

DOI: 10.11830/ISSN.1000-5013.201603051



双策略高效视频编码帧间模式 快速决策算法

罗志伟, 冯桂

(华侨大学 信息科学与工程学院, 福建 厦门 361021)

摘要: 为了减少编码复杂度, 提出针对编码单元(CU)级和预测单元(PU)级的帧间快速算法. 首先, 利用当前编码块的 3 个空间邻近已编码块的深度信息, 提前跳过某些 CU 的编码模式的检测. 其次, 利用当前编码块执行完 Inter_2N×2N 模式后, 得出的残差块的纹理信息跳过某些 PU 的编码模式检测. 实验结果表明: 文中算法可以节省 45.03% 的编码时间; BD-BR(bjontegaard delta bit rate)造成 1.49% 的损失; 算法能够很大程度节省编码时间; 引入的率失真性能损失忽略不计.

关键词: 高效视频编码; 快速选择; 当前块; 邻近块; 残差纹理

中图分类号: TN 919.81 **文献标志码:** A **文章编号:** 1000-5013(2018)02-0262-06

Fast Decision Algorithm of Inter-Frame Mode Using Double Methods for High Efficient Video Coding

LUO Zhiwei, FENG Gui

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

Abstract: In order to reduce coding complexity, we proposed inter-frame fast decision algorithm for coding unit (CU) and prediction unit (PU). Firstly, using three spatially adjacent coding block depth information of the current coding block to skip certain CU encoding mode detection. Then, using the texture information of the residual block after performed the Inter_2N×2N mode to skip certain PU encoding mode detection. Experimental results show that the algorithm can save coding time 45.03%, while 1.49% of the loss of BD-BR (bjontegaard delta bit rate). The algorithm proposed can also largely save coding time, and rate-distortion performance loss is negligible.

Keywords: high efficient video coding; fast decision; current block; neighboring block; residual texture

为满足人们对高清视频的需求, 在视频感知质量相当的情况下, 高效视频编码(high efficient video coding, HEVC)相比于 H. 264, 比特率要减少 50%^[1]. HEVC 带来更小码率的同时, 却比 H. 264 具有更大的复杂度. 就帧间预测模式而言, 它支持 4 种对称预测模式, 4 种非对称预测模式(AMP), 穷尽计算出每个深度下模式选择过程, 并根据最小率失真代价选出最优组合. 就复杂度而言, 它消耗的时间是非常巨大的, 降低 HEVC 编码器的复杂度非常有必要. 国内外很多科研工作者为了优化 HEVC 编码器, 从帧内帧间的预测做了大量的优化^[2-10]. 本文采集空域邻近编码单元(CU)的信息决策当前 CU 的深度级, 同时, 针对预测单元(PU)分割的优化, 模拟残差块的纹理分布决策跳过某些 PU 的编码模式检测.

收稿日期: 2016-03-16

通信作者: 冯桂(1960-), 女, 教授, 博士, 主要从事信号与信息处理、通信系统信息安全理论、多媒体通信、计算机视觉、模式识别与影像理解的研究. E-mail: fengg@hqu.edu.cn.

基金项目: 福建省自然科学基金资助项目(2016J01306)

1 HEVC 帧间预测简介

HEVC 的编码单元的大小从 $64\text{ px}\times 64\text{ px}$ 至 $8\text{ px}\times 8\text{ px}$,使用四叉树的递归分割结构进行编码. 其四叉树编码过程如下. 首先,进行最大编码单元(largest coding unit,LCU)的编码,此时 CU 尺寸为 $64\text{ px}\times 64\text{ px}$,深度为 0,对 CU 穷尽帧间预测模式选择,并选择最小的率失真代价预测模式. 其次,将 CU 分割为 4 个子 CU,深度为 1,对 4 个子 CU 分别进行预测模式选择. 根据选择最小的 rdcost 原则,得到每个子 CU 的预测模式,直到 CU 的深度为 3,大小为 $8\text{ px}\times 8\text{ px}$ 时才结束分割. 最后,对不同深度下的 CU 进行选择,将 4 个 $8\text{ px}\times 8\text{ px}$ 子 CU 的 rdcost 之和同相对应的 $16\text{ px}\times 16\text{ px}$ 子 CU 的 rdcost 作比较,若前者较小,则选择 $8\text{ px}\times 8\text{ px}$ 的分割模式;否则,选择 $16\text{ px}\times 16\text{ px}$ 分割模式,直到 CU 的大小为 $64\text{ px}\times 64\text{ px}$,选出 LCU 的最佳分割模式.

