

doi: 10.11830/ISSN.1000-5013.201704021



HEVC 帧内预测编码的 快速 CU 决策算法

吴良堤, 冯桂

(华侨大学 信息科学与工程学院, 福建 厦门 361021)

摘要: 提出一种快速编码单元(CU)修剪方法和 CU 深度范围决策方法. 在快速 CU 修剪方法中, 根据贝叶斯决策准则, 基于残差信息 Hadamard 变换绝对值之和(SATD)提前结束 CU 的修剪; 在 CU 深度范围决策方法中, 根据当前 CU 深度与空间邻近 CU 深度的相似性决策当前 CU 深度级范围. 通过在 HM10.1 的实验证明: 文中方法相比于 HM10.1 的标准代码, 能降低 35.9% 的编码时间, 且 BDBR 的增加仅为 1.039%.

关键词: 高效视频编码; 帧内编码; 编码单元; 残差信息; Hadamard 变换

中图分类号: TN 919.81 **文献标志码:** A **文章编号:** 1000-5013(2017)04-0556-06

Fast CU Decision Algorithm for HEVC Intra Prediction Coding

WU Liangdi, FENG Gui

(College of Information Science and Engineering, Huaqiao University, Xiamen 361021, China)

Abstract: In this paper, a fast coding unit (CU) pruning and depth levels decision method is proposed. The early CU pruning method is performed according to a Bayes decision rule method based on the absolute sum of Hadamard transformed coefficients of the residual signal (SATD). Based on the depth information resemblance between spatio adjacent CUs and the current CU, some depths can be excluded. Experimental results show that the proposed algorithm achieves 35.9% encoder time saving on average compared to the original coding in HM10.1 with only 1.039% bit rate increment in coding performance.

Keywords: high efficiency video coding; intra coding; coding unit; residual information; absolute sum of Hadamard transformed

2013 年, JCT-VC 制定了高效视频编码(HEVC), 它采用混合编码结构^[1]. HEVC 引入了新的编码工具, 使其具有更高的压缩能力, 同时也导致了编码复杂度的提高. 为了提高 HEVC 的实时性, 研究者们提出了 3 类快速帧内编码算法: 快速 CU 尺寸决策算法、快速模式决策算法和 RQT 优化算法. 在快速 CU 尺寸决策算法上, Shen 等^[2]将 CU 分割公式化, 提出基于贝叶斯准则的快速 CU 决策; Shen 等^[3]根据邻近 CU 提出一个预测当前 CU 深度的公式, 跳过某些 CU 深度级搜索; Kim 等^[4]设定一个基于 RD 代价的阈值来提前终止当前 CU 的分割; Xu 等^[5]利用灰度级自相关函数来决策当前 CU 的深度范围, 同时利用固定的残差信息 Hadamard 变换绝对值之和(SATD)作为阈值来提前终止 CU 修剪过程. 在快速模式决策算法方面, Ting 等^[6]采用了变换域的边缘检测法, 减化 RMD 模式个数; Fang 等^[7]通过计算特定角度的方向能量减少进行 RMD 过程的预测模式; Yan 等^[8]通过对 RMD 后的模式进行分组合并, 减少 RDO 候选模式个数. 在 RQT 优化算法中, Teng 等^[9]利用合并分割决策过程代替原来的深度

收稿日期: 2016-09-04

通信作者: 冯桂(1960-), 女, 教授, 博士, 主要从事图像信息处理的研究. E-mail: fenggui@hqu.edu.cn.

基金项目: 福建省自然科学基金资助项目(2014J01242, 2016J01306)

优先模式决策过程;Zhang 等^[10]根据不同尺寸 CU 的最大帧间 RQT 深度在编码性能上不相同,提出一种帧间模式的自适应 RQT 选择. 上述算法在测试中都取得了一定的效果. Cho 等^[11]提出以 RMD 后的代价和 RDO 后的代价为阈值,利用贝叶斯准则自适应地对 CU 进行快速修剪和分割. 基于此,本文以 SATD 为阈值,利用贝叶斯准则来自适应地提前终止 CU 修剪过程.

