

## 高性能视频编码帧间预测的单元划分优化算法

单娜娜\* 周巍 段哲民 魏恒璐

(西北工业大学电子信息学院 西安 710072)

**摘要:** 新一代的高性能视频编码(HEVC)通过应用各种先进技术来大幅提高视频编码的性能,然而,这些方法也大大增加了整个编码过程,尤其是预测编码阶段的计算复杂度。该文提出一种应用于高性能视频编码帧间预测的快速单元划分算法。该算法基于 HEVC 灵活的四叉树结构,根据被编码单元的运动特性来确定所编码信息的阈值,并以此作为单元划分的提前终止条件,从而减少了不必要的单元划分操作和率失真代价的相关计算,达到节省编码时间和降低编码复杂度的目的。实验结果表明,在峰值信噪比(PSNR)损失仅为 0.0418 dB 的情况下,所提算法可以平均降低 46.1%的编码时间。

**关键词:** 高性能视频编码; 编码单元; 帧间预测; 率失真代价

**中图分类号:** TN919.81

**文献标识码:** A

**文章编号:** 1009-5896(2016)05-1194-08

**DOI:** 10.11999/JEIT150843

## An Effective CU Splitting Algorithm in Inter Prediction of HEVC

SHAN Nana ZHOU Wei DUAN Zhemin WEI Henglun

(Department of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract:** High Efficiency Video Coding (HEVC) provides better compression performance by adopting many new techniques. However these tools also increase the computational complexity of the prediction, which consumes most of encoder computations. This paper proposes an effective splitting algorithm in inter prediction of HEVC. Based on the motion homogeneity of coding unit, a certain threshold is used to decide whether the coding unit should be split into smaller ones. So the unnecessary calculation decreases rapidly. By skipping some specific sub-CUs, the coding complexity is dramatically improved. Experimental results show that the proposed technique can save 46.1% coding time in average with negligible loss of coding efficiency, and the decrease of PSNR is only 0.0418 dB.

**Key words:** High Efficiency Video Coding (HEVC); Coding Unit (CU); Inter prediction; Rate distortion cost

### 1 引言

新一代视频编码标准,即高性能视频编码(High Efficiency Video Coding, HEVC)标准,是由 VCEG (Video Coding Experts Group)和 MPEG(Moving Picture Experts Group)联合成立的视频编码联合组(Joint Collaborative Team on Video Coding, JCT-VC)开发制定的<sup>[1]</sup>。HEVC 目前已经基本实现了在相同的质量下,压缩率比 H.264/AVC 提高一倍的目标,但是如何进一步提升编码效率和降低编码复杂度仍然是该领域内极具挑战性的热点问题。

在针对提高编码单元(CU)划分效率的有关问题上,人们进行了大量的研究,文献[2]提出了借助图像梯度信息提前选择 CU 深度范围的快速选择算法。文献[3]将率失真(Rate-Distortion, RD)代价作为提前终止 CU 划分的条件。文献[4]根据统计结果,提出在 CU 层上提前终止划分的优化算法。文献[5]通过分析时空相关 CTU 的深度信息,减少深度搜索范围。文献[6]首先基于 Sobel 边缘检测算子缩小 CU 遍历的深度范围,然后计算 RD 代价函数的阈值并以此提前终止 CU 的划分。文献[7]利用 CU 的四叉树特点和其时空相关性提出了 3 种自适应模式选择策略。文献[8]根据时域上相邻两帧相同位置编码单元 PU 模式间的相关性,跳过 CU 的某些冗余划分模式。文献[9]提出一种 PMD(Pyramid Motion Divergence)算法来进行 CU 划分模式的判定。文献[10]算法跳过前一帧或者相邻 CU 中较少使用的划分模式,根据 CU 特性来跳过某些划分。

本文算法首先根据被编码单元的运动特性将编

收稿日期: 2015-07-13; 改回日期: 2015-12-11; 网络出版: 2016-02-03

\*通信作者: 单娜娜 helensnn@hotmail.com

基金项目: 国家自然科学基金(60902101), 教育部新世纪优秀人才计划(NCET-11-0824), 中央高校基本科研业务费专项资金(3102014JCQ01057)

Foundation Items: The National Natural Science Foundation of China (60902101), New Century Excellent Talents in University (NCET-11-0824), The Fundamental Research Funds for the Central Universities (3102014JCQ01057)

码单元所在的区域分类,从而将编码运动信息所需要的比特数作为判断是否进行下一步划分的阈值初值,再根据其对应率失真代价的变化计算阈值补偿,并进行自适应调整,从而达到节省编码时间,降低编码复杂度的目的。

## 2 HEVC 中的划分

HEVC 将编码帧分成若干编码树块 CTB (Coding Tree Block), CTB 是进行预测、变换、量化、熵编码等一系列处理的基本单元。相同位置的一个亮度 CTB 和两个色度 CTB,以及它们的语法元素组成了编码树单元 CTU (Coding Tree Unit)<sup>[11,12]</sup>。

HEVC 同 H.264/AVC 一样,仍然采用了分块的编码方式,但是块的尺寸和划分方式可以自适应

地进行变化,其中包含 3 类单元,即编码单元 CU (Coding Unit),预测单元 PU (Prediction Unit) 和变换单元 TU (Transform Unit)。

