



研究与开发

基于最小CU代价的AVS3帧内预测并行算法

张全¹, 王顺¹, 刘洋毅², 段昶³, 彭博¹

- (1. 西南石油大学计算机与软件学院, 四川 成都 610500;
2. 四川警察学院智能警务四川省重点实验室, 四川 泸州 646000;
3. 西南石油大学电气信息学院, 四川 成都 610500)

摘 要: 为解决第三代音视频编码标准(audio video coding standard 3, AVS3)帧内预测的耗时问题, 提出一种基于最小编码单元(coding unit, CU)代价的帧内预测并行算法。首先, 将图像划分为最小CU。然后, 利用原始像素作为参考, 并行计算所有最小CU的帧内模式代价。最后, 用代价组合的方式快速计算出其他CU的帧内模式优先级, 选择最优的15个模式进入粗略模式决策(rough mode decision, RMD)阶段。此外, 为减少方法引入的误差, 提出了3种优化策略。在预测前对原始像素进行预处理, 使其更贴合重构像素; 修改帧内预测的代价函数, 以更准确地估计每种模式的优先级; 大尺寸CU使用顶层的CU代价作为参考, 减少CU组合累积的误差。实验结果表明, 在码率仅下降0.35%的情况下, 整体编码的计算时间减少了27%, 有效地减少了帧内预测的耗时并保证了编码质量。

关键词: AVS3; 帧内预测; 并行计算

中图分类号: TP919.81

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2025027

AVS3 intra frame prediction parallel algorithm based on minimum CU cost

ZHANG Quan¹, WANG Shun¹, LIU Yangyi², DUAN Chang³, PENG Bo¹

1. School of Computer Science and Software Engineering, Southwest Petroleum University, Chengdu 610500, China
2. Intelligent Policing Key Laboratory of Sichuan Province, Sichuan Police College, Luzhou 646000, China
3. School of Electrical Engineering and Information, Southwest Petroleum University, Chengdu 610500, China

Abstract: To address the time-consuming issue of audio video coding standard 3(AVS3) intra frame prediction, an intra frame prediction parallel algorithm based on the cost of the minimum coding unit (CU) was proposed. Firstly, the images were divided into the smallest CUs. Secondly, the original pixels were used as references to calculate the intra-mode cost of all smallest CUs in parallel. Finally, the intra-mode priorities of other CUs were quickly calculated by means of cost combination, and the optimal 15 modes were selected to enter the rough mode decision (RMD) stage.

收稿日期: 2024-11-25; 修回日期: 2024-12-10

通信作者: 刘洋毅, liuyangyi_ioe@163.com

基金项目: 智能警务四川省重点实验室开放课题 (No.ZNJW2023KFZD003, No.ZNJW2024KFMS003)

Foundation Items: The Open Project of Intelligent Policing Key Laboratory of Sichuan Province (No.ZNJW2023KFZD003, No.ZNJW2024KFMS003)



Additionally, three optimization strategies were proposed to reduce the errors caused by the proposed method. The original pixels were preprocessed before prediction to better approximate reconstructed pixels; the cost function of intra-frame prediction was modified to more accurately estimate the priority of each mode; and for larger CUs, the cost of the top-level CU was used as a reference to reduce cumulative errors during CU combination. Experimental results demonstrate that while the code rate is reduced by only 0.35%, the overall encoding time is reduced by 27%, effectively reducing the time required for intra frame prediction and ensuring the encoding quality.

Key words: AVS3, intra prediction, parallel computing

0 引言

第三代音视频编码标准 (audio video coding standard 3, AVS3) 是中国自主研发的最新一代音视频编码标准, 相较第二代音视频编码标准 (audio video coding standard 2, AVS2), 引入了多项新的编码技术, 使得视频压缩率显著提升, 码率节省了约 30%^[1]。然而, 这些技术的引入也大幅度地增加了编码的计算复杂度^[2], 导致编码器的实际应用受到一定限制。计算复杂度的急剧上升成为 AVS3 广泛应用的主要障碍之一。因此, 如何在保证压缩率和视频质量的前提下, 降低 AVS3 编码的计算复杂度, 已成为当前视频编解码技术领域的一个重要研究方向。近年来, 许多研究人员都对视频编解码的快速算法进行了深入研究, 这些方法大致可以分为对编码单元 (coding unit, CU) 的划分进行提前终止、对编码技术 (如预测算法) 进行优化和利用多线程并行计算的能力优化编码任务的分配。

在 CU 划分策略方面, Chen 等^[3]利用图像纹理变化对 CU 划分深度进行预测, 跳过了纹理简单区域的复杂划分, 减少了不必要的计算负担。然而, 这种方法在复杂场景下可能存在精度下降的风险。Chen 等^[4]基于 CU 的梯度和方差评估其复杂度, 并跳过低复杂度的 CU 划分, 该策略通过引入图像统计特征提高了计算效率。Xu 等^[5]采用频域纹理提取指导 CU 划分, 针对特定频域信息做出更高效的分块决策。Song 等^[6]应用深度学习技术预测 CU 的最佳深度, 但训练过程和模型

推理的计算开销较大。罗小龙等^[7]利用相邻 CU 划分信息对当前 CU 的划分进行预测, 结合阈值策略, 进一步优化了计算流程。陈燕辉等^[8]提出了一种基于 CU 子块方向特性与空间复杂度的快速划分算法, 利用 CU 中间区域与边缘区域的纹理复杂度对划分进行提前决策。Hoangvan 等^[9]设计了一种提前终止划分的卷积神经网络来预测划分。

