTM-7.0 的率失真性能比较

Fig. 8　The rate distortion performance comparison between the proposed algorithm and the VTM-7.0

3　结　论

本文提出了一种基于子块纹理差异的 VVC 快速 CU 划分算法。该算法首先通过不同划分方向的子块的复杂度差异提前判断 MTT 划分的方向;然后根据相邻子块间的纹理差异进一步判断是否跳过 TT 划分。实验测试结果表明,与原始平台VTM-7.0相比,本文算法在 *BDBR* 仅增加 1.26%的情况下,能够平均减少 47.24%的编码时间,说明本文算法在计算复杂度降低和编码效率损失之间取得了良好的平衡,对 VVC 的实际应用具有较大的现实意义。下一步可以对纹理特征与 CU 划分深度之间的联系进行研究,以进一步提高编码速度。

参考文献:

[1] SULLIVAN G. Versatile video coding (VVC) arrives [C]//IEEE International Conference on Visual Communications and Image Processing, December 01-04, 2020, Macau, China. New York: IEEE, 2020: 1-1.

[2] HUANG Y W, AN J C, HUANG H, et al. Block partitioning structure in the VVC standard[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 31(10): 3818-3833.

[3] PFAFF J, FILIPPOV A, LIU S, et al. Intra prediction and mode coding in VVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 31(10): 3834-3847.

[4] MEUEL H, OSTERMANN J. Analysis of affine motion-compensated prediction in video coding[J]. IEEE Transactions on Image Processing, 2020, 19(1): 7359-7374.

[5] BROSS B, CHEN J L, OHM J R, et al. Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC)[J]. Proceedings of the IEEE, 2021,109(9): 1463-1493.

[6] PAKDAMAN F, ADELIMANESH A M, GABBOUJ M, et al. Complexity analysis of next-generation VVC encoding and decoding[C]//IEEE International Conference on Image Processing, October 25-28, 2020, Abu Dhabi, United Arab Emirates. New York: IEEE, 2020: 3134-3138.

[7] SALDANHA M, SANCHEZ G, MARCON C, et al. Complexity analysis of VVC intra coding[C]//IEEE International Conference on Image Processing, October 25-28, 2020, Abu Dhabi, United Arab Emirates. New York: IEEE, 2020: 3119-3123.

[8] LU J B, PENG Z J, SHU Z J, et al. Fast CU partition and angle mode decision for VVC intra coding[J]. Journal of Optoelectronics · Laser, 2021, 32(11): 1171-1179.
卢嘉彬,彭宗举,束争杰,等. 面向 VVC 帧内编码的快速 CU 划分和角度模式决策[J]. 光电子 · 激光, 2021, 32(11):1171-1179.

[9] CUI J, ZHANG T, GU C C, et al. Gradient-based early termination of CU partition in VVC intra coding[C]//Data Compression Conference, March 24-27, 2020, Snowbird, UT, USA. New York: IEEE, 2020: 103-112.

[10] FAN Y B, CHEN J N, SUN H M, et al. A fast QTMT partition decision strategy for VVC intra prediction[J]. IEEE Access, 2020, 8(1):107900-107911.

[11] ZHANG H C, LI Y, LI T S, et al. Fast GLCM-based intra block partition for VVC[C]//Data Compression Conference, March 23-26, 2021, Snowbird, UT, USA. New York: IEEE, 2021: 382-382.

[12] LI T Y, XU M, TANG R Z, et al. DeepQTMT: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC[J]. IEEE Transactions on Image Processing, 2021, 30(1):5377-5390.

[13] HE Q, WU W X, LUO L, et al. Random forest based fast CU partition for VVC intra coding[C]//IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, August 4-6, 2021, Chengdu, China. New York: IEEE, 2021:1-4.

[14] WU G Q, HUANG Y, ZHU C, et al. SVM based fast CU partitioning algorithm for VVC intra coding[C]//IEEE International Symposium on Circuits and Systems, May 22-28, 2021, Daegu, Korea. New York: IEEE, 2021:1-5.

[15] ZHANG Q W, GUO R X, JIANG B, et al. Fast CU decision-making algorithm based on DenseNet network for VVC[J]. IEEE Access, 2021, 9(1): 119289-119297.

作者简介:

李　强　(1968—),男,硕士,副教授,研究生导师,主要从事音视频信号处理方面的研究.