

基于纹理特征的 HEVC 帧内编码快速算法^①

汤进, 彭勇

(江南大学 物联网工程学院, 无锡 214122)

通讯作者: 汤进, E-mail: 786765884@qq.com

摘要: 为了降低高效视频编码 (HEVC) 帧内编码的复杂度, 提出了一种基于编码块纹理特征的帧内编码快速算法. 首先, 对当前编码单元 (CU) 进行预处理, 获取 CU 的纹理复杂程度和方向特性. 其次, 根据纹理复杂度决定部分 CU 是否划分, 跳过率失真代价计算, 减少不必要的划分与裁剪. 然后, 根据纹理方向减少帧内预测模式的数量, 降低帧内预测过程的复杂度. 实验结果表明, 在全 I 帧模式下快速算法与 HM10.0 相比, 码率 (BR) 上升 0.649%, 峰值信噪比 (PSNR) 降低 0.059 dB, 帧内编码时间平均减少 56.18%.

关键词: 高效视频编码; 纹理特征; 编码单元划分; 预测模式选择

引用格式: 汤进, 彭勇. 基于纹理特征的 HEVC 帧内编码快速算法. 计算机系统应用, 2018, 27(7): 150-155. <http://www.c-s-a.org.cn/1003-3254/6454.html>

Fast Intra Coding Algorithm for HEVC Based on Texture Property

TANG Jin, PENG Yong

(School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China)

Abstract: In order to reduce the complexity of intra encoding in High Efficiency Video Coding (HEVC), a fast algorithm for intra encoding based on coded block texture features is proposed. Firstly, the texture complexity and direction characteristics are extracted from the current Coding Unit (CU) by using a preprocessing method. Secondly, according to the texture complexity, the algorithm can determine whether to divide some CU, skip the complex rate of distortion calculation, and cut down the CU unnecessary division and cutting. Then, according to the texture direction, the algorithm can decrease the number of intra prediction modes, which can reduce the computational complexity of intra prediction. On the basis of the experimental results, compared with HM10.0, the proposed algorithm can save 56.18% coding time on average, with the Bit Rate (BR) increases of 0.649%, and the Peak Signal-to-Noise Ratio (PSNR) losses lower 0.059 dB, respectively.

Key words: High Efficiency Video Coding (HEVC); texture feature; Coding Unit (CU) partitioning; prediction mode decision

由于视频信息具有真实、具体、生动、全面、信息量大等特点, 人们对各种视频信息的需求与日俱增, 使得大数据中海量视频信息的占比已过半, 因此, 保证海量视频信息的高效、优质、标准化的压缩日显重要^[1]. H.264/AVC^[2]视频编码标准逐渐不能满足应用需

求, 所以 H.265/HEVC^[3]视频编码标准应运而生.

HEVC 视频编码性能的高效, 是以更高的算法复杂度换取的. 相对于 H.264, 它在很多地方有了革命性的变化. HEVC 放弃了之前编码标准中使用的宏块, 采用了具有二叉树递归划分的编码单元 (CU)、预测单

① 基金项目: 江苏省产学研联合创新资金 (BY2013015-35)

Foundation item: Joint Innovation Foundation of University and Research Institute of Jiangsu Province (BY2013015-35)

收稿时间: 2017-11-30; 修改时间: 2017-12-21; 采用时间: 2017-12-27; csa 在线出版时间: 2018-06-27

元 (PU) 和变换单元 (TU). 这让编码更符合视频图像细节的特性. 但同时也显著增加了计算复杂度^[4]. 所以在不影响 HEVC 编码效果的前提下, 有效降低编码复杂度是目前研究的热点.