在预测算法优化方面, Wang 等^[10]提出了一种基于 CU 纹理方向选择帧内预测模式的方法, 能够显著减少模式遍历的复杂度。Ren 等^[11]利用相邻 CU 的相关性来提高预测模式的参考价值, 减少了模式搜索的次数。Wu 等^[12]采用决策树优化仿射预测计算量, 但此方法的效果依赖于决策树的构建精度。Cao 等^[13]通过直接使用非首编码 CU 的第一预编码结果, 减少了编码时间, 但在一定程度上牺牲了压缩率。Huang 等^[14]提出了一种基于离散正余弦变换的快速率失真代价计算方法, 显著地降低了编码成本的复杂度。Zhang 等^[15]通过自适应运动矢量分辨率优化运动估计复杂度, 有效地减少了编码时间。

此外, 有一些研究者将编码任务重新划分, 利用多线程的方式并行地处理每个编码任务, Chen 等^[16]提出了一种基于波前的并行方案, 重新定义了块的编码顺序, 为并行算法修改了更合适的编码方案。Papaioannou 等^[17]在波前并行的基础上对更细粒度的层面进行并行, 并权衡了 2 种方案的有效结合点, 进一步提升了并行性。Chen 等^[18]及 Cai 等^[19]分别基于 FPGA 异构加速和硬件

架构优化实现帧内编码的加速,但其方法依赖于特定硬件,灵活性相对较低。

综上所述,现有研究多聚焦于对编码过程的剪枝,然而在AVS3复杂的划分和预测模式下,剪枝法容易导致较大误差。基于硬件的并行方案虽然能够提升编码速度,但缺乏通用性和灵活性。为此,本文提出一种基于最小CU代价的AVS3帧内预测并行算法,主要贡献如下。

(1) 针对AVS3的编码耗时问题,本文提出了一种基于代价组合的帧内预测并行算法。先将图像划分成若干最小CU,每个最小CU使用原始像素作为参考像素进行帧内预测,打破CU间的顺序依赖,使其更适应并行化处理,在较低质量损失的情况下显著降低了编码复杂度。在进入粗略模式决策(rough mode decision, RMD)阶段前,利用预先计算的最小CU代价,通过自底向上组合的方式快速计算当前CU的帧内模式优先级。该方法重点关注帧内模式代价排序的准确性,而非代价计算的精确度,通过提前筛选出代价更低的模式进入RMD阶段,减少了RMD模式遍历的开销。

(2) 对本文的并行方案提出3种减少误差的方法。首先,本文针对原始像素与重构像素之间的差异问题,提出了一种预处理方法。在使用原始像素构建参考像素前,先对原始像素进行预处理,使其更贴合重构像素,从而降低使用原始像素带来的误差;其次,针对原始代价函数在代价组合中对预测模式代价的冗余累加问题,提出了一种新的代价函数,能更准确地估计每种模式的优先级;最后,针对利用最小CU计算大块CU代价会有较大误差的问题,对CU进行分类,在计算大块CU代价时使用其顶层的CU代价作为参考,减少组合最小CU带来的误差。

1 AVS3帧内预测

相比于AVS2,AVS3进一步扩展了CU的尺寸范围,将最小尺寸从 8×8 缩小至 4×4 ,最大尺

寸从 64×64 增加至 128×128 。同时,在原有的二叉树(quad tree, QT)划分基础上,新增了水平和垂直方向的扩展二叉树(extended quad tree, EQT)划分,以及水平和垂直方向的一元树(binary tree, BT)划分,AVS3划分方式如图1所示。此外,AVS3将AVS2中的33种帧内预测模式扩展至66种^[20],AVS3帧内预测模式如图2所示。

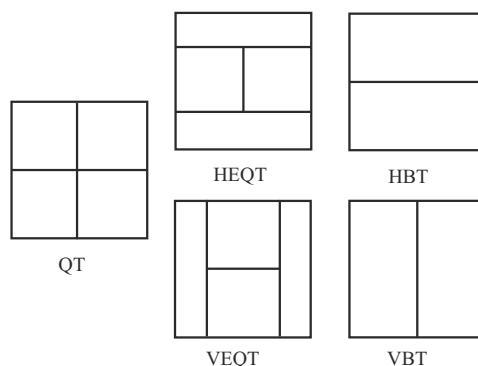


图1 AVS3划分方式

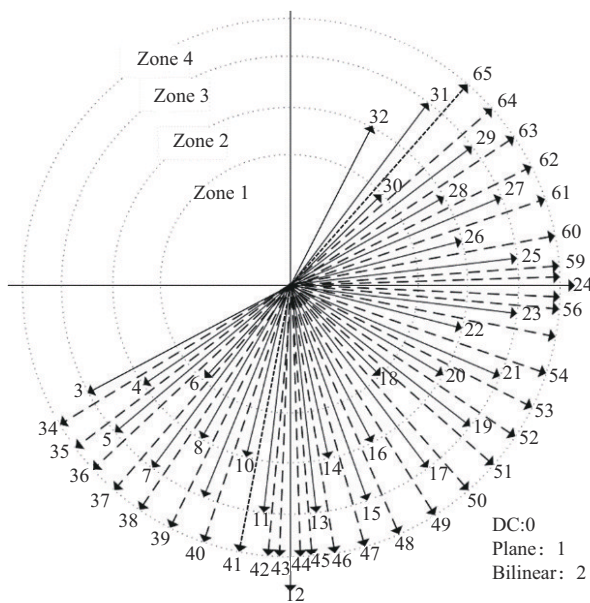


图2 AVS3帧内预测模式

在AVS3的编码过程中,每一帧首先被划分成尺寸为 128×128 的最大编码单元(largest coding unit, LCU),然后递归地将每个LCU进一步划分为更小的CU。在每次划分过程中,会尝试不同的划分树结构,并对每个CU进行预测,在



