

## HEVC 帧间预测编码单元深度快速选择算法

张盛峰,汪 仟,黄 胜,肖 傲

(重庆邮电大学 光通信与网络重点实验室,重庆 400065)

**摘 要:**为降低高效视频编码(HEVC)帧间预测编码的计算复杂度,提出一种快速编码单元(CU)深度选择算法。利用当前 CU 与时空域相邻 CU 的深度信息,初步预测当前 CU 的深度范围,并根据当前 CU 与邻近 CU 的运动矢量信息对当前 CU 深度选择做进一步终止判断。实验结果表明,该算法与 HEVC 测试模型(HM16.0)相比,在低时延和随机访问 2 种编码配置下,码率分别仅增加 0.93% 和 1.01%,而编码时间减少了 29.1% 和 30.3%。  
**关键词:**高效视频编码;帧间预测;计算复杂度;编码单元;深度选择;运动矢量

**中文引用格式:**张盛峰,汪 仟,黄 胜,等. HEVC 帧间预测编码单元深度快速选择算法[J]. 计算机工程,2018,44(10):258-263.

**英文引用格式:**ZHANG Shengfeng,WANG Qian,HUANG Sheng,et al. Fast coding unit depth selection algorithm for inter-frame prediction of HEVC[J]. Computer Engineering,2018,44(10):258-263.

### Fast Coding Unit Depth Selection Algorithm for Inter-frame Prediction of HEVC

ZHANG Shengfeng,WANG Qian,HUANG Sheng,XIAO Ao

(Key Laboratory of Optical Communication and Network,

Chongqing University of Posts and Telecommunication,Chongqing 400065,China)

**[Abstract]** In order to reduce the computational complexity of High Efficiency Video Coding (HEVC) inter-frame prediction,a fast Coding Unit (CU) depth selection algorithm is proposed in this paper. Firstly, the spatio-temporal correlation of depth information among the current CU and adjacent CU is used to preliminary forecast the depth of the current CU. Secondly, the CU depth of current CU is judged further for stop based on the Motion Vector (MV) information of current CU and adjacent CU. The experimental results demonstrate that, compared with the HEVC test model(HM16.0), under two coding structures of low-delay and random-access, the proposed algorithm increases bit rate by 0.93% and 1.01% and saves coding time by 29.1% and 30.3%, respectively with the same objective quality.

**[Key words]** High Efficiency Video Coding (HEVC); inter-frame prediction; computational complexity; Coding Unit(CU); depth selection; Motion Vector(MV)

**DOI:**10.19678/j.issn.1000-3428.0048149

#### 0 概述

相比于 H. 264/AVC,新一代视频编码标准 HEVC<sup>[1]</sup>在相同质量视频的条件下可以减少 50% 左右的视频码流<sup>[2]</sup>,但编码计算复杂度也成倍增加,严重阻碍了其在多媒体领域的快速应用<sup>[3]</sup>。因此,如何减少帧间预测编码过程的计算复杂度是该领域内的热点问题。

为提高帧间预测过程编码效率并减少计算复杂度,人们进行了大量的研究。文献[4]先对时空邻近快速编码单元(Coding Unit, CU)的深度进行加权计算,预测当前 CU 的深度,然后通过分类讨

论预测当前 CU 的深度范围。但由于该算法采用固定的权值,因此对于不同视频序列误差较大。文献[5]利用当前 CU 与时空邻近 CU 的深度相关性预测当前 CU 的纹理复杂度,从而确定当前 CU 的深度范围。该算法虽然可以保证良好的编码性能,但降低的编码时间比较有限。文献[6]对邻近编码树单元(Coding Tree Unit, CTU)运动矢量信息提前终止当前 CU 的划分,但由于视频序列间的差异性,该方法对于不同序列降低的编码复杂度差异较大。文献[7]通过光流法计算金字塔运动发散(Pyramid Movement Diverges, PMD)的特征值,从而决定当前 CU 的划分,但未考虑运动矢量的相关