1 HEVC 帧内预测编码概述

HEVC 编码中引入 3 种单元:编码单元(CU)、预测单元(PU)和变换单元(TU). 相对于 H. 264 固定大小的宏块,HEVC 提供的 CU 尺寸从 64×64 到 8×8,对应深度(Depth)级分别从 0 到 3. CU 的分割及对应的四叉树结构,如图 1 所示. 在 HEVC 测试模型(HM)中,帧内预测编码主要由 CU 分割和 CU 修剪完成. CU 分割过程如下:首先,CU 从 Depth=0 开始进行预测编码,获得率失真代价(RD Cost);然后,将 CU 分割成 4 个等大的子 CU,此时 Depth=1,并对当前 CU 进行预测编码得其 RD Cost;以此类推,直至 CU 深度为 3 时,结束分割.

CU 修剪过程如下:从 Depth=3 的 CU 开始,比较 4 个 8×8 子 CU 的 RD Cost 之和是否小于其父 CU 的 RD Cost,若小于,则选择子 CU;否则,选择父 CU;以此类推,直至 Depth=0. 32×32 CU 的分割与修剪示意图,如图 2 所示. 图 2 中:灰色阴影为修剪过程;其他为分割过程.

HEVC 帧内预测提供了 35 种预测模式,如图 3 所示. 帧内模式决策过程:首先,进行粗选择(RMD),即对 35 种预测模式进行低复杂度的 RD Cost 计算,选出其中代价最小的 N 个,作为率失真优化(RDO)的初始候选模式集;然后,判断 MPMs 是否包含在初始候选模式集中,若不包含则将相应模式添加到 RDO 初始候选模式集中,反之,则不添加;其次,对 N 到 $N+2$ 个候选模式用式(1)计算 RD Cost,选出其中 RD Cost 最小的作为最佳预测模式;最后,利用 RQT 来得到最佳 TU 尺寸,即

$$J = SSE + \lambda_{mode} \cdot R_{mode}. \tag{1}$$

式(1)中:SSE 表示原始图像块与预测图像块差的平方和; λ_{mode} 为拉格朗日乘子; R_{mode} 为编码预测模式信息所需的比特数.

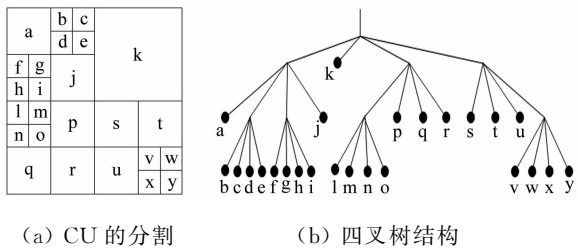


图 1 CU 的分割及对应的四叉树结构

Fig. 1 Partitioning CU and corresponding quadtree structure

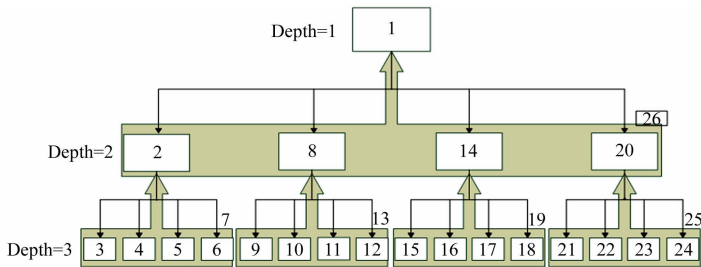


图 2 32×32 CU 的分割与修剪

Fig. 2 32×32 CU splitting and CU pruning process

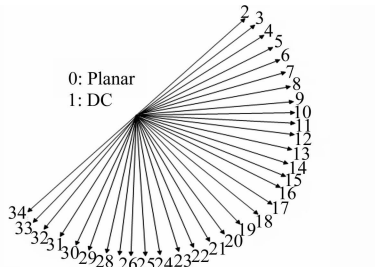


图 3 帧内预测模式及方向

Fig. 3 Intra prediction modes and directions

2 快速帧内 CU 尺寸决策算法

帧内预测编码复杂度增大的原因主要有两方面:CU 分割与修剪的复杂化和模式决策的复杂化. 为了降低帧内编码的复杂度,根据当前 LCU 与空间邻近 LCU 深度的相似性,缩小 LCU 分割的深度范围,同时提出了基于贝叶斯准则利用 SATD 提前结束 CU 修剪过程.