帧内编码是视频编码的一个重要组成部分. HEVC 中 CU、PU 和 TU 的四叉树递归划分和 35 种帧内预测模式, 导致了编码复杂度极高. 相关学者提出了很多帧内快速编码算法, 主要可以分为两大类: 1) 根据当前 CU 周围相邻已编码重建单元的编码结果, 对当前 CU 的最佳划分方式进行估计, 跳过一些可能性较小的结构; 2) 根据视频图像本身的纹理特性, 通过一些预处理方式, 实现各深度下 CU 的早划分或早裁剪或减少 PU 预测模式数量. Zhang 等^[5]通过分析空间上相邻的已编码完成的 CU 和时间上前一帧相同位置的 CU 与当前 CU 的关系, 对当前 CU 进行快速分割; Zhao 等^[6]根据相邻 CU 的结构相似性, 当前 CU 的划分结构和预测模式直接利用相邻相似度高的已编码 CU 的结果; 文献^[7,8]用当前 CU 内像素方差值表示 CU 复杂程度, 根据复杂程度决定 CU 分割; 文献^[9,10]利用 Sobel 算子对当前 CU 进行边缘检测, 进而分析纹理复杂程度, 根据纹理复杂程度决定 CU 是否划分. 蔡婷婷等^[11]根据 CU 在水平、垂直、45°和 135°方向上的全局边缘复杂度来判断 CU 是否划分, 并通过比较水平和垂直方向的边缘复杂度大小减少预测模式数量, 但全局边缘复杂度计算比较复杂, 且预测模式数量减少较少.

目前多数算法是单一的从 CU 的尺寸选择或 PU 预测模式的快速决策入手. 通过分析 HEVC 帧内编码过程, 本文提出一种基于编码块纹理特征的 CU 尺寸提早划分, PU 预测模式快速决策算法. 通过一种预处理算法获取当前 CU 的纹理复杂度和方向性. 根据纹理复杂度决定部分 CU 的划分方式, 根据纹理方向性减少 PU 预测模式的数量. 从 CU 和 PU 两方面来降低帧内编码的复杂度.

1 HEVC 帧内编码复杂度分析

在 HEVC 测试软件 HM 中, 一帧图像首先被分割成多个最大编码单元 (LCU), 对每个 LCU 以光栅扫描的顺序进行编码. 单个 LCU 可以进一步划分成多个 CU. 深度 (Depth) 为 0, 1, 2, 3 分别对应 CU 的尺寸为 64×64, 32×32, 16×16, 8×8. 当深度为 0, 1, 2 时, PU 尺寸与 CU 相同, 当深度为 3 时, PU 尺寸可以是 8×8, 也

可以 4×4. 因此, 在 CU 与 PU 划分组合的过程中, 需要进行 $4^0 + 4^1 + 4^2 + 4^3 + 4^4 = 341$ 次递归划分. 递归划分过程如图 1 所示, LCU 先自上而下分割成 4 个尺寸为 32×32 的子 CU (sub-CU), 每个 sub-CU 还可以继续分割成 4 个 sub-CU, 直到深度为 3 为止. 然后自下而上进行 CU 裁剪. 对每个 sub-CU 分别进行 35 种帧内预测模式搜索, 找到率失真代价最小的最佳预测模式, 并将 4 个 sub-CU 的率失真代价和与其父 CU 的率失真代价进行比较决定父 CU 是否划分. 若父 CU 的率失真代价小于 4 个 sub-CU 的率失真代价和, 则这个父 CU 不分割成 4 个 sub-CU, 反之则分割.

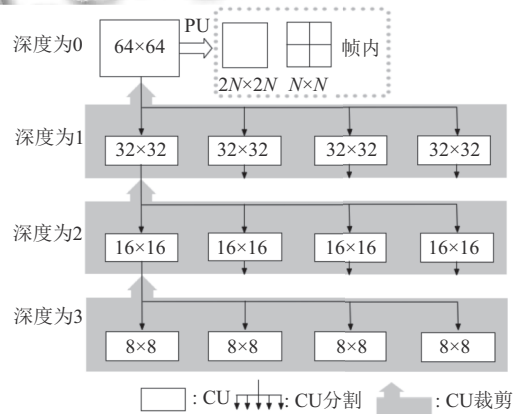


图 1 HEVC 四叉树递归划分示意图

每一个 CU 都需要进行 PU 模式选择. 图 2 是 HEVC 帧内预测的 35 种预测模式示意图.

