



研究与开发

基于 SIMD 的 AVS3 并行率失真优化量化算法

唐毅欣¹, 黄晓峰¹, 唐然¹, 周洋¹, 崔燕², 殷海兵¹

(1. 杭州电子科技大学通信工程学院, 浙江 杭州 310018;

2. 浙江省经济信息中心, 浙江 杭州 310006)

摘要: 针对第三代音频视频标准 (the third generation audio video coding standard, AVS3) 中的率失真优化量化 (rate-distortion optimization quantization, RDOQ) 过程的编码效率问题, 提出一种基于单指令多数据 (single instruction multiple data, SIMD) 的并行 RDOQ 算法。首先, 在最优系数决策 (optimal coefficient decision, OCD) 阶段, 通过优化, 仅保留扫描线内的依赖关系。然后, 在最后一个非零位置决策 (last non-zero position decision, LNPd) 阶段, 基于分治策略, 通过将 Zig-Zag 扫描线分解成多个独立扫描线, 实现每条扫描线上最优系数位置的并行计算。最后, 采用 SIMD 指令进行并行加速, 以提高整个率失真优化量化的计算效率。实验结果表明, 该算法在 AI 配置下, 可以降低 29.46% 的编码时间, 而 BD-Rate 仅损失 0.25%。

关键词: AVS3; 率失真优化量化; 单指令多数据; 并行加速

中图分类号: TN919.81

文献标志码: A

doi: 10.11959/j.issn.1000-0801.2024155

SIMD-based parallelized rate distortion optimized quantization for AVS3

TANG Yixin¹, HUANG Xiaofeng¹, TANG Ran¹, ZHOU Yang¹, CUI Yan², YIN Haibing¹

1. School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China

2. Zhejiang Economic & Information Centre, Hangzhou 310006, China

Abstract: To improve the coding efficiency of rate-distortion optimization quantization (RDOQ) for the third-generation audio video coding standard (AVS3), a parallelized RDOQ algorithm based on single instruction multiple data (SIMD) was proposed. Firstly, in the optimal coefficient decision (OCD) stage, the dependencies within the scan line were retained through optimization. Secondly, in the last non-zero position decision (LNPd) stage, the Zig-Zag scan line was split into various independent lines based on the partitioning strategy. The optimal coefficients on each scan line were allowed to be calculated in parallel by the proposed method. Finally, SIMD instructions were utilized for parallelized acceleration to boost calculation efficiency in the RDOQ process. Experimental results show that the

收稿日期: 2024-01-27; 修回日期: 2024-04-08

通信作者: 黄晓峰, xfhuang@hdu.edu.cn

基金项目: 国家科技重大专项 (No.2021ZD0109802); 国家自然科学基金资助项目 (No.61901150)

Foundation Items: The National Science and Technology Major Project(No.2021ZD0109802), The National Natural Science Foundation of China (No.61901150)

proposed algorithm can achieve a 29.46% coding time reduction with only 0.25% BD-Rate loss for AI configuration.

Key words: AVS3, RDOQ, SIMD, parallelized acceleration

0 引言

超高清 (ultra-high definition, UHD) 视频编码的出现为用户更加丰富的视觉体验提供了技术支持^[1]。目前, 互联网 70% 以上的信息来源于图像和视频。但第二代音频视频标准 (the second generation audio video coding standard, AVS2) 已经不能满足人们在实际应用中的需求。因此, 中国数字音视频编解码技术标准工作组制定了新一代视频标准^[2]。AVS3 与 AVS2 相比, 编码效率有了显著的提高。

AVS3 在 AVS2 的基础上, 采用了几种更加先进的编码工具。例如, 与 AVS2 相比, AVS3 采用了更加灵活的块划分结构, 将块进一步划分为多个变换单元和预测单元, 同时还采用了基于历史的运动矢量预测^[3]。部分学者围绕如何利用这些编码工具降低码率和计算复杂度展开研究。Huang 等^[4]主要采用传统的基于块的混合编码方法, 在块结构的设计上取得一定进展。Fu 等^[5]主要提出一种改进的环路滤波算法, 提高了编码效益。姚英彪等^[6]提出基于图像空间相关性与纹理的高效视频编码 (high efficiency video coding, HEVC) 块划分快速算法, 降低了编码时间。祁伟等^[7]和孟翔等^[8]通过使用统计建模的方式降低模式决策复杂度。

除了以上研究, RDOQ 也是 AVS3 中一项非常关键的技术, 它旨在通过量化过程中权衡失真和码率, 以提高编码效率。一般而言, 量化优化算法主要分为两大类: 旁路跳过 RDOQ^[9]和快速 RDOQ^[10-13]。旁路跳过 RDOQ 算法主要通过提前决策判断是否进入 RDOQ 函数, 减少 RDOQ 函数调用次数以降低计算复杂度; 快速 RDOQ 算法旨在简化 RDOQ 中的复杂运算, 从而降低 RDOQ 计

算量。

对于旁路跳过 RDOQ 算法的研究, Xu 等^[14]和 Wang 等^[15]提出一种基于全零块的算法来跳过不必要的 RDOQ 过程, 即当一个变换块内所有变换系数的和小于某个值时, 选择性地跳过 RDOQ 过程。Wang 等^[16]研究一种硬决策量化方法, 设计一个基于贝叶斯分类原理的死区偏移模型来近似得出 RDOQ 的输出值。Wang 等^[17]、Lee 等^[18]和 Yin 等^[19]建立自适应量化模型来模拟 RDOQ 的过程。

对于快速 RDOQ 算法的研究, Yang 等^[20]分析了零块中的变换系数特征, 设计一种自适应零块检测模型。Yin 等^[21]提出一种在两个候选量化值之间进行精确码率估计的高效 RDOQ 算法, 有效地降低了计算复杂度。Cui 等^[22]提出通过对混合拉普拉斯分布的变换系数的统计量进行建模, 以避免复杂的率失真 (rate-distortion, RD) 代价计算。He 等^[23]通过直接计算 RD 代价差的一种快速计算算法来有效节约量化时间。Canh 等^[24]提出利用神经网络, 通过预量化值来预测 RDOQ 输出值的方法。

尽管上述文献提出一些通过优化 RDOQ 过程中的算法来降低复杂度和码率的方法, 但尚未涉及 RDOQ 的并行化过程。Igarashi 等^[25]提出一种在 GPU 上的双向并行扫描算法, 在一定程度上可以完成 RDOQ 过程的并行化, 但造成了较大的编码损失。Zhao 等^[26]提出一种基于 Z 型 (Zig-Zag) 扫描线的快速算法, 对 RDOQ 过程进行了并行优化, 但也造成了一定的编码损失。因此, 针对上述问题, 本文提出一种快速 RDOQ 算法, 通过并行化和 SIMD 加速来降低 RDOQ 计算复杂度, 其主要贡献如下。