### 2.1 编码单元 CU

HEVC 采用灵活的四叉树结构来进行 CTU 的划分,被划分的 CU 尺寸可以从  $64 \times 64$ ,  $32 \times 32$ ,  $16 \times 16$ ,  $8 \times 8$  之间变化,其划分深度相应的分别为 0, 1, 2, 3。其中  $64 \times 64$  大小的 CU 被称为 LCU (Largest Coding Unit),如果一个 CTU 不进行分解,那么它就仅包含一个  $64 \times 64$  的 LCU,当一个 CU 一直递归的被划分为  $8 \times 8$  大小时,到达其最小的划分 SCU (Smallest Coding Unit)。

图 1 显示了某一帧的某个 CTU 的划分情况,图 2 是其对应的四叉树结构。实线表示的是 CU 的边界,虚线表示的是 PU 的边界。

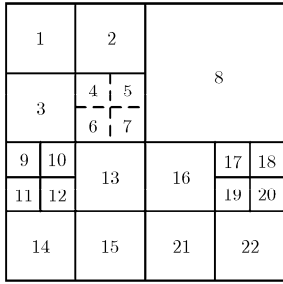


图 1 某个 CTU 的划分

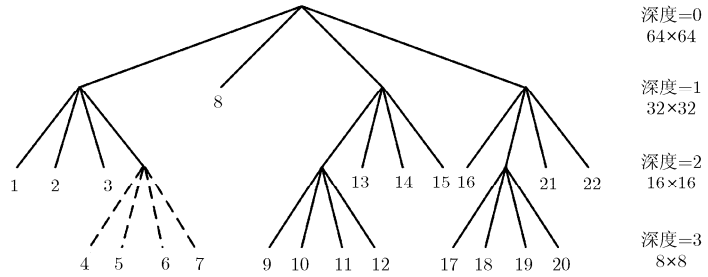


图 2 CTU 划分对应的四叉树表示

### 2.2 预测单元 PU

CTU 在被划分为 CU 的同时,也在进行预测单元(PU)的划分。它们同样也是按照四叉树结构灵活地、递归地来进行划分的,其中 PU 的尺寸大小可以从  $64 \times 64$  到  $4 \times 4$ ,当 CU 是  $8 \times 8$  时,PU 可以是  $8 \times 8$  也可以是  $4 \times 4$ 。

HEVC 中的预测单元是进行预测的基本单位,一个 CU 可以按照所计算的最佳率失真代价被划分为 1 个、2 个或者 4 个 PU,其划分可以是对称的,也可以是非对称的。如图 3 所示,一个  $2N \times 2N$  ( $N$  可以是 32, 16, 8, 4) 的 CU 可以有 8 种划分方式,其中,前 4 种是对称划分,后 4 种是非对称划分。在进行预测时,可以在这 8 种划分方式中任意选择,假如选择了非对称划分方式,则达到较为精确的匹配,从而不需要再进一步划分。

### 2.3 率失真优化

基于 Shannon 的率失真编码理论,视频压缩算法的效率是由编码产生的比特和压缩产生的失真共同来衡量的,通过在两者之间的折中,达到在一定视频质量的条件下使用尽可能少的编码比特,或者

在一定码率要求下尽可能降低视频失真的目的<sup>[13-15]</sup>。

$$\text{SSE} = \sum (f(x, y) - f'(x, y))^2 \quad (1)$$

$$J_m = (\text{SSE}_{\text{lu}} + \omega_{\text{ch}} \cdot \text{SSE}_{\text{ch}}) + \lambda_m \cdot B_m \quad (2)$$

式(1)中的  $f(x, y)$  和  $f'(x, y)$  分别是位于  $(x, y)$  处的原始像素值和重构像素值,  $\text{SSE}_{\text{lu}}$  和  $\text{SSE}_{\text{ch}}$  (Sum of Squared Error) 分别表示亮度和色度的平方误差和,  $\omega_{\text{ch}}$  用来定义亮度和色度平方误差和的权重。式(2)通过引入拉格朗日因子  $\lambda_m$ , 将视频的率失真优化问题转化为无约束问题,因此在给定码率  $R_c$  下就可以确定  $\lambda_m$ , 从而寻找到最优的  $B_m$ , 也就是给定编码模式的编码比特数。

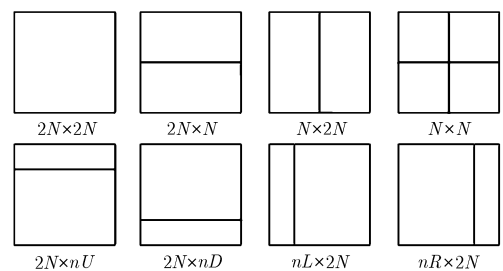


图 3 8 种划分模式

## 2.4 标准的划分流程

对于当前 CU 来说, 首先根据 2.3 节式(2)分别计算其在 SKIP, INTER 和 INTRA 几种模式下的率失真代价  $J_m$ , 并选择其中最优的率失真代价记为该 CU 的 cost。