帧间预测模式,如图 1 所示. HEVC 针对当前深度的 CU 先进行 skip/merge 模式预测,接着进行各个 PU 分割预测,即对 $\text{PART}_{2N\times 2N}$, $\text{PART}_{N\times 2N}$, $\text{PART}_{2N\times N}$ 进行模式选择,若深度不为 3,还将进行非对称模式预测(AMP),即对模式 $\text{PART}_{NL\times 2N}$, $\text{PART}_{NR\times 2N}$, $\text{PART}_{2N\times NU}$, $\text{PART}_{2N\times NB}$ 进行选择,根据最小的 rdcost 选出最优预测模式. 其中, $\text{PART}_{N\times N}$ 模式只有在 CU 等于所允许的最小尺寸时才启用^[11].

2 CU 级深度预测

根据文献[6],当前 LCU 跟其相邻空域已编码的 LCU 存在着相当强的相关性. 若一个 LCU 的空域邻近 LCU 采用的编码深度较小,往往会用较大尺寸的编码块编码;反之,则用较小尺寸的编码块编码. 对 LCU 定义一个分类,若一个 LCU 的最大深度不超过 1,则定义它为简单块;若 LCU 的最大深度为 3,则定义它为复杂块.

利用当前 LCU 的空域相邻 LCU 决策合适的深度,估计空域预测的准确性,选取 Class A~E 的 9 个序列测试,这些序列的图像内容分别包含纹理平坦及纹理复杂,QP 为 32,用邻近 LCU 深度信息预测当前 LCU 的深度的命中率,如表 1 所示. 表 1 中:Depth 0~2 的含义是邻近 LCU 均为简单块时,选择的最优深度为 0~2 的概率;Depth 1~3 的含义是邻近块均为复杂块时,选择的最优深度为 1~3 的概率. 由表 1 可知:若邻近 LCU 是简单块,则当前 LCU 没有选择深度 3 的概率为 100%;若邻近 LCU 是复杂块,则当前 LCU 没有选择深度 0 的概率在 55.6%~96.4%波动.

因此,文中采用的算法 1 如下:若左边 LCU、上边 LCU、左上 LCU 均为简单块,则跳过当前 LCU 在深度 3 下的帧间编码模式选择过程;若左边 LCU、上边 LCU、左上 LCU 均为复杂块,则跳过当前 LCU 在深度 0 下的帧间编码模式选择过程.

表 1 用邻近 LCU 深度信息预测当前 LCU 的深度的命中率

Tab. 1 Hit-rates of using adjacent LCU depth information for prediction of current LCU depth				%			
序列	简单块和复杂块出现的概率和	Depth 0~2	Depth 1~3	序列	简单块和复杂块出现的概率和	Depth 0~2	Depth 1~3
BasketballDrill	86.1	100	80.3	Johnny	86.9	100	68.8
BasketballPass	73.2	100	55.6	Kimono1	69.4	100	71.7
BlowingBubbles	74.9	100	96.4	Cactus	82.6	100	80.6
PartyScene	83.8	100	75.6	Traffic	87.9	100	75.6
FourPeople	86.7	100	64.8				