- 在 OCD 阶段, 采用一种基于扫描线并行



的率失真优化量化算法来构建局部依赖,打破相邻数据的依赖性,并将候选系数固定。

- LNP阶段,基于分治策略将Zig-Zag扫描线拆分成多个单独的扫描线,实现每条扫描线最优系数的并行计算。
- 对于整个RDOQ过程,采用SIMD指令进行并行处理,并使计算效率与性能损失有一个良好的折中。

1 RDOQ过程

RDOQ是一种软量化决策技术,其原理是将率失真优化(rate-distortion optimization, RDO)算法与量化过程结合,从而对码率和失真进行权衡。AVS3中RDOQ流程如图1所示,在AVS3中,RDOQ过程主要有以下4个阶段:预量化阶段、OCD阶段、LNP阶段及零值设置阶段。

1.1 预量化阶段

在预量化阶段,RDOQ基于量化参数(quantization parameter, QP)为变换块(transform block, TB)中每个变换系数执行预量化操作,以获得初始量化系数 l_i ,预量化计算式如下。

$$l_i = \text{round}\left(\frac{c_i}{Q_{\text{step}}} + f\right) \quad (1)$$

其中, c_i 为变换系数, Q_{step} 为量化步长,与QP相关,round表示向下取整操作, f 为量化的偏移量。

1.2 OCD阶段

在OCD阶段,候选量化值需要基于RD代价进行比较以确定每个量化系数的最优值,候选值推导见表1,显示了量化系数与候选量化值之间的关系。

表1 候选值推导

$ u_i $	候选值
0	0
2	0, 1, 2
$\vdots$	$\vdots$
N	0, N , $N-1$

按照Zig-Zag扫描顺序遍历TB中的每一个系数,Zig-Zag扫描如图2(a)所示。

针对每个系数,对每个候选的量化值计算其RD代价,选择RD代价最小的一个作为扫描索引 i 上最优量化系数 l_i^* ,具体计算式如下。

$$l_i^* = \underset{k=0, \dots, m-1, m}{\operatorname{argmin}} \{D(c_i, l_{i,k}) + \lambda \times R(l_{i,k})\} \quad (2)$$

其中, D 表示失真, R 表示候选量化值对应的码率,与依赖run和level值相关, c_i 为Zig-Zag扫描顺序中第 i 个变换系数, λ 为拉格朗日算子, $l_{i,k}$ 表示预量化系数 l_i 对应的第 k 个候选量化值。

1.3 LNP阶段及零值设置阶段

在LNP阶段,TB块中的系数按照Zig-Zag扫描顺序遍历,通过计算累加RD代价与先前的系数比较,选择两者中RD代价较低的位置作为最后一个非零位置。重复该过程,直到扫描到TB中最后一个系数。

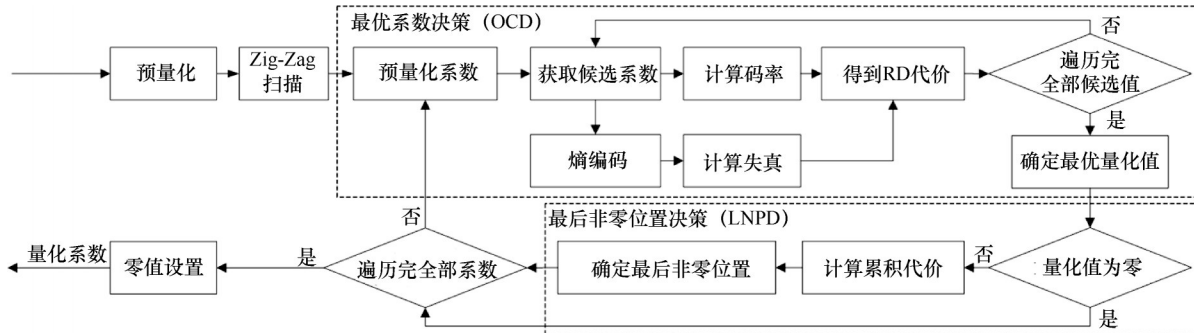


图1 AVS3中RDOQ流程

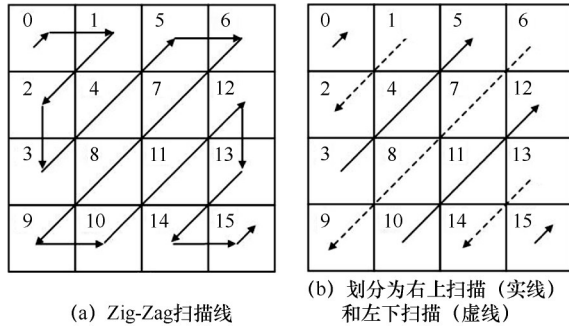


图2 扫描方式示意图

在零值设置阶段,将最后一个非零系数位置后的系数全部设为零,然后输出最终的量化系数。

2 基于扫描线并行的RDOQ算法

原始的RDOQ过程只能实现单输入单输出,串行过程大幅度增加了编码时间。为了能够采用SIMD指令实现算法加速,需要对RDOQ算法进行并行优化。若直接在RDOQ算法中执行列扫描并行处理,由于系数间上下文依赖的影响,编码时必然会造成较大的损失。基于上述问题,本节提出了一种基于扫描线并行的RDOQ算法。在并行OCD决策中,固定了部分扫描线依赖值,进一步减小性能损失。在并行LNPD决策中,基于一种分治策略将Zig-Zag扫描线划分成多条单独的扫描线,并同时处理多个扫描线,通过并行计算每条扫描线上局部最优位置以确定整个转换单元(transform unit, TU)中的最优非零位置。

2.1 OCD并行化

依赖值在计算RD代价时起到比较重要的作用,其中最为重要的依赖值是run和level。level表示量化后的系数值,run表示两个非零系数之间零的个数,run和level值会对后续的RD代价产生影响。

