

doi: 10.11830/ISSN.1000-5013.201604019



# 采用已编码信息的 HEVC 帧间快速模式决策算法

许东旭, 林其伟, 董晓慧

(华侨大学 信息科学与工程学院, 福建 厦门 361021)

**摘要:** 针对高效率视频编码(HEVC)帧间预测高额的计算复杂度,提出一种利用已编码块的相关信息进行快速帧间预测的算法.首先,在每个深度级上,利用当前编码单元联合其时空上最近的所有已编码单元的运动矢量(MV)长度,提前决定  $2N \times 2N$  的预测单元(PU)分割尺寸.其次,利用当前深度层获得的率失真代价(RDcost),结合离线统计出设置的阈值,终止满足条件的编码单元的进一步分割进程.实验结果表明:文中算法可以节省 37.8% 的编码时间,并取得与原始算法几乎相同的率失真性能.

**关键词:** 高效率视频编码; 帧间预测; 编码单元; 快速模式选择; 深度

**中图分类号:** TP 391      **文献标志码:** A      **文章编号:** 1000-5013(2016)04-0486-06

## Fast Inter Mode Decision Algorithm Using the Information of Encoded Coding Units for HEVC

XU Dongxu, LIN Qiwei, DONG Xiaohui

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

**Abstract:** In order to further reduce the great computational complexity for high efficiency video coding (HEVC) inter prediction, a novel algorithm using the information of encoded CUs has been proposed in this paper. Firstly, at each depth level, the lengths of all the MVs which are from current CU and spatially or temporarily adjacent CUs are used to decide  $2N \times 2N$  partition early. Secondly, by using the minimum RDcost of current CU and the off-line pre-set threshold values, the split of CUs which meet the requirements can be early termination. Experimental results show that the proposed algorithm can save an average of 37.8% of the encoding time with negligible loss of coding efficiency compared with the original algorithm.

**Keywords:** high efficiency video coding; inter prediction; coding unit; fast mode decision; depth

高效率视频编码(HEVC)作为新一代高效的视频编码标准,允许更大的编码单元尺寸,最大尺寸可以达到  $64 \times 64$ ,且首次分离了 3 种编码单元类型:编码单元(CU)、预测单元(PU)、变换单元(TU).与 H.264 相比较,在保证相同视频质量的条件下,HEVC 可减少 50% 的比特率<sup>[1]</sup>.但与此同时也引入了巨大的计算复杂度,例如,为了得到最优的 CU,HEVC 需穷尽地递归计算各种各样 CU,PU,TU 的组合.在每个深度级上,对于帧内预测来说,每个 PU 又需要进行高达 35 种帧内预测模式的遍历选择;对于帧间预测来说,也需进行较多 PU 模式的决策.因此,HEVC 离实时应用为时尚远.为了早日使 HEVC 投入市场应用,很多研究者围绕帧间预测优化和帧内预测优化这 2 个角度做了大量的努力<sup>[2-9]</sup>.本文对 CU 层与 PU 层同时进行优化,提出一种利用已编码信息的 HEVC 帧间快速模式决策算法.

**收稿日期:** 2014-08-07

**通信作者:** 林其伟(1957-),男,副教授,主要从事视频编码与网络通信的研究. E-mail:qwlin@hqu.edu.cn.

**基金项目:** 福建省自然科学基金资助项目(2012J01275)

# 1 HEVC 帧间预测流程

HEVC 提出的编码单元四叉树的递归分割技术,如图 1 所示.图 1 中:最大的 CU 称为编码树单元 (CTU).