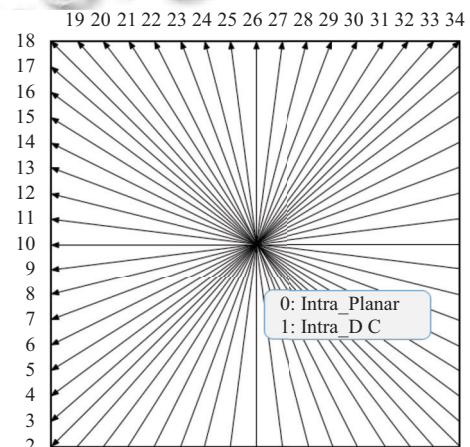


图 2 35 种帧内预测模式示意图

首先, 利用公式 (1) 对 PU 的 35 种预测模式进行粗选 (Rough Mode Decision, RMD).

$$J_{RMD} = SATD + \lambda_{pred} \times Bit \quad (1)$$

其中, $SATD$ 表示残差信息 Hadamard 变换绝对值之和, λ_{pred} 为拉格朗日乘子, Bit 为编码码率. 当 PU 尺寸是 8×8 和 4×4 时分别粗选出 8 种预测模式; 当 PU 尺寸是 64×64 , 32×32 , 16×16 时分别粗选出 3 种预测模式^[12]. 然后再从周围已编码的块中提取出当前 PU 的最有可能模式 (Most Probable Mode, MPM). 将粗选出的模式和 MPM 一起加入到候选列表中, 进行率失真优化 (RDO) 计算, 确定这个 PU 最佳的预测模式. RDO 计算公式 (2) 如下:

$$J_{RDO} = SSE + \lambda_{mode} \times Bit \quad (2)$$

其中, SSE 为残差平方和, λ_{mode} 为拉格朗日乘子, Bit 为编码所需码率. 在这个过程中一共需要进行 $35 \times (4^0 + 4^1 + 4^2 + 4^3 + 4^4) = 11\,935$ 次 J_{RMD} 的计算, 然后至少需要进行 $3 \times (4^0 + 4^1 + 4^2) + 8 \times (4^3 + 4^4) = 2623$ 次 RDO 计算. 这些计算导致帧内编码的复杂度极高^[13].

图 3 是 HM10.0 (HEVC Test Model) 实际编码一帧图像的 CU 划分图. 从图 3 中, 我们可以看出纹理平坦的区域如背景墙壁和地面, 适合用大尺寸的 CU 进行编码. 而纹理复杂的区域如人脸和篮球, 适合用较小尺寸的 CU 编码. 所有可以根据图像的纹理特征避免一些不必要的递归划分.

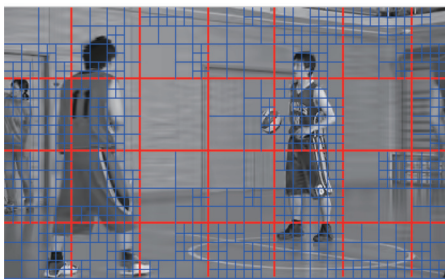


图 3 HM10.0 实际编码一帧图像的 CU 划分图

2 HEVC 帧内编码快速算法

2.1 编码块纹理特征提取函数的提出

在 HEVC 帧内编码的过程中, 图像的纹理特征与 CU 的划分方式以及 PU 预测模式有着非常紧密的联系. 所以, 本文提出一种纹理特征提取函数, 提取编码块水平、垂直、 45° 和 135° 方向上的纹理复杂度. 计算公式如 (3)~(6).

