



研究与开发

基于统计建模的 HEVC 快速率失真估计算法

孟翔, 殷海兵, 黄晓峰

(杭州电子科技大学通信工程学院, 浙江 杭州 310018)

摘要: 针对高效视频编码 (HEVC) 率失真优化过程中复杂的码率与失真计算, 提出一种高准确性的快速率失真估计模型。该算法评估熵编码中不同语法元素的贡献百分比, 并逐个开发语法元素的码率模型以构建整个 TU 级模型。首先, 对系数进行自适应加权, 以区分不同系数对码率的不均匀贡献。然后, 定义位置参数以描述一个块内非零系数可能的分布模式。通过统计方式对大量样本的模型参数进行微调, 得出最终线性码率模型。最后, 建立了变换域失真模型来省略不必要的重构过程。实验结果表明, 所提算法在 RA 配置下, 可以实现 49.76% 的复杂度降低, 而 BD-BR 仅增加 1.74%。

关键词: 高效视频编码; 率失真优化; 码率预估; 失真预估; 视频编码

中图分类号: TN919.81

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2021021

Statistical modeling based fast rate distortion estimation algorithm for HEVC

MENG Xiang, YIN Haibing, HUANG Xiaofeng

College of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China

Abstract: To reduce the coding computational complexity of the rate and distortion calculations in HEVC, a fast TU level rate model with higher accuracy was proposed. The contribution percentages of different syntax elements in entropy coding were evaluated, and syntax element-wise accurate rate models were individually developed to construct the whole TU level model. In addition, coefficient levels were weighted adaptively to distinguish the nonuniform contributions of different coefficients in terms of rate profiling. Moreover, position-wise parameter was defined to depict the distribution patterns for possible non-zero coefficients within one block. The final linear rate model was developed by fine-tuning the model parameters from great amounts of samples in a statistical way. Finally, the transform domain distortion model was also established to bypass unnecessary reconstruction process. Experimental results show that the proposed algorithm can achieve 49.76% complexity reduction with 1.74% BD-BR increase for RA configuration.

Key words: HEVC, RDO, rate estimation, distortion estimation, video coding

收稿日期: 2020-02-24; 修回日期: 2020-07-02

基金项目: 国家自然科学基金资助项目 (No.61972123, No.61901150); 浙江省自然科学基金资助项目 (No.LY19F020043, No.Q19F010030)

Foundation Items: The National Natural Science Foundation of China (No.61972123, No.61901150), Zhejiang Provincial Natural Science Foundation of China (No.LY19F020043, No.Q19F010030)

1 引言

随着各种对高清视频的需求, JCT-VC 提出了新一代高效视频编码 (HEVC) 标准^[1], 其采用高级编码工具, 包括基于四叉树的编码单元 (coding unit, CU)、变换单元 (transform unit, TU) 以及预测单元 (prediction unit, PU)。与 H.264^[2]相比, 在相同图像质量情况下, HEVC 可以节省高达 50% 的比特率。

CU/PU/TU 的组合方案可以大大提高编码效率, 但是基于率失真优化 (rate distortion optimization, RDO)^[3]的模式决策都需要为大量候选模式进行 RDO 成本计算, 其中涉及前向变换、量化、逆量化、逆变换和熵编码, 这会造成极大的计算复杂度, 阻碍了有效并行技术, 导致硬件实现效率降低。

一些学者针对如何减少码率和失真的计算复杂度问题展开研究, Zhao 等^[4]提出广义高斯分布模型来估计码率, 使用在量化中丢掉的比特来估计失真; Tu 等^[5]和 Wang 等^[6]采用量化系数之和与其对应系数坐标来估计码率; Liu 等^[7]提出一种基于二元分类的线性模型来估计码率。Sun 等^[8]尝试简化 RDO 过程, 使用变换系数来估计码率和失真, Chen 等^[9]和 Sharabayko 等^[10]采用信息熵来估计码率。考虑到硬件资源, 一些学者提出了有利于硬件实现的快速 RDO 算法^[11-14], 虽然节省了大量的硬件资源, 但是算法过于简单造成了严重的性能损失。现有工作并未充分探索熵编码特性, 这会导致压缩质量显著下降^[15]。

本文的主要贡献如下: 对每个语法元素进行码率分析, 并根据结果建立码率模型; 提出一种加权量化系数, 能够较好地反映码率信息; 采用建模的方式估计头信息码率; 从变换域建模估计失真, 避免多余的重构过程; 避免了上下文概率状态的实时更新, 有利于硬件实现。

2 码率与失真估计算法

HEVC 原始的码率与失真计算流程如图 1 所示, 其中熵编码技术会涉及上下文状态的实时更新, 系数之间产生较强的依赖性, 这不利于硬件并发的实现并会产生很大的编码复杂度, 失真计算会经过冗长的重构过程, 进一步加剧了编码的复杂性。

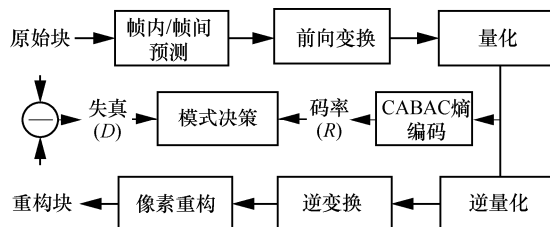


图 1 HEVC 原始码率失真计算流程

为了简化编码的复杂度, 采用码率与失真预估的方式来替代原有的计算过程。引入码率失真算法后的计算过程如图 2 所示, 可以看出码率失真预估算法大大简化了原始 HEVC 算法的流程, 同时避免了上下文概率状态的实时更新和冗长的重构过程, 更有利于硬件并发与流水线技术的实施。