**基金项目:**国家自然科学基金(61371096,61571072);国家留学基金(201708500034);重庆市基础与前沿研究计划项目(cstc2015jcyjA4)。

**作者简介:**张盛峰(1977—),男,副教授,主研方向为视频编码、光网络、内容中心网络;汪 仟,硕士研究生;黄 胜,教授;肖 傲,硕士研究生。

**收稿日期:**2017-07-28 **修回日期:**2017-11-09 **E-mail:**1013165736@qq.com

性,导致编码性能较差。文献[8]利用视频图像的纹理信息确定当前 CU 的运动特征。文献[9]利用邻近 CU 的深度信息预测当前 CU 的深度信息。文献[10-11]利用时空域运动矢量的相关性确定图像的纹理信息和运动特征,从而加速 CU 的划分。文献[12]利用邻近 CU 率失真(RD)代价的相关性提出快速算法,加速帧间 CU 划分过程。

以上快速算法均利用视频的时间和空间相关性,都是从单方面考虑深度信息或运动矢量信息,未将两者相结合,导致编码性能较差。为进一步提高帧间预测的编码效率,本文结合当前 CU 与邻近 CU 深度信息和运动矢量信息的相关性,提出一种 HEVC 帧间预测 CU 深度快速选择算法。首先利用当前 CU 深度与邻近 CU 深度的时空相关性,找出相邻 CU 深度的最大值与最小值,预测当前 CU 的整体范围,然后根据相邻 CU 运动矢量信息对当前 CU 深度选择做进一步的终止判断。

### 1 HEVC 帧间预测的复杂度分析

H. 264 采用固定的编码单元  $16 \times 16$ , HEVC 采用灵活的四叉树编码结构,编码深度从 0 到 3 依次遍历选择。HEVC 编码结构如图 1 所示。

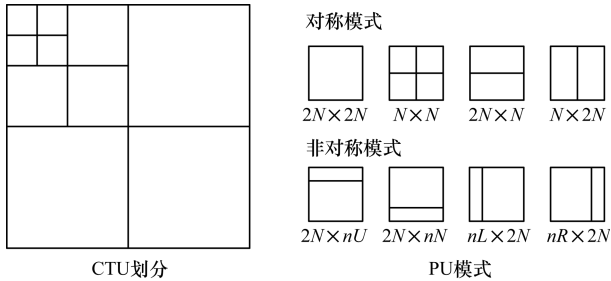


图 1 HEVC 编码结构

HEVC 编码过程如下:

1) 从尺寸为  $64 \times 64$  的 CU 开始,遍历所有 PU 模式并计算其率失真代价<sup>[13]</sup>,选择当前 CU 的最佳预测模式。PU 的 RD 代价计算为:

$$J_{mode} = (S_{luma} + w_{chroma} S_{chroma}) + \lambda_{mode} B_{mode} \quad (1)$$

其中,  $S_{luma}$  和  $S_{chroma}$  分别表示亮度和色度的平方误差和,  $w_{chroma}$  为权重因子,  $\lambda_{mode}$  为拉格朗日算子,  $B_{mode}$  表示编码当前模式时传输信息所需的比特数。

2) 将该 CU 进一步划分为 4 个  $32 \times 32$  的子 CU,深度为 1,同理可得每个子 CU 的 RD 代价,如此递归地划分下去,直至子 CU 的尺寸为  $8 \times 8$ ,即深度为 3,不再继续划分。

3) 从深度为 3 的 CU 开始进行 CU 块裁剪过程,如果 4 个  $8 \times 8$  大小的子 CU 的 RD 代价之和小于其对应  $16 \times 16$  大小的 CU 的 RD 代价,则选择  $8 \times 8$  的 CU 类型进行裁剪,否则,选择  $16 \times 16$  的 CU。如此