帧内预测阶段会遍历所有帧内预测模式, 计算出每一种模式的率失真代价 (rate-distortion cost, RD-Cost), 最后根据 RD-Cost 自底向上选择出最佳的编码方案。尽管这些复杂的划分结构和丰富的预测模式显著地提高了编码质量, 但也大幅增加了算法的计算复杂度, 因此有必要对帧内预测研究快速算法以降低编码的复杂度。

2 并行帧内预测算法

本文方法首先将一帧图像划分为最小 CU, 在异步环境下使用原始像素作为参考像素, 并行计算并存储每个最小 CU 的 66 种帧内预测模式代价。主线程按照原始编码流程进行编码操作, 在帧内预测的 RMD 阶段前, 通过组合当前编码 CU 所包含的最小 CU 的模式代价来快速获取当前 CU 的模式代价, 选择最优的模式进入 RMD 阶段, 以此降低 RMD 阶段计算的耗时。同时, 本文针对为实现并行而使用原始像素作为参考像素产生的误差、代价函数对模式代价的冗余累加问题以及通过组合最小 CU 计算大块 CU 代价的误差问题进行分析和解决。

2.1 基于代价组合的并行方案

在 AVS3 的编码流程中, CU 在帧内预测阶段会先通过 RMD 计算 66 种帧内模式的代价, 然后选择代价最小的 3 个模式进入率失真优化 (rate-distortion optimization, RDO) 阶段, 在 RDO 阶段对 3 个模式进行更精确的代价计算, 选出代价最小的模式作为当前 CU 最终的预测模式。RMD 阶段并不需要准确地计算每种模式的代价, 只需要获取每种模式代价的排序, 选择代价更低的模式进入 RDO 阶段即可。因此, 本文利用 AVS3 以 CU 为单位进行帧内预测这一特性, 利用大 CU 所包含的小 CU, 对大 CU 进行帧内模式优先级的估计。大 CU 每种模式的预测代价等于其包含的所有小 CU 对应模式代价之和, 依次计算出大 CU 66 种模式代价并进行排序, 选择代

价较低的模式进入 RMD 阶段。以此减少 RMD 阶段的遍历开销。在 AVS3 中, 最小 CU 的尺寸为 4×4 , 因此, 本文将图像划分成 4×4 的 CU 进行帧内预测, 并组合 4×4 的 CU 获取其余尺寸 CU 的帧内模式代价。此外, 采用并行的方式对一帧内所有 4×4 CU 进行帧内预测, 这能够极大地减少帧内预测的耗时。

然而, 在 AVS3 标准中, CU 的编码需要按照特定顺序执行, 当前 CU 的参考像素依赖于左侧和上方已编码的 CU, 这给 CU 间的并行带来了挑战。为解决这一问题, 本文首先采用原始像素作为所有 CU 的参考像素, 以打破参考依赖, 实现 CU 间的并行处理。但由于原始像素与重构像素的差异, 这也引入了误差, 这一问题将在第 2.2 节中详细探讨。

在编码器开始对一帧图像进行编码时, 首先将该帧划分为多个 4×4 的最小 CU。使用 OpenMP (open multi-processing) 技术, 利用多核 CPU 对所有 CU 并行执行帧内预测, 并存储每个 4×4 CU 的 66 种帧内模式代价。需要注意的是, 4×4 CU 的并行处理采用异步执行模式。在此过程中, 编码器将继续进行数据预处理和 CU 划分, 直到第一块 CU 开始执行帧内预测。并行执行的结果在该时刻进行同步, 以确保后续处理的正确性。

基于代价组合的并行帧内预测算法如图 3 所示, CU 不再按照传统流程遍历所有预测模式, 而是按模式号累加其包含的所有 4×4 CU 帧内代价。最终, 将这些累加结果按照代价从低到高排序, 形成当前 CU 的帧内模式代价表。当前 CU 将直接参考该代价表, 选择代价最低的 n 个模式进入 RMD 阶段。当 n 的取值足够大时, 改进的帧内预测结果将与标准帧内预测完全一致, 但加速效果不明显。通过实验验证, 本文将 n 的初始取值设定为 15, 并在 HPM15.0 (AVS3 参考软件) 上全 I 帧配置下进行测试, 测试量化参数 (quantization parameter, QP) 序列为 27、32、38、45, 初始并