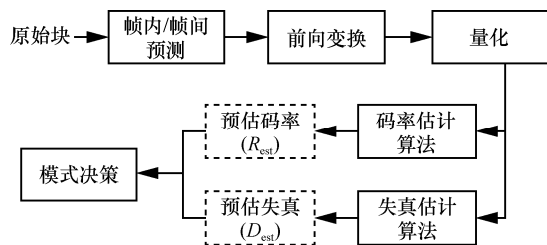


图 2 引入码率失真算法后的计算过程

在 RDO 过程中, 码率一般由两部分组成:

$$R = R_{\text{head}} + R_{\text{res}} \quad (1)$$

其中, R_{head} 表示编码头信息所需码率, R_{res} 代表编码量化系数所需码率。在编码量化系数时, 需要编码多个语法元素。

首先, 变换单元会被分为若干个 4×4 大小的系数块组 (coefficient group, CG), 按反向扫描顺



序首先扫描到的非零系数为最后一个非零系数位置 (last coefficient position, LCP), 编码器需要对 LCP 信息进行编码, 然后对每个 CG 进行判断, 代码子块标志 (code sub block flag, CSBF) 表示该 CG 内是否存在非零系数, 有效系数标志 (significant coefficient flag, SCF) 表示当前系数是否非零, 系数符号标志 (coefficient sign flag, CSF) 表示当前系数是否为正, 前 8 个非零系数中, 系数大于 1 (greater than 1, G1) 表示当前系数绝对值是否大于 1, 对于第一个大于 1 的系数, 系数大于 2 (greater than 2, G2) 表示该系数绝对值是否大于 2, 最后系数剩余部分 (remaining, RM) 会被编码。

为了更加直观地分析码率, 本文统计了在不

同 TU 大小和不同量化参数 (quantization parameter, QP) 下, 各个语法元素所占的码率比重, 如图 3 所示, 编码模式为随机访问 (random access, RA) 模式, 可以看出, 在 TU 4×4 和 TU 8×8 大小下, LCP 语法元素占有较大比重, 分别为 32.2% 与 25.8%, G1 与 CSF 语法元素码率占比较为均匀, SCF 码率综合占比最大, 因此如何着重分析码率占比较大的语法元素非常关键。

3 头信息码率预估

3.1 帧间预测跳过标志的码率估计

跳过模式是合并模式的一种特殊情况, 其 CU 内不包含任何残差信息。对于每个单元, CU 跳过标志被用来表示该 CU 是否为跳过模式。其

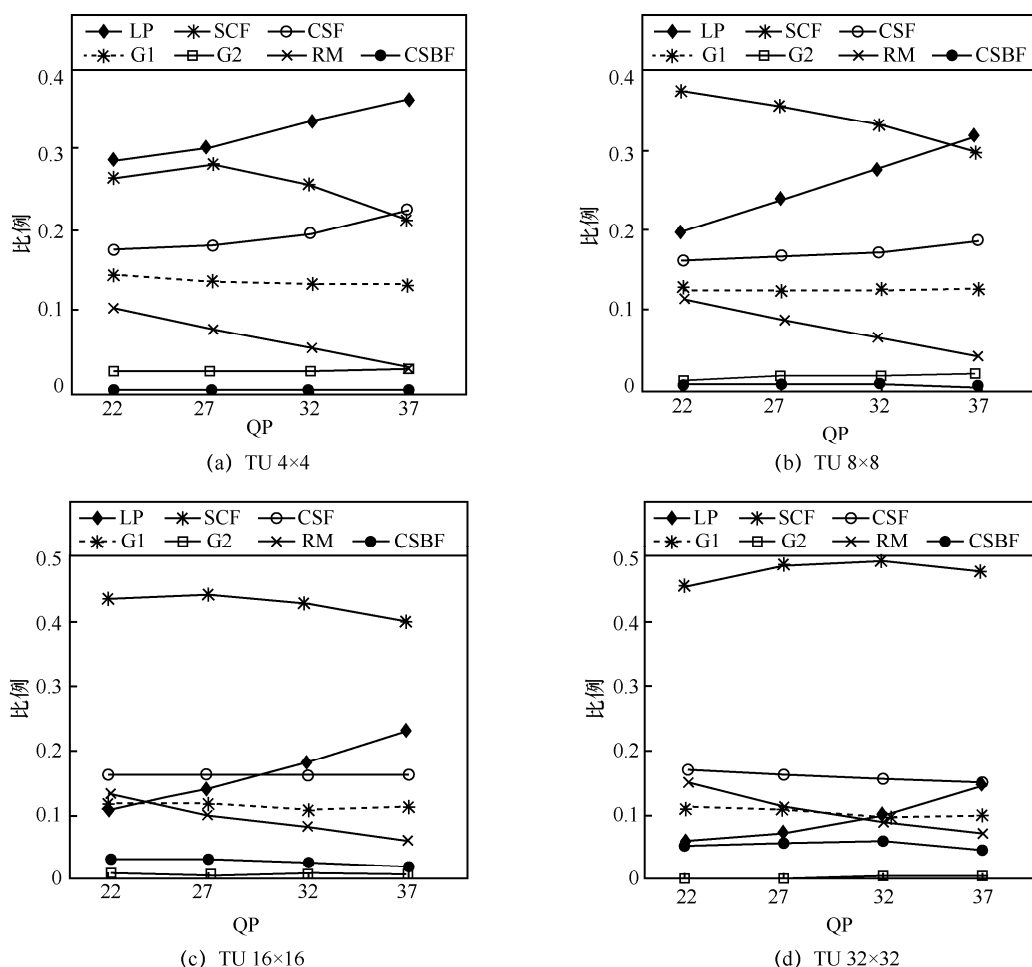


图 3 TU 级别下各语法元素码率比重

根据相邻 CU 跳过标志的不同会使用 3 种上下文模型。本文实验发现跳过标志所消耗的码率均值在不同 QP 下有着较强的线性关系。当相邻单元均为非跳过模式时, 码率与 QP 之间的关系如图 4 所示。