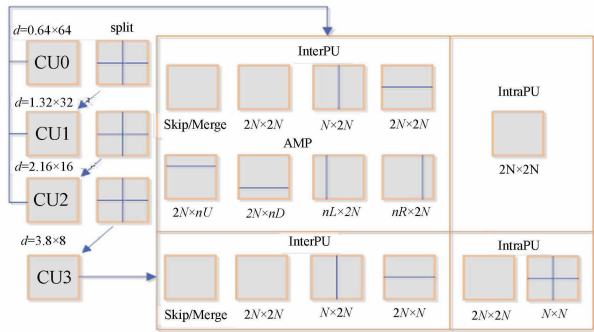


图 1 HEVC 四叉树递归分割结构及各深度下的预测模式
Fig. 1 HEVC quadtree recursive partitioning structure and prediction mode in each depth

3 残差块纹理预测模式

经过 $\text{Inter_}2N \times 2N$ 模式预测后,可以得到当前深度下的残差块,为了判定 CU 内的运动情况,将该 CU 的残差块作如下处理:将一个残差 CU 分割成 16 个残差子 CU(深度为 0,1,2,3 的残差块分别分割成 16 个 $16\text{ px} \times 16\text{ px}$, $8\text{ px} \times 8\text{ px}$, $4\text{ px} \times 4\text{ px}$, $2\text{ px} \times 2\text{ px}$ 残差子块).

残差子块的平均值和偏离平均值的绝对差之和(SADA),分别为

$$\left. \begin{aligned} \text{average}_{i,j} &= \sum_{x=0}^{K-1} \sum_{y=0}^{K-1} \text{resi_pixel}(x,y), \\ \text{SADA}_{i,j} &= \sum_{x=0}^{K-1} \sum_{y=0}^{K-1} | \text{resi_pixel}(x,y) - \text{average}_{i,j} |. \end{aligned} \right\} \tag{1}$$

式(1)中: K 为残差子块的宽度和高度; $\text{resi_pixel}(x,y)$ 为残差子块在 (x,y) 位置的像素值; $\text{average}_{i,j}$ 为第 (i,j) 个残差子块的像素平均值; $\text{SADA}_{i,j}$ 为第 (i,j) 个残差子块的 SADA,使用 SADA 代替方差,从而模拟残差子块内像素值的变化情况. 如果某个残差子块的 SADA 小于阈值 ThSA ,则认为该子块是平滑的,即

$$\text{ThSA} = (\text{ct} + \text{depth}) \times (1 \ll (4 - \text{depth}) \ll (4 - \text{depth})). \tag{2}$$

式(2)中: depth 为当前深度; ct 为设定的一个常量; $(1 \ll (4 - \text{depth}) \ll (4 - \text{depth}))$ 为残差子块内的像素数目.

残差子块的 SADA 和平均值可估计块内的运动情况,若它们均小于某个阈值(ThSA 的 ct 设为 4, ThAvg 设为 7)^[5]. 采用如下算法 2.

- 策略 1
- 1) 若残差块的 16 个残差子块的平均值的绝对值小于给定的阈值 ThAvg .

2) 如果当前块的深度为 0,至少有 15 个残差子块的被认为是平滑的;或者当前块的深度为 1,2,3 时,16 个残差子块均被认为是平滑的.

若策略 1 的 1),2) 均满足,则提前选择 $\text{Inter_}2N \times 2N$ 为最佳模式,跳过剩余的帧间预测模式. 如果策略 1 的条件不成立,才进入策略 2.

把 4 个空间邻近的残差子块合并成一个虚拟块(VB),使得一个残差块具有 4 个虚拟块. 如果一个虚拟块中有超过两个残差子块是不平滑的,则认为该虚拟块是不平滑的. 残差块内的不平滑虚拟块分布情况,如图 2 所示.