$$VarH = \frac{1}{N \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |Y(i, j) - Y_H(i)| \quad (3)$$

$$VarV = \frac{1}{N \times N} \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} |Y(i, j) - Y_V(j)| \quad (4)$$

$$Var45^\circ = \frac{1}{N \times N} (Var45^\circ_L + Var45^\circ_R) \quad (5)$$

$$Var135^\circ = \frac{1}{N \times N} (Var135^\circ_L + Var135^\circ_R) \quad (6)$$

在公式 (3) 和 (4) 中, $VarH$ 和 $VarV$ 分别表示水平和垂直方向的纹理复杂度 (Variance), $Y(i, j)$ 为编码块在第 i 行第 j 列的像素值, $Y_H(i)$ 为第 i 行像素的均值, $Y_V(j)$ 为第 j 列像素的均值, N 表示当前编码块的尺寸. 在公式 (5) 和 (6) 中, $Var45^\circ$ 和 $Var135^\circ$ 分别表示 45° 和 135° 方向上的纹理复杂度. 以 CU 尺寸 64×64 为例说明特征提取函数, 如图 4 所示, 将尺寸为 64×64 编码块的所有像素点划分到 45° 和 135° 方向. 纹理复杂度的计算将沿对角线分成左右两部分用 $Var45^\circ_L$ 、 $Var45^\circ_R$ 、 $Var135^\circ_L$ 和 $Var135^\circ_R$ 表示. 公式 (5) 和公式 (6) 可以进一步分解成公式 (7)~(10). 其中, N 表示当前 CU 的尺寸, $Y_{45^\circ}(k)$ 表示从左上角到右下角第 k ($0 \leq k \leq 2 \times N - 1$) 个 45° 方向上像素点的均值, $Y_{135^\circ}(k)$ 表示从左下角到右上角第 k ($0 \leq k \leq 2 \times N - 1$) 个 135° 方向上像素点的均值, 绝对值中的被减数表示沿着第 k 个 45° 或 135° 方向上相应位置处的像素值.

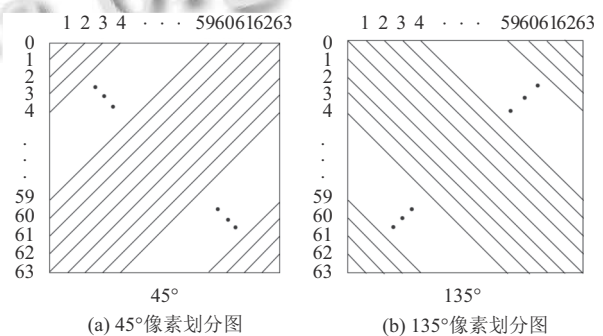


图 4 45° 和 135° 编码块像素点划分示意图

$$Var45^\circ_L = \sum_{k=0}^{N-1} \sum_{i=0}^k |Y[(i \times N), (k - i)] - Y_{45^\circ}(k)| \quad (7)$$

$$Var45^\circ_R = \sum_{k=N}^{2 \times N - 2} \sum_{i=k-(N-1)}^{N-1} |Y[(i \times N), (k - i)] - Y_{45^\circ}(k)| \quad (8)$$

$$Var135^{\circ}_L = \sum_{k=0}^{N-1} \sum_{j=0}^k |Y[(N-1+j-k) \times N, j] - Y_{135^{\circ}}(k)| \quad (9)$$

$$Var135^{\circ}_R = \sum_{k=N}^{2 \times N - 2} \sum_{j=k-(N-1)}^{N-1} |Y[(N-1+j-k) \times N, j] - Y_{135^{\circ}}(k)| \quad (10)$$

图像纹理简单区域, 像素点的值比较接近, 那么 $VarH$ 、 $VarV$ 、 $Var45^{\circ}$ 、 $Var135^{\circ}$ 的值都会很小. 如果图像纹理存在方向性, 比如沿水平方向, 那么水平方向的像素值相差就不会很大, 则 $VarH$ 的值最小, $VarV$ 、 $Var45^{\circ}$ 和 $Var135^{\circ}$ 相对要大. 所以通过分析 $VarH$ 、 $VarV$ 、 $Var45^{\circ}$ 和 $Var135^{\circ}$ 可以得出编码块的复杂度和方向特性.