一个 LCU 划分的具体流程如图 4 所示。

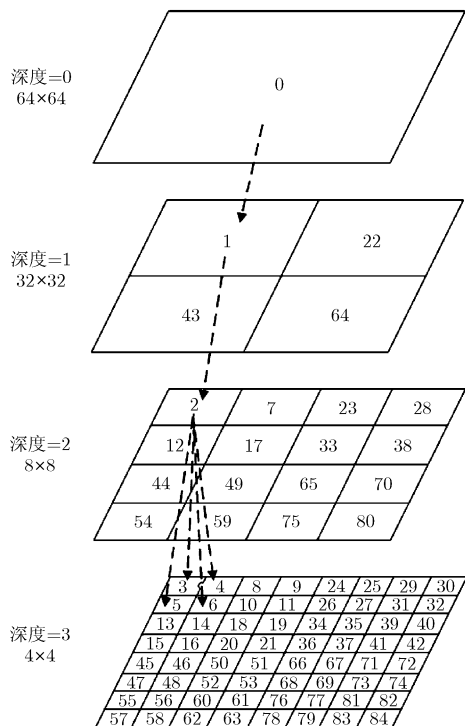


图4 标准的单元划分模式

(1)计算深度为  $i$  的 CU 被划分前的 cost, 记为  $\text{cost}_{\text{unsplit}}$ ;

(2)分别计算被划分为  $i+1$  深度后 4 个子 CU 的 cost, 然后对它们求和, 记为  $\text{cost}_{\text{split}}$ , 假如  $i+1$  深度不是最大深度, 还需要逐层递归进行向下的搜索, 直到到达最大深度;

(3)将划分之前和划分之后计算出来的率失真代价  $\text{cost}_{\text{split}}$  和  $\text{cost}_{\text{unsplit}}$  进行比较: 若  $\text{cost}_{\text{split}} < \text{cost}_{\text{unsplit}}$ , 则最佳单元划分模式为划分后的更小层次的 CU; 否则, 若  $\text{cost}_{\text{split}} \geq \text{cost}_{\text{unsplit}}$ , 则该 CU 不需要进行划分;

(4)按照 Z 扫描顺序逐级遍历所有 CU, 循环执行步骤(1)~(3), 直到该 LCU 的所有层次的每个 CU 都被计算并且判断完毕, 整个 LCU 的最佳划分方式被确定下来。

HEVC 中的单元划分过程是遍历, 并且逐层递归进行的<sup>[16,17]</sup>, 一个  $64 \times 64$  大小 LCU 需要计算和比较  $1+4+4 \times 4+4 \times 4 \times 4=85$  个不同大小的 CU。

## 3 CU 的快速划分算法

### 3.1 快速划分算法的思想

在以往的研究工作中, 有许多方法用来降低有关单元划分的编码复杂度。如提案文献[18-20]分别通过设置 CFM(CBF Fast Mode), ESD(Early SKIP Detection)和 ECU(Early CU)等特殊参数来判断是否满足提前终止划分的条件, 以便在不太影响整个编码效率的前提下, 跳过某些判决和划分。

如 2.3 节中所述, 标准中的 CU 划分是在 4 种深度级别上分别逐级递归计算 cost, 然后从中选择较小的 cost 作为最佳模式。按照原始的标准算法进行全深度遍历的话, 需要进行大量的计算和比较, 在得到精确划分方式的同时, 也带来了巨大的编码复杂度。图 5 是不同量化参数 QP 下最优划分的分布情况, 大量最终划分深度为 0, 1 和 2 的 CU 或 PU 仍然会进行深度为 3 的计算和比较工作, 而且随着 QP 变大, 最优划分为最小深度的可能性还会变得更小。假如能够适当减少这部分的划分工作, 则可以大幅提高编码速度。

### 3.2 CU 划分的自适应阈值

BasketballDrill( $832 \times 480$ ) 是标准测试序列之一, 首先, 通过分析它的码流文件, 如图 6 所示, 划分深度比较大的 CU 主要集中于篮球运动员、篮球以及篮球架和篮筐这几个区域, 其他区域的 LCU 几乎没有做划分。因此, 可以将视频中的每一帧按照编码运动信息所需比特数的多少, 分为运动剧烈区域和运动平缓区域, 并可以得到如下结论: 对于运动剧烈区域, 由于运动信息比较丰富, 需要使用较多的编码比特数和较大的 CU 深度才可以进行精确的匹配; 反之, 对于运动平缓区域, 因运动信息比较少, 则使用较少的编码比特数和较小的 CU 深度就可以精确地进行预测。

接着分析 CU 划分时各个编码运动信息比特数的出现次数, 也就是不同编码比特数出现的频率。如图 7 所示, 可以得到编码运动信息所需比特数主要集中在 0 到 100 之间, 大于 100 bit 数的只有极少出现频率。

最后, 针对测试序列 BasketballDrill( $832 \times 480$ ) 各帧中的每个 CU, 统计它们在某个编码运动信息比特数下不再进行下一步划分的概率。结果如表 1 所示, CU 的编码比特数越小, 那么它不再进行下一步划分的概率越大, 也就是说该 CU 最优划分模式的深度越小。