在依赖保留过程中,由于按列输入的影响,部分坐标块无法获得前一个依赖值,因此提出一种基于扫描线并行的优化算法来实现并行化处

理,即将Zig-Zag扫描线划分为右上扫描线和左下扫描线。具体流程如图2(b)所示。

对于右上扫描方向 $((u+v) \bmod 2 = 0)$ 上的坐标点,最初的run和level均可以通过初始化获得,即处理每个系数时,会根据它在矩阵中水平方向上的前一个系数的值进行计算。利用run和level进行最优系数决策过程可获得最优系数。在下次循环时,会将前一循环获得的最优系数作为level值输入,具体根据式(4)获得新的level值,run值依据式(3)获得。

$$\text{run}_i = \begin{cases} 0 & , (u+v) \bmod 2 = 0 \text{ 且 } l_i \neq 0 \\ 1 & , (u+v) \bmod 2 = 1 \\ \text{run}_{i-1} + 1, & (u+v) \bmod 2 = 0 \text{ 且 } l_i = 0 \end{cases} \quad (3)$$

$$\text{level}_i = \begin{cases} 5 & , (u+v) \bmod 2 = 1 \text{ 且 } l_i = 0 \\ |l_i| & , (u+v) \bmod 2 = 1 \text{ 且 } l_i \neq 0 \\ |l_{i-1}^*| & , (u+v) \bmod 2 = 0 \end{cases} \quad (4)$$

其中, (u,v) 为扫描索引值 i 的坐标指数,run_{*i*}为扫描索引值 i 上的run值,level_{*i*}为扫描索引值 i 上的level值, l_i 为式(1)中的扫描索引值 i 上的预量化系数, l_{i-1}^* 为式(2)中计算的上一个非零最优系数。

对于左下扫描方向 $((u+v) \bmod 2 = 1)$ 上的坐标点,数据依赖性是在并行性的障碍。由于扫描线是固定从左往右进行的,因此根据扫描的方向,当前坐标点无法获得前一个坐标点对应的run和level。以对角线为例进行说明,划分为右上扫描(实线)和左下扫描(虚线),如图2(b)所示:在索引值9处首先进行列扫描,随后扫描索引值8处,但索引值8处的依赖值需要索引值7处提供,所以依赖不能满足并行列扫描的要求,因此需要提前对run和level值进行赋值。run根据式(3)赋值,level根据式(4)赋值。这种并行运算模式确保了run和level计算都可以独立进行,减少了对于整个矩阵扫描次序的依赖,提高了算法的并行处理能力。



通过优化算法, 数据依赖关系仅仅存在于扫描线内, 即同一列中的所有行系数都可以并行处理, 为后续的SIMD指令并行处理做了准备。

2.2 LNPDP并行化

在此部分的处理中, 采用分治策略, 将Zig-Zag扫描线拆分成多个单独的扫描线, 各个求解, 分而治之, 在降低计算量的同时提高了计算效率。

对于每条扫描线, 参与计算的数据主要包括: 最后一个位置横纵坐标last_X和last_Y, 最后一个非零位置标志代价 J_{last_one} , 扫描线到当前点的累加代价 J_{cur_pos} , 从扫描线开始到最优非零位置的代价 J_{end_pos} 。

对于扫描线上的RD代价计算, 右上扫描线前后位置直接通过累加RD代价比较, 当 $\Delta J < 0$ 时, 则认为当前位置最优。具体计算式如下。

$$\Delta J = (J_{cur_pos}^k + J_{last_one}^k) - (J_{end_pos}^{k-1} + J_{last_one}^{k-1}) \quad (5)$$

其中, k 和 $k-1$ 是扫描线上相邻的两个量化系数, k 表示当前系数。

而左下扫描线无法直接通过累加RD代价进行比较, 因此当计算RD代价时, 需要采取一种差值比较方法计算, 从而满足正向累加RD代价比较仍然符合“当 $\Delta J < 0$ 时, 则当前位置最优”的判断条件。先假设整个扫描线的累加代价为 J_{all} , 它是扫描线上所有系数的RD代价之和, 为恒定值。在从右上往左下逐个扫描扫描线上的各位置点时, 会得到累加代价 J_{cur_pos} , 然后通过 $J_{all} - J_{cur_pos}$ 就得到了正向(右上方向)累加代价。具体计算式如下。

$$\Delta J = \left[J_{all} - (J_{cur_pos}^k + J_{last_one}^k) \right] - \left[J_{all} - (J_{end_pos}^{k-1} + J_{last_one}^{k-1}) \right] \quad (6)$$

经推导得:

$$\Delta J = (J_{end_pos}^{k-1} + J_{last_one}^{k-1}) - (J_{cur_pos}^k + J_{last_one}^k) \quad (7)$$

2.3 并行化处理的总体架构

在完成预量化过程之后, 采用列扫描顺序对预量化系数进行并行计算。每次计算多组预量化系数数据, 在多组数据比较下确定最优量化系数。之后对非零位置进行并行决策, 确定扫描线上最优非零位置。最后进行末尾系数置零操作。

基于扫描线并行的RDOQ算法流程如下。

输入 变换系数 c_i

输出 量化系数

步骤1 根据式(1)计算预量化系数 l_i 。

步骤2 确定式(3)和式(4)中run和level值。

步骤3 将run和level值代入式(2)中确定最优量化系数。

步骤4 执行步骤2和步骤3, 确定TB中最优量化系数。

步骤5 根据式(5)和式(7)确定当前扫描线上的最优非零位置。

步骤6 执行步骤5, 确定TB中最后一个非零位置。

步骤7 最后一个非零位置后的量化系数全部设为零。

对于并行架构中改进的RDOQ, 基于扫描线并行的RDOQ流程如图3所示。

综上, 本节实现了RDOQ算法并行优化, 为SIMD并行处理奠定了基础。

3 SIMD指令并行处理

SIMD指令并行是一种计算机并行技术, 它可以在同一时刻使用单个指令对多个数据执行相同的操作, 从而实现高效的并行计算^[27]。为了提高RDOQ算法的计算效率, 本文基于第2节提出的基于扫描线并行的RDOQ算法, 采用Intel中SIMD内联函数对RDOQ并行算法进行加速优化。部分常用SIMD优化函数指令见表2。

3.1 预量化阶段

在预量化阶段, RDOQ基于式(1)推导出

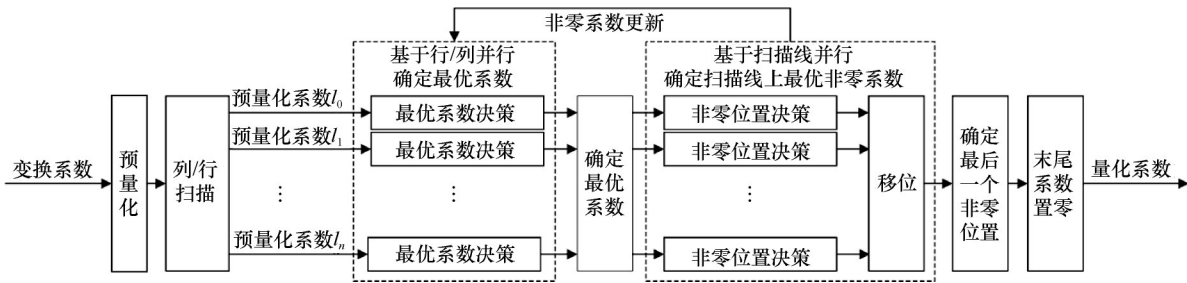


图3 基于扫描线并行的RDOQ流程

表2 部分常用SIMD优化函数指令

函数指令	指令作用描述
<code>_mm_setr_epi32</code>	创建一个包含4个32位整数的SSE128寄存器
<code>_mm_mullo_epi32</code>	将a, b中对应的4个32位整数进行乘法运算
<code>_mm_min_epi32</code>	将a, b中对应的4个32位整数进行比较, 选取每对整数中较小的输出
<code>_mm_add_epi32</code>	将a, b中对应的4个32位整数进行加法运算
<code>_mm_blendv_epi8</code>	根据mask的值选择a, b中的相应元素进行混合并返回结果

