高效率视频编码帧内预测编码单元划分快速算法

齐美彬[①③] 陈秀丽[*①] 杨艳芳[②] 蒋建国[①③] 金玉龙[①] 张俊杰[①]
[①]合肥工业大学计算机与信息学院 合肥 230009)
[②](合肥工业大学电子科学与应用物理学院 合肥 230009)
[③](教育部安全关键工业测控技术工程研究中心 合肥 230009)

摘 要：高效率视频编码(HEVC)采用基于编码单元(CU)的四叉树块分区结构，能灵活地适应各种图像的纹理特征，显著提高编码效率，但编码复杂度大大增加，该文提出一种缩小深度范围且提前终止 CU 划分的快速 CU 划分算法。首先，在学习帧中，基于 Sobel 边缘检测算子计算一帧中各深度边缘点阈值，缩小后面若干帧中 CU 遍历的深度范围；同时，统计该帧中各 CU 划分为各深度的率失真(RD)代价，计算各深度的 RD 代价阈值。然后，在后续视频帧中，利用 RD 代价阈值在缩小的深度范围内提前终止 CU 划分。为了符合视频内容的变化特性，统计参数是周期性更新的。经测试，在平均比特率增加仅1.2%时，算法时间平均减少约59%，有效提高了编码效率。

关键词：高效率视频编码(HEVC)；帧内预测；Sobel 算子；率失真代价
中图分类号：TN919.8 **文献标识码**：A **文章编号**：1009-5896(2014)07-1699-07
DOI: 10.3724/SP.J.1146.2013.01148

Fast Coding Unit Splitting Algorithm for High Efficiency Video Coding Intra Prediction

Qi Mei-bin[①③] Chen Xiu-li[①] Yang Yan-fang[②]
Jiang Jian-guo[①③] Jin Yu-long[①] Zhang Jun-jie[①]
[①](*School of Computer and Information*, *Hefei University of Technology*, *Hefei* 230009, *China*)
[②](*School of Electronic Science & Applied Physics*, *Hefei University of Technology*, *Hefei* 230009, *China*)
[③](*Engineering Research Center of Safety Critical Industrial Measurement and Control Technology*, *Ministry of Education*, *Hefei* 230009, *China*)

Abstract: High Efficiency Video Coding (HEVC) adopts a quadtree-based Coding Unit (CU) block partitioning structure which is flexible to adapt to various texture characteristics of images and can improve the coding efficiency significantly; however, it also introduces a great computation complexity. This paper proposes a fast CU splitting algorithm which can narrow CU depth range and early terminate the CU splitting. Firstly, in learning frame, based on the Sobel edge detection operator, this study obtains edge point threshold at each CU depth level to narrow the CU traversal depth range in the next several frames. Meanwhile, this paper introduces the statistical of Rate-Distortion (RD) cost of CU in the frame and calculates RD threshold of each depth level. Then, in the subsequent video frames, RD threshold is utilized to terminate early the CU splitting in the narrow depth range. The statistical parameters are periodically updated to cope with varying video content characteristics. The experimental results show that the proposed algorithm is able to save 59% encoding time on average with only 1.2% increasing on bit-rate which can observably improve the coding efficiency.

Key words: High Efficiency Video Coding (HEVC); Intra prediction; Sobel operator; Rate-Distortion (RD) cost

1 引言

高效率视频编码(High Efficiency Video Coding, HEVC)是目前最新的视频编码标准[1]，其编码效率大大优于以前的标准。HEVC 实现高编码效率的一个重要方法是采用灵活的四叉树结构来组织编码单元(Coding Unit, CU)、预测单元(Prediction Unit, PU)和变换单元(Transform Unit, TU)，但计算 3 个单元的每一种组合的率失真(RD)代价给编码器带来了高计算复杂度[2]。

CU 是帧间/帧内预测区域划分的基本单元，每个 CU 由一个亮度块和两个相应的色度块及相关语法元素组成。就亮度而言，CU 大小范围为 64×64 (Largest Coding Unit, LCU) 到 8×8(Smallest

2013-07-31 收到，2014-03-07 改回
安徽省科技攻关计划项目(1301b042023)资助课题
*通信作者：陈秀丽 xiulichen0927@163.com