综上所述, 可以得出: 编码比特数主要集中在 100 以下的范围内, 而且在运动较平缓的区域内,

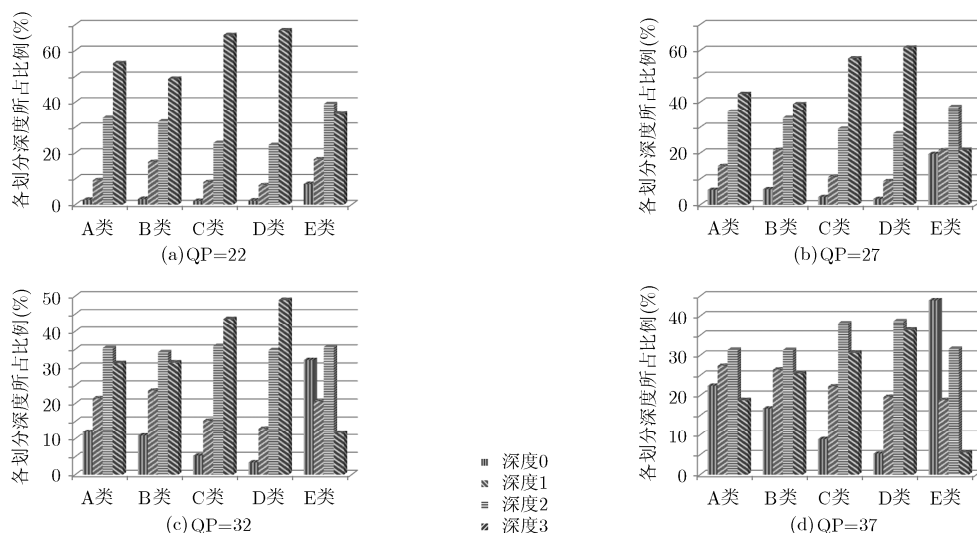


图 5 不同 QP 下最优划分模式的深度分布

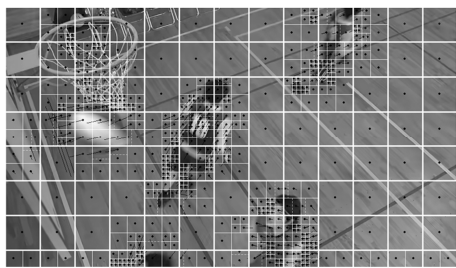


图 6 BasketballDrill(832×480)的 CU 划分和运动矢量

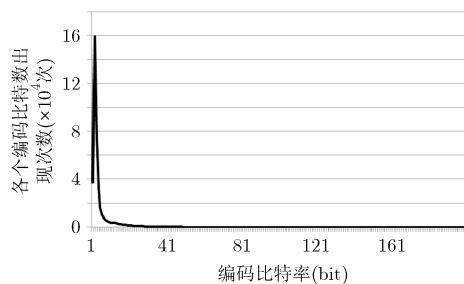


图 7 BasketballDrill(832×480)中 CU 的编码比特数统计图

编码比特数较小, CU 不再进行下一步划分的概率相对较大。因此, 可以设定某种阈值, 当编码运动

信息所需要的比特数小于该阈值时, 则可以跳过划分。即

$$\text{splitflag} = \begin{cases} \text{unsplit}, & \text{bits} \leq \text{Bits}_{\text{TH}} \\ \text{split}, & \text{bits} > \text{Bits}_{\text{TH}} \end{cases} \quad (3)$$

CU 编码比特和是否划分存在着密切的相关性, 通过表 1 中的数据可以得知, 在编码比特等于某个特定的值时, 可以得到当前 CU 不再进行划分的概率。假如将编码比特数的阈值设置的很小, 例如设为 20 的时候, 不划分的概率已经能够达到 98%, 因此对编码质量几乎没有影响; 反之, 假如将编码比特数的阈值设置过大, 虽然编码时间减少的比较多, 但是引起的失真也变大。

为了在编码时间和编码质量之间做出权衡, 本文提出了一种基于自适应编码比特阈值的划分判定模型, 并以此为基础计算出当前 CU 划分的编码比特阈值, 从而提高 CU 划分算法的效率。

设  $\Delta \text{cost}$  表示一个 CU 以某个比特值作为划分阈值后与最优划分模式相比所增加的 cost,  $\Delta \text{cost}$  和比特值之间的关系可以表示为

$$\Delta \text{cost} = F(\text{Bits}) = a + b \cdot \text{Bits} + c \cdot \text{Bits}^2 + \dots \quad (4)$$

表 1 测试序列 BasketballDrill(832×480)中 CU 的编码比特和不划分概率(%)

QP	比特数 B									
	5	10	15	20	25	30	35	40	45	50
22	99.97	99.32	98.16	97.86	97.41	96.88	96.35	95.81	95.27	94.72
27	99.98	99.48	98.59	98.10	97.49	96.87	96.20	95.54	94.92	94.32
32	99.79	98.96	98.14	97.70	97.20	96.73	96.32	95.96	95.64	95.36
37	99.96	99.33	98.24	97.75	97.24	96.77	96.36	95.94	95.62	95.29