2.1 空间邻近 LCU 决策当前 LCU 深度范围

2.1.1 观察与分析 在帧内预测中,LCU 分割深度级与图像纹理有很大的关系. 进一步分析可知:当前 LCU 的分割深度级与空间邻近(左边、上边、左上、右上)LCU 的分割深度级具有极大的相似性. 当前 LCU 与空间邻近 LCU 的空间位置关系,如图 4 所示.

为了量化空间邻近 LCU 与当前 LCU 深度的相似度,定义相似度计算公式为

$$S_X = \frac{\sum_{i=1}^N (\text{cur}(i) = \text{nei_X}(i))}{N}, \quad X = \text{L}, \text{U}, \text{LU}, \text{RU}. \quad (2)$$

式(2)中:cur(*i*)为当前第 *i* 个 LCU 的深度;nei_X(*i*)为与当前第 *i* 个 LCU 邻近(*X*=L 表示左邻近,以此类推)的 LCU 的深度;*N* 为统计的当前 LCU 的总个数;*S_L*,*S_U*,*S_{LU}*,*S_{RU}* 分别为与当前 LCU 左邻近、上邻近、左上邻近和右上邻近 LCU 的相似度. 当前 LCU 的深度与邻近 LCU 的深度相等时,cur(*i*)=nei_X(*i*)为 1;否则为 0. 因此,*S_X* 的取值为[0,1].

经过上述定义后,取各分辨率的序列,计算当前 LCU 与各邻近 LCU 的相似度. 每个序列在不同的量化参数下编码,量化参数选择为 22,27,32,37. 其中:PeopelOnStreet,Cactus,BasketballDrill,BasketballPass 属于纹理复杂、运动较剧烈的序列;Traffic,BQMall,BQSquare 属于纹理较均质、平坦,运动变化较小的序列;FourPeople,AristenAndSara 属于背景较简单、前景较复杂的序列. 得到当前 LCU 与邻近 LCU 相似度 *S_L*,*S_U*,*S_{LU}*,*S_{RU}* 分别为 0.795,0.794,0.788,0.763. 由此可知:当前 LCU 与左邻近 LCU、上邻近 LCU、左上邻近 LCU、右上邻近 LCU 的相似度呈递减趋势. 因此,可以利用空间邻近 LCU 的深度级跳过当前 LCU 不太可能的深度级搜索.

2.1.2 算法内容 记当前 LCU 空间邻近 4 个 LCU 的深度分别为:LDepth(左边)、UDepth(上边)、LUDepth(左上)、RUDepth(右上).

将 LDepth,UDepth,LUDepth,RUDepth 的异同分两大类:LDepth,UDepth,LUDepth,RUDepth 有任意两个深度相等;LDepth,UDepth,LUDepth,RUDepth 的深度级各不相等.

对于第一大类,又可分成两小类:LDepth≠UDepth 和 LDepth=UDepth. 当 LDepth=UDepth 时,当前 LCU 有极高的可能性选择与其相同的深度级作为当前 LCU 的最佳分割深度级. 为了减少这种情况引起的编码性能的损失,增加一个额外的深度级进行搜索. 此额外深度级 *e_Depth* 定义为

$$e_Depth = \begin{cases} LDepth + 1, & LDepth \leq 2, \\ LDepth - 1, & LDepth = 3. \end{cases} \quad (3)$$

当 LDepth≠UDepth 时,可通过判断空间邻近 LCU 各深度级的个数决策当前 LCU 的深度级搜索. 若空间邻近 LCU 中深度 0 和深度 1 的个数之和多于深度 2 和深度 3 的个数之和,则将当前 LCU 搜索深度级最大值设为 2;反之,则将当前 LCU 搜索深度级最小值设为 1.

对于第二大类,当 LDepth,UDepth,LUDepth 只要有一个的深度为 3,此时当前 LCU 选择深度级 0 为最佳分割的概率非常小. 因此,将当前 LCU 的深度级最小值设为 1;反之,将当前 LCU 的深度级最大值设为 2.

