



基于深度学习的快速 QTMT 划分

彭双¹, 王晓东¹, 彭宗举^{1,2}, 陈芬²

(1. 宁波大学信息科学与工程学院, 浙江 宁波 315211; 2. 重庆理工大学电气与电子学院, 重庆 400054)

摘要: 与之前的编码标准相比, 多功能视频编码 (versatile video coding, VVC) 进一步提高了压缩效率。嵌套多类树的四叉树 (quadtree with nested multi-type tree, QTMT) 结构是提高编码增益的关键之一, 同时极大地增加了编码复杂度。为降低 VVC 编码复杂度, 提出了一种基于深度学习的快速 QTMT 划分方法。首先, 提出了注意力-非对称卷积结构来预测划分模式的概率。然后, 基于阈值提出了快速划分模式决策。最后, 提出了编码性能与时间的代价函数来求解最优阈值, 提出了阈值决策方法。实验表明, 算法在不同档次下的时间节省分别为 48.62%、52.93%、62.01%, BDBR 分别为 1.05%、1.33%、2.38%。结果表明, 算法的时间节省和率失真性能优于其他快速算法。

关键词: VVC; QTMT; 快速划分决策; 深度学习

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2021062

Fast QTMT partition decision based on deep learning

PENG Shuang¹, WANG Xiaodong¹, PENG Zongju^{1,2}, CHEN Fen²

1. Faculty of Information Science and Engineering, Ningbo University, Ningbo 315211, China

2. Faculty of Electrical and Electronics, Chongqing University of Technology, Chongqing 400054, China

Abstract: Compared with the predecessor standards, versatile video coding (VVC) significantly improves compression efficiency by a quadtree with nested multi-type tree (QTMT) structure but at the expense of extremely high coding complexity. To reduce the coding complexity of VVC, a fast QTMT partition method was proposed based on deep learning. Firstly, an attention-asymmetric convolutional neural network was proposed to predict the probability of partition modes. Then, the fast decision of partition modes based on the threshold was proposed. Finally, the cost of coding performance and time was proposed to obtain the optimal threshold, and the threshold decision method was proposed. Experimental results at different levels show that the proposed method achieves an average time saving of 48.62%/52.93%/62.01% with the negligible BDBR of 1.05%/1.33%/2.38%. Such results demonstrate that the proposed method significantly outperforms other state-of-the-art methods.

Key words: VVC, QTMT, fast partition decision, deep learning

收稿日期: 2020-09-30; 修回日期: 2021-04-10

基金项目: 国家自然科学基金资助项目 (No.61771269, No.61620106012); 浙江省自然科学基金资助项目 (No.LY20F010005); 宁波市自然科学基金资助项目 (No.2019A610107); 重庆理工大学科研启动基金资助项目 (No.2020ZDZ029, No.2020ZDZ030)

Foundation Items: The National Natural Science Foundation of China (No.61771269, No.61620106012), The Natural Science Foundation of Zhejiang Province (No.LY20F010005), The Natural Science Foundation of Ningbo (No.2019A610107), The Scientific Research Foundation of Chongqing University of Technology (No.2020ZDZ029, No.2020ZDZ030)



1 引言

随着超高清、高动态、宽色域和全景视频等技术的出现, 高效视频编码 (high efficiency video coding, HEVC)^[1] 的压缩效率明显不足, 因而多功能视频编码 (versatile video coding, VVC)^[2] 被提出。在相同客观质量条件下, VVC 的比特率约为 HEVC 的 50%。嵌套多类树的四叉树 (quadtree with nested multi-type tree, QTMT)^[3] 划分结构是 VVC 编码增益提高的关键^[4], 但同时带来了编码复杂度的急剧上升。在帧内配置模式下, VVC 编码复杂度约为 HEVC 的 18 倍^[5]。此外, 与之前编码标准中帧内编码复杂度远低于帧间编码不同, VVC 中帧内编码复杂度却高于帧间编码, 前者约为后者的 1.3 倍^[6]。因此, 降低 VVC 帧内编码复杂度具有重要意义。

目前已有许多学者对 HEVC 和 VVC 低复杂度编码进行了研究。研究方法主要分为基于统计和基于学习的方法。在基于统计的快速编码算法中, 姚英彪等^[7]结合空域相关性和纹理信息提出了快速划分方法。Kuo 等^[8]提出一种基于时空域编码单元 (coding unit, CU) 的决策方法, 利用已编码相邻编码树单元 (coding tree unit, CTU) 和同位 CTU 深度信息, 加权预测当前 CTU 的深度范围。Jamali 等^[9]利用绝对变换残差和 (sum of absolute transformed difference, SATD) 代价对率失真代价 (rate-distortion cost, RDC) 进行估计, 从而减少进行率失真优化 (rate distortion optimization, RDO) 的模式, 同时将预测的 RDC 建模为正态分布, 通过置信区间进一步改进预测效果。Huang 等^[10]为平衡率失真 (rate distortion, RD) 性能与编码时间, 通过优化率失真复杂度对编码模式进行决策, 其中, RD 性能和编码时间通过提取特征来估计。参考文献[8,10]虽然能有效降低编码复杂度, 但仅针对 HEVC 中 QT 划分进行决策, 对 VVC 编码中 QTMT 划分并不适用。Lei 等^[11]