式(4)的后半部分是  $F$  的泰勒展开式, 为简化计算, 省略式(4)中的高次项, 记为

$$\Delta \text{cost} = a + b \cdot \text{Bits} \quad (5)$$

又因为当  $\text{Bits} = 0$  的时候,  $\Delta \text{cost} = 0$ , 所以  $a = 0$ , 即

$$\Delta \text{cost} = b \cdot \text{Bits} \quad (6)$$

根据需要设置  $\Delta \text{cost}$  的上限  $\gamma$ , 其对应的比特值是  $\alpha$ 。在实际应用中,  $\Delta \text{cost}$  在  $\gamma$  附近波动, 因此有必要对编码比特的阈值做适当的调整。假如  $\Delta \text{cost}$  超过了设定的上限  $\gamma$ , 则编码比特的阈值需要减小, 从而减少跳过的 CU 划分, 以增加编码质量; 反之, 假如  $\Delta \text{cost}$  小于设定的上限  $\gamma$ , 则编码比特的阈值需要增大, 从而增加跳过的 CU 划分, 以减少编码时间。

根据式(5), 阈值的变化量为

$$\Delta \text{Bits} = \frac{1}{b} \cdot (\Delta \text{cost} - \gamma) \quad (7)$$

因此, 当前 CU 的阈值为

$$\text{Bits}_{\text{TH}} = \alpha + \Delta \text{Bits} \quad (8)$$

即

$$\text{Bits}_{\text{TH}} = \alpha + \beta \cdot (\Delta \text{cost} - \gamma) \quad (9)$$

其中,  $\beta = 1/b$ 。

### 3.3 快速划分算法的流程

快速划分算法的流程如图 8 所示, 具体分为以下步骤:

(1)初始化当前 CU 的信息;

(2)分别计算出当前 CU 在 Merge  $2N \times 2N$ , Inter  $2N \times 2N$ , Inter  $N \times 2N$ , Inter  $2N \times N$  这 4 种模式下的 cost;

(3)假如 AMP 为 true, 再分别计算当前 CU 在 Inter  $2N \times nU$ , Inter  $2N \times nD$ , Inter  $nL \times 2N$ , Inter  $nR \times 2N$  模式下的 cost;

(4)比较当前 CU 在以上各个模式下的 cost, 取最小值, 记为当前 CU 最优预测模式;

(5)采用本文算法计算出划分的阈值  $\text{Bits}_{\text{TH}}$ , 并根据式(9)进行自适应的调整;

(6)将步骤(4)下的 bits 与步骤(5)中计算出来的  $\text{Bits}_{\text{TH}}$  进行比较, 假如  $\text{bits} \leq \text{Bits}_{\text{TH}}$ , 该 CU 的深度即为当前深度, 不再进行划分; 反之, 假如  $\text{bits} > \text{Bits}_{\text{TH}}$ , 该 CU 按照 HM 标准算法, 需要进行下一步的划分, 深度+1;

(7)返回到步骤(1)。

## 4 实验结果分析

实验得到编码比特数 30 可以作为判断是否进行划分的阈值初值, 自适应改变阈值步长  $\beta = 0.4$ ,

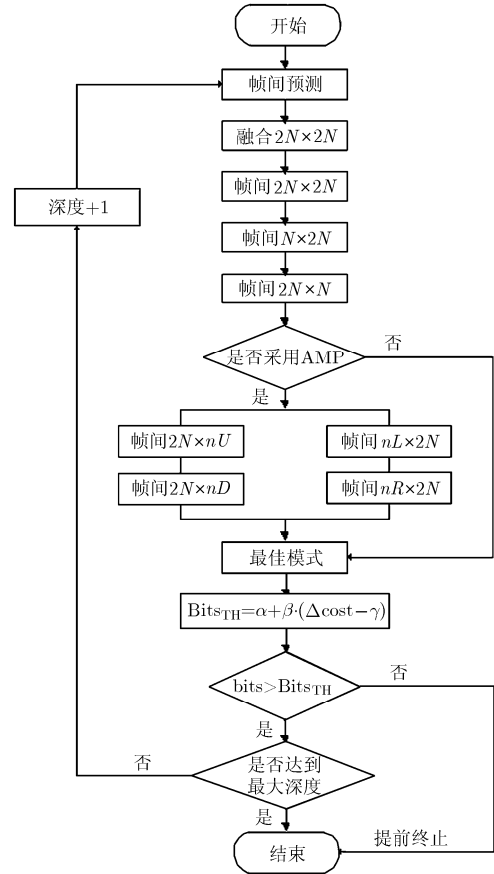


图8 CU 划分流程

阈值每变化 1 bit, 对应 cost 增量  $\gamma = 12$ 。代入式(9)中, 可以得到根据 cost 变化自适应调整的编码比特阈值:

$$\text{Bits}_{\text{TH}} = 30 + 0.4 \cdot (\Delta \text{cost} - 12) \quad (10)$$

表 2 是采用本文算法, 针对标准的 5 个不同级别的 12 个测试序列在 HM-13.0 上进行测试的实验结果, 其中配置模式使用的是 encoder\_lowdelay\_main, 量化参数 QP 分别设置为 22, 27, 32 和 37。用时间( $T$ )来衡量编码效率, 用比特率(BR)和亮度的峰值信噪比(PSNR)来衡量编码性能。在时间减少的百分比  $T$ , 比特率增加的百分比 BDBR 和亮度分量峰值信噪比的减少量 BDPSNR 3 个方面与文献[9,21]中的数据进行比对。通过表 2 中的数据可以看出, 本算法与文献[9]的算法相比较, BDBR 的增量减少了 0.8%, BDPSNR 的损失减少了 0.0274 dB, 编码所使用的时间也降低了 0.5%。与文献[21]中的算法相比较, 本文算法 BDBR 的增加了 1%, BDPSNR 减少了 0.036 dB, 编码时间减少了 25.2%的。也就是说, 本文算法可以在编码效率上有较大的提升, 而在编码性能上仅有少量的下降。