为了证明上述算法中利用空间邻近 LCU 深度预测当前 LCU 深度搜索范围的合理性,取与上述一致的测试序列,统计其命中率. 统计结果如表 1 所示. 通过表 1 分析可知:利用空间邻近 LCU 跳过当前 LCU 不必要的深度级搜索,平均命中率达到 94.1%,足以说明所提出的利用空间邻近

LCU 决策当前 LCU 的深度范围能精确、有效地排除不必要的深度级搜索.

2.2 基于贝叶斯准则利用 SATD 提前结束 CU 修剪的算法

由上帧内预测编码概述可知,HEVC 引入了 RMD 算法,其计算代价表示为

$$J_{\text{RMD}} = \text{SATD} + \lambda_{\text{pred}} \cdot R_{\text{pred}}. \quad (4)$$



图 4 当前 LCU 的空间邻近 LCU
Fig. 4 Spatial neighbor LCU of current LCU

表 1 空间邻近 LCU 预测当前 LCU 深度范围命中率

Tab. 1 Hit-ratio of spatial neighbor LCU to predicted depth of current LCU

序列	总的 LCU 块数	错判的 LCU 块数	命中率/%
PeopleOnSrreet	200 000	9 083	95.4
Traffic	200 000	15 916	92.0
Cactus	102 000	6 858	93.4
BasketballDrill	20 800	257	98.7
BQMall	20 800	292	98.5
BasketballPass	5 600	341	93.9
BQSquare	5 600	22	99.6
FourPeople	48 000	3 472	92.7
Aristen & Sara	48 000	7 988	83.3
平均	—	—	94.1

式(4)中: SATD 为残差信息绝对 Hadamard 变换之和; λ_{pred} 为拉格朗日乘子; R_{pred} 为编码预测模式信息所需的比特数. SATD 的定义为

$$\text{SATD} = \sum_{i=0}^N \sum_{j=0}^N \frac{|d_{i,j}|}{2}.$$

(5)

式(5)中: N 为残差信息块的大小; $d_{i,j}$ 为残差信息经 Hadamard 变换后位于 (i, j) 的值, 其定义式为

$$d = \mathbf{H}(\mathbf{S} - \mathbf{C})\mathbf{H}^T.$$

(6)

式(6)中: \mathbf{S} 为原始图像块; \mathbf{C} 为重构图像块; \mathbf{H} 为 Hadamard 矩阵.

2.2.1 动机与统计 将当前 CU 不分割标记为 w_1 , 分割标记为 w_2 , 则由贝叶斯公式可得

$$P(w_j | \text{SATD}) = \frac{P(\text{SATD} | w_j) \cdot P(w_j)}{P(\text{SATD})}.$$

(7)

式(7)中: $P(\text{SATD} | w_j)$ 为在已知分割(或不分割)下 SATD 的条件概率; $P(w_j)$ 为分割(或不分割)的先验概率, 表示为

$$P(w_j) = \frac{N_{w_j}}{N_{w_1} + N_{w_2}}, \quad j = 1, 2.$$

(8)

$P(\text{SATD})$ 表示 SATD 的先验概率, 表示为

$$P(\text{SATD}) = \sum_{j=1}^2 P(\text{SATD} | w_j) \cdot P(w_j).$$

(9)

由式(7)可知: 为了利用贝叶斯准则判断当前 CU 分割与否, 需先统计相关的参量, 如表 2 所示.

表 2 各 CU 级下需获取的参数
Tab. 2 Needed parameters in each CU depth level

参量来源	参数	性质	参量来源	参数	性质
由原始 HM10.1 获得的参量	N_{w_1}	不分割 CU 的总数	统计、计算的 更新参量	$P(w_1)$	分割 CU 的先验概率
	N_{w_2}	分割 CU 的总数		μ_{SATD}	分割 CU 的 SATD 均值
	$\text{SATD} w_2$	分割 CU 的 J_{SATD}		σ_{SATD}	分割 CU 的 SATD 标准差

为使算法具备较好的适应性, 文中相关参量都是在线周期性更新, 更新周期为帧率.