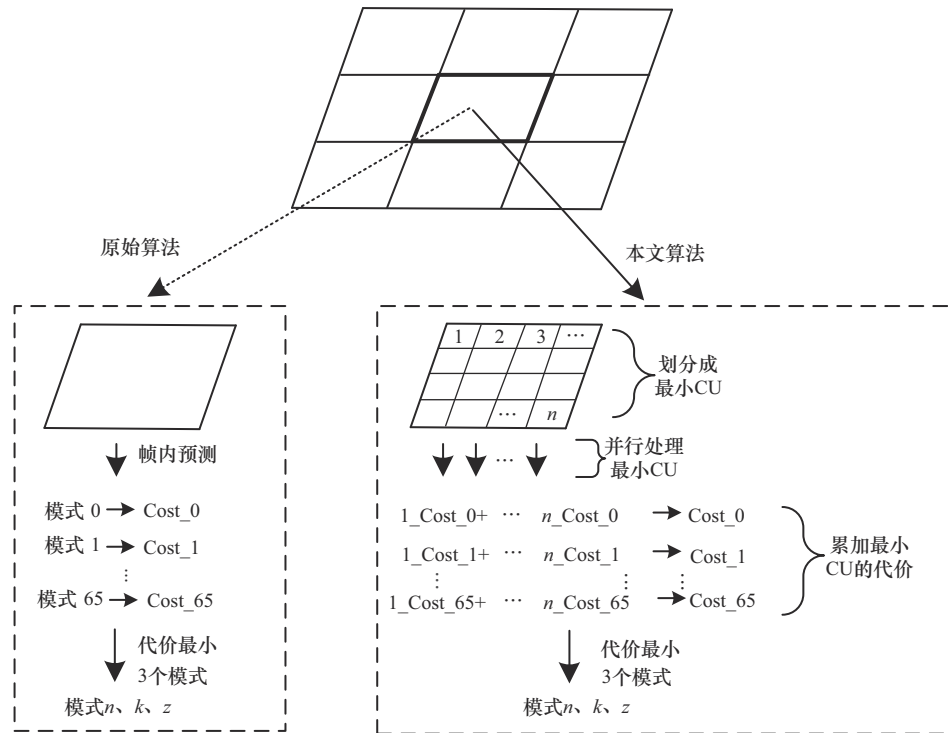


图3 基于代价组合的并行帧内预测算法

行算法结果见表1, 其中, BD-Rate (Bjontegaard delta rate)^[21]是编码性能的衡量指标, 其正值代表编码性能损失, ΔT 表示本文方法相较于原始算法在整体耗时上节省的百分比。可以看出本文算法减少了27%的耗时, 耗时减少的原因主要是本文算法减少了RMD阶段遍历的模式数量, 但在一定程度上增加码率, 码率上涨的分析和优化将在第2.2、2.3、2.4节中详细阐述。

表1 初始并行算法结果

视频序列	BD-Rate	ΔT
ParkScene	1.23%	27%
BasketballDrive	1.32%	28%
Vidyo3	1.17%	26%
Flowervase	1.28%	27%
BQSquare	1.26%	29%

2.2 参考样本的优化

为打破参考像素依赖问题, 本文采用原始像素作为CU的参考像素。然而, 这种做法可能引

入误差, 尤其在QP较大时, 高频信息的丢失会导致样本变得更加模糊。为减少这种误差, 本文尝试了2种处理原始像素的方法: 一种是直接对原始图像进行预处理, 如低通滤波或去除高频信息; 另一种是在构造参考样本时, 对每个CU使用的原始像素进行优化。

针对第一种方法, 本文分别使用低通滤波、下采样和离散余弦变化 (discrete cosine transform, DCT) 后再量化的方法对原图进行预处理, 原图预处理结果如图4所示。其中, 重建图是根据AVS3编码过程中构建参考像素的方法重建的结果, 作为第一种方法的参考标准, 可以看出低通滤波、下采样、DCT变化后再量化的方法均和重建图有较大差距, 且用其作为参考像素的编码质量低于直接使用原图的编码质量。其原因在于对原始帧进行预处理时, 每个像素的值取决于周围所有像素, 且一次计算出整张参考图像, 而在标准编码流程中, CU的参考像素仅取决于左侧和上侧已编码的像素。此外, 即使是相同位置的

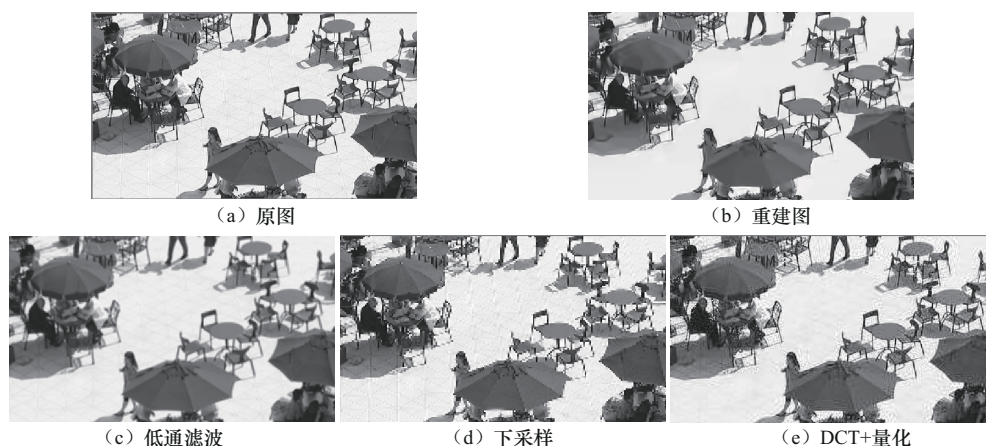


图4 原图预处理结果

CU，由于编码顺序不同，参考像素也会有所差异，即每个CU参考像素的计算是根据CU的编码顺序逐步计算得到的。因此，直接对原始帧进行预处理无法有效贴合原始编码流程中计算的参考像素，且如果所采用的处理方法过于复杂，还会带来额外的计算开销。