2.2 CU 分割快速算法

本文 CU 快速分割算法是通过上述四个方向上计算出来的编码块的复杂度与阈值进行比较, 决定当前编码块是否要分割. 所以阈值的选取至关重要, 在 HM10.0 中加入特征提取函数, 让 HM10.0 在正常编码测试序列 ClassA~ClassE 的过程中分析阈值的取值. 当 QP=32 时, 通过大量仿真实验表明, 选取阈值 $Var_Thre=6$ 时, 编码效果最佳. 本文提出的 CU 快速分割算法, 具体实现步骤如下:

首先计算当前编码块的 $VarH$ 、 $VarV$ 、 $Var45^{\circ}$ 和 $Var135^{\circ}$. 求出这四个值里面的最大值 $VarMax$ 和最小值 $VarMin$, 如式 (11) 和式 (12).

$$VarMax = \max(VarH, VarV, Var45^{\circ}, Var135^{\circ}) \quad (11)$$

$$VarMin = \min(VarH, VarV, Var45^{\circ}, Var135^{\circ}) \quad (12)$$

然后通过 $VarMax$ 和 $VarMin$ 与 Var_Thre 进行比较决定部分 CU 是否划分. 如果 $VarMax$ 小于 Var_Thre , 可以认为当前编码块是纹理简单区域, 则 CU 不用进行子块划分. 如果 $VarMin$ 大于 $1.25 \times Var_Thre$, 则可以认为当前编码块为纹理复杂区域, 直接跳过率失真代价计算, 将 CU 划分成 4 个 Sub-CU. 如果不满足前面的两个条件, 则按照 HM10.0 的算法进行正常流程的 CU 划分的判断. 这样挑选出编码过程中根据纹理特征本可避免递归划分的 CU, 直接进行划分判断, 所以本算法可以大大减少 CU 二叉树划分的时间.

本文快速 CU 分割算法, 实际分割的一帧图像如图 5 所示. 满足纹理复杂区域用小尺寸 CU 编码, 纹理简单区域用大尺寸 CU 编码. 与图 3 标准算法分割图相比, 省去了一些不重要位置的详细划分, 不会影响重

建图像的视觉效果. 但更为简单的划分方式, 可以很好地缩短 CU 划分的时间.

2.3 PU 预测模式优化算法

由前面分析可知最佳预测模式一般是与纹理方向保持一致. 所以先对 35 种预测模式进行分类, 通过 $VarMin$ 判断当前编码块最佳预测模式落在哪个类中, 然后只在这个类中进行寻找. 将图 2 中的 0-34 种预测模式分成四类, 可得 C_0-C_3 候选集^[14]. 如表 1 所示.

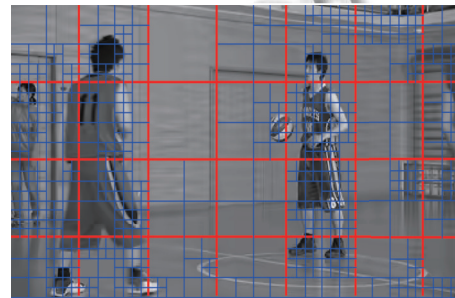


图 5 本文算法实际编码一帧图像的 CU 划分图

表 1 角度预测模式候选集

Class C_i	Judgement condition	Texture direction	Subset of modes chosen for RMD calculation
C_0	$VarMin=VarH$	水平	0,1,6-14
C_1	$VarMin=Var45^{\circ}$	45°	0,1,2-5,30-34
C_2	$VarMin=VarV$	垂直	0,1,22-30
C_3	$VarMin=Var135^{\circ}$	135°	0,1,14-22