Coding Unit, SCU)，深度级范围是0到3，每级划分成4个大小相等的块[2]。

HEVC 测试模型(HEVC test Model, HM)的 CU 决策过程，使用所有可能的 CU 大小和预测模式，并依据拉格朗日乘数找到最低的率失真代价[3]。然而，这种遍历方法将导致较高的计算复杂度，限制了 HEVC 编码器的实际应用。因此，有必要研究快速算法,在尽量不影响编码性能的前提下降低 CU 划分的计算复杂度[4,5]。

现有的帧内预测快速算法一般基于图像的纹理特性或空间相关性。文献[6]提出一个两阶段的 PU 大小决策快速算法，第1阶段根据视频内容分析 LCU 和它的4个子块经下采样后的纹理复杂度，滤除 LCU 和它的子块不需要的 PU。第2阶段，在帧内编码过程中利用已编码的相邻块的 PU 跳过当前块小的 PU 侯选者。文献[7]根据贝叶斯估计方法，统计各深度 CU 不划分的低复杂度 RD 代价和划分的 RD 代价，实现各深度 CU 的早划分和早裁剪。文献[8]将 CU 划分和 PU 预测优化结合起来提出一种 CU 划分方法，该方法利用当前块的左、左上、上和右上边的块判断当前块的深度范围。

对于帧间预测，文献[9]提出一个CU提早终止方法，如果当前CU节点在当前CU深度选择SKIP模式作为最佳预测模式，则跳过下面子CUs的编码过程。文献[10]在文献[9]的基础上应用早期跳过(skip)模式检测取得进一步加速。文献[11]阐述了非跳过(non-skip)模式下，利用CBF(Coded Block Flag)来实现CU的提前终止。文献[9–11]利用SKIP模式或帧间预测的运动数据信息寻找CU提前终止条件，终止CU进一步的划分。文献[12]基于图像的时间和空间相关性先判定CU的深度范围，再根据帧间预测特性判断CU划分的深度，跳过一些CU级上不必要的运动估计。

在以上帧内、帧间快速算法中，文献[6]和文献[8]是利用图像的纹理特性或时间、空间相关性进行预测，跳过一些深度级上 CU 的预测，文献[7]和文献[9–11]则基于统计的思想获得阈值判断 CU 划分的深度级。它们或者只考虑图像本身的特性，或者只统计划分过程的信息，而本文受文献[12]的启发，将图像的纹理特性和 HM 模型中 CU 划分的统计特性相结合，提出一种实现 CU 最佳划分的快速算法。

本文将视频帧分为学习帧和应用帧。在学习帧中，首先基于 Sobel 边缘检测算子计算一帧中各深度级 CU 的边缘点数，建立深度级与边缘点数之间的对应关系，缩小后继帧中 CU 遍历的深度范围；同时，统计该帧中各 CU 划分为各深度的 RD 代价，计算各深度的 RD 代价阈值。在应用帧中，利用在学习帧中得到的 RD 代价阈值，在缩小的深度范围内提前终止 CU 划分。

2 HM 中 CU 划分过程

前面提到，HM 中的 CU 决策过程，是遍历所有的 CU 划分深度，选择率失真 RD 代价最小的模式作为最佳的划分方式。每个预测模式的 RD 代价定义为

$$J_{\text{mode}} = (\text{SSE}_{\text{luma}} + w_{\text{chroma}} \cdot \text{SSE}_{\text{chroma}}) + \lambda_{\text{mode}} \cdot B_{\text{mode}} \tag{1}$$

SSE_{luma}, $\text{SSE}_{\text{chroma}}$ 分别为亮度和色度信号的原始输入图像块和预测块之间的误差平方和，λ_{mode} 是拉格朗日乘数，B_{mode} 是对相应 CU 块的候选模式进行编码的总的比特数。对于帧内预测，w_{chroma} 值为0.57[3]。