首先,进行 CTU 四叉树的递归划分分割;当其不划分时,CU 尺寸为  $64\times 64$ ,深度为 0;对其进行预测编码,得其率失真代价.然后,将该 CU 分割为 4 个子 CU,每个子 CU 的尺寸为  $32\times 32$ ,深度为 1;对其进行预测编码,可得各自的率失真代价(RDcost),对这 4 个子 CU 分别进行进一步的分割,如此递归地划分下去;当子 CU 的尺寸为  $8\times 8$  时,即深度为 3 时,结束分割过程.最后,对分割的 CU 进行选择,依次比较求得的各个 CU 的 RDcost,选择具有最小 RDcost 的 CU 尺寸作为最优的分割模式.比较 4 个  $8\times 8$  子 CU 的代价和是否小于其对应的  $16\times 16$  的 CU 代价,如果小于,选择  $8\times 8$  的 CU 分割模式;否则,选择  $16\times 16$  的分割模式,如此比较下去,直至当前 CU 的深度为 0.

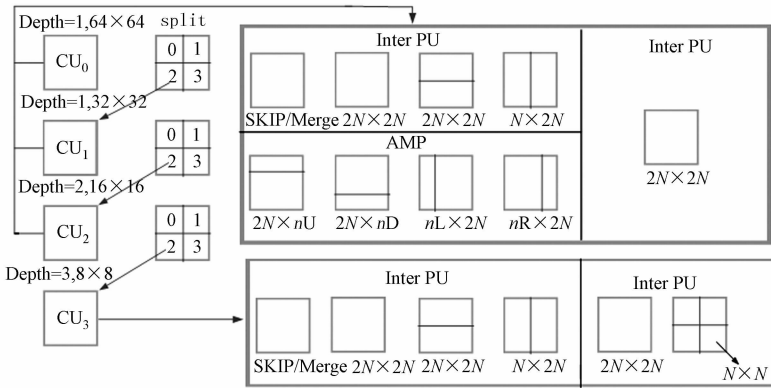


图 1 编码单元四叉树的递归分割技术

Fig. 1 Recursively split technology of quadtree of coding-units

图 1 中:在每个深度级上,HEVC 需要从 SKIP/Merge 模式、帧内预测 (intra PU) 模式及帧间各种 PU 分割尺寸,如  $2N\times 2N(N=32,16,8,4)$ ,  $2N\times N$ ,  $N\times 2N$  中选出具有最小 RDcost 的模式作为最优的预测模式.如果当前深度不是最小深度(深度 3),则还需对所有 AMP 分割模式,即对  $2N\times nU$ ,  $2N\times nD$ ,  $nL\times 2N$  和  $nR\times 2N$  进行运动估计,综合所有得出的 RDcost,从中选出最优的预测模式.在 HEVC 中,  $N\times N$  模式只能在最小的 CU 尺寸时才被允许,如果最小 CU 的尺寸是  $8\times 8$ ,则帧间  $N\times N$  模式在 HEVC 中是被禁止的<sup>[10]</sup>,而在帧内是允许的.

## 2 快速帧间模式决策算法

### 2.1 基于 RDcost 的 CU 分割提前终止算法

$$J_{mode} = (SSE_{luma} + w_{chroma} \cdot SSE_{chroma}) + \lambda_{mode} \cdot B_{mode}. \tag{1}$$

式(1)中:  $B_{mode}$  为编码当前块需要的比特数;  $\lambda_{mode}$  为拉格朗日因子;  $SSE_{luma}$  为当前的亮度块与参考块之间的均方差值和;  $SSE_{chroma}$  为当前的色度块与其参考块之间的均方差值和;  $w_{chroma}$  为色度块的加权因子;  $J_{mode}$  为求得的率失真代价.

利用式(1)的代价函数作为评价准则,HEVC 原始帧间预测以递归的方式对各种编码分割进行率失真优化(RDO)计算,求得最优的编码分割模式.直观上,1 个 CU 如果分割成 4 个子 CU,那么该 CU 的  $J_{mode}$  一般会较大;如果该 CU 没有进一步分割成 4 个子 CU,那么该 CU 的  $J_{mode}$  一般较小.因此,提出以下假设:RDcost 分布可能具有一定的规律性,可以利用该规律约束某些块不必要的分割进程.