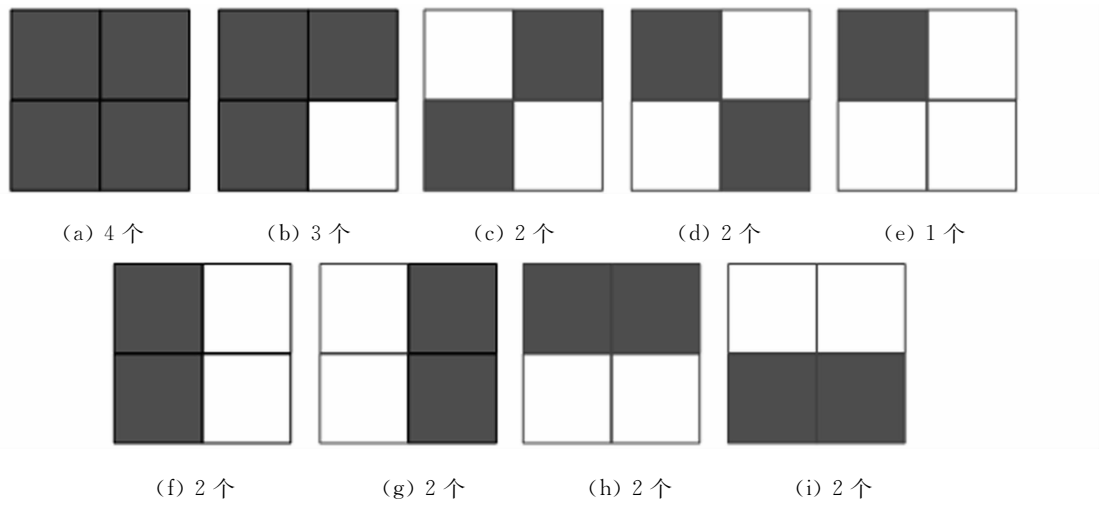


图 2 残差块内不平滑 VB 的分布

Fig. 2 Distribution of unsmooth VB in residual block

策略 2 1) 若当前深度不为 3,残差块内不平滑的 VB 呈图 2(a)~(e)分布,则跳过剩余的 PU 编码模式的检测,直接分割当前深度的 CU;若当前深度为 3,残差块内不平滑的 VB 呈图 2(a)~(d)分布,跳过 $\text{Inter_}2N \times N$, $\text{Inter_}N \times 2N$.

2) 若不平滑的 VB 等于 2,且只分布在残差块内的上边或下边,如图 2(h), (i) 所示,则跳过 Inter_

$N \times 2N$. 如果 CU 在经过 $\text{Inter_}2N \times N$ 后的最佳预测模式是 $\text{Inter_}2N \times 2N$, 需要重新进行 $\text{Inter_}N \times 2N$. 同理, 可得不平滑的 VB 只分布在残差块内的左边或右边.

QP 为 32, 策略 1, 2 的命中率, 如表 2 所示. 策略 1 的命中率(η)是帧间模式选择 $2N \times 2N$ 的概率; 策略 2 的命中率是预测模式的选择满足条件 1), 2) 下的概率. 由表 2 可知: 策略 1 的命中率极高, 均在 98% 之上; 策略 2 的命中率偏低, 主要原因是因为其划分略微粗糙, 导致率失真性能部分损失.

表 2 策略 1 和 2 的命中率
Tab. 2 Hit-rates of strategy 1 and 2

序列	η (策略 1)	η (策略 2)	序列	η (策略 1)	η (策略 2)
BasketballDrill	99.4	60.7	Johnny	99.6	40.0
BasketballPass	99.6	60.3	Kimono1	99.2	53.4
BlowingBubbles	98.8	46.9	Cactus	99.5	46.2
PartyScene	98.6	49.8	Traffic	99.3	50.6
FourPeople	99.7	46.2			

4 算法流程

总的算法描述, 如图 3 所示. 算法有如下 4 个步骤.