若当前编码块的纹理平坦, 不存在纹理方向性, 那么 $VarMax$ 与 $VarMin$ 比较接近, 即 $VarMax-VarMin < 0.1 \times VarMin$, 则将 DC、Planar 和 MPM 模式进行 RDO 计算, 选出最佳模式. 若 $VarSecMin-VarMin < 0.1 \times VarMin$, 其中 $VarSecMin$ 为 $VarH$ 、 $Var45^{\circ}$ 、 $VarV$ 和 $Var135^{\circ}$ 中的第二小值, 说明当前编码块存在两个纹理方向, 将这两个方向对应的候选集 C_i 和 C_j 作为先选模式, 根据当前深度粗选出 J_{RMD} 值较小的前 N (数量同文献[12]) 个候选模式和 MPM 模式一起进行 RDO 计算. 其它情况下, 说明当前编码块只存在一个纹理方向, 若此时编码块的深度小于等于 2, 则将候选集 C_i 先进行 RMD 计算粗选出 J_{RMD} 值较小的前 3 种候选模式和 MPM 模式一起进行 RDO 计算. 若当前编码块深度大于 2, 则将候选集 C_i 和 MPM 模式一起进行 RDO 计算, 选出最佳模式.

2.4 帧内编码快速算法的流程

帧内编码快速算法的流程如图 6.