如同文献[11]的 J_{SATD} 服从高斯随机分布, 文中也先假设 SATD 服从高斯分布. 从这一假设出发, 统计了 SATD 与 $P(\text{SATD} | w_2)$, SATD 与 $P(\text{SATD} | w_1)$ 的关系. 经过使用各种测试序列进行大量的实验, 证明了 SATD 近似服从高斯分布. 以 BasketballDrill 这一序列为例, 其中, SATD 以 200 为间隔, 统计了 QP=37, 深度为 2 时的实际分布, 并给出其拟合的高斯模型, 如图 5 所示.

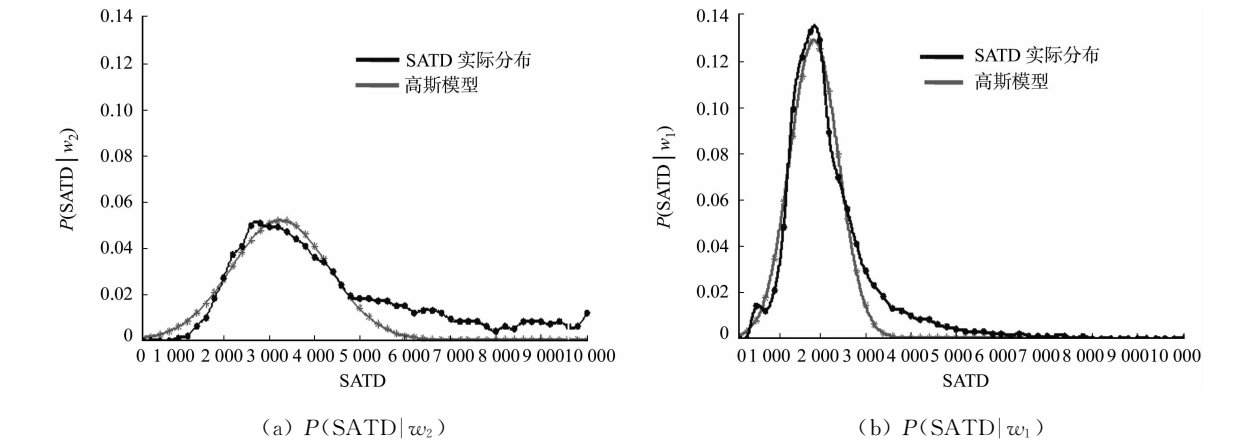


图 5 BasketballDrill 序列的 RDcost 实际分布及其拟合高斯模型

Fig. 5 BasketballDrill sequence actual distribution of RD cost and fit Gaussian distribution

2.2.2 阈值的计算 CU 提前修剪的贝叶斯决策, 如图 6 所示. 由图 6 可知: α_{ST} 表示在 CU 继续分割时被判断为不再分割的概率. 在每一个 CU 的深度级上, 当最小 SATD 小于预设阈值 SATD_TH 时, 则执行提前修剪算法. 由图 6 易得

$$\alpha_{\text{ST}} = \int_{-\infty}^{\text{SATD_TH}} P(x | w_2) \cdot P(w_2) dx.$$

(10)

式(10)中; $x=\text{SATD}$. 将式(10)变换为

$$\frac{\alpha_{\text{ST}}}{P(w_2)} = \int_{-\infty}^{\text{SATD_TH}} P(x | w_2) \mathrm{d}x. \tag{11}$$

由上述分析可知,SATD 服从高斯分布,为了得到给定 α_{ST} 下的 SATD_TH,需在更新阶段获得均值 μ_{SATD} 与方差 σ_{SATD}^2 ,通过这些参量来得到阈值 SATD_TH. 将式(11)标准化,则可得

$$\text{SATD_TH} = \mu_{\text{SATD}} + z \cdot \sigma_{\text{SATD}}. \tag{12}$$

式(12)中; z 为服从标准正态分布变量积分的上界,可由查找标准正态分布表得到其值. 即

$$\frac{\alpha_{\text{ST}}}{P(w_2)} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z \exp\left(-\frac{u^2}{2}\right) \mathrm{d}u. \tag{13}$$

大量实验证明,取 $\alpha_{\text{ST}}=0.1$ 时,在编码时间与码率损失之间可以达到较好的平衡.