为证明提出的假设,对各种序列进行大量的实验,以研究 RDcost 随着不同序列的分布情况.限于篇幅,选取其中一个典型的序列为例进行说明. PartyScene 序列在量化参数(QP)为 37 时,尺寸为  $16\times 16$  的 CU 中分割的 CU 和不分割的 CU 各自 RDcost 的取值分布情况,如图 2 所示.图 2 中:  $n$  为 CU 数.由图 2 可知:不分割 CU 的 RDcost 集中在相对较小值的范围内,并呈现如正态分布一样的“单峰”特性;分割 CU 的 RDcost 分散在相对较大值的范围内,几乎覆盖图 2 整个横坐标轴.以上的统计结果是文

中设计基于 RDcost 的 CU 分割提前终止算法的动机。

假如取图 2 中标注的 Threshold 为不分割 CU 的阈值,一旦当前深度的 CU 的最优 RDcost 小于 Threshold,则可终止当前 CU 的进一步分割处理,即跳过余下的深度计算,这将显著地降低编码的计算复杂度。

从以上论述可知:Threshold 在率失真性能损失与编码复杂度减少之间起到权衡的作用.一方面,如果 Threshold 取得太大,则可以得到可观的编码时间减少量,但率失真性能的损失也会加大;另一方面,如果 Threshold 取得太小,虽然可以使率失真性能的损失微小,但相应减少的编码时间减少量将相当有限。

综上,阈值选取原则:使统计序列在当前深度下 CU 的不分割概率达到 90%左右.同时,由于 RDcost 依赖于 QP,所以阈值还需随着 QP 自适应变化.得出的三组阈值(分别对应于  $64\times 64,32\times 32,16\times 16$  的 CU),分别为

$$\text{Thr}_{64} = 2\,270 \times \exp(0.890\,7 \times \text{QP}),$$

(2)

$$\text{Thr}_{32} = 722.2 \times \exp(0.109\,6 \times \text{QP}),$$

(3)

$$\text{Thr}_{16} = 228.5 \times \exp(0.113\,6 \times \text{QP}).$$

(4)

选取 5 个不同分辨率及运动特征各不相同的序列,证明式(2)~(4)所列阈值的有效性(节 2.1 算法).QP 为 22,37,序列全部编码 20 帧,采用 HM 10.1 原始的帧间 CU 尺寸决策,在当前 CU 满足上述三组阈值条件下,统计不分割的概率,实验结果如表 1 所示.表 1 中:“—”表示满足条件的 CU 块数很少,基本不会影响编码器的率失真性能与编码速度。

由表 1 可知:提出的基于 RDcost 的 CU 分割提前终止算法的命中率( $\eta$ ),对于大部分测试序列在不同 QP 下都达到了 90%以上。

表 1 基于 RDcost 的 CU 分割提前终止算法命中率

Tab. 1 Hit-rates of RDcost checking based on early split termination algorithm

| 分割模式  | QP | $\eta/\%$                      |                      |                           |                     |                     | $\eta_{ave}/\%$ |
|-------|----|--------------------------------|----------------------|---------------------------|---------------------|---------------------|-----------------|
|       |    | BasketballDrive<br>1 920×1 080 | PartyScee<br>832×480 | BasketballPass<br>416×240 | BQSquare<br>416×240 | Vidyo1<br>1 280×720 |                 |
| 64×64 | 22 | —                              | —                    | 97.4                      | —                   | 78.3                | 87.9            |
|       | 37 | 97.5                           | —                    | —                         | —                   | 97.8                | 97.7            |
| 32×32 | 22 | 90.8                           | 98.7                 | 93.4                      | —                   | 85.3                | 92.1            |
|       | 37 | 99.4                           | 98.7                 | 99.5                      | 98.0                | 99.3                | 99.0            |
| 16×16 | 22 | 97.6                           | 92.3                 | 96.0                      | 80.7                | 96.4                | 92.6            |
|       | 37 | 99.1                           | 99.8                 | 99.9                      | 99.8                | 100.0               | 99.7            |