类推,直至深度为 0 的 CU,一个 CTU 深度选择过程结束<sup>[14]</sup>。以 CTU 尺寸为  $64 \times 64$  为例,最大编码深度为 3,仅对 Inter 模式的复杂度进行分析,当编码深度为 0 时,需要计算 7 次 RD 值;当编码深度为 1 时,则需要计算  $4 \times 7 = 28$  次 RD 值;当编码深度为 2 时,则需要计算  $4 \times 4 \times 7 = 112$  次 RD 值;当编码深度为 3 时,则需要计算  $4 \times 4 \times 4 \times 4 = 256$  次 RD 值( $8 \times 8$  尺寸的 CU 其 PU 模式只有对称模式)。综上,一个 CTU 需计算 RD 值  $7 + 28 + 112 + 256 = 403$  次,才能确定最佳深度和 PU 模式。一个 CTU 的 RD 代价值计算次数如表 1 所示。

表 1 一个 CTU 的 RD 值计算次数

CU 尺寸	CU 个数	帧间 PU 模式个数	RD 值计算次数
$64 \times 64$	1	7	7
$32 \times 32$	4	7	28
$16 \times 16$	16	7	112
$8 \times 8$	64	4	256

在一幅分辨率为  $1\,920 \times 1\,080$  的视频图像中,需要的 RD 值计算次数为  $1\,920 \times 1\,080 / 64 \times 403 \approx 204\,019$  次,由此可知,HEVC 深度选择的遍历需要大量计算,因此,如果能对其 CU 深度进行快速选择,可大幅降低编码复杂度,提高编码效率。

### 2 编码单元深度快速选择算法

#### 2.1 CTU 四叉树划分统计分析

为降低 HEVC 帧间预测编码过程的计算复杂度,本文选取 3 组不同视频序列(分辨率、运动矢量、纹理特征)来统计分析 CTU 四叉树的特点。其中: Basketballpass( $416 \times 240$ ) 序列中物体运动剧烈且纹理较复杂; Traffic( $2\,560 \times 1\,600$ ) 序列中物体运动适中; Johnny( $1\,280 \times 720$ ) 序列纹理相对简单。该过程在 HEVC 参考软件 HM-16.0 进行实现,量化参数(Quantization Parameter, QP)分别取 22、27、32 和 37,编码 100 帧(配置为 LDP)。

表 2 给出了 3 组视频序列的深度分布情况。从中可以看出, Depth0 ~ Depth3 平均所占比例为分别为 47.78%、27.84%、17.44% 和 6.94%。Johnny 序列有 71.13% 的 CU 选择 Depth0 作为最佳深度, Depth3 所占的比例只有 2.55%, 这表明运动量较小且纹理较简单序列的 CTU 划分深度相对较小。由于 CU 深度分布的不均匀,因此在 HEVC 的编码过程中,没有必要遍历所有的深度,可以跳过不可能的深度,从而减少帧间编码过程的计算复杂度。

表2 CU深度分布

%

视频序列	QP	Depth0 占比	Depth1 占比	Depth2 占比	Depth3 占比
Basketballpass	22	15.86	33.87	30.56	19.71
	27	19.37	35.71	31.24	13.68
	32	22.68	38.89	29.75	8.68
	37	25.74	41.54	25.87	6.85
	平均值	20.91	37.50	29.36	12.23
Traffic	22	32.45	31.28	24.12	12.15
	27	47.63	28.75	15.96	7.66
	32	57.89	26.84	12.14	3.13
	37	67.21	23.56	8.02	1.21
	平均值	51.30	27.61	15.06	6.03
Johnny	22	57.38	25.62	11.56	5.44
	27	69.56	19.86	7.68	2.90
	32	75.86	15.87	6.70	1.57
	37	81.73	12.34	5.66	0.27
	平均值	71.13	18.42	7.90	2.55
总体平均值		47.78	27.84	17.44	6.94

## 2.2 CU划分的提前终止算法

数字视频是若干有联系的图像在时间上连续播放形成的,在很短的时间内,视频相邻帧间中存在很强的时空相关性,因此,可以利用此相关性对当前CU的深度进行快速选择,减少CU深度划分的复杂度。

根据相邻帧间CU深度信息的时空相关性,本文对当前帧和参考帧中CU进行标记,如图2所示,其中,Cur-CU表示当前CU,Col-CU表示时间相邻的参考帧中与Cur-CU相同位置的CU,L-CU、U-CU、LU-CU、RU-CU分别表示位于当前CU左方、上方、左上方和右上方的时空邻近CU。