相较之下，第二种方法更符合原始编码流程中参考像素的计算方式，即在每个CU进行帧内预测前计算其参考像素。参考像素的计算如图5所示，在使用上侧参考像素进行预测时，对CU上方4行的原始像素值进行加权求和得到新的一行像素替代重构像素作为参考像素。使用4行像素的原因是过少的行数无法充分利用邻域像素，而过多的行数会增加计算复杂度，且本文采用 4×4 的CU作为并行单位，过多的行数会使得靠近图像边界的CU在获取原始像素数时需要额外处理。4行原始像素的权重分配基于与待预测像素的距离，距离越近的像素影响越大，因此赋予较高权重，左侧参考像素的计算则遵循相同的原理。实验表明，按照由近到远权重递减的规律设置的权重值均能在第2.1节初始方案基础上获得收益，在多组权重组合的测试中，由近到远 $(1/2, 1/5, 1/5, 1/10)$ 的分配在降低BD-Rate方面表现最优，因此，本文采用该权重组合优化参考像素。

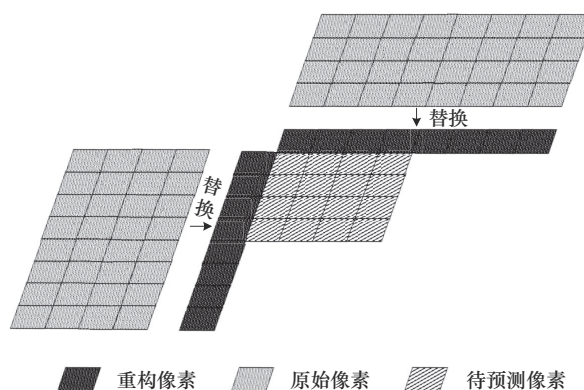


图5 参考像素的计算

优化参考样本后的结果见表2，展示了在第2.1节初始方案的基础上增添本节的优化方案后得到新的实验结果。结果显示，相较于初始方案，优化参考样本能够显著地提升编码质量。这是因为本文参考像素的优化遵循了AVS3原始编码流程中参考像素的计算流程，为每一个CU单独计算参考像素，且使用邻近像素加权平均，较好地模拟了原始编码流程中原始像素和重建像素的差异。

表2 优化参考样本后的结果

视频序列	BD-Rate	ΔT
ParkScene	0.74%	26%
BasketballDrive	0.87%	28%
Vidyo3	0.68%	26%
Flower vase	0.71%	27%
BQSquare	0.65%	28%

2.3 代价函数的优化

帧内预测阶段会选出帧内模式中 RD-Cost 最小的预测模式，RD-Cost 的计算如式 (1) 所示。

$$J = D + \lambda R \quad (1)$$

其中， J 为率失真代价， D 为原始块和预测块之间计算的失真，RDO 使用差值平方和 (sum of squared difference, SSD) 来表示 D ， R 为编码所需要的比特数， λ 为拉格朗日因子。RDO 的计算是非常耗时的，因此在 RDO 前会先对所有模式使用 RMD 进行粗选，RMD 阶段代价的计算复杂度更低，RMD 粗选后再对代价最小的 3 个模式执行 RDO。

由式 (1) 可知，CU 帧内模式的编码代价由 2 部分组成，一是预测值和原始值的差距，反映了使用当前模式预测得到的图像质量；二是当前模式的编码代价，反映了要得到当前预测值需要编码的额外信息量。在基于代价组合的并行方案中，每个 CU 都需要计算 66 种模式的预测代价，每种模式的预测代价由当前 CU 包含的所有 4×4 CU 对应模式预测代价累加得到，如式 (2) 所示。

$$J = \sum_{i=0}^m (D_i + \lambda_i R_i) \quad (2)$$

其中， m 表示当前 CU 包含的 4×4 CU 数量。

通过式 (2) 得到当前 CU 所有帧内模式的代价并进行排序，根据排序结果选择进入 RMD 的模式。但式 (2) 仍然有需要优化的地方，在基于组合的方案中，代价函数中像素差值 D 的累加是符合逻辑的，因为当前 CU 是由 4×4 CU 组合而成，像素差值的累加代表当前 CU 不同区域差值的累加。模式的编码代价的累加则是冗余的，因为模式代表的是根据参考像素得到预测值的一种计算方式，并不跟 CU 的尺寸有直接联系，因此，将代价的计算方式更改为式 (3)。

$$J = \sum_{i=0}^m D_i + \lambda R \quad (3)$$

在修正的方案中，CU 通过组合方式计算每个帧内模式的像素差 D ，然后结合当前 CU 的拉格朗日因子 λ 和基于经验估计的当前模式编码所需的比特数 R 计算当前模式的代价。为了验证优化代价函数对降低 BD-Rate 的有效性，本文在第 2.1 节初始方案的基础上增添本节的优化方案，优化代价函数后的结果见表 3。结果显示，相较于初始方案，优化代价函数能明显地提升编码质量，这是因为优化后的代价函数能更准确地根据 4×4 CU 的模式代价计算当前 CU 模式代价的优先级，从而更精准地选择代价更低的模式进入后续编码阶段。

表 3 优化代价函数后的结果

视频序列	BD-Rate	ΔT
ParkScene	0.72%	28%
BasketballDrive	0.66%	28%
Vidyo3	0.71%	27%
FlowerVase	0.67%	27%
BQSSquare	0.69%	28%