3 结果与分析

文中提出的快速帧内 CU 尺寸决策算法在 HM10.1 上实现. 实验配置参照文献[12],使用 QP=22,27,32,37,LCU 尺寸为 64×64 ,最大深度级为 4. 测试使用 JCT-VC 推荐的 5 个类别序列:ClassA, ClassB,ClassC,ClassD,ClassE. 所用计算机处理器为 AMD Athlon(tm) 2×2 B28. 使用 BDBR, BDPSNR^[13] 来衡量编码性能,用平均节省时间(ATS)来衡量编码复杂度的降低,即

$$\text{ATS} = \frac{1}{4} \sum_{i=1}^4 \frac{\text{Time}_{\text{HM10.1}}(\text{QP}_i) \text{Time}_{\text{pro}}(\text{QP}_i)}{\text{Time}_{\text{HM10.1}}(\text{QP}_i)} \times 100\%. \tag{14}$$

式(14)中: $\text{Time}_{\text{HM10.1}}(\text{QP}_i)$, $\text{Time}_{\text{pro}}(\text{QP}_i)$ 分别为原始 HM10.1 与文中算法在不同 QP 下的编码时间. 文中算法与文献[3],[4]比较的实验结果,如表 3 所示. 文中算法与文献[5]ClassE 测试序列的对比结果,如表 4 所示.

表 3 文中算法与文献[3],[4]算法比较结果

Tab. 3 Performance comparison between proposed and literature [3],[4]

序列	文中算法			文献[3]算法			文献[4]算法		
	BDPNR/dB	BDBR/%	ATS/%	BDPNR/dB	BDBR/%	ATS/%	BDPNR/dB	BDBR/%	ATS/%
ClassA	-0.054	1.086	-40.7	-0.031	0.636	-23.3	-0.047	0.949	-34.4
ClassB	-0.035	1.090	-37.2	-0.032	0.981	-26.7	-0.021	0.683	-21.1
ClassC	-0.045	0.836	-29.3	-0.018	0.349	-18.1	-0.017	0.333	-20.0
ClassD	-0.029	0.529	-22.9	-0.016	0.296	-10.2	-0.007	0.126	-16.4
ClassE	-0.071	1.654	-49.3	-0.072	1.696	-28.6	-0.150	3.339	-55.6
平均	-0.047	1.039	-35.9	-0.034	0.791	-21.4	-0.048	1.086	-32.5

表 4 文中算法与文献[5]算法 ClassE 测试序列的比对

Tab. 4 Coding performance compared with ClassE in literature [5]

序列	文中算法			文献[5]算法		
	BDPNR/dB	BDBR/%	ATS/%	BDPNR/dB	BDBR/%	ATS/%
FourPeople	-0.046	0.875	-44.4	-0.085	1.646	-51.7
ClassE Johnny	-0.081	2.211	-45.8	-0.279	7.565	-62.2
Kristen & Sara	-0.087	1.875	-57.6	-0.194	4.216	-66.1
平均	-0.071	1.654	-49.3	-0.186	4.756	-60.0

通过表 3 可知:文中算法平均能节省 35.9% 的编码时间,引起 1.039% 的 BDBR 增加. 同时,从表 3 中可以发现:相较于文献[3],文中算法以增加微量的 BDBR 为代价,平均多节省了 14.5% 的编码时间;与文献[4]相比较,文献[4]算法对 ClassE 的测试序列码率增加比较大,平均为 3.339%;而文中算法对 18 个测试序列编码性能的损失都没那么严重,最大仅为 1.654%;同时可以发现两者的平均率失真性能几乎相等,但文中算法编码时间减少的更多,平均多降低了 3.4%. 这说明文中算法与文献[4]相比,所