表3是配置文件分别采用encoder\_randomaccess\_

表2 本文算法与文献[9,21]算法实验结果的比对(Low-Delay)

序列	本文算法			文献[9]算法			文献[21]算法		
	$T(\%)$	BDBR(%)	BDPSNR(dB)	$T(\%)$	BDBR(%)	BDPSNR(dB)	$T(\%)$	BDBR(%)	BDPSNR(dB)
Traffic	-51.5	1.3	-0.0429	-46.3	2.5	-0.1023	-25.4	0.4	-0.0126
Kimono	-46.2	0.6	-0.0210	-47.0	1.6	-0.0612	-18.7	0.0	0.0010
BQTerrace	-49.6	1.0	-0.0203	-51.5	0.6	-0.0237	-22.2	0.1	-0.0016
BasketballDrive	-42.2	0.5	-0.0109	-51.9	3.9	-0.0851	-20.4	0.1	-0.0014
RaceHorsesC	-20.0	0.8	-0.0358	-31.8	2.6	-0.0934	-14.1	0.2	-0.0094
BQMall	-34.3	1.2	-0.0519	-47.3	3.2	-0.1074	-19.9	0.2	-0.0101
BasketballDrill	-37.5	0.8	-0.0329	-34.7	1.7	-0.0665	-19.1	0.4	-0.0153
BasketballPass	-43.8	1.6	-0.0774	-41.1	1.8	-0.0829	-19.2	0.1	-0.0071
BQSquare	-40.7	1.2	-0.0537	-24.2	1.1	-0.0513	-14.7	0.2	-0.0091
Vidyo1	-65.4	0.7	-0.0226	-58.4	1.8	-0.0550	-19.2	0.0	0.0025
Vidyo3	-60.6	2.1	-0.0661	-52.5	1.1	-0.0386	-29.2	0.0	0.0020
Vidyo4	-61.5	2.1	-0.0661	-60.6	2.1	-0.0631	-28.3	0.2	-0.0083
平均值	-46.1	1.2	-0.0418	-45.6	2.0	-0.0692	-20.9	0.2	-0.0058

表3 本算法与文献[7]算法实验结果的比对(Random-Access和Low-Delay)

类别	序列	Random-Access				Low-Delay			
		本文算法		文献[7]算法		本文算法		文献[7]算法	
		BDBR(%)	$\Delta T(\%)$	BDBR(%)	$\Delta T(\%)$	BDBR(%)	$\Delta T(\%)$	BDBR(%)	$\Delta T(\%)$
Class A	Traffic	1.5	-61.2	0.9	-59.3	1.3	-51.5	-	-
	Kimono	0.6	-41.9	0.7	-42.8	0.6	-46.2	0.8	-42.7
Class B	BQTerrace	1.3	-55.8	0.7	-58.8	1.0	-49.6	0.9	-57.6
	BasketballDrive	1.8	-48.2	0.6	-48.7	0.5	-42.2	1.0	-47.8
Class C	RaceHorsesC	2.0	-26.2	1.0	-34.6	0.8	-20.0	1.0	-36.6
	BQMall	1.1	-42.7	0.9	-52.2	1.2	-34.4	1.2	-52.2
	BasketballDrill	1.9	-42.6	0.1	-46.3	0.8	-37.5	0.7	-45.2
Class D	BasketballPass	2.2	-48.3	0.4	-51.4	1.5	-43.9	1.3	-49.6
	BQSquare	1.1	-51.5	0.5	-53.5	1.2	-40.7	1.0	-45.8
Class E	Vidyo1	0.8	-70.0	-	-	0.7	-65.4	-	-
	Vidyo3	1.1	-66.1	-	-	2.0	-60.7	-	-
	Vidyo4	0.6	-65.9	-	-	2.0	-61.6	-	-
	平均值	1.3	-51.7	0.6	-49.7	1.1	-46.1	1.0	-47.2

randomaccess\_main和encoder\_lowdelay\_main两种方式下, 本文算法与文献[7]算法的实验结果比对情况, 该测试基于 HM-13.0 平台, 使用 5 个不同级别的 12 个测试序列, 量化参数 QP 分别设置为 22, 27, 32 和 37, 用时间( $T$ )来衡量编码效率, 编码比特率的增加量(BDBR)来衡量编码性能。在 Random-Access 配置下, 本文算法时间平均减少 51.7%, BDBR 平均增加 1.3%, 本文算法可以减少更多的编

码时间, 而编码性能仅有少量的下降; 在 Low-Delay 配置下, 本算法时间平均减少 46.1%, BDBR 平均增加 1.1%, 本算法与文献[7]算法在编码效率和编码性能上基本达到了同样的效果。