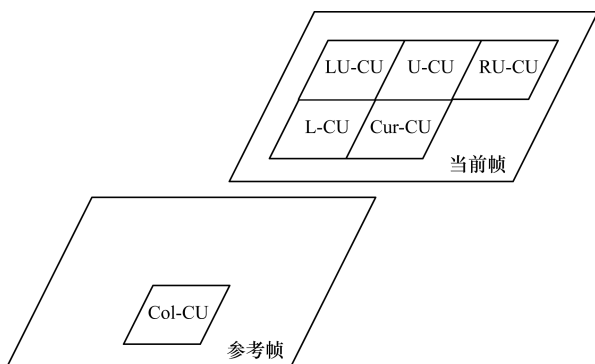


图2 相邻CU示意图

表3给出了时空邻近CU深度最大值 $D_{\max}$ 与深度最小值 $D_{\min}$ 差值小于等于2的统计情况。从中可以看出,3组不同视频序列时空邻近CU深度最大值与深度最小值的差值小于等于2所占比例平均值都达到95%以上。

表3  $D_{\max} - D_{\min} \leq 2$  情况占比

%

QP	Basketballpass	Traffic	Johnny
22	94.60	96.10	96.70
27	95.30	94.80	97.30
32	96.40	95.40	98.50
37	96.80	97.20	99.20
平均值	95.78	95.88	97.93

根据上述分析,本文提出的算法在Col-CU、L-CU、U-CU、LU-CU、RU-CU存在时,首先找出相邻CU中深度最大值 $D_{\max}$ 与深度最小值 $D_{\min}$ 的情况,若 $D = D_{\max} - D_{\min} \leq 2$ ,则初步预测当前CU的深度范围取值为 $[D_{\min}, D_{\max}]$ ,否则,正常编码。

在初步预测当前CU的深度范围取值为 $[D_{\min}, D_{\max}]$ ,从深度最小值 $D_{\min}$ 开始,本文利用与当前CU空域相邻的 $8 \times 8$ 大小CU块(如图3所示)运动矢量的差值MVD来判断当前块与邻近CU相关性强弱。首先,通过式(2)求出当前CU与邻近CU的平均MV,再利用式(3)求出运动矢量差值MVD,如果MVD小于或等于某一阈值 $T$ ,表示当前CU与周围邻近CU具有相同的运动信息,则当前CU与邻近CU相关性较强,当前CU的最佳深度值可以借鉴周围邻块的深度信息,停止对当前CU的划分;如果MVD大于某一阈值 $T$ ,表示当前CU与邻近CU相关性较弱,则将当前CU划分为4个子CU,深度 $D = D_{\min} + 1$ 。用以上方法计算4个子CU与周围邻近CU的运动矢量差值,若小于等于阈值 $T$ ,则深度 $D_{\min} + 1$ 为当前最佳划分深度,否则 $D_{\max}$ 为当前最佳划分深度,编码结束。

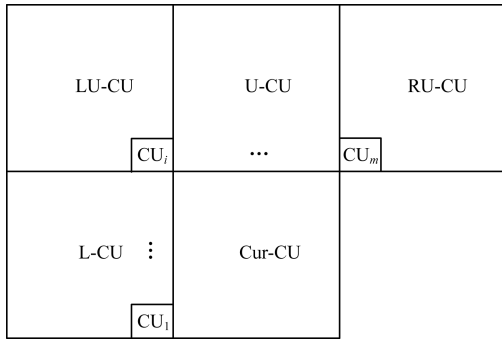


图 3 当前 CU 与空域相邻 CU

平均运动矢量计算<sup>[15]</sup>公式为:

$$MV = \frac{1}{2N \times 2N} \sum_{i=0}^{2N-1} \sum_{j=0}^{2N-1} \sqrt{MV_x(i,j)^2 + MV_y(i,j)^2} \quad (2)$$