HM 中，基于 RD 代价确定最佳 CU 划分的处理顺序如图1所示。CU 的递归划分是按深度优先，自上而下的方式进行的。先计算深度为0的 CU 即 LCU 的代价，然后计算其 Z 扫描顺序下第1个深度为1的子 CU 的 RD 代价,接着是该深度为1的子 CU 在 Z 扫描顺序下第1个深度为2的子 CU 的 RD 代价，这样一直按深度增加往下进行。对深度为3的 CU，会比较 PART_2N×2N 和 PART_N×N 两种模式的 RD 代价，选取 RD 代价较小的预测方式。达到最大深度以后，如图1中深度3向上箭头指示，将4个深度为3的 SCU 的 RD 代价总和与对应深度为2的 CU 的 RD 代价进行比较，选取代价较小的，接着按图中处理序号逐级计算再逐级向上进行比较，遍历整个 LCU 所有的划分深度，确定最佳即 RD 代价最小的划分方式。图1是一个 LCU 的处理顺序，只具体写出了一个32×32的 CU 的划分顺序，另外3个是一致的。

由图1可以看到，整个 CU 划分过程是在4个深度级范围内逐级比较 RD 代价的，并选择 RD 代价

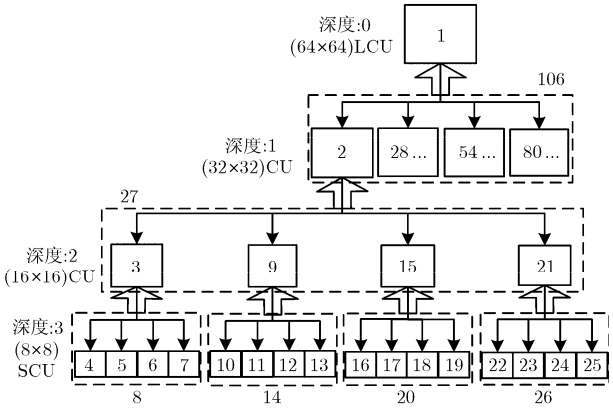


图1 一个 LCU 划分时具体处理顺序

最小的作为最佳划分结果。这种遍历方法，需进行106次的 RD 代价相关计算和比较，虽然可以获得更精确的图像划分结果，但也给 HEVC 编码器带来了巨大的复杂度，限制了 HEVC 的实际应用，因此本文提出了基于 CU 划分的快速算法。

3　基于 Sobel 边缘检测算子的自适应深度范围选择

在 HM 模型中，CU 深度级范围为0到3，一般小的深度级适用于平滑的图像区域，大的深度级适用于运动剧烈或纹理丰富的区域。可以用图像边缘点的多少来估计 CU 的纹理复杂程度，并根据纹理复杂程度来预测 CU 的深度范围，从而跳过某些深度级上 CU 的预测和相关 RD 代价计算，节省编码时间。本文采用 Sobel 边缘检测算子来获得各 CU 的边缘点数目，用边缘点数目表示 CU 的复杂程度。

3.1 利用 Sobel 算子获得 LCU 边缘点数

将 Sobel 算子与图像作卷积，分别计算出横向、纵向梯度 $\boldsymbol{G}_x$ 和 $\boldsymbol{G}_y$，用 $\boldsymbol{A}$ 代表原始图像，如式(2)所示。

$$\boldsymbol{G}_x=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}*\boldsymbol{A},\quad \boldsymbol{G}_y=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}*\boldsymbol{A} \tag{2}$$

再按式(3)计算各点的梯度值，得到梯度图。

$$\boldsymbol{G}=\sqrt{\boldsymbol{G}_x^2+\boldsymbol{G}_y^2} \tag{3}$$

得到梯度图后，再用最大类间方差(otsu)法对梯度图进行二值化分割，梯度值大于阈值的像素点作为边缘点。统计一个 LCU 中的边缘点个数，边缘点数目越多则认为该块越复杂，划分的深度级则越大。otsu 方法得到的是全局最优的分割阈值，而 CU 划分要能反映图像的局部纹理特性，若使用全局最优的阈值分割可能导致某些局部边缘点丢失。为防止边缘点丢失，将一帧图像按 LCU 扫描顺序划分为大致相等的4块，如图2所示。用 otsu 算法分别计算每一块的分割阈值 $\mathrm{Gfour}_i, i=0,1,\cdots,3$。各 LCU 利用本块的分割阈值进行二值化分割，得到各 LCU 内的边缘点数。