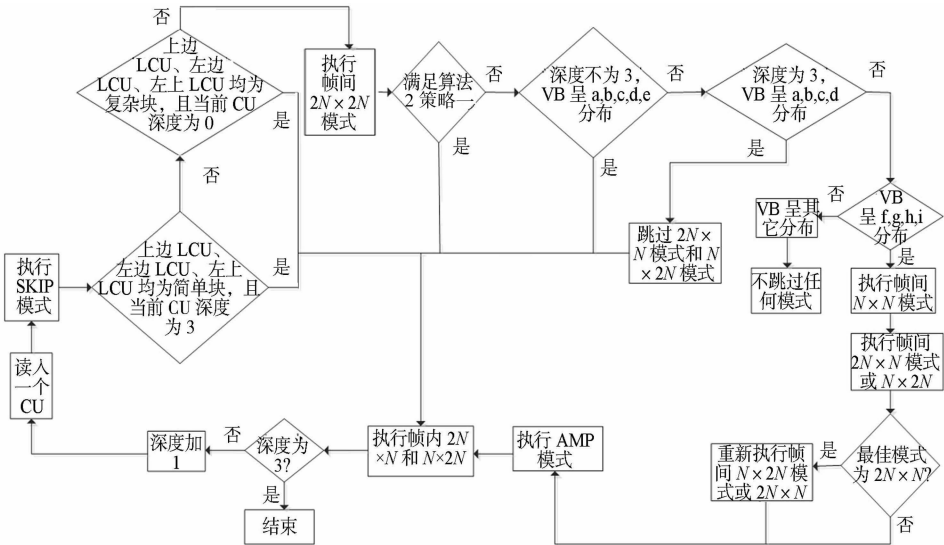


图 3 总的算法流程框图
Fig. 3 Flow diagram of total algorithms

- 步骤 1 读入一个 CU 后执行 skip 模式, 判断 3 个空域 LCU 的纹理复杂度是否满足算法 1 的条件, 若是, 则转至步骤 4; 否则, 转至步骤 2.
- 步骤 2 计算 CU 的 $\text{Inter_}2N \times 2N$, 判断是否满足算法 2 的策略 1 的条件 1) 和 2), 若满足, 转至步骤 4; 否则, 转至步骤 3.
- 步骤 3 判断残差块内的 VB 分布状况是否满足算法 2 的策略 2, 若满足, 执行策略 2 的算法; 否则, 转到步骤 4.
- 步骤 4 判断当前深度是否为 3, 若是, 则结束当前 CU 的模式预测; 否则, 深度加 1, 转到步骤 1.

5 实验结果与分析

实验采用的测试模型为 HM16.0; 测试硬件为英特尔酷睿 2 四核 Q9400@; CPU 为 2.66 GHz; 内存为 3.21 GB; 使用的计算机为 Window 7 旗舰版; 编译器为 Visual Studio 2010. 对 HEVC 官方规定 Class A~E 的 18 个序列进行测试^[12], 序列统一设定为 50 帧, 量化参数 QP 分别为 22, 27, 32, 37. 使用 BD-BR(bjontegaard delta bit rate)和 BD-PSNR(bjontegaard delta peak signal-to-noise rate)^[13]衡量率失真性能; ΔT 衡量复杂度的减少程度; $(\text{BD-BR})/\Delta T$ 衡量每减少 1% 的计算复杂度. 因为不同算法具有

不同的 BD-BR 与 ΔT , $(\text{BD-BR})/\Delta T$ 能够最有效地比较不同算法的性能. BD-BR 是在同等客观质量条件下, 两种方法的码率的节省情况. BD-PSNR 是在给定的同等码率下, 两种方法的 Y-PSNR 的差异. ΔT 的定义为

$$\Delta T = \frac{1}{4} \sum_{i=1}^4 \frac{T_{\text{HM16.0}}(\text{QP}_i) - T_{\text{pro}}(\text{QP}_i)}{T_{\text{HM16.0}}(\text{QP}_i)} \times 100\%.$$

(3)

式(3)中: $T_{\text{HM16.0}}(\text{QP}_i)$ 为原始算法 HM16.0 编码时间; $T_{\text{pro}}(\text{QP}_i)$ 为文中提出的算法编码时间.

选择文献[7],[8]中的算法与文中算法作比较. 在随机访问(RA)配置文件下运行的程序结果, 如表 3 所示. 由表 3 可知: 文中算法平均可以节省 45.03% 的编码时间, 同时, 带来可以忽略的率失真性能的损失; 文中算法与文献[7]相比, 编码时间减少量约为 5.5%, 率失真性能降低了 0.48% 的 BD-BR, $(\text{BD-BR})/\Delta T$ 也减小 1.67; 文中算法与文献[8]相比, 虽然编码时间的减少量略小于文献[8], 但是带来的率失真性能的损失也小于文献[8], $(\text{BD-BR})/\Delta T$ 比文献[8]减小了 0.54, 文中算法的性能略优于文献[8]; 文中引入的复杂度极小, 且只有在 CU 级算法不成立的时候才进入 PU 级的算法.