通过简化粗选模式决策估计不同划分方向的 SATD 代价, 将子 CU 不同划分方向的 RDC 作为当前 CU 的估计代价, 最后根据估计的 SATD 代价和 RDC 综合决定当前 CU 的最优划分方向。Chen 等^[12]和 Fan 等^[13]通过提取方差和梯度特征, 对特征设定阈值来决策提前终止和跳过不可能的模式, 最终仅对一个模式进行 RDO, 极大降低了编码复杂度。Park 等^[14-15]利用中间编码信息对跳过模式和终止运动估计进行决策。在参考文献[14]中, 通过对三叉树划分进行统计分析, 提出将已编码 CU 代价作为概率决策特征, 从而决定是否跳过三叉树划分。在参考文献[15]中, 利用已编码 CU 的最佳模式对终止仿射运动估计进行决策, 利用运动矢量方向缩减仿射运动参考帧的数量。参考文献[12-13]仅在单一的 CU 级实施决策, 参考文献[14]仅跳过三叉树划分, 而参考文献[15]仅对仿射运动估计进行决策, 以上方法节省的时间都非常有限。

在基于学习的快速编码算法中, Liu 等^[16]改进了传统复杂度特征并作为分类器输入, 通过两个支持向量机分类器将 CU 分为 3 类, 并且利用惩罚因子平衡性能损失与时间复杂度。Chen 等^[17]引入非对称卷积核来提高特征提取能力, 通过置信区间平衡复杂度与性能损失, 极大降低了 HEVC 编码复杂度, 同时, 通过预测最小 RDO 候选模式数量来加速模式决策过程。Katayama 等^[18]充分利用了 CU 的空域相关性, 将相邻 CU 亮度信息作为卷积神经网络 (convolutional neural network, CNN) 的输入, 实现了 HEVC 低复杂度帧内编码, 但不利于硬件编码。Kim 等^[19]建立了基于 CNN 的多级二分类模型来决策 CU 划分, 将 CU 亮度信息作为模型的输入, 并将编码的中间信息作为外部特征输入模型。Xu 等^[20]建立了分层 CU 划分映射来描述 CTU 的划分情况, 提出了提前终止的分层 CNN 模型, 在保证编码性能的前提下显著降低了 HEVC 帧内编码复杂度。Tang 等^[21]

将可变池化 CNN 应用到 VVC 帧内编码中, 使得 CNN 能适应不同形状的 CU。然而, 在池化层中丢失了特征信息, 导致最终模型的预测结果较差。Yang 等^[22]通过人工方式提取特征, 将特征输入决策树模型进行训练, 最终决定 CU 是否划分。虽然决策树模型能充分挖掘出特征与划分之间的潜能, 但特征选取受人为因素影响较大。Fu 等^[23]利用了贝叶斯分类器, 将是否跳过垂直模式建模为二分类问题, 以当前 CU、子 CU 划分模式和角度模式作为分类器的输入。Amestoy 等^[24]提出一种可调节的机器学习方法, 通过调整风险区间的大小来控制 VVC 的性能损失。参考文献[16,20]针对 HEVC 均提出了有效的快速编码方案, 对基于学习的 VVC 快速编码方法具有一定的启发意义。参考文献[16,22,24]均通过人为选取特征, 不能充分挖掘 CU 的特征信息。

上述方法主要将划分决策建模为分类问题, 而实际问题中分类边界通常不明确, 因此本文将划分决策建模为回归问题。同时考虑到深度学习在编码各领域取得的突出效果^[25], 本文结合深度学习模型来预测划分模式的概率, 提出了一种基于深度学习的快速 VVC 划分决策方法。实验结果表明, 所提出算法在保证编码性能几乎不损失的同时极大地降低了编码复杂度。

2 快速 QTMT 划分方法

本文首先分析了 QTMT 的复杂度, 然后提出注意力-非对称卷积网络 (attention asymmetric CNN, AA-CNN) 结构来预测划分模式的概率, 最后提出了基于 AA-CNN 的快速模式决策模型。

2.1 QTMT 复杂度分析

VVC 中采用 QTMT 结构使得 CU 能适应不同图像内容。在 VVC 编码过程中, 每个 CU 先按照四叉树 (quad tree, QT) 划分, 再在 QT 叶节点处按多类树 (multi-type tree, MT) 递归划分。MT 划分结构包括 4 种划分模式, 分别为垂直二

叉树 (vertical binary tree, VB)、水平二叉树 (horizontal binary tree, HB)、垂直三叉树 (vertical ternary tree, VT) 和水平三叉树 (horizontal ternary tree, HT)。除上述划分的模式以外, 还有不划分的模式 Intra, 这些模式组成了 QTMT 的划分模式列表 (partition mode list, PML)。最优模式 m^* 为 PML 中 RDC 最小的模式, 计算式如下:

$$m^* = \min_{m \in \text{PML}} \text{RDC}_m \quad (1)$$

其中, $\text{PML} = \{1, 2, 3, 4, 5, 6\}$, 分别对应 {Intra, QT, HB, VB, HT, VT}, RDC_m 的计算式如下:

$$\text{RDC}_m = D_m + \lambda \cdot R_m \quad (2)$$

其中, D_m 和 R_m 分别为编码模式 m 产生的失真和所需要的比特, λ 为控制 RD 性能的拉格朗日乘子。实际码流中仅模式 m^* 被编码, 因此 QTMT 结构包括了大量冗余模式。虽然 VVC 快速划分策略^[26]能跳过部分划分模式, 但仅对纹理较为平坦的区域有效。