3.2 基于边缘点数目的 CU 深度范围预测

在学习帧中获得各 LCU 的边缘点数的同时，还利用 HM 模型得到各 LCU 的划分深度级，从而得到各个深度级对应的边缘点数范围。实验中发现，深度级为 0 和深度级为 1 的 CU 的边缘点数很接近，因此将 HM 中最终划分为深度 0 和 1 的各 LCU 边缘点数放在一起求出边缘点数的均值和标准差，并计算两者之和得到阈值 $\mathrm{TH\_Edge}_1$。同时，由深度为 2 的各 LCU 边缘点数获得阈值 $\mathrm{TH\_Edge}_2$。在后续应用帧中，若当前 LCU 边缘点数小于等于 $\mathrm{TH\_Edge}_1$，说明其纹理较简单，可划分为深度级 0 或深度级 1；若当前 LCU 边缘点数大于 $\mathrm{TH\_Edge}_1$ 且小于等于 $\mathrm{TH\_Edge}_2$，则该 LCU 可以划分为深度级 2；若当前 LCU 边缘点数大于 $\mathrm{TH\_Edge}_2$，则该 LCU 可以划分为深度级 3。

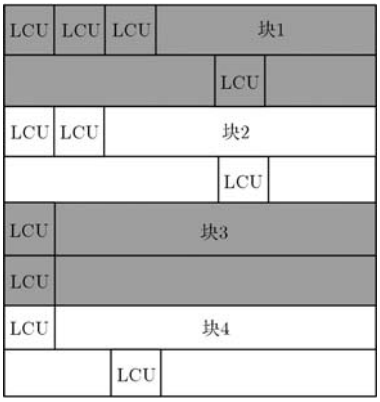


图2 一帧图像划分为4块

另外，边缘点数并不能精确估计深度级，只能预测深度级的范围，因此修正边缘点数与深度级范围之间的对应关系，具体如表 1 所示。

表1 边缘点数目与深度级范围的对应关系

LCU 中边缘点数目		深度级范围
$\le \mathrm{TH\_Edge}_1$	/	0 1 2
$> \mathrm{TH\_Edge}_1$	$\le \mathrm{TH\_Edge}_2$	1 2 3
	$>\mathrm{TH\_Edge}_2$	2 3

在学习帧中统计 LCU 边缘点数目和最佳划分深度级，获得边缘点数目与深度级范围之间的对应关系(即表1所示关系)。在学习帧之后的应用帧中，计算各 LCU 的边缘点数目后，根据表1来确定 LCU 所属的深度范围。假设序列帧率为30帧/s，可以将第1帧设为学习帧，其余29帧为应用帧，并一直交替下去。

4　基于 RD 代价的 CU 划分提前终止

HM 的 CU 划分过程，是在4个深度级范围内逐级比较 RD 代价，并选择 RD 代价最小的划分作为最佳划分。本文在由 LCU 边缘点确定的2个或3个深度级范围内进行 CU 划分。为进一步加快 CU 划分速度，采用基于 RD 代价的 CU 划分提前终止策略。

一般情况下，若 CU 当前深度的 RD 代价越小，则说明当前划分方式的效果越好，该 CU 继续划分的可能性越小。也就是说，若 CU 当前深度 RD 代

价小于某个阈值，就无需继续划分，这就是本文的“CU 划分提前终止策略”。

在学习帧中，除了得到边缘点数目与深度级范围的对应关系外，还利用 HM 得到各 LCU 的划分深度及对应的 RD 代价，并计算每个深度级所对应的 RD 代价的均值及标准差，再按照式(4)来计算用于提前终止的 RD 代价阈值。在此后的应用帧中，从最小深度级开始计算当前深度级的 RD 代价，若当前深度级 CU 的 RD 代价小于对应深度级的 RD 代价阈值，则该 CU 不继续划分，提前终止，否则继续按深度增加逐级比较。这样，就大大减小了 RD 代价的计算量。

$$\mathrm{Th\_RD}_k = \mathrm{Mean}_k + \lambda^* \mathrm{Devi}_k \tag{4}$$

式(4)中，λ用于调整阈值的大小，一般取$0 \le \lambda \le 1$，随着λ的增加，序列编码时间效率会有较大增加，码率增加较小，而信噪比几乎不变；k表示深度级，$0 \le k \le 3$；Mean_k，Devi_k分别为深度级k所对应的 RD 代价的均值及标准差。$\mathrm{Th\_RD}_0$，$\mathrm{Th\_RD}_1$，$\mathrm{Th_RD}_2$用于对应深度 CU 的提前终止，而$\mathrm{Th_RD}_3$用于 PU 的提前终止，即当 CU 划分到深度为3时，在 PU 为 Intra_2N×2N 模式下，若当前 RD 代价小于获得的深度3的阈值$\mathrm{Th_RD}_3$，则不执行深度3时 PU 为 Intra_N×N 模式的相关 RD 代价计算。