表 3 文中算法与文献[7],[8]算法实验比较结果

Tab. 3 Result comparison between proposed algorithm and literature [7],[8]

类别	序列	文献[7]算法			文献[8]算法			文中算法		
		BD-BR/%	ΔT /%	$\frac{\text{BD-BR}}{\Delta T}$	BD-BR/%	ΔT /%	$\frac{\text{BD-BR}}{\Delta T}$	BD-BR/%	ΔT /%	$\frac{\text{BD-BR}}{\Delta T}$
Class A	Traffic	2.05	43.2	4.75	2.03	56.13	3.62	1.30	51.81	2.50
	PeopleOnStreet	2.24	42.5	5.27	1.24	44.44	2.79	1.52	38.05	3.99
Class B	Kimono1	—	—	—	2.04	56.13	3.63	1.15	40.95	2.80
	ParkScene	—	—	—	1.46	47.28	3.09	1.25	47.62	2.60
	Cactus	2.98	40.2	7.41	1.77	48.44	3.65	1.29	45.47	2.85
	BQTerrace	—	—	—	2.1	45.07	4.66	0.81	47.98	1.70
	BasketballDrive	2.35	42.4	5.54	2.21	50.96	4.34	1.62	46.78	3.40
Class C	RaceHorsesC	—	—	—	—	—	—	2.44	36.83	6.62
	BQMall	1.98	37.2	5.31	1.58	45.96	3.44	2.37	42.67	5.57
	PartyScene	—	—	—	1.3	36.63	3.55	1.72	37.54	4.58
	BasketballDrill	1.86	36.7	5.06	1.70	47.58	3.57	1.33	43.03	3.10
Class D	RaceHorses	1.13	36.5	3.09	1.21	33.37	3.72	2.24	33.75	6.66
	BQSquare	1.04	37.4	2.78	0.84	30.67	2.74	1.29	40.06	3.22
	BlowingBubbles	—	—	—	0.89	28.82	3.09	1.67	34.61	4.85
	BasketballPass	—	—	—	1.19	36.63	3.25	1.71	44.52	3.86
Class E	FourPeople	2.07	39.8	5.20	2.83	66.35	4.27	0.64	58.10	1.10
	Johnny	1.95	39.1	4.99	2.48	70.93	3.50	1.32	60.84	2.17
	KristenAndSara	—	—	—	2.29	68.23	3.35	1.12	59.94	1.86
平均值		1.97	39.5	4.98	1.98	51.45	3.85	1.49	45.03	3.31

文中算法与原始算法 HM16.0 应用在序列 BQTerrace(Class B, 1 920 px×1 080 px)与序列 RaceHorsesC(Class C, 832 px×480 px)下的率失真曲线图, 如图 4 所示. 图 4 中: r 为比特率. 图 4(a), (b)