2.2 划分模式概率预测

2.2.1 数据集构建

不完全的 PML 会影响模式的真实分布, 使得模型的预测准确率降低。本文建立训练数据集时, 仅采用具有完整 PML 的 CU。从 A1、A2、B、C、D 和 E 6 类标准测试序列中分别挑选了序列 Campfire、ParkRunning3、Cactus、BQMall、BasketballPass 和 Johnny, 每个序列选取 60 000 个数据样本, 其中大小为 32×32 和 16×16 的 CU 各 30 000 个, 并利用 3 次插值将 16×16 的 CU 上采样为 32×32 的 CU。数据集中样本分别用 4 个量化参数 (quantization parameter, QP) 22、27、32 和 37 进行编码, 总样本数量为 $60\,000 \times 6 \times 4 = 1\,440\,000$, 每个样本包括 CU 亮度信息、CU 大小和 QP, 样本标签为 CU 最优划分模式。数据集被随机地划分为训练集 (9/10) 和校验集 (1/10)。

2.2.2 AA-CNN 结构

本文采用 CNN 来预测划分模式的概率, 结合



MT 划分非对称性和特征重要性,提出了 AA-CNN 结构。MT 划分非对称性体现在子 CU 形状非正方形,为提高非对称特征的提取能力,本文引入了非对称卷积核。注意力模块可实现特征通道^[27]和卷积核^[28]的权重分配,因此在本文中引入该模块。本文通过压缩原始图像来获取注意力,以便从全局控制特征权重的分配,如图 1 所示,首先将 32×32 的 CU 展开为 1×1024 的向量,再通过 1024×128 和 $128 \times N$ 两个维度逐渐减小的全连接层进行压缩得到 $1 \times N$ 的注意力向量,其中 N 为被控制的特征向量通道数,并且 QP 作为外部特征被加入全连接层,最后通过 Softmax 函数进行激活。此外,为了加快模型的收敛速度,激活函数引入了参考文献[28]中所使用的温度控制,即对 Softmax 函数的输入除以一个温度系数 τ , τ 随着训练次数发生改变。 τ 初始值为 30,使得各特征权重相当,可有效加快收敛速度,随着训练次数的增加, τ 逐渐减小,当 $\tau=1$ 时激活函数退化为原始 Softmax 函数。

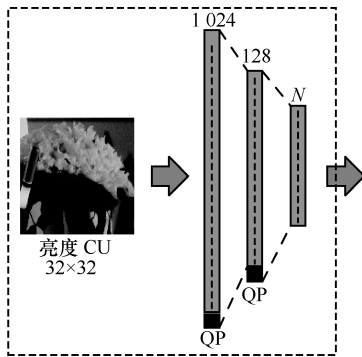


图 1 注意力模块

本文设计的 AA-CNN 结构如图 2 所示,分为数据预处理、特征提取、特征拟合以及结果输出 4 部分,具体如下。

(1) 获取亮度 CU 并将大小缩放至 32×32 ,再进行归一化处理。

(2) 该结构包括 L1、L2 和 L3 3 个非重叠卷积层,每层包含 3 个分支,第一层各分支的卷积核大小分别为 4×2 、 4×4 和 2×4 (滤波器数量为 16),

第二层和第三层各分支的卷积核大小均分别为 2×1 、 2×2 和 1×2 (滤波器数量为 24 和 32)。与对称卷积核相比,非对称卷积核可以提取不同方向的特征,更加适应 MT 划分。此外,通过压缩输入 CU 获得注意力向量,再根据注意力向量对每个特征通道进行权重分配。每个注意力模块对应一个卷积模块,共 9 个,分别为注意力 1~9,该模块的引入可有效提高网络容量并消除特征冗余。

(3) 将每个分支提取的特征展开并拼接为 1×2176 的特征向量,然后通过 3 个大小分别为 2176×128 、 128×96 和 96×6 的全连接层 L4、L5 和 L6 进行特征拟合,并且在每个全连接层中将 QP 作为外部特征加入特征向量。