其中,  $2N \times 2N$  表示 CU 的大小,  $MV_x(i,j)$  和  $MV_y(i,j)$  分别表示 CU 在  $(i,j)$  处的运动矢量的  $x$  和  $y$  分量。运动矢量差值计算公式为:

$$MVD = \frac{1}{m} \sum_{i=1}^m |MV_{cur} - MV_i| \quad (3)$$

其中,  $m$  表示与当前 CU 相邻  $8 \times 8$  大小 CU 块的个数,  $MV_{cur}$  表示当前 CU 的平均运动矢量,  $MV_i$  表示  $CU_i (i = 1, 2, \dots, m)$  的平均运动矢量。

根据多次实验观察得到合适的门限值  $T = 0.2$ , 当  $MVD > T$  时, 表示当前 CU 与邻近 CU 相关性较弱, 则将当前 CU 划分成 4 个子 CU, 进行下一步判断; 当  $MVD \leq T$  时, 表示当前 CU 与邻近 CU 具有相近的运动矢量, 相关性较强, 停止划分。

CU 深度快速决策算法整体流程如图 4 所示。

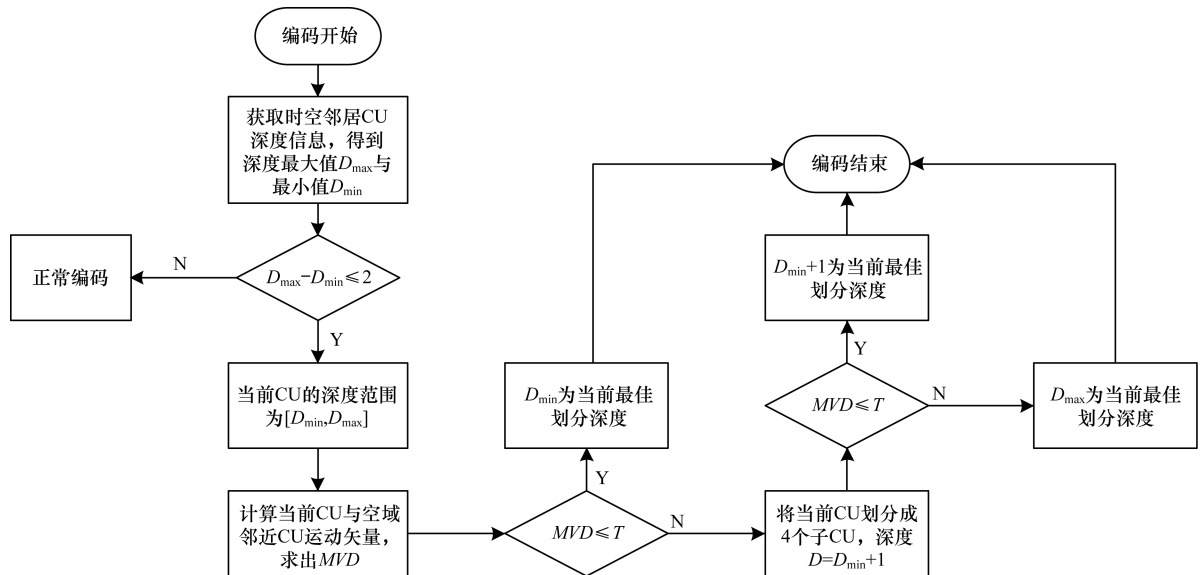


图 4 CU 深度快速算法流程

### 3 实验与结果分析

本文算法在 HEVC 参考软件 HM-16.0<sup>[16]</sup> 进行实现, 主要参数为: 最大编码单元尺寸为  $64 \times 64$  (深度  $d = 3$ ), 编码配置为低时延和随机访问, 量化参数 (QP) 为 22、27、32 和 37, 编码 100 帧。仿真结果通过相同质量下的码率  $BDBR$  和减少的编码时间  $TS$  2 种指标进行评价, 计算公式如下:

$$\Delta BDBR = \frac{BDBR - BDBR_{HM16.0}}{BDBR_{HM16.0}} \times 100\% \quad (4)$$

$$\Delta TS = \frac{T - T_{HM16.0}}{T_{HM16.0}} \times 100\% \quad (5)$$

根据以上的设定进行实验, 本文算法与标准算法的对比结果如表 4 所示。从中可以看出, 与 HM16.0 标准算法相比, 本文算法在低时延和随机访问配置下  $BDBR$  分别提高了 0.93% 和 1.01%, 同时编码时间平均减少了 29.1% 和 30.3%。对于低时