2.2 基于 MV 长度的快速 PU 模式决策

2.2.1 算法设计的动机 一般来说,自然序列中存在大量的静止背景区域,这些区域每层深度上选中最大的  $2N\times 2N$  的 PU 分割尺寸的概率一般很高.为了证明上面的假设,选择 5 个典型序列,QP 选取 24,30,序列统一编码 30 帧,统计各深度上选中  $2N\times 2N$  分割尺寸的概率,实验结果如表 2 所示。

由表 2 可知:在每个深度级上,最优分割尺寸为  $2N\times 2N$  的概率极高,且随着 QP 的增加而增加.在 HM 当中,首先进行 SKIP 模式的计算,接着进行  $2N\times 2N$  的运动估计,同时 SKIP 模式也是  $2N\times 2N$  分割.因此,假如在  $2N\times 2N$  运动估计后,采用有效的策略提前决定当前深度层 CU 最优分割为  $2N\times 2N$ ,则可以合理地跳过余下 PU 模式( $2N\times N,N\times 2N$ ,AMP,帧内预测)的 RDO 计算,这将显著地降低编码的计算复杂度。

2.2.2 基于 MV 长度提前决定  $2N\times 2N$  PU 分割 运动静止的编码单元的 MV 一般很小,且很可能会选择较大的分割尺寸,如  $2N\times 2N$  分割.由于自然序列时空上的连续性,文中利用 6 个 MV 所组成的

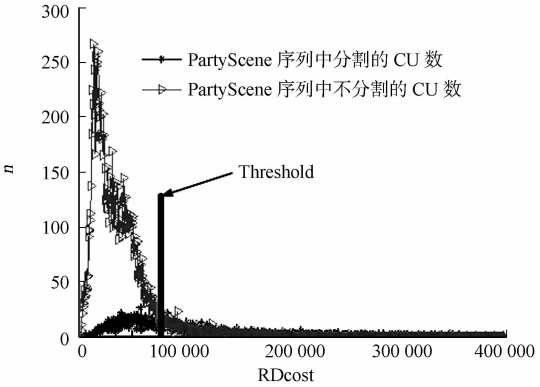


图 2 PartyScene 序列 RDcost 取值分布图  
Fig. 2 Distribution of RDcost  
in PartyScene sequence

表 2 各深度上  $2N \times 2N$  成为最优分割的概率  
Tab. 2 Probabilities of  $2N \times 2N$  PU partition being optimal partition at different depth levels

| 序列                             | 深度 | $\eta/\%$ |       | 序列                    | 深度 | $\eta/\%$ |       |
|--------------------------------|----|-----------|-------|-----------------------|----|-----------|-------|
|                                |    | QP=24     | QP=30 |                       |    | QP=24     | QP=30 |
| BasketballDrive<br>1 920×1 080 | 0  | 45.7      | 60.5  | Kimono<br>1 920×1 080 | 0  | 33.1      | 44.8  |
|                                | 1  | 65.6      | 77.3  |                       | 1  | 64.9      | 73.5  |
|                                | 2  | 79.3      | 86.8  |                       | 2  | 83.4      | 8.9   |
|                                | 3  | 87.4      | 93.2  |                       | 3  | 88.1      | 93.4  |
| PartyScene<br>832×480          | 0  | 33.3      | 33.5  | Vidyo1<br>1 280×720   | 0  | 70.5      | 80.1  |
|                                | 1  | 33.9      | 42.6  |                       | 1  | 83.8      | 91.2  |
|                                | 2  | 50.7      | 67.2  |                       | 2  | 93.9      | 97.3  |
|                                | 3  | 74.3      | 85.1  |                       | 3  | 98.3      | 99.4  |
| Johnny<br>1 280×720            | 0  | 68.3      | 81.5  |                       |    |           |       |
|                                | 1  | 82.1      | 91.9  |                       |    |           |       |
|                                | 2  | 93.2      | 97.8  |                       |    |           |       |
|                                | 3  | 98.0      | 99.5  |                       |    |           |       |

集合判断当前编码单元是否属于运动静止的区域,提前决定  $2N \times 2N$  的分割尺寸,如图 3 所示. 具体有以下 4 个步骤.