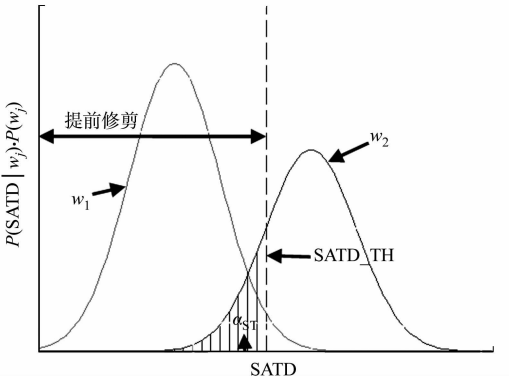


图 6 CU 提前修剪的贝叶斯决策

Fig. 6 Bayes decision for early CU pruning

有的测试序列都能在编码时间与码率之间达到更好的平衡。

由表 4 可知:虽然文中算法在编码时间降低上略有不足,但其对 ClassE 序列平均 BDBR 的增加为 1.654%;而文献[5]对 ClassE 序列平均 BDBR 增加了 4.765%,在 Johnny 测试序列中达到了 7.565%。这是比较难以接受的。通过这一点的对比,更能突显出文中算法对序列的自适应性。

4 结束语

提出了基于贝叶斯准则使用 SATD 提前终止 CU 修剪过程,同时利用空间邻近 LCU 与当前 LCU 深度级的相似性决策当前 LCU 深度范围的快速 CU 尺寸决策算法。该算法相比于原本的 HM10.1,平均能节省 35.9%的编码时间,而引起的平均码率增加只有 1.039%。下一步的研究方向主要是在帧内模式决策上进行改进,从而获得更大的编码复杂度降低。

参考文献:

[1] SULLIVAN G J, HAN W J, WIEGAN D T. 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] SHEN Xiaolin, YU Lu, CHEN Jie. Fast coding unit size selection for HEVC based on Bayesian decision rule[C]// Picture Coding Symposium. Krakow: IEEE Press, 2012: 453-456.

[3] SHEN Liquan, ZHANG Zhaoyang, AN Ping. Fast CU size decision and mode decision algorithm for HEVC intra coding[J]. IEEE Transactions on Consumer Electronics, 2013, 59(1): 207-213.

[4] KIM J, CHOE Y, KIM Y G. Fast coding unit size decision algorithm for intra coding in HEVC[C]// IEEE International Conference on Consumer Electronics. Las Vegas: IEEE Press, 2013: 637-638.

[5] XU Dongxu, LIN Qiwei, DONG Xiaohui. Fast intracoding unit size decision algorithm for high-efficiency video coding[J]. Journal of Electronic Imaging, 2014, 23(1): 013012.

[6] TING Y C, CHANG T S. Fast intra prediction algorithm with transform domain edge detection for HEVC[C]// IEEE Asia Pacific Conference on Circuits and Systems. Kaohsiung: IEEE Press, 2012: 144-147.

[7] FANG Cemin, CHANG Yuanteng, CHUNG Wenhao. Fast intra mode decision for HEVC based on direction energy distribution[C]// IEEE 17th International Symposium on Consumer Electronics. Hsinchu: IEEE Press, 2013: 61-62.

[8] YAN Shunqing, HONG Liang, HE Wei-feng, *et al.* Group-based fast mode decision algorithm for intra prediction in HEVC[C]// Eighth International Conference on Signal Image Technology and Internet Based Systems. Naples: IEEE Press, 2012: 225-229.

[9] TEND S W, HANG H M, CHEN Y F. Fast mode decision algorithm for residual quadtree coding in HEVC[C]// IEEE Visual Communications and Image Processing. Tainan: IEEE Press, 2011: 1-4.

[10] ZHANG Yongfei, ZHAO Mingfei. An adaptive RQT mode selection algorithm for HEVC[C]// 5th International Congress on Image and Signal Processing. Chongqing: IEEE Press, 2012: 173-177.

[11] CHO S, KIM M. Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2013, 23(9): 1555-1564.

[12] BOSSEN F. Common HM test conditions and software reference configuration[C]// 12th JCT-VC Meeting. Geneva: JCTVC-L1100, 2013: 1-3.

[13] BJONTEGAARD G. Calculation of average PSNR difference between RD-curves[C]// 13th VCEG-M33 Meeting. Austin: [s. n.], 2001: 290-294.

(责任编辑: 黄晓楠 英文审校: 吴逢铁)