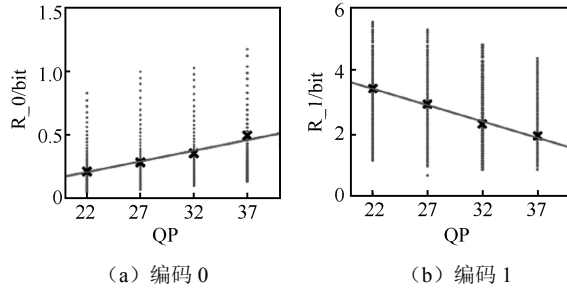


图 4 跳过标志码率与 QP 的关系

其中, R_0 和 R_1 分别表示编码 0 (非跳过模式) 和 1 (跳过模式) 所消耗的码率。实验测试环境为 RA 编码模式, 测试序列为 BasketballPass, 测试帧数为 30。图 4 中的×表示各 QP 下码率消耗的平均值。事实上, 随着 QP 的增加, CU 更有可能被编码为跳过模式。因此, 编码 1 消耗的比特逐渐减少, 编码 0 消耗的比特逐渐增加。

因此可以根据 QP 进行建模, 式 (2) 中定义了跳过标志的码率估计模型。

$$R_{\text{skip}} = \begin{cases} \alpha_0 \text{QP} + \beta_0 \\ \alpha_1 \text{QP} + \beta_1 \end{cases} \quad (2)$$

其中, α_0 与 β_0 表示编码 0 时的模型系数, α_1 与 β_1 表示编码 1 时的模型系数, 在相邻单元都为非跳过模式时, $\alpha_0=0.016\ 89$, $\beta_0=0.183\ 3$, $\alpha_1=-0.107\ 3$, $\beta_1=5.814$ 。当相邻单元只有一个为跳过模式或都为跳过模式时, 其码率模型可以采用相同的方法分析。

3.2 预测模式的码率估计

在头文件码率中, 编码预测模式所消耗码率占有较大比例, 对亮度块来说, 最优预测模式从两类中选出, 一类是来自相邻单元的 3 种高概率模式 (most probability mode, MPM), 另一类为低概率模式 (low probability mode, LPM)。prev_intra_luma 会被用来描述该模式是否为 MPM

模式, 与跳过标志建模类似, QP 与码率关系如图 5 所示。

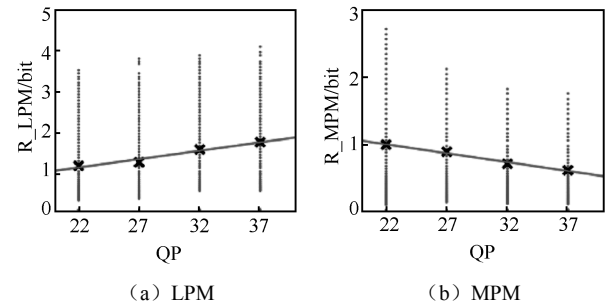


图 5 预测模式 QP 与码率的关系

最后该部分模型如式 (3) 所示:

$$R_{\text{luma}} = \begin{cases} \gamma_0 \text{QP} + \delta_0, \text{LPM} \\ \gamma_1 \text{QP} + \delta_1, \text{MPM} \end{cases} \quad (3)$$

其中, $\gamma_0=0.041\ 73$, $\delta_0=0.138\ 4$, $\gamma_1=-0.026\ 8$, $\delta_1=1.578$ 。

对于色度块, 从 5 个候选模式中选择最佳模式。intra_chroma_pred_mode 用于描述最佳模式是否与最佳亮度模式相同, 该部分建模类似于 prev_intra_luma。

4 残差系数码率估计

4.1 基于加权量化系数的量化系数和

在现有工作中, Sheng^[16]采用量化系数总和 (sum quantized coefficient, SQC) 作为系数部分码率估计的特征。但是, 当 TU 的 SQC 相同时, 其码率消耗会有较大的差异。造成这一误差的原因之一是: 在编码过程中, 上下文概率模型转换会导致严重的系数间串行依赖。为解决此问题, 本文提出加权量化系数总和 (sum weighted quantized coefficient, SWQC)。

在图 3 中, SCF 语法元素总体比其他语法元素占据更大的比例。因此, 本文重点分析因编码 SCF 语法元素而产生的系数依赖性。实际上, SCF 的上下文模型选择受许多因素影响, 包括当前块是否为亮度块、下方和右侧的 CSBF 值等。因此, 本文将 CG 分为 7 类, 如图 6 所示。

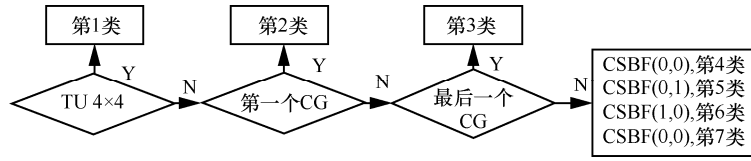


图6 加权量化系数的分类方法

在图6中, CSBF(0,0)代表下方和右方CG的CSBF值均为0。TU 4×4为第1类。对于第一个CG, 存在DC系数并且为低频区域, 非零系数多于其他CG, 因此将第一个CG分为第2类。最后一个非零CG具有LP语法元素, 分为第3类。根据下方和右方CSBF取值分为第4、5、6、7类。

本文使用线性回归方法确定权重, 使用CG级别码率进行训练。不同组将获得不同的权重, 加权量化系数总和累积作为TU级码率估计的特征。第3类中有一个特殊情况, 最后一个非零CG具有LP语法元素, 其码率消耗主要来自LP信息, 为了简化算法, 将第3类的权重设置为1。该部分码率模型如式(4)所示:

$$R_{wqc} = \sum_{i=1}^{N^2/16} \begin{cases} \sum_{j=1}^{16} |L_{ij}|, \text{第3类} \\ \sum_{j=1}^{16} |W_{kj} L_{ij}|, \text{其他} \end{cases} \quad (4)$$

其中, W_{kj} 代表权重系数, k 表示由图6确定的CG类别, L_{ij} 代表量化系数, N 为TU块大小。在QP为32时, 各TU大小下 R_{wqc} 与码率之间关系如图7所示。