每个变换系数的量化系数。但在实际的量化过程中, 需要通过缩放比较来确保量化后的系数值不超过32位整型的范围, 最终推导计算式如下:

$$|l_i| = (\min(|c_i| \times q_value, \text{Min}_{\text{mum}})) \gg q_bits \quad (8)$$

$$\text{Min}_{\text{mum}} = \text{INT32_MAX} - (1 \ll (q_bits - 1)) \quad (9)$$

$$l_i = \text{sign}(c_i) \times |l_i| \quad (10)$$

其中, l_i 为 Zig-Zag 扫描顺序中第 i 个量化系数, c_i 为 Zig-Zag 扫描顺序中第 i 个变换系数, INT32_MAX 为32位最大值的常数, q_bits 为位宽和TB大小所确定的数, q_value 为QP和TB大小所确定的数。

基于SIMD的预量化如图4所示。 q_bits 和 q_value 在图4中分别用 qb 和 v 表示。TB中竖直方向上连续4个变换系数被加载到寄存器中用于计算。本文具体采用了 `_mm_setr_epi32` 函数, 将指定地址的连续4个32位整数存入一个SSE128位寄存器中。对预量化部分的SIMD优化主要围绕式(8)进行, 本文通过使用 `_mm_mullo_epi32`、`_mm_min_epi32`、`_mm_add_epi32` 等函数同时计算出4个量化系数的绝对值。最后使用 `_mm_blendv_epi8` 函数, 根据变换系数的正负计

算得到最终的量化系数。同时, 在实际量化过程中, 由于 q_bits 、 q_value 的值仅与位宽、TB和QP大小有关, 即在同一个TB中 q_bits 、 q_value 、 Min_{mum} 的值保持不变, 因此选择将上述3个数提前存储至寄存器中, 减少每次遍历时对于数据的加载。

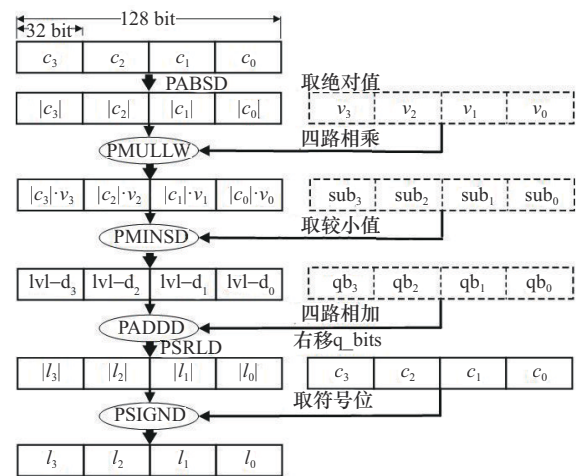


图4 基于SIMD的预量化

3.2 OCD阶段

该阶段要通过码率和失真来计算每个候选量化值的RD代价, 根据RD代价在候选量化值中选取最优量化系数。首先要计算出每个候选量化值

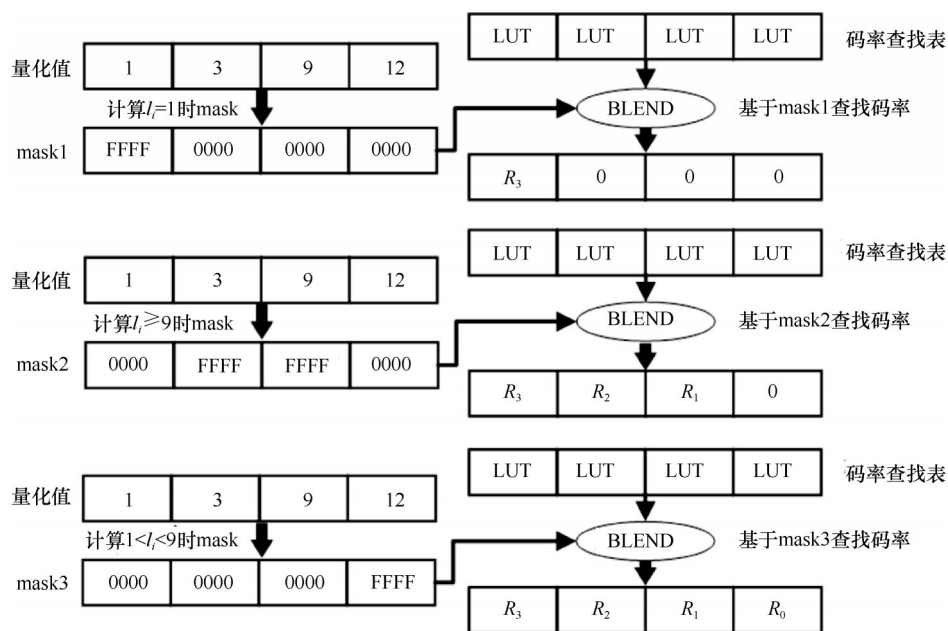


图5 量化值非零时码率计算流程

未编码的RD代价，即量化值为零时的RD代价。在实际量化过程中，当量化值为零时，采用一般量化计算RD代价；当量化值不为零时，采用RDOQ计算RD代价。码率计算的流程较为复杂，因此在HEVC参考软件HM或者AVS3参考软件HPM中都用了—个表格来代替真实的熵编码过程，即查找表。在码率计算中，根据run值选择查找表中的码率。本文通过使用`_mm_cmpeq_epi32`等函数将run值与其他值进行计算和比较，并将结果保存在mask寄存器中，然后使用`_mm_blendv_epi8`函数根据mask寄存器中的值选择查找表中的码率。量化值非零时码率计算流程如图5所示，图5中用 R_i 表示码率，以量化值[1, 3, 9, 12]为例。由于RD代价的最大位宽为64位，但码率的最大位宽只有32位，因此在计算中利用`_mm256_cvtepi32_epi64`函数将码率的最大位宽扩展为64位。在失真计算中，通过使用`_mm256_mul_epi32`、`_mm256_srli_epi32`等函数进行计算，失真计算流程如图6所示，图6中的Dist代表失真， $l_{vl}-d_i$ 代表量化过程产生的误差值，scale代表失真权重。