2.4 代价组合方案的优化

在帧内预测阶段，CU 会以最大 64×64 的尺寸开始预测。在以代价组合的方式获取 64×64 CU 的帧内预测代价时，需要统计其包含的 256 个 4×4 CU。当需要进行代价组合的 4×4 CU 数量较多时，由于每个 4×4 CU 在组合的时候都会因为参考像素的不同等产生一定误差，帧内模式代价排序的准确性会受到一定影响。但这种误差在尺寸较小的 CU 的代价组合中是被允许的，因为较少的误差并不会影响帧内模式的排序；但对于大尺寸的 CU 而言，数百个 4×4 CU 间误差的累加会在一定程度上影响帧内模式的排序，从而导致代价最低的模式没有被选中。为了减少上述误差，本文对使用代价组合方法的 CU 尺寸进行限制，组合方式优化方案如图 6 所示，当 CU 长 $\times$ 宽 $\geq 32 \times 32$



时, 视其为大尺寸CU, 否则视为小尺寸CU。小尺寸的CU将使用4×4的CU组合帧内预测代价, 而大尺寸CU直接使用其顶层64×64 CU的帧内预测结果, 具体方法为: 在编码前将图像划分为64×64 CU, 并行执行每一个64×64 CU的帧内预测, 预测结果将直接用于后续出现的大尺寸CU。在第2.1节初始方案的基础上增加本节方案进行实验, 优化组合方案后的结果见表4。相较于初始方案, 优化代价组合方案能进一步提高了编码质量, 原因在于其减少了使用4×4 CU获取大尺寸CU帧内预测代价时引入的误差。

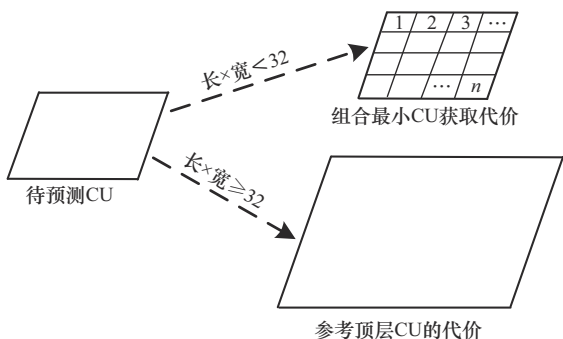


图6 组合方式优化方案

表4 优化组合方案后的结果

视频序列	BD-Rate	ΔT
ParkScene	0.98%	28%
BasketballDrive	1.03%	29%
Vidyo3	1.01%	27%
Flowervase	0.94%	28%
BQSquare	1.12%	29%

3 实验结果与分析

将本文提出的基于代价组合的并行帧内预测方法和3种优化方案相结合, 合并算法流程如图7所示, 每一帧图像在编码开始前都会被异步地进行4×4和64×64的CU划分, 同一尺寸下的CU会被多线程并行地执行帧内预测。主线程仍然按照原始编码流程进行编码, 当执行第一个CU的帧

内预测时, 异步执行的计算结果将被同步到主线程, 根据当前CU的尺寸选择代价计算方式。当CU长×宽<32×32时, 通过组合4×4 CU的帧内代价获取当前CU的帧内预测代价, 当CU长×宽≥32×32时, 使用顶层64×64 CU的帧内代价获取当前CU的帧内预测代价。算法在HPM15.0上进行测试, 测试QP序列为27、32、38、45, 实验设备参数为CPU: Intel(R) Core(TM) I7-8750H, GPU: NVIDIA GeForce GTX 1050Ti。本文算法与HPM15.0对比结果见表5, ΔT 定义如式(4)所示。

$$\Delta T = \frac{1}{4} \sum_{QP_i} \frac{T_{HPM(i)} - T_{prop(i)}}{T_{prop(i)}} \times 100\% \quad (4)$$

其中, $T_{HPM(i)}$ 表示HPM15.0在第*i*个QP下的编码整体耗时, $T_{prop(i)}$ 表示本文的算法在第*i*个QP下的编码整体耗时。

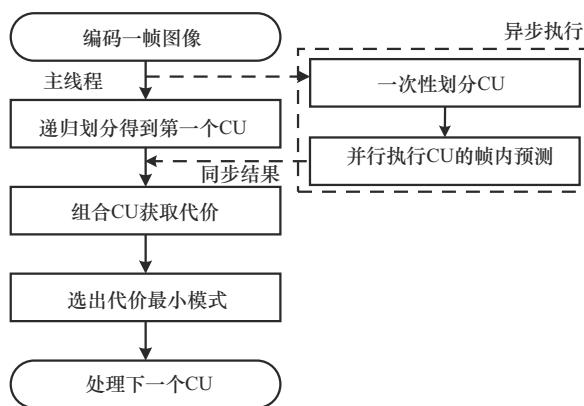


图7 合并算法流程

从表5可以看出, 本文方法相较于HPM15.0, 在平均0.35%的码率损失下, 获得了27%的耗时节省。序列BasketballDrive RD曲线如图8所示, 可以看出, 在视频BasketballDrive上, 本文算法与HPM15.0算法的RD曲线几乎重叠, 说明两者编码率失真性能近似。此外, 为直观地展示解码后图像的视觉效果, 本文在视频BasketballDrive上将提出的方法和HPM15.0的方法进行对比, 解码图像对比如图9所示, 从视觉