延配置下 Blowing Bubbles 序列, 本文算法效果相对其他视频序列要好, 这是因为视频序列纹理相当简单并且视频图像中的背景区域相当单一, 本文算法获得了更好的计算性能。对于 Race Horses 和 Party Scene 视频序列, 由于序列中运动剧烈, 且背景区域也在进行相对平移, 复杂运动区域所占比例较大, 因此本文算法的效果较差。

为了评估本文算法的效果, 将其与文献[4]算法进行对比, 结果如表 5 所示, 其中, 斜杠前为  $\Delta BDBR$ , 斜杠后为  $\Delta TS$ 。从中可以看出: 在 LDP 和 RA 配置下, 文献[4]算法分别减少了 27.6% 和 29.3% 的编码时间; 在 LDP 配置下, 文献[4]算法对视频序列 Race Horses 和 BQ Square 的编码时间仅减少了 14.4% 和 15.9%。这表明文献[4]提出的算法对于背景相对简单和运动剧烈的视频序列效果不是很好, 本文算法对于视频序列纹理相对简

单或者运动剧烈的视频序列效果相比文献[4]提出的算法要好。在 LDP 和 RA 配置下,本文算法分

别平均降低了 29.1% 和 30.3% 的编码时间,而  $BDBR$  损失较小,与文献[4]算法相近。

表 4 本文算法与 HM16.0 标准算法的性能比较

%

视频序列	图片尺寸	低时延配置		随机访问配置	
		$\Delta BDBR$	$\Delta TS$	$\Delta BDBR$	$\Delta TS$
Traffic	2 560 × 1 600	—	—	1.20	-32.4
People On Street	2 560 × 1 600	—	—	1.09	-33.4
Kimono1	1 920 × 1 088	0.64	-29.6	1.02	-33.6
Park Scene	1 920 × 1 088	0.72	-35.8	1.02	-35.8
Tennis	1 920 × 1 088	1.02	-31.4	0.96	-33.0
Basketball Drive	1 920 × 1 088	0.82	-34.5	0.80	-34.2
Cactus	1 920 × 1 088	0.85	-29.5	0.82	-30.8
BQ Terrace	1 920 × 1 088	1.11	-32.6	1.04	-32.2
Race Horses	832 × 480	1.32	-24.5	1.58	-24.1
Basketball Drill	832 × 480	1.25	-29.4	1.18	-31.7
BQ Mall	832 × 480	1.46	-28.6	1.77	-28.4
Mobisode2	832 × 480	0.84	-32.8	0.82	-33.9
Party Scene	832 × 480	0.76	-22.5	0.72	-26.0
Race Horses	416 × 240	0.83	-15.4	0.79	-17.0
Basketball Pass	416 × 240	0.96	-27.6	0.89	-26.8
Blowing Bubbles	416 × 240	0.65	-36.4	0.62	-35.7
BQ Square	416 × 240	0.82	-27.2	0.78	-26.5
平均值		0.93	-29.1	1.01	-30.3