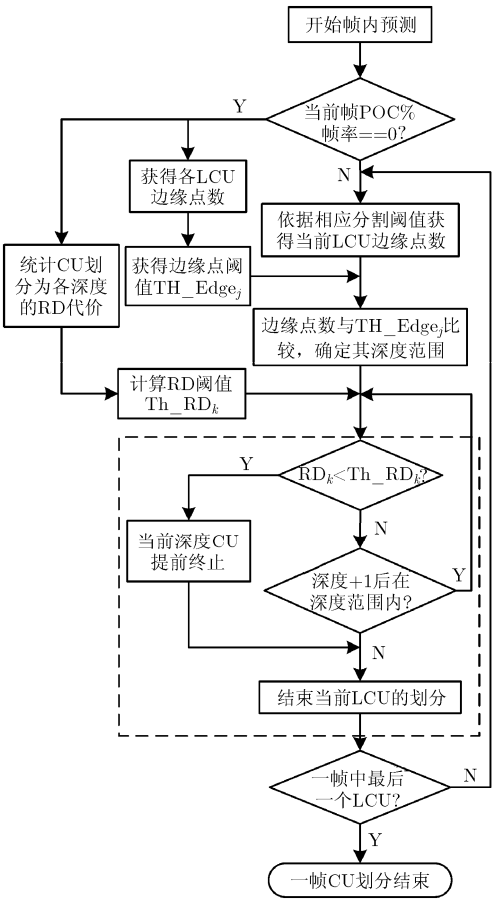


图 3 程序整体框图

5 CU 快速划分算法流程

对于整个算法流程，首先是学习帧基于梯度获得一帧图像中各深度的边缘点阈值，在统计边缘点的同时也统计一帧中各LCU最终划分时各子CU的 RD 代价，并对统计得到的各个深度级 RD 代价求均值和标准差，获得用于判断的 RD 代价阈值。然后在应用帧中，先依据边缘点阈值获得当前 LCU 需遍历的深度范围，再在该深度范围内，比较当前深度 CU 的 RD 代价和 RD 代价阈值的大小，若小于 RD 代价阈值则 CU 提前终止。图3是程序的整体流程图，具体步骤为：

(1)首先是学习帧，对每个 LCU 依据 Sobel 边缘检测算子计算各像素点的梯度值，获得梯度图后，将梯度图按 LCU 扫描顺序划分为4块；

(2)利用 otsu 算法计算每块的分割阈值Gfour_i，$i = 0,1,\cdots,3$(取计算得到值的0.75)，并依据该阈值判断该块内各 LCU 的边缘点数；

(3)依据 HM 中该帧图像各 LCU 最终划分深度将 LCU 的边缘点数分组(深度 0 和 1 并为一组)，分别计算各深度边缘点数的均值和标准差，剔除边缘点数目落在标准差之外的 CU，重新计算，将再次计算的均值和标准差之和作为对应深度级的阈值$\mathrm{Th_Edge}_j, j = 1,2$；

(4)在统计各 LCU 梯度的同时，统计该帧中各 CU 最终划分时的 RD 代价，并计算各深度级的 RD 代价均值和标准差，获得 RD 代价阈值，阈值计算公式为式(4) ；

(5)学习帧结束后，进入应用帧，根据步骤(3)中得到的阈值判断当前 LCU 的边缘点数属于哪个划分深度，获得遍历的深度范围，然后将递归过程中当前深度 CU 的 RD 代价RD_k与对应深度的阈值$\mathrm{Th_RD}_k, k = 0,1,\cdots,3$进行比较，若小于阈值，则终止当前 CU 划分，CU 最终划分深度为k。否则，深度加1，继续比较下一深度的 RD 代价和相应阈值，直至最大深度。然后进入下一个 LCU 的比较，再次执行步骤(5)。若执行到学习帧，则返回到步骤(1)。