(4) 采用 Softmax 函数激活预测值并输出 1×6 的预测概率向量,分别对应 PML 中各模式为最优概率。

交叉熵用于度量两个概率分布间的差异性,差异越大交叉熵越大,反之越小。因此,本文采用交叉熵作为训练 AA-CNN 的损失函数,计算式如下:

$$\text{Loss} = H(p, q) + H(1 - p, 1 - q) \quad (3)$$

其中, $H(\cdot)$ 为交叉熵运算符, p 和 q 为预测和标签向量。

2.3 模式决策

2.3.1 快速模式决策

为有效平衡编码性能与时间,应在尽可能保留最优模式的情况下保留最少模式数量,本文所提出的快速模式决策流程如图 3 所示。首先调用 AA-CNN 模型得到预测概率向量 p ; 再按概率对 PML 进行降序排序,可表示为:

$$(L, p') = \text{Sort}(\text{PML}, p) \quad (4)$$

其中, L 和 p' 分别为排序后的模式列表和预测向量; 然后累加概率,并设定阈值 φ 确定最小保留模式数量 n , 计算式如下:

$$n = \arg \min_{x \in \{1, 2, 3, 4, 5, 6\}} x \quad \text{s.t.} \sum_{k=1}^x p'_k \geq \varphi \quad (5)$$

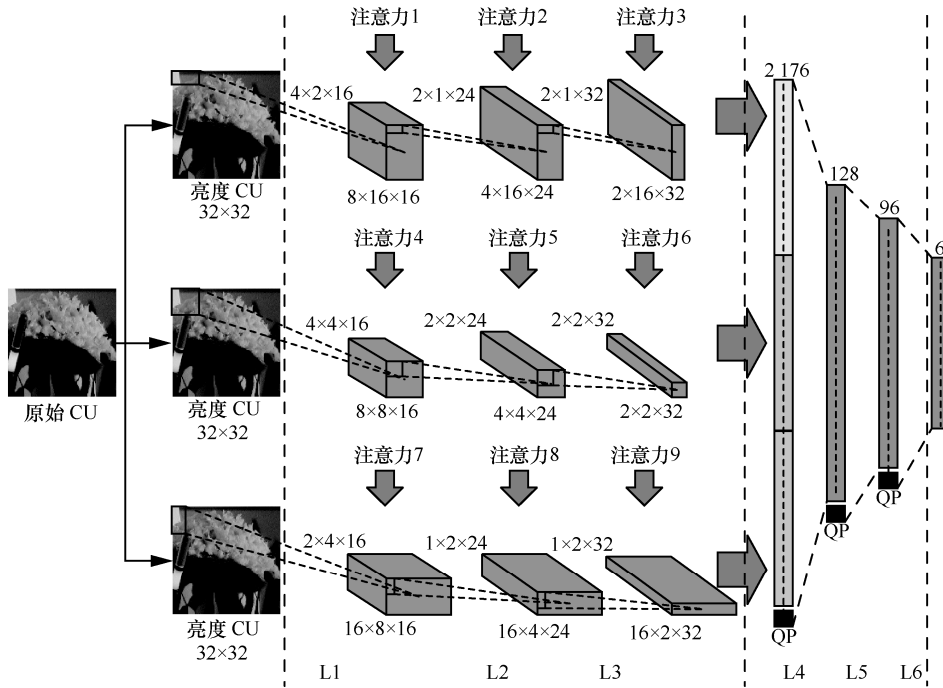


图2 AA-CNN 结构

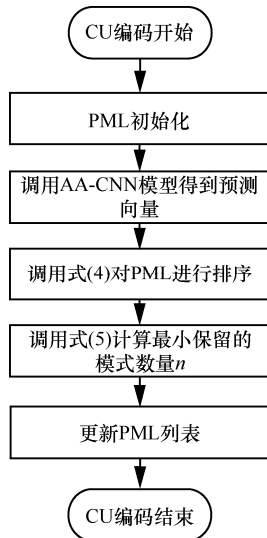


图3 快速模式决策流程

其中, x 为可保留模式数量, p_k' 为 p' 中第 k 个元素; 最后, 根据 n 更新 PML 为 L' , $L' = \{L_i | 1 \leq i \leq n\}$, L_i 为 L 中第 i 个元素。因此, 本文提出的快速编码方法仅对 n 个模式进行 RDO, $n \in [1, 6]$, 极大地降低了编码复杂度, 并可通过 φ 来控制性能的损失。

2.3.2 阈值决策

为选取最佳阈值 φ^* 来平衡编码性能与时间,

本文提出了性能与时间的代价函数。模型预测准确率 α 能表征编码性能, α 越大则性能越高; L' 的模式数量 n 能表征时间节省 (time reduction, TR), n 越小则时间节省越多。因此, 定义代价函数如下:

$$\text{Cost}(\alpha, n) = \alpha - \omega \cdot n \quad (6)$$

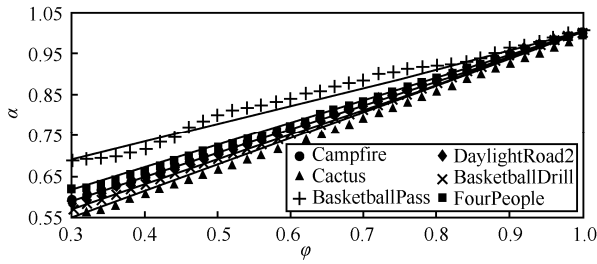
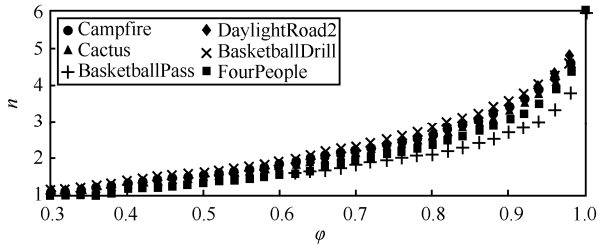
其中, ω 为两者的平衡因子, 当 $\omega < 1$ 时, 可以节省更多的时间; 当 $\omega > 1$ 时, 性能更高。