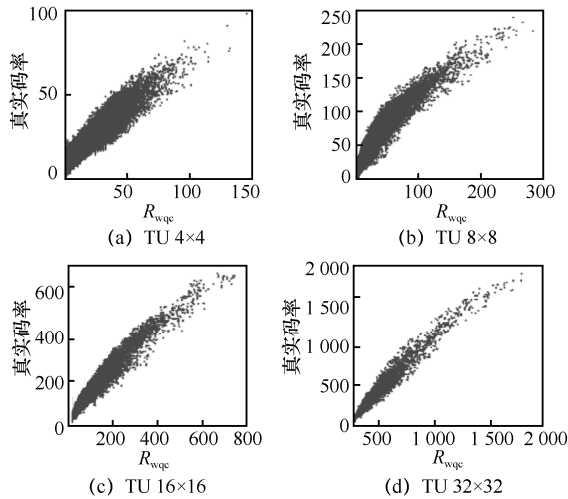


图7 TU级别加权量化系数总和与码率的关系

权重系数从序列BasketballPass获得。将其应用于序列BQsquare, 其与TU级别码率的相关系数(R^2)与均方差(MSE)见表1, 其中拟合方式均为线性拟合, 拟合公式为 $y = ax + b$, 其中 a 、 b 为模型参数。可以看到SWQC比SQC具有更小的均方差和更高的相关系数。这证明了使用SWQC在码率估计方面将获得更高的精准度。

表1 均方差与相关系数

TU尺寸	R^2 (SQC)	R^2 (SWQC)	MSE(SQC)	MSE(SWQC)
4×4	0.837 9	0.886 3	3.942 1	3.300 6
8×8	0.859 9	0.867 4	12.179 3	11.114 1
16×16	0.916 9	0.946 7	30.019 8	23.192 8
32×32	0.938 6	0.958 5	80.730 2	63.990 1

4.2 位置参数以及其他特征参数

在熵编码中, 高概率事件将消耗较少的比特, 而低概率事件将消耗较多的比特。经过变换、量化等步骤后, 非零系数出现在低频区域的概率要大于高频区域。因此, 码率消耗将随着系数分布的不同而有着较大的变化。本文定义 η_c 为CG级别最后一个非零系数的正向扫描位置。当 η_c 较小时, 表示非零系数分布在低频区域。 η_z 表示CG内第一个与最后一个非零系数之间的零系数个数。当 η_z 较小时, 表示非零系数分布更加集中。这两个特征可以很好地描述系数的分布, 该部分码率定义如下:

$$R_{zpos} = \sum_{m=1}^{N^2/16} \eta_{c,m} + \eta_{z,m} \quad (5)$$

QP为32时, 其与码率的关系如图8所示。

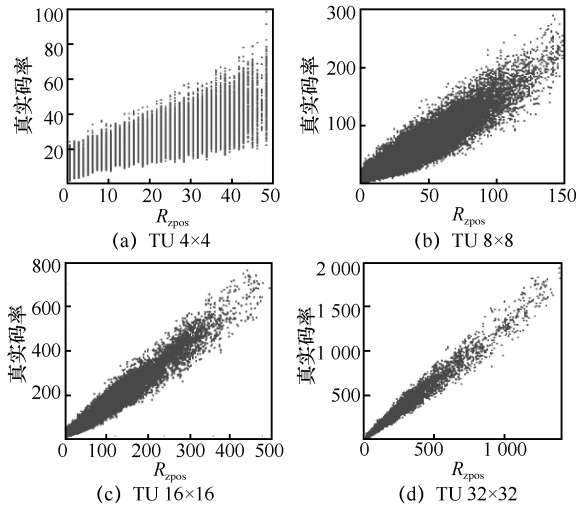


图8 TU级别位置信息与码率的关系

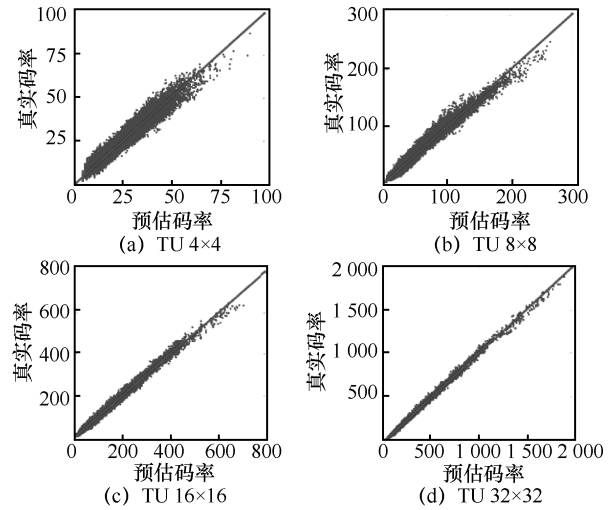


图9 预估码率与真实码率的关系

在 TU 4×4 与 TU 8×8 两种情况下, 编码 LCP 位置信息时, 其码率占比较大, 由图 3 可知其占比分别达到 32.2% 与 25.8%, 但是 TU 16×16 与 TU 32×32 情况下, 其码率占比只有 16.7% 与 9.1%。在考虑到模型复杂度与硬件资源的情况下, 本文只针对 TU 4×4 与 TU 8×8 情况考虑 LCP 位置信息, LCP 语法元素码率组成如下:

$$R_{\text{LCP}} = R_{\text{pre}} + R_{\text{suf}} \quad (6)$$

其中, R_{pre} 表示前缀码, 采用常规编码模式; R_{suf} 为后缀码, 采用等概率编码模式; 本文定义特征 η_b 为前缀码和后缀码的总码元个数, 并使用它来估计编码 LCP 信息所消耗的码率。

此外, 在不同 TU 大小下, 各个语法元素码率占比波动较大, 但 CSF 语法元素却很平稳, 占比平均为 17.1%, 因为该语法元素采用等概率编码方式, 因此本文定义特征 η_n 为非零系数个数并纳入考虑范围。

4.3 最终码率模型以及码率估计算法实现过程

最终模型如下:

$$R_{\text{Est}} = \theta_0 + \theta_1 R_{\text{wqc}} + \theta_2 R_{\text{zpos}} + \theta_3 \eta_b + \theta_4 \eta_n \quad (7)$$