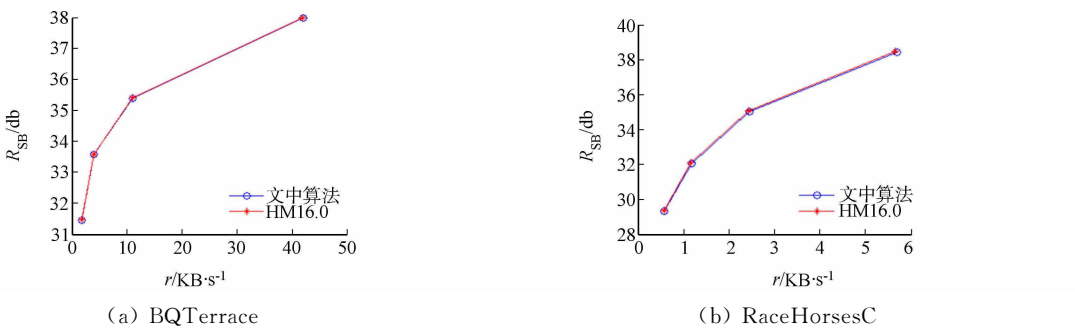


图 4 BQTerrace 与 RaceHorsesC 序列的率失真曲线

Fig. 4 Rate-distortion curves of BQTerrace and RaceHorsesC

在文中算法下编码分别能获得较好的率失真性能及较差的率失真性能. 由图 4 可知: 图 4(a), (b) 应用文中算法的率失真曲线与应用原始算法的率失真曲线几乎重合, 即在不同的比特率上, 文中算法与原始算法能够获得几乎相同的峰值信噪比.

6 结束语

针对新一代视频编码标准 HEVC 所带来的高额复杂度, 提出两种针对帧间预测模式选择的算法, 具体包括: 在 CU 级利用空域信息跳过当前 CU 模式选择; 在 PU 级利用残差块的特性得出运动复杂度, 从而跳过某些帧间的 PU 模式选择. 实验结果表明: 文中算法能够减少 45.03% 的编码复杂度, 同时, 取得了与原始算法相近的率失真性能, 适合于需要实时应用需求的环境.

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] CHOI K, PARK S H, JANG E S. Coding tree pruning based CU early termination[C]//JCTVC-F092, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 6th Meeting. Torino: [s. n.], 2011: 1-4.
- [3] YANG J, KIM J, WON K, *et al.* Early SKIP detection for HEVC[C]//JCTVC-G543, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 7th Meeting. Geneva: [s. n.], 2011: 1-6.
- [4] GWEON R, LEE Y L. Early Termination of CU encoding to reduce HEVC complexity[J]. IEICE Transactions on Fundamentals of Electronics, Information and Communication Engineers, 2012, 95(7): 1215-1218. DOI: 10.1587/transfun. E95. A. 1215.
- [5] ZHAO Wenjun, ONOYE T, SONG Tian. Hierarchical structure-based fast mode decision for H. 265/HEVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2015, 25(10): 1651-1664. DOI: 10.1109/TCSVT. 2015. 2395751.
- [6] SHEN Liqian, 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.
- [7] HUANG Xinpeng, ZHANG Qiuwen, ZHAO Xiaoxin, *et al.* Fast inter-prediction mode decision algorithm for HEVC [J]. Signal, Image and Video Processing, 2016: 1-8. DOI: 10.1007/s11760-016-0887-4.
- [8] ZHANG Yun, KWONG S, WANG Xu. Machine learning-based coding unit depth decisions for flexible complexity allocation in high efficiency video coding[J]. IEEE Transactions on Image Processing, 2015, 24(7): 2225-2238.
- [9] ZHANG Yongfei, LI Zhe, LI Bo. Gradient-based fast decision for intra prediction in HEVC[J]. IEEE Visual Communications and Image Processing, 2012, 42(4): 1-6. DOI: 10.1109/VCIP. 2012. 6410739.
- [10] LEE J H, JANG K S, KIM B G, *et al.* Fast intra mode decision algorithm based on local binary patterns in high efficiency video coding[C]//IEEE International Conference on Consumer Electronics. Las Vegas: [s. n.], 2015: 270-272.
- [11] YUAN Yuan, KIM I K, ZHENG Xiaozhen, *et al.* Quadtree based nonsquare block structure for inter frame coding in high efficiency video coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1707-1719. DOI: 10.1109/TCSVT. 2012. 2223037.
- [12] BOSSEN F. Common test conditions and software reference configurations[C]//JCTVC-L1100, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 12th Meeting. Geneva: [s. n.], 2013: 1-4.
- [13] BJØNTEGAARD G. Calculation of average PSNR differences between RD-curves[C]//ITU-Telecommunications Standardization Sector Study Group 16 Question 6 Video Coding Experts Group, VCEG-M33. Austin: [s. n.], 2001: 1-4.