**步骤 1** 在当前深度层进行完  $2N \times 2N$  的运动估计后,收集所有 MV 信息(图 3).

**步骤 2** 计算每个 MV 的长度,即  
$$l(MV_i) = |x_i| + |y_i|, \quad i = 1, 2, 3, 4, 5, 6. \quad (5)$$

**步骤 3** 找出 MV 长度集合中最大的 MV 长度,即  
$$L = \max\{l(MV_1), l(MV_2), l(MV_3), l(MV_4), l(MV_5), l(MV_6)\}.$$

**步骤 4** 判断  $L$  是否小于文中给出的阈值. 若是,则终止当前深度层余下 PU 模式的 RDO 计算,直接编码下一个深度的 CU;否则,对当前深度层余下的 PU 模式进行 RDO 计算.

经过大量的实验统计,并参考 HEVC 的官方测试条件<sup>[11]</sup>,得出 4 个 QP(22,27,32,37)下的 MV 阈值为 0,2,4,6. 为了进一步证明文中基于 MV 长度提前决定  $2N \times 2N$  分割算法的有效性,采用 5 个统计序列(表 1),序列统一编码 20 帧,统计该算法的命中率,统计结果如表 3 所示.

表 3 文中基于 MV 长度提前决定  $2N \times 2N$  分割算法的命中率  
Tab. 3 Hit-rates of MV checking based early  $2N \times 2N$  determination algorithm

| 序列              | $\eta/\%$ |       |       |       |
|-----------------|-----------|-------|-------|-------|
|                 | QP=22     | QP=27 | QP=32 | QP=37 |
| BasketballDrive | 74.6      | 86.5  | 93.7  | 97.2  |
| PartyScene      | 97.9      | 91.6  | 93.7  | 96.3  |
| BasketballPass  | 98.7      | 98.2  | 97.9  | 98.2  |
| BQSquare        | 93.8      | 89.7  | 94.8  | 97.5  |
| Vidyo1          | 98.9      | 99.2  | 99.6  | 99.8  |

2.3 总的算法步骤

快速的帧间模式决策算法必须考虑错误传播的问题. 因此,提出的 CU 级优化算法在 I 帧中不执行;提出的 PU 级优化算法需在  $MV_2 \sim MV_6$  中,至少 3 个存在时才执行. 总的算法有以下 5 个步骤.

**步骤 1** 计算 SKIP 模式,计算  $2N \times 2N$  模式. 利用式(5),(6)判断  $L$  是否满足文中提出的阈值条件. 若是,则当前深度层最优的 PU 模式为 SKIP/Merge,或者  $2N \times 2N$  模式中具有较小 RDcost 的模式,转至步骤 3;否则,转至步骤 2.

**步骤 2** 依次计算  $N \times 2N$  模式, $2N \times N$  模式,所有 AMP 模式,进行帧内预测. 转至步骤 3.

**步骤 3** 比较所有模式计算得出的 RDcost,选择最小 RDcost 对应的模式作为当前深度层最优的

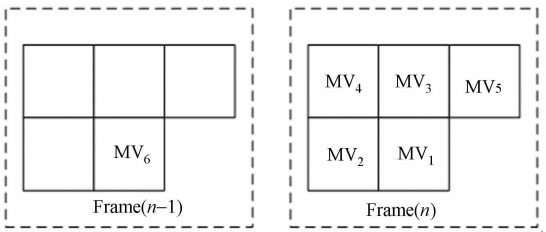


图 3 所有 MV 的位置关系  
Fig. 3 Spatial relationships of all the MVs

模式. 转至步骤 4.