表 4 和表 5 分别是序列 BasketballDrill(832×480)在 Random-Access 和 Low-Delay 两种配置方式下, 使用 HM-13.0 算法和使用本文算法时, CU 的最优划分深度分布情况。由此看出, 本文中的算法

表4 本文算法与HM-13.0最优划分深度的分布情况(Random-Access)(%)

QP	HM-13.0 算法				本文算法				两者差异			
	深度 0	深度 1	深度 2	深度 3	深度 0	深度 1	深度 2	深度 3	深度 0	深度 1	深度 2	深度 3
22	1.92	8.19	23.18	66.71	2.54	8.97	23.89	64.61	0.61	0.79	0.71	-2.11
27	3.40	10.98	28.31	57.31	4.88	12.30	29.64	53.19	1.48	1.32	1.33	-4.13
32	6.08	15.98	33.48	44.46	9.17	18.92	31.66	40.25	3.09	2.93	-1.82	-4.21
37	10.41	22.80	33.40	33.39	16.89	24.45	29.68	28.98	6.47	1.64	-3.71	-4.40

表5 本文算法与HM-13.0最优划分深度的分布情况(Low-Delay)(%)

QP	HM-13.0 算法				本文算法				两者差异			
	深度 0	深度 1	深度 2	深度 3	深度 0	深度 1	深度 2	深度 3	深度 0	深度 1	深度 2	深度 3
22	1.46	8.70	23.89	65.95	1.83	9.41	24.48	64.29	0.36	0.71	0.59	-1.67
27	2.98	10.72	29.47	56.84	3.91	11.69	32.31	52.09	0.93	0.98	2.84	-4.75
32	5.29	15.17	35.92	43.62	7.62	18.74	40.05	33.59	2.33	3.57	4.12	-10.03
37	8.91	22.24	38.17	30.67	14.21	28.66	36.97	20.16	5.30	6.42	-1.21	-10.51

主要是减少了深度较大的划分比例,增加了深度较小的划分比例,从而达到降低编码复杂度,提升编码速度的目的。两表中的数据还表明,随着量化参数 QP 的增大,最优划分为深度 0 的 CU 增加的百分比在变大,与此同时,最优划分为深度 3 的 CU,其减少的百分比也在变大。这种划分深度增减程度的变化,正是说明随着视频细节要求的降低, CU 划分时的深度可以变小,也就是说,根据视频的应用场景和自身特点,利用本文算法自适应改变编码比特阈值的步长,在编码性能和编码复杂度两者之间做出权衡的做法是合理的。

图 9 是将 QP 设置为 32 时, HM-13.0 算法和本文算法对于序列 BasketballDrill(832×480)中某一帧运动较剧烈细节部分 CU 进行划分的效果比对。图 10 是将 QP 设置为 32 时,使用 HM-13.0 算法和本文算法对于序列 BasketballDrill(832×480)中某一帧的全部 CU 进行划分的效果比对。结合图 9 和图 10 可以看出,在运动较为复杂的区域,本文算法对于细节的处理和 HM 算法基本是一致的,而对于运动较为平缓的区域,本算法可以做到使用较少的划分就可以将视频信息比较完整地表达出来,由此本算法具有更高的编码效率。

## 5 结束语

本文提出了一种快速单元划分算法:首先,根据统计数据判断被预测单元的运动特性,确定编码运动信息所需比特的阈值,将当前编码区域划分为运动剧烈区域和运动平缓区域,然后以此作为单元划分的提前终止条件阈值的初始值,再根据由此引

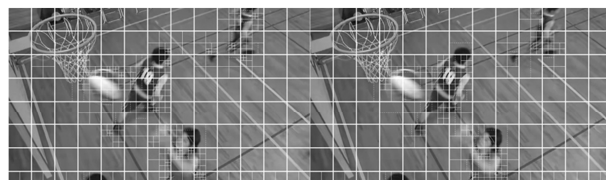


(a) HM算法



(b)本文算法

图9 BasketballDrill(832×480)中某一帧 CU 的划分情况(局部)



(a) HM算法

(b)本文算法

图10 BasketballDrill(832×480)中某一帧CU的划分情况(整帧)

起的率失真代价变化进行自适应的调整,通过调整后的阈值和相应的判定算法来减少编码单元遍历次数和率失真代价的计算量。实验结果表明,在峰值信噪比(PSNR)损失仅为 0.0418 dB 的情况下,本文算法可以平均降低 46.1% 的编码时间。

## 参考文献

- [1] SULLIVAN G J, OHM J, 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. doi: 10.1109/TCSVT.2012.2221191.
- [2] 罗零. HEVC 快速 CU 大小选择与 SKIP 模式早期判断算法[D]. [硕士论文], 西安电子科技大学, 2014.