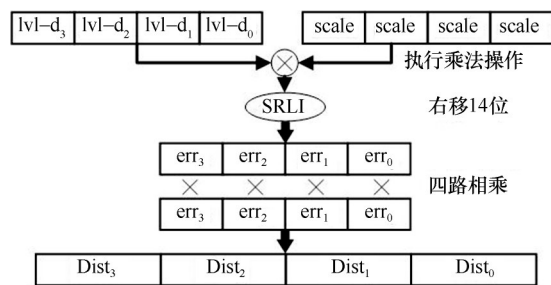


图6 失真计算流程

在OCD过程中，由表1可知，需要分别计算量化值为 $|I_i|-1$ 、 $|I_i|$ 、0的RD代价，并选择RD代价较小的一个作为最优量化系数。但在本文提出的并行算法中，为了避免冗余计算，将码率计算拆分为两种情况：4个量化系数均为0；量化系数不为零。全零检测流程如图7所示。最优系数的比较过程如图8所示，根据比较得到的最小RD代价来确定最优量化系数。在全零检测中利用

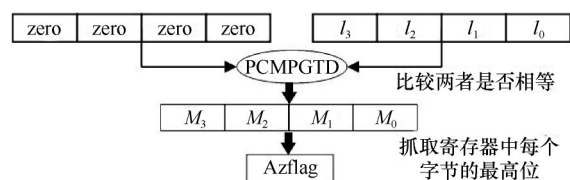


图7 全零检测流程

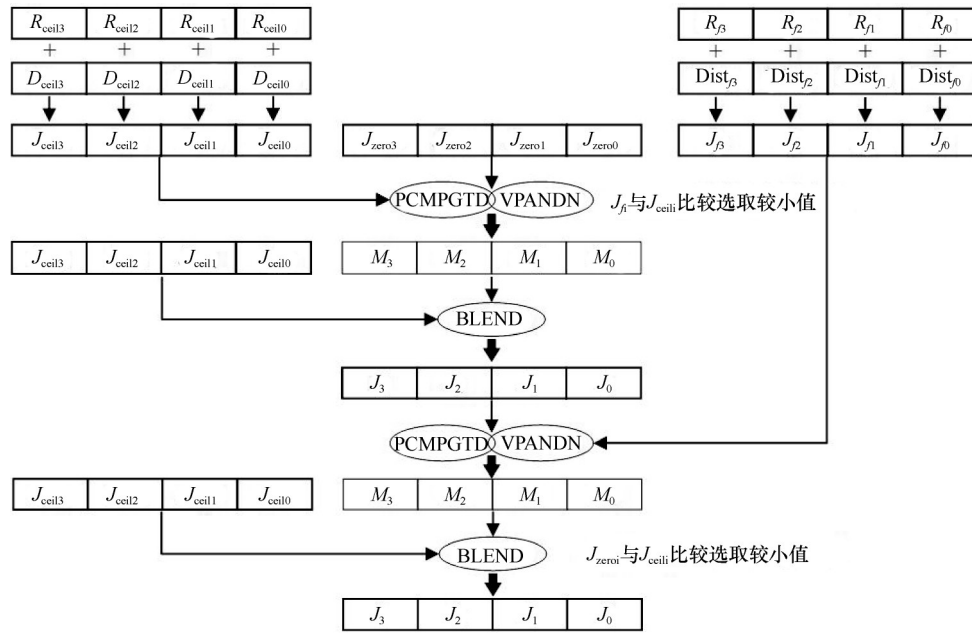


图8 最优系数的比较过程

`_mm_movemask_epi8` 函数提前对存入寄存器的4个量化系数进行判断。若全为零,则不需要进行两次比较来确定最优系数,即跳过图5和图8的计算过程。

在OCD过程中,最棘手的部分是计算候选值的RD代价,在寄存器中存着4个并不完全相同的量化系数,这对并行化处理造成了一定的困难。虽然在上文采取了一定措施来降低并行化的困难,但计算过程仍然存在较大的冗余。由表1可知,量化系数大于2,需要比较2次RD代价,量化系数为2,需要比较1次,等于零则不比较;同时在SIMD指令中不存在具有最小值比较功能的指令。综上所述,在该算法中默认进行两次比较获得RD代价最小的量化系数。用SIMD选择最优系数的例子如图9所示,第1次选择量化值大于零的RD代价与未编码的RD代价比较,第2次再将两者中的较小值与量化值大于1的RD代价比较。需注意图9中虚线箭头所指向的候选值不参与比较。

3.3 LNP阶段及零值设置阶段

在LNP阶段仍然是对4个数据进行并行操

作,在该阶段中的关键步骤是数据旋转操作,因为在进行决策时,需要将同一扫描线上的数据在寄存器中对齐。旋转操作的实现流程如图10所示,展示了4路扫描线上相邻数据进行RD代价比较。由于在x86平台下没有指令具有旋转功能,

候选值	比较过程				条件
l_i	2	0	5	1	1st: $l_i > 0$
0	0	0	0	0	
l_{i-1}	1	-1	4	0	2nd: $l_i > 1$

图9 用SIMD选择最优系数的例子

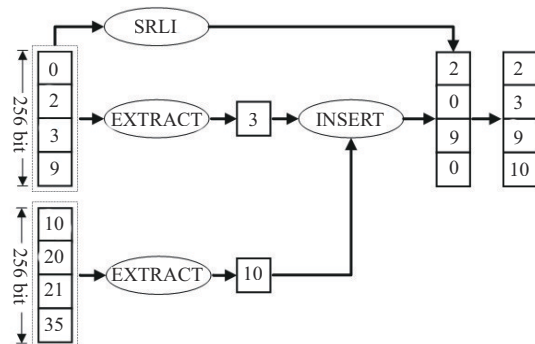


图10 旋转操作的实现流程



因此在该算法中采用了 `_mm256_srli_si256` 函数、`_mm256_extract_epi64` 函数、`_mm256_insert_epi64` 函数来实现数据旋转操作。