其中, η_b 只有在 TU 大小为 4×4 与 8×8 的情况下才会使用; $\theta_0 \sim \theta_4$ 为模型参数, 由线性回归训练得到。图 9 为 QP 等于 32 时, R_{Est} 与真实码率之间的关系, 测试序列为 BasketballPass。可以看出, 两者之间有着很强的线性关系。

码率估计算法流程如下。

输入 量化系数矩阵 L , 4×4 权重矩阵 W_k 。

输出 预估码率 R_{Est} 。

说明 量化系数矩阵 L 尺寸取决于当前 TU 块尺寸, 取值范围为 4×4、8×8、16×16、32×32。 K 取值范围为 1~7, 由图 6 所示方法决定。步骤 3 中 C 为系数组矩阵, 大小为 4×4。

步骤 1 根据式 (2)、式 (3) 计算头信息码率。

步骤 2 遍历系数矩阵 L , 确定 LCP 位置并计算 η_b 参数。

步骤 3 当前 TU 大小为 4×4, 计算 $R_{\text{wqc}} = \sum \sum L \circ W_1$; 否则计算 $R_{\text{wqc}} = \sum_i \sum \sum C_i \circ W_k$ 。

步骤 4 统计 η_c 参数, 同时记录零系数的个数 η_z , 根据式 (5) 计算出 R_{zpos} 。

步骤 5 统计非零系数个数 η_n , 遍历 L 结束。

步骤 6 如果 TU 尺寸为 4×4、8×8, 根据式 (7) 得出 R_{Est} ; 否则令 $\eta_b = 0$, 代入式 (7) 得出 R_{Est} 。码率估计算法流程如图 10 所示。

5 失真预估

在 HEVC 中, 计算量化过程如下:

$$Q_{ij} = (y_{ij} \cdot Sc + f) \gg \text{iQBits} \quad (8)$$

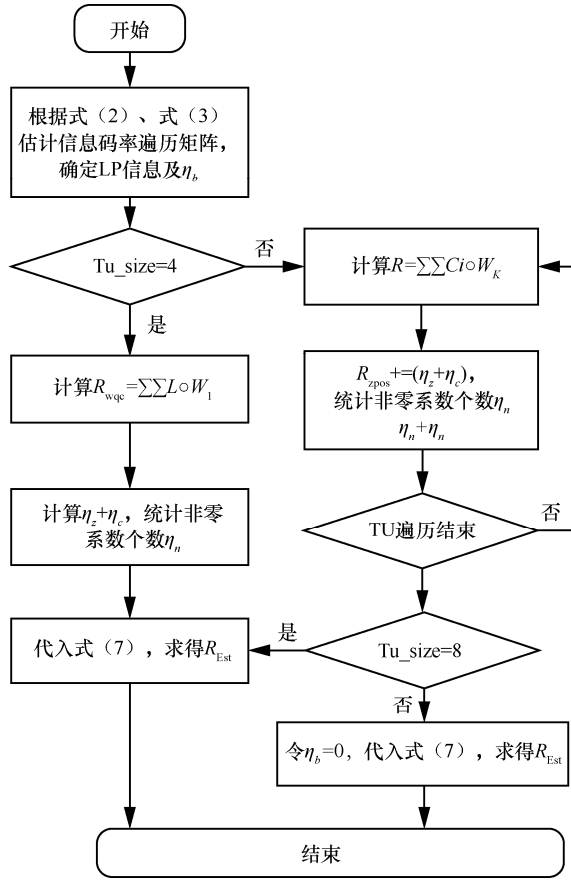


图10 码率估计算法流程

其中, Q_{ij} 为量化值, y_{ij} 为变换系数, $Sc = 2^{qbits} / Q_{step}$ 是和 QP 有关的缩放系数。在变换域中, 失真可以由式 (9) 统计:

$$D = \left[\left(\frac{(y - y_i)}{2^{iQbits}} \right) \cdot Q_{step} \right]^2 \quad (9)$$

其中, y 为变换系数, y_i 为重构后的变换系数。为了简化重构过程, 本文分别定义了原始缩放后系数 $P_{ij} = y_{ij} \times Sc$ 和近似重构后的变换系数 $T_{ij} = Q_{ij} \ll iQbits$, 并统计了两者间的差异, 定义如下:

$$D_{dif} = |P_{ij} - Q_{ij} \ll iQbits| \quad (10)$$

因此被缩放后失真近似为:

$$D_e \approx \sum_{i,j} D_{dif}^2 = \sum_{i,j} (P_{ij} - Q_{ij} \ll iQbits)^2 \quad (11)$$

将式 (11) 代入式 (9):

$$D = \left[\left(\frac{D_e}{2^{2iQbits}} \right) \cdot Q_{step}^2 \right] \quad (12)$$

其中, $iQbits = qbits + shift$, 化简后有:

$$D_{Est} = \alpha(QP) \left(\frac{D_e}{(Sc)^2} \gg 2 \cdot shift \right) \quad (13)$$

其中, $shift$ 是与 TU 大小有关的偏移量。为了更精准地估计码率, 本文引入与 QP 相关的模型参数 $\alpha(QP)$ 其在 QP 取值为 22、27、32、37 时分别等于 0.86、0.89、0.92、0.95。预估失真与真实失真关系如图 11 所示。

失真估计算法具体流程如下。

输入 量化系数矩阵 Q , 变换系数矩阵 Y 。

输出 预估失真 D_{Est} 。

步骤 1 遍历系数矩阵 Q , 计算 $T_{ij} = Q_{ij} \ll iQbits$, 得到近似重构后的变换系数。

步骤 2 计算 $d_{ij} = (P_{ij} - Q_{ij} \ll iQbits)^2$, 为该系数缩放后失真。

步骤 3 去除缩放比例, 由式 (13) 得到

$$D_{ij} = \frac{d_{ij}}{(Sc)^2} \gg 2shift。$$

步骤 4 矩阵 Q 遍历结束, 引入 $\alpha(QP)$ 模型参数, 累加得 $D_{Est} = \alpha(QP) \sum \sum D_{ij}$ 。