- LUO Ji. Fast CU size selection and early decision of SKIP mode algorithms for HEVC[D]. [Master dissertation], Xidian University, 2014.
- [3] ZHANG Hao and MA Zhan. Fast intra mode decision for high efficiency video coding[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2014, 24(4): 660–668. doi: 10.1109/TCSVT.2013.2290578.
- [4] 张和仙. 下一代视频编码标准 HEVC 帧间预测优化算法研究[D]. [硕士论文], 西安电子科技大学, 2013.
- ZHANG Hexian. Research on optimization of Inter prediction in HEVC[D]. [Master dissertation], Xidian University, 2013.
- [5] ZHANG Yongfei, WANG Haibo, and LI Zhe. Fast coding unit depth decision algorithm for inter frame coding in HEVC[C]. Data Compression Conference (DCC), Snowbird, 2013: 53–62.
- [6] 齐美彬, 陈秀丽, 杨艳芳, 等. 高效率视频编码帧内预测编码单元划分快速算法[J]. 电子与信息学报, 2014, 36(7): 1699–1705. doi: 10.3724/SP.J.1146.2013.01148.
- QI Meibin, CHEN Xiuli, YANG Yanfang, *et al.* Fast coding unit splitting algorithm for high efficiency video coding Intra prediction[J]. *Journal of Electronics & Information Technology*, 2014, 36(7): 1699–1705. doi: 10.3724/SP.J.1146.2013.01148.
- [7] SHEN Liquan, ZHANG Zhaoyang, and LIU Zhi. Adaptive inter-mode decision for HEVC jointly utilizing inter-level and spatiotemporal correlations[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2014, 24(10): 1709–1722. doi: 10.1109/TCSVT.2014.2313892.
- [8] 李元, 何小海, 钟国韵, 等. 一种基于时域相关性的高性能视频编码快速帧间预测单元模式判决算法[J]. 电子与信息学报, 2013, 35(10): 2365–2370. doi: 10.3724/SP.J.1146.2013.00028.
- LI Yuan, HE Xiaohai, ZHONG Guoyun, *et al.* A fast Inter-frame prediction unit mode decision algorithm for high efficiency video coding based on temporal correlation[J]. *Journal of Electronics & Information Technology*, 2013, 35(10): 2365–2370. doi: 10.3724/SP.J.1146.2013.00028.
- [9] XIONG Jian, LI Hongliang, WU Qingbo, *et al.* A fast HEVC inter CU selection method based on pyramid motion divergence[J]. *IEEE Transactions on Multimedia*, 2014, 16(2): 559–564. doi: 10.1109/TMM.2013.2291958.
- [10] SHEN Liquan, LIU Zhi, ZHANG Xinpeng, *et al.* An effective CU size decision method for HEVC encoders[J]. *IEEE Transactions on Multimedia*, 2013, 15(2): 465–470. doi: 10.1109/TMM.2012.2231060.
- [11] KIM I K, MIN J, LEE T, *et al.* Block partitioning structure in the HEVC standard[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22(12): 1697–1706. doi: 10.1109/TCSVT.2012.2223011.
- [12] HELLE P, OUDIN S, BROSS B, *et al.* Block merging for quadtree-based partitioning in HEVC[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22(12): 1720–1731. doi: 10.1117/12.945932.
- [13] SULLIVAN G J and WIEGAND T. Rate-distortion optimization for video compression[J]. *IEEE Signal Processing Magazine*, 1998, 15(6): 74–90.
- [14] 李斌. 面向高性能视频编码标准的率失真优化技术研究[D]. [博士论文], 中国科学技术大学, 2013.
- LI Bin. Research on rate-distortion optimization techniques for HEVC standard[D]. [Ph.D. dissertation], University of Science and Technology of China, 2013.
- [15] ORTEGA A and RAMCHANDRAN K. Rate-distortion methods for image and video compression[J]. *IEEE Signal Processing Magazine*, 1998, 15(6): 23–50.
- [16] BROSS B, HAN W J, OHM J R, *et al.* High efficiency video coding (HEVC) text specification draft 10 (for FDIS & Last Call)[C]. Joint Collaborative Team on Video Coding (JCTVC) of ITU-T and ISO/IEC, JCTVC-L1003\_v34, Geneva, 2013: 1–310.
- [17] MCCANN K, BROSS B, HAN W J, *et al.* High efficiency video coding (HEVC) test model 10 (HM10) encoder descriptions[C]. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T and ISO/IEC, JCTVC-L1002\_v3, Geneva, 2013: 1–39.
- [18] CHOI K, PARK S H, and JANG E S. Coding tree pruning based CU early termination[C]. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T and ISO/IEC, JCTVC-F092, Torino, 2011: 1–4.
- [19] YANG J, KIM J, WON K, *et al.* Early SKIP detection for HEVC[C]. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T and ISO/IEC, JCTVC-G543, Geneva, 2011: 1–6.
- [20] GWEON R H, LEE Y L, and LIM J. Early termination of CU encoding to reduce HEVC complexity[C]. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T and ISO/IEC, JCTVC-F045, Torino, 2011: 1–4.
- [21] HAN S L, KIM K Y, KIM T R, *et al.* Fast encoding algorithm based on depth of coding-unit for high efficiency video coding [J]. *Optical Engineering*, 2012, 51(6): 067402–067402-11. doi: 10.1117/1.OE.51.6.067402.
- 单娜娜: 女, 1979 年生, 博士生, 研究方向为视频编码。
- 周 巍: 男, 1979 年生, 副教授, 研究方向为视频编码、集成电路分析与设计。
- 段哲民: 男, 1953 年生, 教授, 主要从事数据采集与信号处理、集成电路分析与设计、电工理论与新技术等领域的教学和科研工作。
- 魏恒璐: 男, 1990 年生, 博士生, 研究方向为视频编码。