4 实验结果

为了评估基于 SIMD 的并行 RDOQ 算法的 RD 性能和时间复杂度,在测试平台 Intel(R) Core (TM) i9-9900K CPU@3.60 GHz 上,使用 AVS3 参考软件 HPM-4.0 进行测试。测试条件遵循 AVS3 通用测试条件^[28]中的 AI (all intra)、RA (random access) 和 LDB (low delay B) 配置。测试了 23 个序列,QP 值分别为 27、32、38 和 45。并行 RDOQ 算法的计算复杂度由时间变化率衡量,编码性能由 BD-Rate^[29]衡量,其负值代表编码性能增益,具体计算式如下。

$$TS_Q = \frac{\text{Time}_{\text{RDOQ}} - \text{Time}_{\text{proposedQ}}}{\text{Time}_{\text{RDOQ}}} \times 100\% \quad (11)$$

$$TS_{\text{Enc}} = \frac{\text{Time}_{\text{ENC}} - \text{Time}_{\text{proposed}}}{\text{Time}_{\text{ENC}}} \times 100\% \quad (12)$$

其中, TS_Q 和 TS_{Enc} 分别为 RDOQ 过程节省的时间占原始 RDOQ 时间的比例和总编码过程节省的时间占原始总编码时间的比例, $\text{Time}_{\text{RDOQ}}$ 和 $\text{Time}_{\text{proposedQ}}$ 分别为原始的 RDOQ 时间和提出的并行算法的 RDOQ 时间, Time_{ENC} 和 $\text{Time}_{\text{proposed}}$ 分别为原始的总编码时间和提出的并行算法的总编码时间。

在 HPM-4.0 平台和 AI、RA 和 LDB 配置下,本文方法的编码结果见表 3。表 3 中,数据包括每个类别中所有视频序列的平均节省时间和编码性能。

表 3 展示了本文在 AI、RA 和 LDB 配置下,

表 3 本文方法的编码结果

类	序列	AI			RA			LDB		
		BDBR	TS_Q	TS_{Enc}	BDBR	TS_Q	TS_{Enc}	BDBR	TS_Q	TS_{Enc}
UHD	Tango2	0.16%	32.36%	9.81%	0.26%	30.56%	7.55%	0.28%	29.92%	8.00%
	Parkrunning3	0.09%	31.12%	8.10%	0.19%	26.04%	6.09%	0.19%	27.64%	7.07%
	Campfire	0.13%	30.42%	8.49%	0.35%	28.21%	7.42%	0.36%	28.32%	6.82%
	DaylightRoad2	0.08%	31.49%	8.31%	0.22%	28.55%	7.25%	0.17%	28.95%	6.95%
	Average	0.12%	31.35%	8.68%	0.26%	28.34%	7.08%	0.25%	28.71%	7.21%
HD	Cactus	0.35%	30.75%	8.22%	0.24%	26.03%	6.47%	0.30%	28.21%	6.65%
	BasketballDrive	0.46%	31.27%	8.32%	0.23%	27.52%	6.51%	0.34%	27.38%	6.45%
	MarketPlace	0.16%	32.01%	10.26%	0.44%	29.69%	7.23%	0.20%	31.04%	7.40%
	RitualDance	0.15%	31.29%	8.24%	0.33%	28.56%	7.63%	0.10%	28.89%	7.10%
	Average	0.28%	31.33%	8.76%	0.31%	27.95%	6.96%	0.24%	28.88%	6.90%
720p	City	0.36%	32.36%	10.00%	0.45%	29.55%	7.30%	0.49%	30.10%	8.12%
	Crew	0.34%	31.52%	8.56%	-0.29	29.08	6.50	0.42%	27.17%	6.40%
	vidyo1	0.15%	31.44%	8.20%	0.33%	28.15%	6.84%	0.20%	26.35%	6.32%
	vidyo3	0.25%	28.52%	8.00%	0.17%	27.56%	6.34%	-0.04%	28.64%	6.86%
	Average	0.28%	30.96%	8.69%	0.17%	28.59%	6.75%	0.27%	28.07%	6.93%
240p	BasketballPass	0.24%	30.59%	8.65%	0.14%	27.84%	6.88%	0.56%	28.42%	6.03%
	BQSquare	0.12%	32.56%	10.51%	0.53%	30.33%	7.08%	0.04%	28.85%	6.82%
	BlowingBubbles	0.36%	29.75%	7.84%	0.24%	27.13%	6.30%	0.41%	27.14%	6.67%
	RaceHorses	0.28%	34.21%	9.58%	0.09%	30.52%	7.99%	0.23%	29.96%	7.88%
	Average	0.25%	31.78%	9.15%	0.25%	28.96%	7.06%	0.31%	28.59%	6.85%
Summary	Overall	0.23%	31.35%	8.82%	0.25%	28.46%	6.96%	0.27%	28.56%	6.97%

分别测试相同序列的结果。本文的方法平均BD-Rate损失为0.25%，而缩短的RDOQ时间为29.46%，总编码时间缩短了7.58%。编码时间的减少仅仅带来了0.25%的BD-Rate损失。这表明本文所提出的方法起到了优化作用。

此外，从每个视频分辨率中选择一个序列，部分序列在AI配置下的RD曲线如图11所示，在图11分别绘制了这些序列的RD曲线。这些曲线表明，本文所提出的方法的BD-Rate和HPM-4.0中原始RDOQ非常接近，并且在不同视频分辨率下的编码性能损失很小。

为了更好地验证本文方法的有效性，进行了与编码性能相关的消融实验，结果见表4。结果表明：对于OCD并行优化，其在AI、RA和LDB配置下的BD-Rate分别损失了0.18%、0.21%和0.21%；对于基于分治策略的LNPD并行优化，其在AI、RA和LDB配置下的BD-Rate分别损失了0.05%、0.04%和0.06%。可以看出，编码性能的损失主要集中在OCD并行优化阶段。主要是

因为在OCD阶段计算中，将扫描线分为两类，仅有一类保留了系数间依赖，而另一类为了并行舍去了依赖关系，这对码率性能造成影响。在基于分治策略的LNPD阶段，存在着很少的编码性能损失。

表4 本文方法BD-Rate的消融实验结果

方法	BD-Rate		
	AI	RA	LDB
OCD并行优化	0.18%	0.21%	0.21%
OCD并行优化+基于分治策略的LNPD	0.23%	0.25%	0.27%

表5给出了本文方法与Xu等^[14]、Lee等^[18]、Igarashi等^[25]和Zhao等^[26]的方法在编码性能和计算复杂度方面的比较结果。对于在参考软件HM平台进行测试的Xu等^[14]、Lee等^[18]和Igarashi等^[25]，直接使用了原始实验结果进行比较；对于在参考软件HPM平台进行测试的Zhao等^[26]，基于HPM平台对其实验结果进行复现。N/A表示上述比较方法未在该配置下进行实验，故没有实验