效果来看, 两者差异较小, 表明本文方法对图像质量影响较小。

表5 本文算法与HPM15.0对比结果

分辨率/像素	视频	BD-Rate	ΔT
3 840×2 160	Tango2	0.34%	27%
	FoodMarket4	0.33%	27%
	Campfire	0.36%	28%
	CatRobot	0.32%	27%
	DaylightRoad2	0.31%	26%
2 560×1 600	PeopleOnStreet	0.41%	29%
	Traffic	0.33%	27%
1 920×1 080	ParkScene	0.37%	28%
	BasketballDrive	0.36%	28%
	Cactus	0.31%	27%
	MarketPlace	0.35%	28%
	RitualDance	0.31%	27%
1 280×720	Vidyo3	0.39%	28%
	Vidyo4	0.33%	26%
	FourPeople	0.37%	27%
	Johnny	0.35%	27%
832×480	FlowerVase	0.32%	27%
	PartyScene	0.34%	27%
416×240	BQSquare	0.36%	28%
	Keiba	0.38%	28%
平均		0.35%	27%

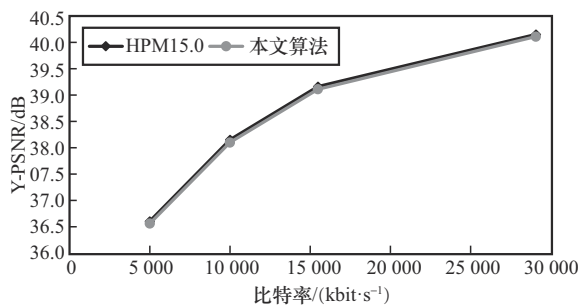


图8 序列BasketballDrive RD曲线



(a) HPM15.0 (b) 本文方法

图9 解码图像对比

为进一步验证本文方法的有效性, 将其与文献[4]、文献[5]以及文献[9]进行对比分析, 本文算法与其他算法对比结果见表6。实验采用 ΔT /BD-Rate作为参考指标, 该指标表示在编码性能损失1%的情况下所能节省的编码时间。结果表明, 在综合考虑BD-Rate和 ΔT 的情况下, 本文算法的 ΔT /BD-Rate值达到了77.14, 显著高于其他对比方法。这说明本文算法在相同的码率增加条件下能够显著节省编码时间, 展现了相较其他方法的优势。

4 结束语

本文提出了一种新的帧内预测优化算法, 该算法通过将一帧图像划分成最小的 4×4 CU, 然后利用多线程并行计算每个CU的帧内代价。同时, 主线程继续编码任务, 并在帧内编码开始前同步计算结果, 利用代价组合快速计算每块CU的帧内预测代价。在此基础上, 进一步提出3种优化策略, 从参考像素的选取、代价计算和组合方法上提高了算法的编码速度。实验结果表明, 该算法在码率仅增加0.35%的情况下, 将编码耗时减少了27%, 相较于其他最新方法展现出显著的优势。然而, 本文算法仍有进一步改进空间, 在未来的研究工作中, 可考虑根据CU纹理对CU进行分类, 为每种纹理类型的CU设计不同的优化策略, 以减少误差。此外, 在并行处理方面, 可以引入细粒度更高的帧内模式级别的并行方案, 以提高算法的速度。通过这些改进和研究有望使该帧内预测算法在性能上得到更大的提升。



表6 本文算法与其他算法对比结果

视频	文献[4]		文献[5]		文献[9]		本文算法	
	BD-Rate	ΔT	BD-Rate	ΔT	BD-Rate	ΔT	BD-Rate	ΔT
Tango2	0.96%	37%	1.12%	24%	1.15%	27%	0.34%	275%
FoodMarket4	0.84%	35%	1.31%	32%	1.24%	29%	0.33%	27%
PeopleOnStreet	1.21%	42%	0.23%	30%	1.01%	28%	0.41%	29%
Traffic	1.18%	42%	0.64%	25%	0.76%	26%	0.33%	275%
ParkScene	0.60%	43%	0.35%	22%	0.61%	35%	0.37%	28%
BasketballDrive	1.60%	49%	0.51%	25%	0.71%	29%	0.36%	28%
Vidyo3	0.80%	40%	0.58%	33%	1.24%	28%	0.39%	28%
FourPeople	0.77%	39%	0.46%	23%	1.17%	33%	0.37%	27%
FlowerVase	1.20%	48%	0.42%	26%	1.15%	32%	0.32%	27%
PartyScene	0.82%	47%	0.44%	25%	1.21%	34%	0.34%	27%
BQSquare	0.10%	34%	0.62%	31%	0.82%	25%	0.36%	28%
Keiba	0.85%	42%	0.37%	25%	0.93%	28%	0.385	28%
平均	0.91%	42%	0.58%	27%	1.01%	31%	0.35%	274%
ΔT /BD-Rate	46.15		46.55		30.69		77.14	

参考文献:

- [1] 张嘉琪, 雷萌, 马思伟. AVS3 视频编码关键技术及应用[J]. 中兴通讯技术, 2021, 27(1): 10-16.
ZHANG J Q, LEI M, MA S W. Key technologies and applications of AVS3 video coding standard[J]. ZTE Technology Journal, 2021, 27(1): 10-16.
- [2] FAN K, CAI Y G, GAO X S, et al. Performance and computational complexity analysis of coding tools in AVS3[C]//Proceedings of the 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). Piscataway: IEEE Press, 2020: 1-6.
- [3] CHEN M F. Improved algorithm for intra-frame prediction based on AVS3[C]//Proceedings of the 2022 International Conference on Informatics, Networking and Computing (ICINC). Piscataway: IEEE Press, 2022: 242-245.
- [4] CHEN C X, LUO X J, YAN Y Y, et al. A fast CU partition decision strategy for AVS3 intra coding[C]//Proceedings of the 2022 IEEE International Conference on Consumer Electronics (ICCE). Piscataway: IEEE Press, 2022: 1-5.
- [5] XU C G, WU Y, CHEN L, et al. Fast CU partition decision algorithm for AVS3 based on frequency domain[C]//Proceedings of the 2022 21st International Symposium on Communications and Information Technologies (ISCIT). Piscataway: IEEE Press, 2022: 195-198.
- [6] SONG W J, LI C X, ZHANG Q W. Fast CU partitioning algorithm for VVC based on CNN and FSVM[J]. IEEE Access, 2024, 12: 52150-52160.
- [7] 罗小龙, 刘子龙. 基于快速编码单元划分和自适应模式决策的 HEVC 效率改进算法[J]. 电子科技, 2020, 33(10): 21-25, 39.
LUO X L, LIU Z L. HEVC efficiency improvement algorithm based on fast coding unit partition and adaptive mode decision[J]. Electronic Science and Technology, 2020, 33(10): 21-25, 39.
- [8] 陈燕辉, 李强, 董阳, 等. 基于 CU 特征差异的 VVC 帧内快速划分算法[J]. 计算机应用研究, 2024, 41(3): 939-943.
CHEN Y H, LI Q, DONG Y, et al. Intra-frame FAST division algorithm of VVC based on CU feature difference[J]. Application Research of Computers, 2024, 41(3): 939-943.
- [9] HOANGVAN X, NGUYENQUANG S, DINHBAO M, et al. Fast QTMT for H.266/VVC intra prediction using early-terminated hierarchical CNN model[C]//Proceedings of the 2021 International Conference on Advanced Technologies for Communications (ATC). Piscataway: IEEE Press, 2021: 195-200.
- [10] WANG Y Z, ZHANG C B, SUN S L. Intra prediction fast algorithm in AVS3 based on image texture characteristics[C]//Proceedings of the 2021 20th International Symposium on Communications and Information Technologies (ISCIT). Piscataway: IEEE Press, 2021: 6-10.
- [11] REN R, CHEN L. Fast algorithm of video coding based on AVS3[C]//Proceedings of the 2020 IEEE 6th International Con-

- ference on Computer and Communications (ICCC). Piscataway: IEEE Press, 2020: 1827-1831.
- [12] WU J C, SUN S L, ZHANG J Q, et al. Decision tree based early termination algorithm for affine prediction in AVS3[C]//Proceedings of the 2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP). Piscataway: IEEE Press, 2023: 1-6.
- [13] CAO X Y, LIN T, ZHAO L P, et al. Fast algorithm based on non-first pre-coding skip for AVS3 intra coding[C]//Proceedings of the 2023 8th International Conference on Signal and Image Processing (ICSIP). Piscataway: IEEE Press, 2023: 259-263.
- [14] HUANG B, CHEN Z F, SU K X, et al. Low-complexity rate-distortion optimization for HEVC encoders[J]. IEEE Transactions on Broadcasting, 2021, 67(3): 721-735.
- [15] ZHANG Y K, XIANG G Q, LI C, et al. A fast adaptive motion vector resolution algorithm for AVS3[C]//Proceedings of the 2024 IEEE International Conference on Consumer Electronics (ICCE). Piscataway: IEEE Press, 2024: 1-5.
- [16] CHEN K J, SUN J, DUAN Y Z, et al. A novel wavefront-based high parallel solution for HEVC encoding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 26(1): 181-194.
- [17] PAPAIOANNOU G I, KOZIRI M G, PAPADOPOULOS P K, et al. Tile based wavefront parallelism in HEVC[C]//Proceedings of the 2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA). Piscataway: IEEE Press, 2020: 1-5.
- [18] CHEN W J, HE Q F, LI S, et al. Parallel implementation of H.265 intra-frame coding based on FPGA heterogeneous platform[C]//Proceedings of the 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). Piscataway: IEEE Press, 2020: 736-743.
- [19] CAI Z Y, GAO W. Efficient fast algorithm and parallel hardware architecture for intra prediction of AVS3[C]//Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE Press, 2021: 1-5.
- [20] 雷萌, 罗法蕾, 王苕社, 等. CE2-related: 帧内角度模式扩展[C]//数字电视音视频编解码技术标准化工作组第七十次会议. 海口: AVS 工作组, 2019: 17-21

LEI M, LUO F L, WANG S S, et al. CE2-related: extension of intra angular modes[C]// Proceedings of the 70th Meeting of the Audio Video Coding Standard Workgroup of China. Haikou: AVS Workgroup, 2019: 17-21.

- [21] BJØNTEGAARD G. Calculation of average PSNR differences between RD-curves: ITU-T VCEG-M33[R]. 2001.

[作者简介]



张全 (1985-), 男, 博士, 西南石油大学计算机与软件学院副教授, 主要研究方向为视频编码、图像处理、并行计算和计算机视觉。



王顺 (1999-), 男, 西南石油大学计算机与软件学院硕士生, 主要研究方向为视频编码、并行计算。



刘洋溢 (1989-), 博士, 四川警察学院智能警务四川省重点实验室副教授, 主要研究方向为图像处理、计算机视觉。



段昶 (1979-), 男, 西南石油大学电气信息学院讲师, 主要研究方向为视频编解码、图像处理、计算机视觉。



彭博 (1980-), 男, 西南石油大学计算机与软件学院教授, 主要研究方向为图像处理、计算机视觉、并行计算。