**步骤 4** 判断当前深度是否为 3. 若是, 转至步骤 1; 否则, 转至步骤 5.

**步骤 5** 执行节 2.1 基于 RDcost 的 CU 分割提前终止算法, 即检查当前深度层最优的 RDcost 是否小于节 2.1 提出的阈值. 若是, 终止当前 CU 的进一步分割; 否则, 将当前 CU 分为 4 个子 CU, 对每个子 CU 转至步骤 1.

3 结果与讨论

实验采用 HM 10.1 官方参考软件, 测试的环境为具有 Intel(R) Core(TM)2 Quad CPUQ9400 @ 2.66 GHz, 4.0 GB 内存的计算机, 采用 VS2008 编译器. 量化参数为 22, 27, 32, 37, 对 HEVC 官方规定的 Class A~E 的 18 个测试序列<sup>[11]</sup>进行全部测试, 序列全部统一编码 50 帧. 快速运动估计(TZsearch)和快速编码器决策(FEN)全开(HM 10.1 默认打开).

采用 BDBR 与 BDPSNR 衡量<sup>[12]</sup>率失真性能, 两者分别表示在同样的客观质量下, 两种方法的码率节省情况, 以及在给定的同等码率下, 两种方法的 Y-PSNR 的差异.

时间改变量  $\Delta t$  定义为

$$\Delta t = \frac{1}{4} \sum_{i=1}^4 \frac{t_{\text{HM 10.1}}(\text{QP}_i) - t_{\text{pro}}(\text{QP}_i)}{t_{\text{HM 10.1}}(\text{QP}_i)} \times 100\%.$$

(7)

式(7)中:  $t_{\text{HM 10.1}}(\text{QP}_i)$ ,  $t_{\text{pro}}(\text{QP}_i)$  分别为原始 HM 10.1 算法及提出的算法在不同 QP 值下的编码时间.

选取文献[7]与文中算法进行对比. 两者在低时延编码配置模式<sup>[11]</sup>(LDP)编码模式下的实验结果, 如表 4 所示.

表 4 文中算法与文献[7]算法的实验结果比较

Tab. 4 Experimental results comparison between our overall algorithm and paper [7]

| 分辨率                    | 序列              | 文中算法   |             |               | 文献[7]算法 |             |               |
|------------------------|-----------------|--------|-------------|---------------|---------|-------------|---------------|
|                        |                 | BDBR/% | Y-BDPSNR/dB | $\Delta t$ /% | BDBR/%  | Y-BDPSNR/dB | $\Delta t$ /% |
| Class A<br>2 560×1 600 | Traffic         | 2.305  | −0.064      | −49.5         | 0.556   | −0.015      | −22.2         |
|                        | PeopleOnStreet  | 1.647  | −0.066      | −25.4         | 0.063   | −0.002      | −17.1         |
|                        | ParkScene       | 0.510  | −0.014      | −30.6         | 0.418   | −0.012      | −20.6         |
| Class B<br>1 920×1 080 | BasketballDrive | 0.586  | −0.010      | −36.2         | 0.406   | −0.008      | −20.1         |
|                        | Cactus          | 1.564  | −0.030      | −37.3         | 0.397   | −0.007      | −20.6         |
|                        | BQTerrace       | 0.374  | −0.009      | −28.2         | 0.144   | −0.003      | −22.2         |
|                        | Kimono1         | 0.581  | −0.018      | −41.2         | −0.271  | 0.008       | −19.3         |
| Class C<br>832×480     | PartyScene      | 0.998  | −0.040      | −20.3         | 0.361   | −0.014      | −16.6         |
|                        | RaceHorses      | 0.075  | −0.002      | −16.7         | 0.092   | −0.003      | −17.1         |
|                        | BQMall          | 2.033  | −0.075      | −38.7         | 0.233   | −0.008      | −21.3         |
|                        | BasketballDrill | 1.908  | −0.066      | −41.6         | 0.436   | −0.015      | −19.9         |
| Class D<br>416×240     | BasketballPass  | 2.435  | −0.104      | −43.8         | 0.543   | −0.024      | −17.6         |
|                        | BQSquare        | 1.328  | −0.048      | −26.8         | 0.194   | −0.008      | −14.3         |
|                        | RaceHorses      | 0.391  | −0.017      | −7.6          | 0.094   | −0.004      | −11.0         |
|                        | BlowingBubbles  | 1.452  | −0.051      | −17.7         | 0.354   | −0.013      | −13.0         |
| Class E<br>1 280×720   | FourPeople      | 3.081  | −0.093      | −73.6         | 0.508   | −0.012      | −28.7         |
|                        | Johnny          | 3.494  | −0.069      | −73.2         | 0.306   | −0.006      | −28.6         |
|                        | KristenAndSara  | 3.422  | −0.090      | −72.0         | 0.489   | −0.011      | −29.1         |
|                        | 平均值             | 1.566  | −0.048      | −37.8         | 0.296   | −0.010      | −20.0         |