由以上可以看出, 失真的计算在进行前向变换量化的时候就已经计算完毕, 省去了逆变换、逆量化、重构等过程。这种做法虽然忽视了逆量化、逆变换因取整和位移等操作造成的误差, 但是这些误差的数值很小, 在可接受范围内, 同时可以节省大量的时间。从图 11 可以看出, 预估失真与真实失真有着很强的线性关系, 这也证明了本文失真估计算法的优越性。

6 实验结果分析

为了评估文中所提出的快速码率失真估计算法, 采用 HEVC 参考代码 HM16.0 进行实验, 实

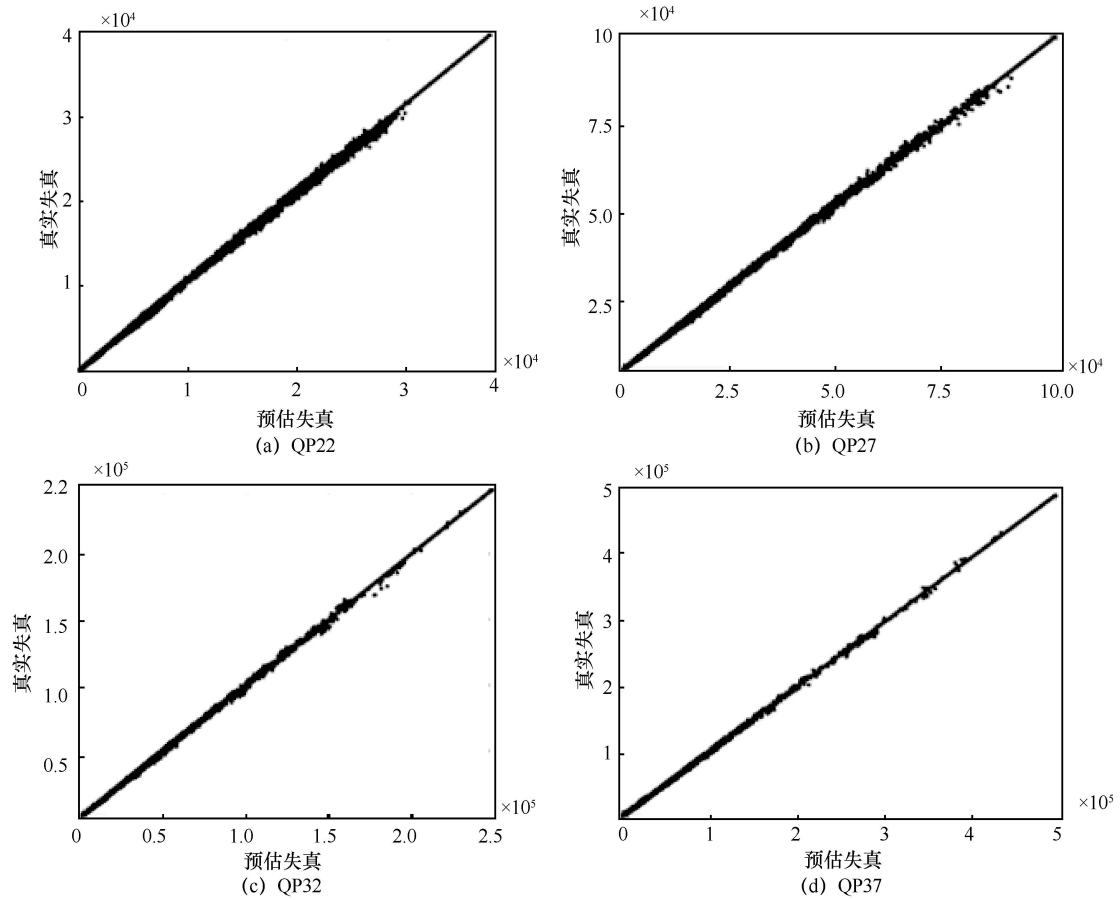


图 11 预估失真与真实失真的关系

验配置为：RA 模式，关闭 RDOQ 功能，关闭变换跳过功能，其余均为默认配置。实验中 QP 取值为：22、27、32、37。实验采用 H.265 标准测试序列^[17]进行测试，码率变化率（BD-BR）^[18]当作评价指标。

为了衡量计算复杂度的变化，本文定义了时间变化率 ΔT 定义如下：

$$\Delta T = \frac{T_{\text{HMRDO}} - T_{\text{ProRDO}}}{T_{\text{HMRDO}}} \times 100\% \quad (14)$$

其中， T_{ProRDO} 为本算法进行 RDO 过程所消耗的时间， T_{HMRDO} 为原始 HEVC 算法进行 RDO 过程所消耗的时间， ΔT 为正表示编码时间的减少，与 HM 算法相比，本文快速算法的率失真性能和整体表现见表 2。其中，*表示仅使用码率预估的性能损失，**表示码率与失真预估的性能损失。表 2

中还列出了近年来典型的码率失真估计算法，并做了比较。

与原始 HM 算法相比，只使用码率估计算法，参考文献[9]的算法 BD-BR 总体上升 1.32%，本文算法 BD-BR 总体上升 0.96%，性能提升 0.36%，对于有较多细节画面的视频序列，例如 PartyScene、BQsquare、BasketballPass 和 SlideEditing，参考文献[9]的算法 BD-BR 分别上升 1.92%、1.68%、1.31%、1.81%，性能损失较为严重，本文算法在这些序列下 BD-BR 分别上升 0.75%、0.43%、0.37%、1.24%，皆优于参考文献[9]。对于平稳的序列，例如 FourPeople、Johnny 和 KristenAndSara，参考文献[9]算法 BD-BR 分别上升 1.17%、1.35%、1.57%。同样地，本文算法 BD-BR 分别上升 1.09%、1.16%、1.12%，因此无论是在