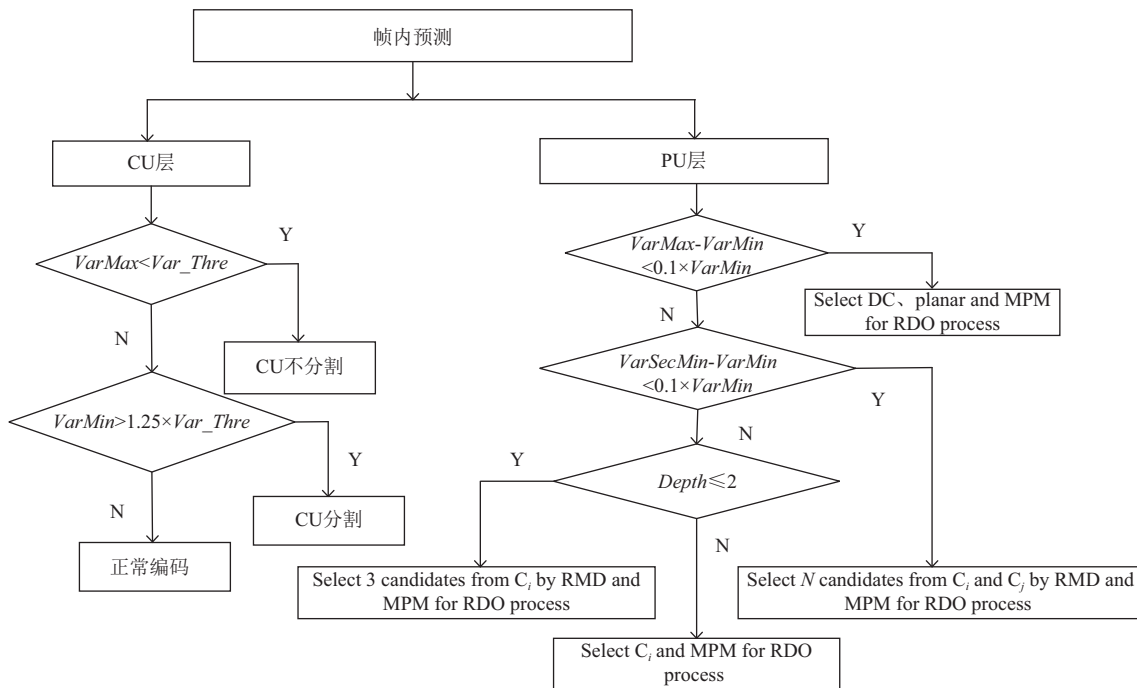


图6 帧内编码快速算法流程图

3 实验结果与分析

为了评估本文快速算法的编码性能, 本文以 HEVC 的参考实现软件 HM10.0 为参照进行试验. 实验硬件平台是 Intel(R) Core(TM) i5-3317U CPU@ 1.70 GHz, 内存为 4 GB, 软件开发环境是 Windows7, Visual Studio 2010. 因为本文只针对帧内编码进行优化, 故采用的编码配置模式为全 I 帧 (AI) 模式, QP 选取 32, 测试时使用 JCT-VC 提供的标准视频测试序列 ClassA~ClassE, 全部统一编码 100 帧. 为了分析算法的性能, 使用节省的编码时间百分比 ΔET 、比特率增量 ΔBR 、峰值信噪比增量 $\Delta PSNR$ 作为衡量标准. 计算公式为 (13)~(15).

$$\Delta ET = (T_{HM10.0} - T_{proposed}) / T_{HM10.0} \times 100\% \quad (13)$$

$$\Delta BR = (BR_{proposed} - BR_{HM10.0}) / BR_{HM10.0} \times 100\% \quad (14)$$

$$\Delta PSNR = PSNR_{proposed} - PSNR_{HM10.0} \quad (15)$$

其中, $T_{proposed}$ 、 $BR_{proposed}$ 、 $PSNR_{proposed}$ 分别表示本文快速算法的编码时间、比特率和峰值信噪比; $T_{HM10.0}$ 、 $BR_{HM10.0}$ 、 $PSNR_{HM10.0}$ 分别表示 HEVC 测试模型 HM10.0 的编码时间、平均比特率和峰值信噪比.

表 2 是本文算法与标准算法 HM10.0 对比的实验结果. 根据表 2, 本文算法在码率升高 0.649%, 峰值信噪比降低 0.059 dB 的情况下, 编码时间平均减少 56.18%.

表 2 本文算法与 HM10.0 的测试结果比较

类型	序列	ΔBR (%)	$\Delta PSNR$ (dB)	ΔET (%)
Class A	Tarffic	0.645	-0.052	56.66
	PeopleOnstreet	0.275	-0.045	48.88
Class B	Kimonol	1.179	-0.050	69.14
	BasketballDrive	1.002	-0.022	46.03
Class C	RaceHorses	0.259	-0.063	55.49
	BasketballDrill	0.515	-0.082	50.45
Class D	BasketballPass	0.737	-0.059	56.07
	BQSquare	-0.072	-0.086	54.45
Class E	Vidyo1	1.006	-0.073	63.81
	Vidyo4	0.947	-0.063	60.82
平均值		0.649	-0.059	56.18

表 3 是本文算法和参考文献其它快速算法与 HM 标准算法的比较结果. 根据表 3 的数据, 本文快速算法与其它快速算法在码率和峰值信噪比相差不大的情况下, 能够节省更多的编码时间.

表 3 本文算法与各快速算法的性能比较

各快速算法	ΔBR (%)	$\Delta PSNR$ (dB)	ΔET (%)
文献[8]	0.6	-0.05	31.9
文献[10]	0.93	-0.04	48.28
文献[11]	0.343	-0.148	48.06
本文算法	0.649	-0.059	56.18

图 7 是本文算法与 HM10.0 的 RD 性能比较. 如图所示, 本文算法在编码视频序列 BasketballPass

(416×240_50)下的RD曲线与HM10.0算法的RD曲线基本重合. 所以本文算法的编码性能与HEVC标准算法相比, 没有明显的下降.

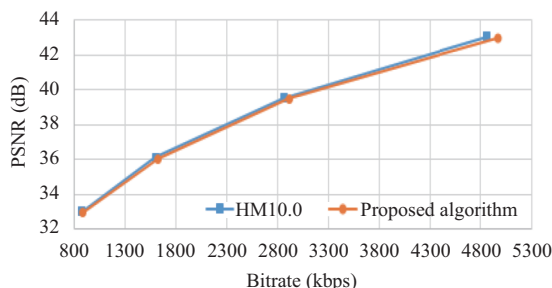


图7 本文算法与HM10.0的RD曲线图

从主观评价上, 本文算法和HM10.0标准算法编码重构后的图像如图8和9. 从这两幅图像的比较中, 可以看出本文算法与HM10.0标准算法实际编码并没有明显差异.