6 实验结果

本文快速算法是在 HM10.0[13]上进行测试的，编码条件是全帧内(All Intra, AI)配置模式。测试时使用 JCT-VC 提供的标准 yuv 测试序列，量化参数 QP 值使用22, 27, 32和37。编码性能使用比特率(Bit Rate, BR)和峰值信噪比(PSNR)度量，计算复杂度

使用消耗的编码时间来度量。现用 ΔBR(%)，$\Delta\mathrm{PSNR}_{\mathrm{Y}}$(dB) 分别表示平均比特率和亮度分量的 PSNR 损失，ΔET(%) 表示编码时间减少百分比，计算公式如式(5)，式(6)和式(7)所示。

$$\Delta\mathrm{BR}=\frac{\mathrm{BR}_{\mathrm{prop}}-\mathrm{BR}_{\mathrm{orig}}}{\mathrm{BR}_{\mathrm{orig}}}\times 100\% \tag{5}$$

$$\Delta\mathrm{PSNR}_{\mathrm{Y}}=\mathrm{PSNR}_{\mathrm{prop}}-\mathrm{PSNR}_{\mathrm{orig}} \tag{6}$$

$$\Delta\mathrm{ET}=\frac{T_{\mathrm{orig}}-T_{\mathrm{prop}}}{T_{\mathrm{orig}}}\times 100\% \tag{7}$$

其中 $\mathrm{BR}_{\mathrm{prop}}$, $\mathrm{PSNR}_{\mathrm{prop}}$ 和 T_{prop} 是本文提出的快速算法的比特率、亮度分量的 PSNR 和总编码时间；$\mathrm{BR}_{\mathrm{orig}}$, $\mathrm{PSNR}_{\mathrm{orig}}$ 和 T_{orig} 是原有的 HM 代码的比特率、亮度分量的 PSNR 和编码时间。

文献[7]方法是基于 RD 代价统计的，有效性较好，且假设各深度 RD 代价数据符合高斯分布，而本文统计 RD 代价均值和标准差的思想也是基于高斯分布假设的，因此将本文方法的结果与文献[7]方法的结果进行了对比。表2中列出了本文算法在$\lambda=0.5$时结果，以及文献[7]算法在$\alpha=0.3$时的结果(表2中的数据是 QP 分别取22, 27, 32和37时得到的均值；本文方法的数据是在 HM10.0上得到的，而文献[7]方法的数据是该文作者在 HM6.0上得到的。由于 HM10.0和 HM6.0的 CU 划分算法完全一致，因此直接使用了原文作者的实验数据)。

从表2可知，与 HM10.0相比，式(4)中$\lambda=0.5$时本文提出的算法效率最高可达67.67%，平均为58.94%，同时带来的平均比特率增加为1.16%, PSNR 损失为0.06 dB，算法几乎没有影响视频质量，综合效果要优于文献[7]算法。另外，表3中还列出了本文算法在$\lambda=0.0$和$\lambda=1.0$的结果，以及文献[7]在$\alpha=0.1$和$\alpha=0.5$的结果。

对于高清视频，本文提出的算法效率更高。表4中是本文算法和文献[7]算法运行 A, B 和 E 类视频得到的各类指标的平均值。可以看到，本文算法在$\lambda=0.5$时，比特率与文献[7]在$\alpha=0.3$时基本一致，但时间效率提高约7%。

需要说明的是，表3和表4中λ与α的取值之间并没有一一对应性，只是说明：随着λ值的增加，本文算法编码时间效率逐渐增加，但比特率也略有增加，而文献[7]算法的结果随α增加与本文有类似趋势(α为显著性水平，具体见文献[7])。

图4是在 QP=22时，序列 KristenAndSara 分别在 HM 算法和本文提出算法中的 CU 划分结果，可以看出本文提出的方法与 HM 的划分基本一致，且对于平坦区域的划分效果更好。

图5是本文提出的算法在$\lambda=0.5$时与 HM 算法的率失真性能比较，由图5(a)可以看出两算法在序列 BasketballDrive 和 KristenAndSara 的 RD 曲线