表2 在 Random Access 下的模型性能

类别	视频序列	参考文献[9]算法	参考文献[8]算法		本文算法		
		BD-BR*	BD-BR**	ΔT^{**}	BD-BR*	BD-BR**	ΔT^{**}
A	Traffic	1.34%	1.99%	38.59%	1.06%	2.36%	40.83%
	PeopleOnStreet	1.45%	2.58%	52.28%	1.28%	2.11%	50.31%
	Nebuta	0.77%	0.98%	42.24%	0.42%	0.96%	59.44%
	SteamLocomotive	0.65%	1.47%	39.62%	0.54%	1.35%	54.81%
B	BQTerrace	1.01%	1.80%	49.56%	0.86%	1.85%	51.73%
	Kimono	1.31%	1.58%	43.22%	1.30%	1.63%	44.09%
	ParkScene	1.22%	2.21%	43.79%	1.27%	2.65%	42.02%
	Cactus	1.42%	2.71%	46.20%	1.22%	2.41%	47.83%
C	BasketBallDrive	1.19%	1.79%	38.87%	1.22%	1.85%	51.07%
	BasketballDrill	1.46%	3.27%	57.45%	1.01%	1.87%	52.50%
	PartyScence	1.92%	2.59%	58.83%	0.75%	1.42%	57.21%
	RaceHorsesC	1.27%	2.72%	60.78%	0.73%	1.66%	50.52%
D	BQMall	1.15%	3.10%	51.14%	1.01%	1.72%	47.96%
	BasketballPass	1.31%	3.14%	45.79%	0.37%	1.25%	51.52%
	RaceHorses	1.23%	3.02%	59.47%	0.82%	1.31%	49.91%
	BQsquare	1.68%	2.72%	60.12%	0.43%	1.12%	42.54%
E	BlowingBubbles	1.44%	3.45%	59.55%	0.77%	1.47%	45.29%
	FourPeople	1.17%	2.82%	33.02%	1.09%	2.07%	44.81%
	Johnny	1.35%	2.71%	32.30%	1.16%	1.96%	58.32%
	KristenAndSara	1.57%	3.05%	31.23%	1.12%	2.01%	54.92%
F	ChinaSpeed	1.19%	2.47%	42.31%	1.21%	1.65%	42.84%
	SlideEditing	1.81%	2.60%	66.32%	1.24%	1.71%	51.45%
	BasketballDrillText	1.02%	2.06%	45.21%	0.79%	1.31%	55.85%
	Slideshow	1.79%	3.12%	39.59%	1.31%	1.95%	46.64%
	平均值	1.32%	2.49%	47.40%	0.96%	1.74%	49.76%

具有较多细节还是较为平稳的视频序列中，本文算法皆优于参考文献[9]提出的算法。

在同时使用码率估计与失真估计算法时，相较于原始 HM 算法，参考文献[8]的算法 BD-BR 总体上升 2.49%，同时 RDO 时间节省 47.40%。对于高分辨率的视频序列，如 Traffic、Nebuta、Kimono，其 BD-BR 分别上升 1.99%、0.98%、1.58%，性能损失较低；对于低分辨率的序列，即 D 类视频序列，其 BD-BR 上升分别为 3.14%、3.02%、2.72%、3.45%，性能损失较大，这说明参

考文献[8]的算法在高分辨率的序列上有着较好的性能。在时间节省方面，参考文献[8]的算法在序列 SlideEditing 和 RaceHorses 可以达到 66.32%和 60.78%，但是在 Johnny 和 KristenAndSara 序列下，其时间节省仅为 32.30%与 31.23%，最大值与最小值之间相差 35%，这说明参考文献[8]的算法时间节省并不均匀，其算法不具有较强的普适性。本文在同时使用码率与失真估计算法时，BD-BR 总体上升了 1.74%，RDO 时间节省 49.76%，相较于参考文献[8]的算法，性能有 0.75%的性能提升，

同时多获得了 2.36% 的时间节省。对于高分辨率的视频序列,如 Traffic、Nebuta、Kimono,其 BD-BR 分别上升 2.36%、0.96%、1.63%;在低分辨率视频序列(D类)下,本文算法 BD-BR 仅上升 1.25%、1.31%、1.12%、1.47%,这证明了本文算法有着更好的率失真性能。在序列 Nebuta 中,本文可以达到 59.44% 的 RDO 时间节省,为测试序列中的最大值;在序列 Traffic 中可以达到 40.82% 的 RDO 时间节省,为最小值,其相差大约 19%,与参考文献[8]算法的 35% 相比,可以证明本文算法节省时间更加均匀,有着更好的普适性。

在 A 类高分辨率视频序列中,由于编码每帧图像所需码率较多,会进行较多的 RDO 模式决策过程,这意味着使用码率失真估计算法会产生较大误差,本文算法在 A 类视频序列性能平均损失仅为 1.69%。在分辨率较低的 D 类序列中,在 RA 模式下,编码每帧所需的码率较少,但这也意味着更多的系数被量化为 0,导致失真计算会有较大的误差,本文算法在 D 类序列上平均损失仅为 1.28%,表 2 数据表明在各个分辨率下,都有着优于参考文献[8]算法和参考文献[9]算法的性能表现。图 12~图 15 为在各个分辨率下的视频序列 R-D 性能曲线,对比了原始的 HM16.0 算法和本文的码率失真估计算法,可以看出本文的算法性能在各个分辨率下都极其接近原始 HM 算法的性能,这也证明了本文算法的优越性。