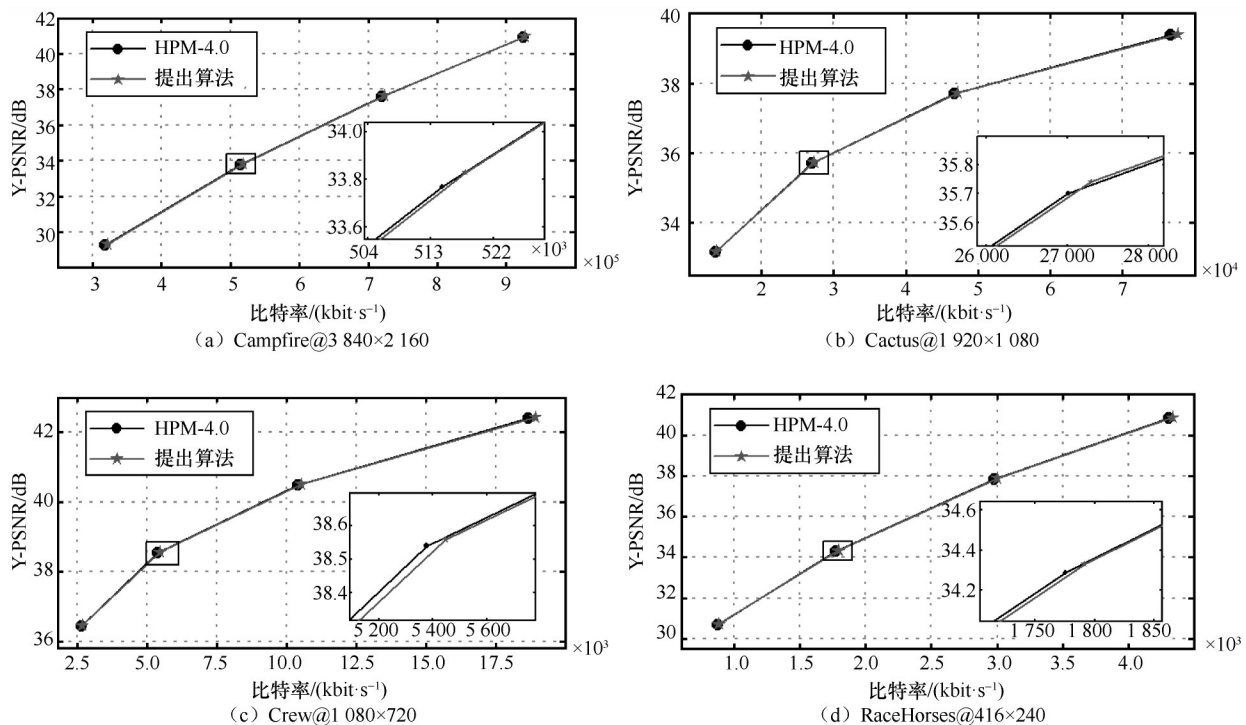


图11 部分序列在AI配置下的RD曲线



表5 本文方法和其他方法编码结果的比较结果

现有工作	参考软件	AI		RA		LDB	
		BD-Rate	TS _Q	BD-Rate	TS _Q	BD-Rate	TS _Q
文献[14]	HM 16.15	0	14.4%	0.04%	15.8%	0.12%	12.9%
文献[18]	HM 11.0	0.09%	14.4%	0.11%	15.8%	0.08%	14.9%
文献[25]	HM 16.0	—	—	2.51%	—	—	—
文献[26]	HPM 4.0	—	—	0.3%	27.6%	0.4%	30.6%
本文方法	HPM 4.0	0.23%	31.53%	0.25%	28.46%	0.27%	28.56%

结果用于复现或比较。可以看出本文方法与其他方法相比,可以在节约更多编码时间的同时,确保性能下降也很小。

Xu等^[14]和Lee等^[18]的方法都涉及对RDOQ过程的优化,在一定程度上降低了RDOQ的复杂度,且造成的编码性能损失较低。Xu等^[14]的方法虽然实现了很低的BD-Rate,但是仅仅降低了14.4%的编码时间;本文方法的BD-Rate虽然略大于Xu等^[14]的方法,但仍然属于可以忽略不计的范围,同时本文方法降低了29.46%的编码时间。本文方法与Lee等^[18]的方法相比,在AI配置下,编码时间的减少从14.4%提升到了31.53%,而BD-Rate的损失相差不大。Igarashi等^[25]的方法提出了一种基于GPU的并行RDOQ算法,但造成了2.51%的显著的BD-Rate损失,而本文方法在CPU测试平台上基于SIMD进行编码,仅造成了0.25%的BD-Rate损失。在Zhao等^[26]的方法中,码率计算时不考虑扫描线上的依赖关系以实现并行;在本文方法中则保留了右上扫描线上的依赖关系。因此Zhao等^[26]的方法虽然在LDB配置下比本文方法降低了更多的编码时间,但是本文方法对于编码性能的影响更小。以上数据和分析均说明本文方法可以在几乎不降低编码性能的情况下有效地减少编码时间。

5 结束语

本文介绍了一种基于SIMD的并行RDOQ算法,用于加速AVS3视频标准中的RDOQ过程。

本文通过构建局部依赖,打破相邻数据的依赖性,将Zig-Zag扫描线拆分成多个单独的扫描线,实现了RDOQ算法的高效并行化;同时采用SIMD对并行RDOQ算法进行加速优化。实验结果表明,与原始HPM-4.0相比,该并行化算法的BD-Rate平均损失为0.25%,计算时间平均减少29.46%。这些结果显示了该算法在视频编码效率和性能方面的显著改进。

参考文献:

- [1] 司武旭. 5G通信技术传输超高清视频的相关研究[J]. 现代工业经济和信息化, 2022, 12(12): 54-55.
- [2] SI W X. Research related to the transmission of ultra-high definition video by 5G communication technology[J]. Modern Industrial Economy and Informationization, 2022, 12(12): 54-55.
- [3] CHOI K, CHEN J L, RUSANOVSKYY D, et al. An overview of the MPEG-5 essential video coding standard in a nutshell[J]. IEEE Signal Processing Magazine, 2020, 37(3): 160-167.
- [4] ZHANG J, JIA C, LEI M, et al. Recent development of AVS video coding standard: AVS3[C]//Proceedings of 2019 picture coding symposium (PCS). Piscataway: IEEE Press, 2019: 1-5.
- [5] HUANG Y W, HSU C W, CHEN C Y, et al. A VVC proposal with quaternary tree plus binary-ternary tree coding block structure and advanced coding techniques[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(5): 1311-1325.
- [6] FU Y B, WANG S, ZHU C, et al. An attention based CNN with temporal hierarchical deployment for AVS3 inter In-loop filtering[C]//Proceedings of the 2022 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE Press, 2022: 501-505.
- [6] 姚英彪, 李晓娟. 基于图像空间相关性与纹理的HEVC块划