表2 本文算法在 AI 配置下的性能

类别	序列	帧数	本文方法($\lambda=0.5$)			文献[7]方法($\alpha=0.3$)		
			ΔBR(%)	$\Delta\mathrm{PSNR}_{\mathrm{Y}}$(dB)	ΔET(%)	ΔBR(%)	$\Delta\mathrm{PSNR}_{\mathrm{Y}}$(dB)	ΔET(%)
A(2560×1600)	PeopleOnStreet	150	1.03	−0.06	60.43	0.95	−0.05	44.51
	Traffic	150	1.53	−0.06	62.00	1.24	−0.04	54.03
B(1920×1080)	BasketballDrive	500	0.99	−0.04	65.12	1.99	−0.03	61.09
	BQTerrace	600	0.58	−0.05	59.96	0.96	−0.02	60.24
	Cactus	500	1.38	−0.04	61.35	1.07	−0.03	47.96
	Kimono	240	3.50	−0.06	66.24	1.18	−0.04	64.80
	ParkScene	240	1.13	−0.05	60.67	0.93	−0.02	47.58
C(832×480)	BasketballDrill	500	0.50	−0.08	58.88	0.60	−0.08	52.75
	BQMall	600	1.79	−0.08	57.61	1.01	−0.05	50.21
	PartyScene	500	0.54	−0.09	49.07	0.26	−0.02	46.28
	RaceHorsesC	300	2.14	−0.08	59.78	1.44	−0.01	64.59
D(416×240)	BasketballPass	500	0.43	−0.04	56.53	0.85	−0.06	50.00
	BlowingBubbles	500	0.38	−0.10	47.97	0.19	−0.02	49.51
	BQSquare	600	0.13	−0.07	44.73	0.36	−0.04	53.62
	RaceHorses	300	0.97	−0.08	53.41	1.34	−0.04	54.55
E(1280×720)	Johnny	600	1.78	−0.04	67.67	2.33	−0.08	65.24
	KristenAndSara	600	1.21	−0.04	65.95	2.74	−0.06	70.57
	FourPeople	600	0.94	−0.06	63.47	0.90	−0.08	50.65
平均值	/	/	1.16	−0.06	58.94	1.13	−0.04	54.90

表3 算法其他参数设置下的平均性能

方法	参数	ΔBR(%)	$\Delta\text{PSNR}_\text{Y}$(dB)	ΔET(%)
本文方法	λ=0.0	0.81	−0.03	51.02
	λ=1.0	1.68	−0.08	63.05
文献[7]方法	α=0.1	0.47	−0.01	45.69
	α=0.5	2.23	−0.08	62.43

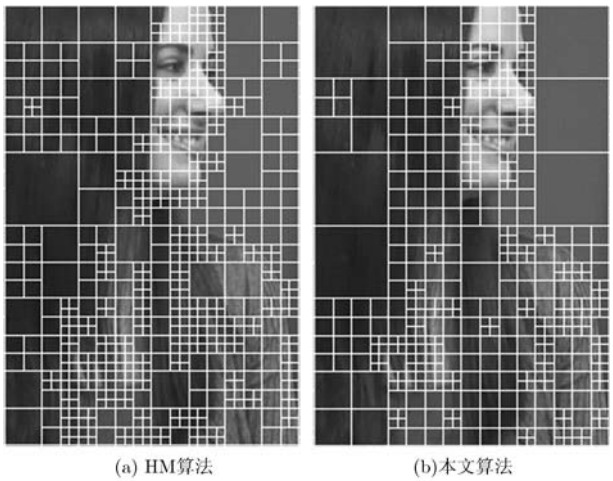
表4 算法在高清视频编码时的性能

类别		ΔBR(%)	$\Delta\text{PSNR}_\text{Y}$(dB)	ΔET(%)
本文方法	λ=0.0	1.00	−0.03	54.93
	λ=0.5	1.41	−0.05	63.29
	λ=1.0	1.83	−0.08	68.76
文献[7]方法	α=0.1	0.56	−0.01	45.29
	α=0.3	1.43	−0.05	56.67
	α=0.5	2.89	−0.08	66.66

是几乎重合的，说明使用本文算法几乎没有性能损失。对于比特率增加较多的序列BQMall和Kimono，如图5(b)所示，其RD曲线和HM的有稍许偏离，但很小，不会对编码图像的质量有很大影响。