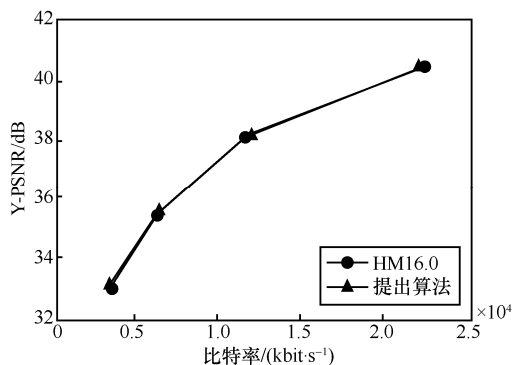


图 12 序列 ParkScene (1 920 dpi×1 080 dpi) 的 R-D 曲线

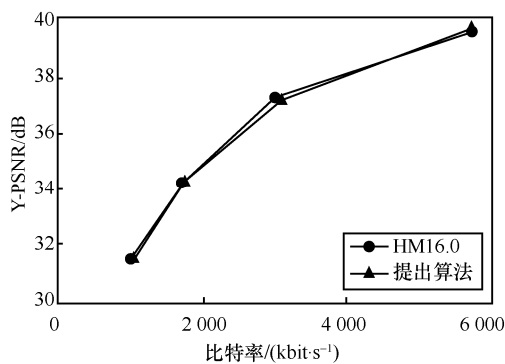


图 13 序列 BQMall (832 dpi×480 dpi) 的 R-D 曲线

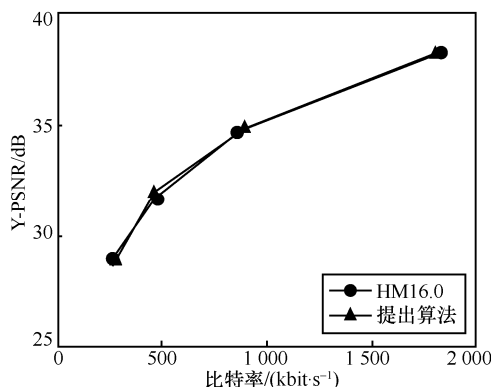


图 14 序列 BQsquare (416 dpi×240 dpi) 的 R-D 曲线

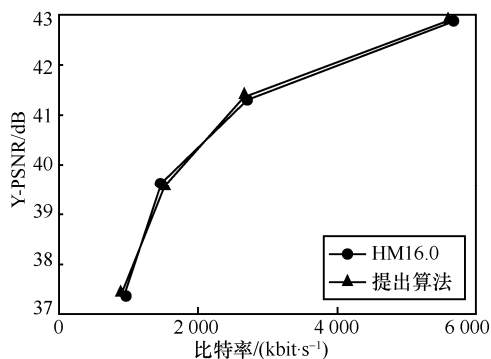


图 15 序列 Johnny (1 280 dpi×720 dpi) 的 R-D 曲线

7 结束语

本文针对 HEVC 率失真优化过程中复杂的码率与失真计算问题,提出一种快速码率失真估计算法。通过实验证明,本文算法与目前典型算法而言,在性能与时间节省方面都有着较大的提升,后续工作主要有两部分:一是对权重系数进行进一步优化,探索其与位置参数的动态关系,使得码率模型能够更精准地估计码率;二是探索码率



与失真模型在全帧内编码中的应用,找出其与帧间编码参数之间的差异,使得模型能够适应更复杂的编码场景。

参考文献:

- [1] 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.
- [2] WIEGAND T, SULLIVAN G J, BJONTEGAARD G, et al. Overview of the H.264/AVC video coding standard[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 560-576.
- [3] SULLIVAN G J, WIEGAND T. Rate-distortion optimization for video compression[J]. IEEE Signal Processing Magazine, 1998, 15(6): 74-90.
- [4] ZHAO X, SUN J, MA S, et al. Novel statistical modeling, analysis and implementation of rate-distortion estimation for H.264/AVC coders[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2010, 20(5): 647-660.
- [5] TU Y K, YANG J F, SUN M T. Efficient rate-distortion estimation for H.264/AVC coders[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2006, 16(5): 600-611.
- [6] WANG Q, ZHAO D, GAO W, et al. Low complexity RDO mode decision based on a fast coding-bits estimation model for H.264/AVC[C]//Proceedings of International Symposium on Circuits and Systems (ISCAS 2005). Piscataway: IEEE Press, 2005.
- [7] LIU Z, GUO S, WANG D. Binary classification based linear rate estimation model for HEVC RDO[C]//Proceedings of IEEE International Conference on Image Processing. Piscataway: IEEE Press, 2015.
- [8] SUN H, ZHOU D, HU L, et al. Fast algorithm and VLSI architecture of rate distortion optimization in H.265/HEVC[J]. IEEE Transactions on Multimedia, 2017, 19(11): 2375-2390.
- [9] CHEN H C, CHANG T S. Fast rate distortion optimization with adaptive context group modeling for HEVC[C]//Proceedings of 2017 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE Press, 2017: 1-4.
- [10] SHARABAYKO M, PONOMAREV O. Fast rate estimation for RDO mode decision in HEVC[J]. Entropy, 2014, 16(12): 6667-6685.
- [11] XIAOFENG H, HUIZHU J, BINBIN C, et al. Fast algorithms and VLSI architecture design for HEVC intra-mode decision[J]. Journal of Real-Time Image Processing, 2015, 12(2): 285-302.
- [12] LANDAN H, HEMING S, DAJIANG Z, et al. Hardware oriented rate-distortion optimization algorithm for HEVC intra-frame encoder[C]//Proceedings of IEEE International Conference on Multimedia & Expo Workshops (ICMEW). Piscataway: IEEE Press, 2015.
- [13] WEIWEI S, YIBO F, LEILEI H, et al. A hardware friendly method for rate-distortion optimization of HEVC intra coding[C]//Proceedings of International Symposium on VLSI Design, Automation and Test (VLSI-DAT). [S.l.:s.n.], 2014.
- [14] ZHANG Y, LU C. High-Performance Algorithm Adaptations and Hardware Architecture for HEVC Intra Encoders[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019(99): 1.
- [15] 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.
- [16] SHENG Z, ZHOU D, SUN H, et al. Low-complexity rate-distortion optimization algorithms for HEVC intra prediction[C]//Proceedings of International Conference on Multimedia Modeling. [S.l.:s.n.], 2014.
- [17] BOSSEN F. Common test conditions and software reference configurations[Z]. 2013.
- [18] BJONTEGAARD G. Calculation of average PSNR differences between RD-curves (VCEG-M33)[Z]. 2001.

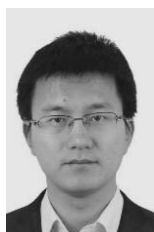
[作者简介]



孟翔 (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。



殷海兵 (1974—), 男, 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。



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