α 与 φ 直接相关, 为建模二者的关系, 本文在 QP=32、CU 大小为 32x32 的条件下对多个序列进行了统计分析。统计结果如图 4 所示, 随着 φ 增加, α 也增加, 其中实线通过线性拟合获得。显然, 对于所有序列两者均符合线性关系。因此, 本文建模 α 为关于 φ 的线性函数, 计算式如下:

$$\alpha = \mu \cdot \varphi + \nu \quad (7)$$

其中, μ 和 ν 是与训练模型和图像相关的参数。

同理, 本文对 n 与 φ 进行了统计分析。统计结果如图 5 所示, 随着 φ 增加, n 的趋势呈现指数增长。因此, 本文将 n 建模为关于 φ 的指数函数, 指数函数有 exp1 和 exp2 两种类型, 计算式

图4 α 随 φ 的变化关系图5 n 随 φ 的变化关系

如下:

$$n = a \cdot b^\varphi + c \quad \text{或} \quad n = a \cdot b^\varphi + c \cdot d^\varphi \quad (8)$$

其中, a 、 b 、 c 和 d 均为与模型和图像相关的参数。

此外, 本文给出了每个模型的拟合系数 R -square, 见表 1。对于 α , 线性函数具有非常高的拟合度; 对于 n , 显然 $\exp 2$ 模型比 $\exp 1$ 模型拟合效果更好, 本文选择 $\exp 2$ 。

表 1 R -square

序列	R -square		
	$\alpha = \mu \cdot \varphi + v$	$n = a \cdot b^\varphi + c$	$n = a \cdot b^\varphi + c \cdot d^\varphi$
Campfire	0.998 3	0.987 2	0.997 8
DaylightRoad2	0.999 5	0.989 6	0.998 8
Cactus	0.998 9	0.989 9	0.998 7
BasketballDrill	0.997 1	0.984 3	0.998 2
BasketballPass	0.982 8	0.966 1	0.995 5
FourPeople	0.999 0	0.983 2	0.997 0

因此代价函数改写为:

$$\text{Cost}(\varphi) = \mu \cdot \varphi + v - \omega \cdot (a \cdot b^\varphi + c \cdot d^\varphi) \quad (9)$$

对于最佳阈值 φ^* 的求解, 令 $\text{Cost}(\varphi) = 0$, 得:

$$\mu = \omega \cdot (a \cdot b^{\varphi^*} \cdot \ln b + c \cdot d^{\varphi^*} \cdot \ln d) \quad (10)$$

为估计式 (10) 中参数 μ 、 a 、 b 、 c 和 d , 本文考虑了序列的时域连续性。通过编码序列第一帧获得样本的预测向量和标签向量, 进而计算得到不同阈值 φ 下的 α 及 n , 最后根据式 (7) 和式 (8) 即可求解参数, 表 2 为部分序列的参数。

表 2 部分序列参数

序列	μ	a	b	c	d
Campfire	0.60	0.58	1.88	8.46×10^{-11}	23.66
DaylightRoad2	0.61	0.57	1.88	2.63×10^{-8}	18.06
Cactus	0.64	0.60	1.88	4.36×10^{-10}	21.93
BasketballDrill	0.62	0.65	1.84	2.43×10^{-8}	17.55
BasketballPass	0.46	0.59	1.63	1.50×10^{-14}	32.44
FourPeople	0.56	0.54	1.83	2.75×10^{-10}	22.66

3 实验结果与分析

3.1 实验配置

本文提出的算法在 VVC 测试平台 (VVC test model, VTM) 7.0^[29]上实现, 采用的学习模型在 TensorFlow 平台上完成, 测试条件采用标准测试配置中的全 I 帧 (all intra, AI)^[30]。测试序列共 22 个, 包括了从类 A1 到类 E 的 6 个类, 其中 A1 和 A2 为 10 bit 超高清序列, 类 B 包括了 10 bit 和 8 bit 高清序列, 类 C~类 F 为不同分辨率的 8 bit 序列。本文的算法性能通过 BDBR^[31]和平均时间节省 ATR 进行测量, ATR 为不同 QP 下 TR 的均值, 计算式如下:

$$\text{ATR} = E\{\text{TR}\} = \frac{1}{4} \sum_{i \in \Omega} \frac{T_o(i) - T_p(i)}{T_o(i)} \quad (11)$$

其中, $T_o(i)$ 和 $T_p(i)$ 分别表示在 $\text{QP}=i$ 时原始平台和所提出算法的编码时间, $\Omega = \{22, 27, 32, 37\}$ 。

3.2 实验结果

本文提出的快速决策算法可通过改变平衡因子 ω 将算法性能调整到不同档次。为了方便对比, 本文通过探索发现 $\omega=0.85$ 时算法性能与参考文献[11,23]的性能相似, $\omega=1$ 时算法的 ATR 与参考