图8 HM10.0标准算法编码重建后的图像



图9 本文算法编码重建后的图像

4 结束语

本文快速算法是根据当前编码单元的纹理特征, 来降低帧内编码算法的复杂度, 节省编码时间. 实验结果显示, 在保证视频质量和码率的情况下, 有效地降低了编码时间. 本文算法考虑了编码块的纹理特征, 并没有考虑到视频信息中存在的时空相关性, 接下来将研究编码单元划分深度的时空相关性进一步降低HEVC视频编码算法的复杂度, 缩减编码时间. 之后, 本文快速算法将应用于智能回收箱项目的视频监控.

参考文献

- 1 朱秀昌, 刘峰, 胡栋. H.265/HEVC: 视频编码新标准及其扩展. 北京: 电子工业出版社, 2016.
- 2 Wiegand T, Sullivan GJ, Bjontegaard G, *et al.* Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 2003, 13(7): 560–576. [doi: 10.1109/TCSVT.2003.815165]
- 3 Bross B, Han WJ, Sullivan GJ, *et al.* High Efficiency Video Coding (HEVC) text specification draft 10(for FDIS & Consent). Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-L1003. 12th Meeting, Geneva, Switzerland. 2013, 1.
- 4 Cho S, Kim M. Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013, 23(9): 1555–1564. [doi: 10.1109/TCSVT.2013.2249017]
- 5 Zhang QW, Zhao JC, Huang XP, *et al.* A fast and efficient coding unit size decision algorithm based on temporal and spatial correlation. *Optik-International Journal for Light and Electron Optics*, 2015, 126(21): 2793–2798. [doi: 10.1016/j.ijleo.2015.07.026]
- 6 Zhao DY, Zhu SP, Gao SP. A novel fast intra-prediction algorithm for high-efficiency video coding based on structural similarity. *Optik-International Journal for Light and Electron Optics*, 2015, 126(23): 4212–4218. [doi: 10.1016/j.ijleo.2015.08.035]
- 7 Nishikori T, Nakamura T, Yoshitome T, *et al.* A fast CU decision using image variance in HEVC intra coding. *Proceedings of the 2013 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, Kuching, Malaysia. 2013. 52–56.
- 8 沈健, 杨静. 基于提前终止编码单元划分的快速帧内预测算法. *计算机应用与软件*, 2016, 33(5): 169–172.
- 9 Belghith F, Kibeya H, Ayed MAB, *et al.* Fast coding unit partitioning method based on edge detection for HEVC intra-coding. *Signal, Image and Video Processing*, 2016, 10(5): 811–818. [doi: 10.1007/s11760-015-0820-2]
- 10 张俊, 项欣光. 基于局部边界特性的HEVC帧内快速预测算法. *计算机应用与软件*, 2015, 32(12): 101–104. [doi: 10.3969/j.issn.1000-386x.2015.12.025]
- 11 蔡婷婷, 陈晓. 高效视频编码帧内快速划分决策算法. *计算机仿真*, 2016, 33(5): 259–263.
- 12 Zhao L, Zhang L, Ma SW, *et al.* Fast mode decision algorithm for intra prediction in HEVC. *Proceedings of the 2011 Visual Communications and Image Processing (VCIP)*. Tainan, China. 2011. 1–4.
- 13 李伟, 王让定, 王家骥, 等. 一种低复杂度的HEVC帧内快速编码算法. *光电子·激光*, 2015, 26(3): 597–604.
- 14 孙学斌, 陈晓冬, 肖禹泽, 等. 基于图像纹理特征的HEVC帧内编码快速算法. *激光与光电子学进展*, 2017, 54(7): 071003.