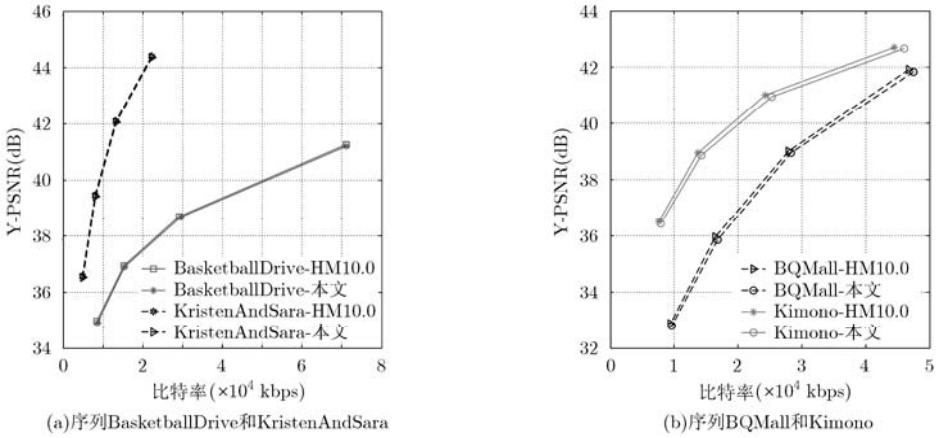
7 结束语

本文中提出了基于 Sobel 算子的深度范围检测和 RD 代价统计相结合的快速算法，将纹理特征和 RD 代价统计结合起来，提高了 CU 划分速度。首先是基于 Sobel 算子计算学习帧各深度级的边缘点阈值，利用该阈值对后续应用帧中 LCU 设定遍历的深度级范围。在计算边缘点数的同时，统计该帧各 LCU 在 HM 中最终划分的 RD 代价，分别计算各个深度的 RD 代价阈值。然后，在边缘点获得的深度范围内，将应用帧各 CU 当前深度 RD 代价和对应深度 RD 代价阈值比较，提前终止 CU 划分。本文算法在 PSNR 几乎不变、平均比特率增加为1.2%的情况下，编码时间平均减少约59%。



(a) HM算法 (b)本文算法

图4 序列 KristenAndSara 在 QP=22时的 CU 划分结果



(a)序列BasketballDrive和KristenAndSara (b)序列BQMall和Kimono

图5 本文算法与 HM10.0的率失真性能比较结果

参 考 文 献

[1] 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 (JCT-VC) of ITU-T and ISO/IEC, JCTVC-L1003_v34, Geneva, 2013.

[2] 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.

[3] McCann K, Bross B, Han W J, *et al*.. High efficiency video coding (HEVC) test model 10 (HM10) encoder description[C]. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T and ISO/IEC, JCTVC-L1002_v3, Geneva, 2013.

[4] Ohm J R, Sullivan G J, Schwarz H, *et al*.. Comparison of the coding efficiency of video coding standards - including high efficiency video coding (HEVC)[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22(12):

1669–1684.

[5] Bossen F, Bross B, Suhring K, *et al*.. HEVC complexity and implementation analysis[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22(12): 1685–1696.

[6] Tian G and Goto S. Content adaptive prediction unit size decision algorithm for HEVC intra coding[C]. Picture Coding Symposium(PCS), Krakow, Poland, 2012: 405–408.

[7] Cho S and 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.

[8] Shen Li-quan, Zhang Zhao-yang, and 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.

[9] 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, Italy, 2011: 1–4.

[10] 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, Switzerland, 2011: 1–6.

[11] Gweon R H, Lee Y L, 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, Italy, 2011: 1–4.

[12] Shen Li-quan, Liu Zhi, Zhang Xin-peng, *et al*.. An effective CU size decision method for HEVC encoders[J]. *IEEE Transactions on Multimedia*, 2013, 15(2): 465–470.

[13] JCT-VC of ISO/IEC and ITU-T. HM Reference Software10.0 [OL]. https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-10.0/, 2013.

齐美彬：男，1969 年生，教授，研究方向为视频编码、运动目标检测与跟踪、DSP 技术.

陈秀丽：女，1987 年生，硕士生，研究方向为视频编码.

杨艳芳：女，1970 年生，副教授，研究方向为数学物理方法.

蒋建国：男，1955 年生，教授，研究方向为数字图像分析与处理、分布式智能系统和 DSP 技术及应用.

金玉龙：男，1989 年生，硕士生，研究方向为目标跟踪.

张俊杰：男，1989 年生，硕士生，研究方向为多目标跟踪.