文献[22-23]的 ATR 相似。本文共给出了 3 个不同档次的结果, 见表 3。其中, $\omega=\infty$ ($\varphi=0$) 为算法能达到最大时间节省点, “*” 表示构建训练集时采用的序列。此外, 实验结果还给出了各档次下 BDBR 和 ATR 的标准差 STD, 其大小可以反映该指标在不同序列下的稳定性。

实验结果表明, 本文算法在略微损失 RD 性能的条件下, 编码时间节省最大能达到 67.83%。同时, 被用于训练模型的序列与其他序列的实验结果高度一致, 这说明所提出算法具有良好的泛

化能力。在 $\omega=0.85$ 、1、10 不同档次下, 算法在不同序列下的 ATR 均值分别达到 48.62%、52.93% 以及 62.01%, 性能损失分别为 1.05%、1.33% 以及 2.38%。可见, 在不同档次下, ATR 和 BDBR 均可达到较好的平衡, 随着档次值的提高, 算法更倾向于时间节省, ATR 及其稳定性增加, 这表明 ATR 的增加是以牺牲 RD 性能及稳定性为代价的。此外, 本文算法对 8 bit 深度序列的效果优于 10 bit 深度序列, 因为 10 bit 深度序列具有更微小的纹理细节, 而在构建数据集时无法区分 bit 深度, 导

表 3 算法性能比较

类	序列	本文所提算法 ($\omega=0.85$)		Lei 等 ^[1] 所提 算法		Fu 等 ^[23] 所提算法		本文所提算法 ($\omega=1$)		Yang 等 ^[22] 所提算法		本文所提算法 ($\omega=\infty$)	
		BD-BR	ATR	BD-BR	ATR	BD-BR	ATR	BD-BR	ATR	BD-BR	ATR	BD-BR	ATR
A1	Tango2	1.19%	40.19%	0.97%	46.80%	1.72%	54.00%	1.52%	44.57%	0.99%	50.98%	2.96%	57.00%
	FoodMarket4	0.73%	26.41%	0.59%	45.90%	1.18%	53.00%	0.84%	27.29%	0.62%	53.53%	2.62%	44.55%
	Campfire*	1.29%	44.37%	0.81%	42.10%	1.04%	46.00%	1.70%	50.92%	0.95%	49.67%	2.84%	59.00%
A2	Catrobot1	1.37%	41.47%	1.19%	41.60%	1.51%	43.00%	1.85%	48.21%	1.05%	44.75%	3.08%	58.43%
	DaylightRoad2	1.33%	50.09%	1.17%	45.40%	1.53%	49.00%	1.84%	55.84%	0.43%	60.83%	2.81%	66.12%
	ParkRunning3*	1.34%	32.57%	0.52%	31.50%	0.54%	51.00%	1.60%	39.18%	0.82%	53.67%	2.37%	52.95%
B	MarketPlace	1.40%	45.40%	0.88%	45.10%	0.76%	52.00%	1.61%	51.18%	—	—	2.99%	64.30%
	RitualDance	1.60%	44.06%	1.10%	45.60%	1.14%	45.00%	2.18%	51.29%	—	—	3.74%	59.89%
	Cactus*	1.32%	51.16%	1.02%	41.90%	0.96%	44.00%	1.60%	56.53%	1.95%	56.66%	2.75%	65.70%
C	BasketballDrive	1.26%	51.75%	0.82%	47.20%	0.95%	49.00%	1.38%	54.86%	2.25%	64.01%	2.75%	64.87%
	BQTerrace	1.51%	54.52%	1.47%	45.50%	0.79%	48.00%	1.96%	59.63%	2.07%	56.07%	2.71%	66.99%
	BasketballDrill	1.54%	48.93%	0.82%	45.70%	1.89%	46.00%	1.84%	52.83%	2.01%	48.19%	3.56%	63.80%
	BQMall*	0.67%	57.70%	0.89%	46.90%	0.87%	39.00%	0.85%	61.26%	2.15%	55.23%	1.59%	67.43%
D	PartyScene	0.54%	56.58%	0.63%	44.20%	0.46%	42.00%	0.63%	58.53%	0.60%	45.73%	1.11%	65.08%
	RaceHorsesC	0.83%	54.58%	2.03%	56.20%	0.71%	42.00%	1.05%	59.05%	1.16%	48.39%	1.77%	65.88%
	BasketballPass*	0.71%	53.90%	0.99%	44.40%	0.70%	43.00%	0.79%	54.25%	2.33%	45.85%	1.65%	61.37%
E	Bqsquare	0.61%	56.22%	0.89%	43.10%	0.47%	44.00%	0.75%	58.24%	0.81%	46.06%	1.32%	64.18%
	BlowingBubbles	0.70%	50.63%	0.91%	47.10%	0.56%	37.00%	0.82%	53.57%	0.77%	41.56%	1.46%	60.41%
	RaceHorses	0.75%	48.76%	1.07%	42.00%	0.75%	39.00%	1.01%	52.44%	0.86%	43.17%	1.76%	59.62%
	FourPeople	0.84%	52.84%	1.42%	51.10%	1.37%	41.00%	1.00%	57.86%	2.75%	57.64%	2.08%	67.83%
	Johnny*	0.92%	53.74%	1.35%	46.20%	1.33%	39.00%	1.54%	60.83%	3.29%	58.98%	2.47%	65.32%
	KristenAndSara	0.74%	53.74%	1.20%	46.30%	1.24%	40.00%	0.99%	56.04%	2.51%	59.19%	1.94%	63.43%
	Average	1.05%	48.62%	1.03%	45.08%	1.02%	44.82%	1.33%	52.93%	1.52%	52.01%	2.38%	62.01%
	STD	0.32%	5.99%	0.25%	2.71%	0.34%	4.07%	0.42%	5.31%	0.76%	5.57%	0.61%	4.08%