由表 4 可知: 文中算法平均可以节省 37.8% 的编码时间, 同时只有 0.048 dB 的 Y-BDPSNR 损失, 以及 1.566% 的 BDBR 增加. 文中提出的 RDcost 和 MV 的阈值, 可以随着不同的应用场合进行灵活的控制. 一方面, 对于实时性要求较高的场合, 可以适当使阈值大些, 相应地编码复杂度也会减少得越多, 但率失真性能的损失也会越大; 另一方面, 对视频质量要求较高的场合, 可以设置让阈值适当小些, 率失真性能也会越好, 但相应地编码时间减少量也会越少. 另外, 文中算法对于平坦序列的优化效果非常可

观. 由表 4 可知: 对于 Class E 序列可以减少 70% 以上的编码时间, 同时率失真性能损失相当微小. 与文献[7]算法对比, 虽然文中算法的率失真性能损失略大, 但其编码时间减少量平均提高了 17.8%.

## 4 结束语

针对新一代视频编码标准 HEVC 帧间预测中高额的计算复杂度, 提出一种新颖的算法优化 HEVC 帧间预测中的模式选择过程. 基于 RDcost 提前终止 CU 的进一步分割进程; 基于已编码块的 MV 长度提前决定  $2N \times 2N$  分割. 实验结果表明: 文中算法可以节省 37.8% 编码时间, 同时保证与原始算法取得几乎相同的率失真性能. 揭示出 RDcost 与 CU 之间, 以及 MV 与 PU 模式之间暗含的关系, 且文中算法跟现阶段大部分已发表的帧间快速算法<sup>[4-6]</sup>是独立正交的, 可与其进一步融合, 更大地降低 HEVC 帧间编码的计算复杂度.

### 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.

[2] ZHANG Hao, MA Zan. Fast intra mode decision for high-efficiency video coding (HEVC)[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2014, 24(4): 660-668.

[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] CHOI K, PARK S H, JANG E S. Coding tree pruning based CU early termination[C]// Joint Collaborative Team on Video Coding of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 6th Meeting. Torino: [s. n.], 2011: 14-22.

[5] GWPEM R, LEE Y L. Early termination of CU encoding to reduce HEVC complexity[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2012, 95(7): 1215-1218.

[6] YANG J, KIM J, WON K, et al. Early SKIP detection for HEVC[EB/OL]. [2014-08-07]. <http://phenix.int-evry.fr/jct/>.

[7] LEE H S, 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(1-11).

[8] 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.

[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, 2013, 16(2): 559-564.

[10] 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.

[11] BOSSEN F. HM 10 common test conditions and software reference configurations[C]// Proceedings of the 12th JCT-VC Meeting. Geneva: [s. n.], 2013: 1-3.

[12] BJONTEGARD G. Calculation of average PSNR differences between RD-curves[EB/OL]. [2014-08-07]. [http://wftp3.itu.int/av-arch/video-site/0104\\_Aus/](http://wftp3.itu.int/av-arch/video-site/0104_Aus/).

(责任编辑: 钱筠      英文审校: 吴逢铁)