- 分快速算法[J]. 电信科学, 2015, 31(1): 38-46.
- 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.
- [7] 祁伟, 殷海兵, 王鸿奎, 等. 基于统计建模的VVC快速码率估计算法[J]. 电信科学, 2022, 38(12): 35-45.
- QI W, YIN H B, WANG H K, et al. Statistical modeling based fast rate estimation algorithm for VVC[J]. Telecommunications Science, 2022, 38(12): 35-45.
- [8] 孟翔, 殷海兵, 黄晓峰. 基于统计建模的HEVC快速率失真估计算法[J]. 电信科学, 2021, 37(1): 58-68.
- MENG X, YIN H B, HUANG X F. Statistical modeling based fast rate distortion estimation algorithm for HEVC[J]. Telecommunications Science, 2021, 37(1): 58-68.
- [9] ZHANG D D, CHEN Y W, IZQUIERDO E. Fast intra mode decision for HEVC based on texture characteristic from RMD and MPM[C]//Proceedings of the 2014 IEEE Visual Communications and Image Processing Conference. Piscataway: IEEE Press, 2014: 510-513.
- [10] WU Q B, XIONG J, LUO B, et al. A novel joint rate distortion optimization scheme for intra prediction coding in H.264/AVC[J]. IEICE Transactions on Information and Systems, 2014, 97(4): 989-992.
- [11] HE D K, WANG J. Rate distortion optimized quantization based on weighted mean squared error for lossy image coding [C]//Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP). Piscataway: IEEE Press, 2014: 5606-5610.
- [12] DENG Y, PENG Q, YANG T W. A rate-distortion optimized real time intra-update method for packet video transmission[J]. Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems, 2003(2): 915-918vol.2.
- [13] LEMMETTI A, KALLIO E, VIITANEN M, et al. Rate-distortion-complexity optimized coding scheme for kvazaar HEVC intra encoder[C]//Proceedings of the 2018 Data Compression Conference. Piscataway: IEEE Press, 2018: 419.
- [14] XU M T, NGUYEN CANH T, JEON B. Simplified level estimation for rate-distortion optimized quantization of HEVC[J]. IEEE Transactions on Broadcasting, 2020, 66(1): 88-99.
- [15] WANG M, XIE X D, FAN H F, et al. Fast rate distortion optimized quantization method for HEVC[C]//Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE Press, 2017: 1-4.
- [16] WANG H, YIN H, SHEN Y. A novel hard-decision quantization algorithm based on adaptive deadzone offset model[C]//Advances in Multimedia Information Processing-PCM 2016: 17th Pacific-Rim Conference on Multimedia, Xi'an, China, September 15-16, 2016, Proceedings, Part II. Springer International Publishing, 2016: 335-345.
- [17] WANG J, YIN H B, GAO Z Y, et al. Improved rate distortion optimized quantization for HEVC with adaptive thresholding[C]//Proceedings of the 2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB). Piscataway: IEEE Press, 2016: 1-4.
- [18] LEE H, YANG S, PARK Y, et al. Fast quantization method with simplified rate - distortion optimized quantization for an HEVC encoder[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2016, 26(1): 107-116.
- [19] YIN H B, CAI H, LU H J. A new all-zero block detection algorithm for high efficiency video coding[C]//Proceedings of the 2017 Data Compression Conference (DCC). Piscataway: IEEE Press, 2017: 470.
- [20] YANG H, YIN H, HUANG X. A fast zero-quantized percentage model for video coding with RDO quantization[C]//Advances in Multimedia Information Processing - PCM 2018: 19th Pacific-Rim Conference on Multimedia, Hefei, China, September 21-22, 2018, Proceedings, Part III 19. Springer International Publishing, 2018: 761-771.
- [21] YIN H B, YANG E H, YU X, et al. Fast soft decision quantization with adaptive preselection and dynamic trellis graph[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2015, 25(8): 1362-1375.
- [22] CUI J, WANG S S, WANG S Q, et al. Hybrid Laplace distribution-based low complexity rate-distortion optimized quantization[J]. IEEE Transactions on Image Processing: a Publication of the IEEE Signal Processing Society, 2017, 26(8): 3802-3816.
- [23] HE J, YANG F Z. High-speed implementation of rate-distortion optimized quantization for H.264/AVC[J]. Signal, Image and Video Processing, 2015, 9(3): 543-551.
- [24] CANH T N, XU M, JEON B. Rate-distortion optimized quantization: A deep learning approach[C]//Proceedings of IEEE High Performance Extreme Computing Conference. Piscataway: IEEE Press, 2018.
- [25] IGARASHI H, TAKANO F, TAKENAKA T, et al. Parallel rate distortion optimized quantization for 4K real-time GPU-based HEVC encoder[C]//Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP). Piscataway: IEEE Press, 2018: 1-4.
- [26] ZHAO J C, YANG F, HUANG X F, et al. Scanline-based fast algorithm and pipelined hardware design of rate-distortion opti-



mized quantization for AVS3[C]//Proceedings of the 2023 IEEE International Conference on Consumer Electronics (ICCE). Piscataway: IEEE Press, 2023: 1-6.

- [27] CHEN K J, DUAN Y Z, YAN L J, et al. Efficient SIMD optimization of HEVC encoder over X86 processors[C]//Proceedings of the Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference. Piscataway: IEEE Press, 2012: 1-4.

- [28] BOSSEN F. Common test conditions and software reference configurations[J]. JCTVC-L1100, 2013, 12(7): 1.

- [29] BJONTEGAARD G. Calculation of average PSNR differences between RD-curves[J]. ITU SG16 Doc. VCEG-M33, 2001.

[作者简介]



唐毅欣 (2003-), 女, 杭州电子科技大学通信工程学院在读, 主要研究方向为视频编解码。



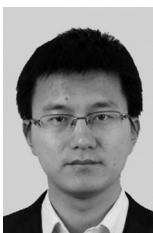
唐然 (1996-), 男, 杭州电子科技大学通信工程学院硕士生, 主要研究方向为视频编解码。



周洋 (1979-), 男, 杭州电子科技大学通信工程学院副教授, 主要研究方向为基于机器学习的视频图像处理、3D视频编解码。



崔燕 (1988-), 女, 浙江省经济信息中心高级工程师, 主要研究方向为视频编解码。



黄晓峰 (1988-), 男, 博士, 杭州电子科技大学通信工程学院副教授, 主要研究方向为数字视频编解码与芯片架构设计。



殷海兵 (1974-), 男, 博士, 杭州电子科技大学通信工程学院教授、在站博士后, 主要研究方向为数字视频编解码、多媒体信号处理、芯片结构设计验证。