致训练模型忽略了部分细节。

本文算法结果与 Lei 等^[11]、Yang 等^[22]及 Fu 等^[23]提出的算法结果进行了对比,对比数据见表3。结果表明,在 $\omega=1$ 时,本文算法无论是时间节省还是 RD 性能均优于 Yang 等^[22]提出的算法。同时,本文算法在相同时间节省情况下, RD 性能更加稳定,这是因为本文算法将模式决策建模为回归问题,通过设定阈值来实现划分模式决策,更易于控制时间节省与性能损失之间的平衡。因此,本文算法可以在节省相同时间的情况下,具有更优的 RD 性能。在 $\omega=0.85$ 时,通过与 Lei 等^[11]和 Fu 等^[23]提出的算法结果对比可知,在具有相同 RD 性能的条件下,本文算法时间节省更多。Fu 等^[23]提出的算法的时间节省与序列分辨率相关,分辨率越高时间节省越多,而本文算法除个别序列外均具有更好的时间节省和 RD 性能。虽然 Lei 等^[11]提出的算法在稳定性和算法性能上能达到较好的平衡,但时间节省更少。

本文进一步分析了所提算法与 VTM7.0 的性能比较,并给出了性能损失最大序列 RitualDance 和最小序列 PartyScene 的 RD 曲线,如图6所示。对于 RitualDance 序列,本文算法的最大性能损失仍能与原始平台保持一致,对于 PartyScene 序列,

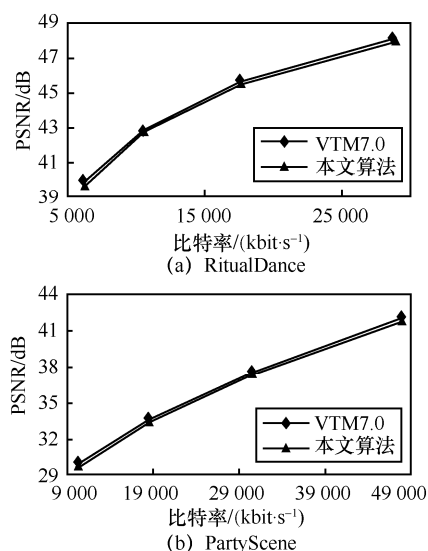


图6 RD 曲线对比

本文算法与 VTM7.0 的 RD 曲线基本重合,表明本文算法的性能损失可以忽略不计。因此,本文算法能极大地降低编码复杂度,并且性能损失与原始平台基本保持一致。

4 结束语

针对 VVC 中 QTMT 划分结构带来的高编码复杂度问题,本文提出了一种快速 QTMT 划分策略,并提供了不同档次的时间节省和性能损失多种组合的灵活调整。首先,提出了基于 AA-CNN 的模式概率预测模型和快速划分决策模型。然后,为获得平衡编码性能与时间的最佳阈值,建立了对应的代价函数,提出了阈值决策模型。最后,将所提算法整合到 VTM7.0 进行实验及分析,并与其他快速算法进行了对比。实验结果表明,本文算法最大平均节省时间达到 62.01%。

参考文献:

- [1] JCT-VC. High efficiency video coding (HEVC) text specification draft 10: JCTVC-L1003[S]. 2013.
- [2] JVET. Meeting report of the 10th JVET meeting: JVET-J1000[S]. 2018.
- [3] JVET. Algorithm description for versatile video coding and test model 2: JEV-T-K1002[S]. 2018.
- [4] 周芸, 胡潇, 郭晓强. H.266/VVC 视频编码图像划分技术研究[J]. 广播与电视技术, 2019, 46(11): 40-44.
- [5] ZHOU Y, HU X, GUO X Q. Research on image partition technology in H.266/VVC[J]. Radio & TV Broadcast Engineering, 2019, 46(11): 40-44.
- [6] JVET. AHG report: test model software development (AHG3): JVET-J0003[S]. 2018.
- [7] PAKDAMAN F, ADELIMANESH M A, GABBOUJ M, et al. Complexity analysis of next-generation VVC encoding and decoding[EB]. 2020. Arxiv: 2005. 10801.
- [8] 姚英彪, 李晓娟. 基于图像空间相关性纹理的 HEVC 块划分快速算法[J]. 电信科学, 2015, 31(1): 38-46.
- [9] YAO Y B, LI X J. Fast block partitioning algorithm for HEVC based on spatial correlation and image texture[J]. Telecommunications Science, 2015, 31(1): 38-46.
- [10] KUO Y, CHEN P, LIN H. A spatiotemporal content-based CU size decision algorithm for HEVC[J]. IEEE Transactions on Broadcasting, 2020, 1(66): 100-112.
- [11] JAMALI M, COULOMBE S. Fast HEVC intra mode decision base on RDO cost prediction[J]. IEEE Transactions on Broadcasting, 2019, 1(65): 109-122.

- [10] HUANG B, CHEN Z, CAI Q, et al. Rate-distortion-complexity optimized coding mode decision for HEVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 3(30): 795-809.
- [11] LEI M, LUO F, ZHANG X, et al. Look-ahead prediction based coding unit size pruning for VVC intra coding[C]//Proceedings of IEEE International Conference on Image Processing. Piscataway: IEEE Press, 2019: 4120-4124.
- [12] CHEN J, SUN H, KATTO J, et al. Fast QTMT partition decision algorithm in VVC intra coding based on variance and gradient[C]//Proceedings of IEEE Visual Communications and Image Processing. Piscataway: IEEE Press, 2019: 1-4.
- [13] FAN Y, CHEN J, SUN H, et al. A fast QTMT partition decision strategy for VVC intra prediction[J]. IEEE Access, 2020(8): 107900-107911.
- [14] PARK S, KANG J. Context-based ternary tree decision method in versatile video coding for fast intra coding[J]. IEEE Access, 2019(7): 172597-172605.
- [15] PARK S, KANG J. Fast affine motion estimation for versatile video coding (VVC) encoding[J]. IEEE Access, 2019(7): 158075-158084.
- [16] LIU X, LI Y, LIU D, et al. An adaptive CU size decision algorithm for HEVC intra prediction based on complexity classification using machine learning[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 1(29): 144-155.
- [17] CHEN Z, SHI J, LI W. Learned fast HEVC intra coding[J]. IEEE Transactions on Image Processing, 2020(29): 5431-5446.
- [18] KATAYAMA T, KURODA K, SHI W, et al. Low-complexity intra coding algorithm based on convolutional neural network for HEVC[C]//Proceedings of International Conference on Information and Computer Technologies. Piscataway: IEEE Press 2018: 115-118.
- [19] KIM K, RO W W. Fast CU depth decision for HEVC using neural networks[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 5(29): 1462-1473.
- [20] XU M, LI T Y, WANG Z, et al. Reducing complexity of HEVC: a deep learning approach[J]. IEEE Transactions on Image Processing, 2018, 10(27): 5044-5059.
- [21] TANG G, JING M, ZENG X, et al. Adaptive CU split decision with pooling-variable CNN for VVC intra encoding[C]//Proceedings of IEEE Visual Communications and Image Processing. Piscataway: IEEE Press, 2019: 1-4.
- [22] YANG H, SHEN L, DONG X, et al. Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 6(30): 1668-1682.
- [23] FU T, ZHANG H, MU F. Fast CU partitioning algorithm for H.266/VVC intra-frame coding[C]//Proceedings of IEEE International Conference on Multimedia and Expo. Piscataway: IEEE Press, 2019: 55-60.
- [24] AMESTOY T, MERCAT A, HAMIDOUCE W, et al. Tunable VVC frame partitioning based on lightweight machine learning[J]. IEEE Transactions on Image Processing, 2020(29): 1313-1328.
- [25] 贾川民, 赵政辉, 王苦社, 等. 基于神经网络的图像视频编码[J]. 电信科学, 2019, 35(5): 32-42.
- JIA C M, ZHAO Z H, WANG S S, et al. Neural network based image and video coding technologies[J]. Telecommunications Science, 2019, 35(5): 32-42.
- [26] WIECKOWSKI A, MA J, SCHWARZ H, et al. Fast partitioning decision strategies for the upcoming versatile video coding (VVC) standard[C]//Proceedings of IEEE International Conference on Image Processing. Piscataway: IEEE Press, 2019: 4130-4134.
- [27] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2018: 7132-7141.
- [28] CHEN Y P, DAI X Y, LIU M C. Dynamic convolution: attention over convolution kernels[C]//Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2020: 11030-11039.
- [29] JVET. Algorithm description for versatile video coding and test model 7: JEV-T-P2002[S]. 2019.
- [30] JVET. JVET common test conditions and software reference configurations: JEV-T-K1010[S]. 2018.
- [31] VCEG. Calculation of average PSNR differences between RD curves: VCEG-M33[S]. 2001.

[作者简介]



彭双 (1995—), 男, 宁波大学信息科学与工程学院硕士生, 主要研究方向为视频信号编码。



王晓东 (1972—), 男, 宁波大学信息科学与工程学院教授, 主要研究方向为视频信号处理与通信。

彭宗举 (1973—), 男, 博士, 重庆理工大学电气与电子学院教授, 主要研究方向为三维视频信号处理与编码。

陈芬 (1973—), 女, 博士, 重庆理工大学电气与电子学院教授, 主要研究方向为三维视频信号处理与编码。