表 5 本文算法与文献[4]算法的性能比较

%

视频序列	低时延配置		随机访问配置	
	文献[4]算法	本文算法	文献[4]算法	本文算法
Traffic	—	—	1.38/-39.4	1.20/-32.4
People On Street	—	—	1.30/-24.6	1.09/-33.4
Kimono1	0.40/-35.6	0.64/-29.6	0.53/-38.7	1.02/-33.6
Park Scene	0.95/-32.5	0.72/-35.8	1.24/-35.5	1.02/-35.8
Tennis	0.65/-34.1	1.02/-31.4	1.21/-32.6	0.96/-33.0
Basketball Drive	1.11/-35.6	0.82/-34.5	1.25/-36.5	0.80/-34.2
Cactus	0.84/-33.7	0.85/-29.5	1.02/-35.7	0.82/-30.8
BQ Terrace	0.87/-35.2	1.11/-32.6	0.86/-36.8	1.04/-32.2
Race Horses	1.10/-20.0	1.32/-24.5	1.68/-20.6	1.58/-24.1
Basketball Drill	1.40/-30.8	1.25/-29.4	1.16/-30.5	1.18/-31.7
BQ Mall	2.13/-26.1	1.46/-28.6	2.74/-28.8	1.77/-28.4
Mobisode2	1.08/-44.3	0.84/-32.8	0.82/-42.6	0.82/-33.9
Party Scene	0.27/-20.6	0.76/-22.5	0.66/-24.6	0.72/-26.0
Race Horses	0.51/-14.4	0.83/-15.4	0.70/-13.5	0.79/-17.0
Basketball Pass	1.93/-18.6	0.96/-27.6	1.57/-18.6	0.89/-26.8
Blowing Bubbles	0.69/-16.7	0.65/-36.4	0.82/-19.1	0.62/-35.7
BQ Square	1.11/-15.9	0.82/-27.2	0.85/-20.1	0.78/-26.5
平均值	1.01/-27.6	0.93/-29.1	1.16/-29.3	1.01/-30.3

#### 4 结束语

为降低 HEVC 帧间预测的编码计算复杂度,本文根据视频帧间时空域的相关性,首先利用时空域邻近 CU 的深度信息初步预测当前 CU 的深度范围,然后利用周围邻块的运动矢量求出运动矢量差值,通过设定阈值进一步判定最小深度是否为当前 CU 的最佳划分深度。实验结果表明,本文算法可有效加快编码速度,并且能够降低帧间预测编码过程中的计算复杂度。如何充分利用当前 CU 与相邻 CU 之间的相关性来改善本文算法的性能,将是下一步的研究方向。

#### 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 ] CORREA G, ASSUNCAO P, AGOSTINI L, et al. Performance and computational complexity assessment of high-efficiency video encoders[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22(12): 1899-1909.
- [ 3 ] 刘 昱,胡晓爽,段继忠. 新一代视频编码技术 HEVC 算法分析及比较[J]. 电视技术, 2012, 36(20): 45-49.
- [ 4 ] SHEN L, LIU Z, ZHANG X, et al. An effective CU size decision method for HEVC encoders [ J ]. IEEE Transactions on Multimedia, 2013, 15(2): 465-470.
- [ 5 ] 周承涛,田 翔,陈耀武. HEVC 编码单元尺寸快速选择算法[J]. 浙江大学学报(工学版), 2014, 48(8): 1451-1460.
- [ 6 ] GARCIA R, KALVA H. HEVC inter-frame skip enhancement at low bit rates[C]//Proceedings of IEEE International Conference on Consumer Electronics. Washington D. C., USA: IEEE Press, 2014: 59-60.
- [ 7 ] XIONG J, LI H, WU Q, et al. A fast HEVC inter CU selection method based on pyramid motion divergence[J]. IEEE Transactions on Multimedia, 2014, 16(2): 559-564.
- [ 8 ] ZHANG Q, YUAN C. Fast inter-frame prediction algorithm of HEVC based on graphic information[J]. Journal of Frontiers of Computer Science and Technology, 2014, 8(5): 537-542.
- [ 9 ] KIM K Y, PARK G H. Fast encoding algorithm based on depth of coding-unit for high efficiency video coding[J]. Optical Engineering, 2012, 51(6): 7402.
- [ 10 ] SHAN N, ZHOU W, DUAN Z, et al. A fast coding unit depth decision algorithm for HEVC inter prediction[C]//Proceedings of International Conference on Frontier of Computer Science and Technology. Washington D. C., USA: IEEE Press, 2015: 316-320.
- [ 11 ] AHN S, LEE B, KIM M. A novel fast CU encoding scheme based on spatiotemporal encoding parameters for HEVC inter coding [ J ]. IEEE Transactions on Circuits and Systems for Video Technology, 2015, 25(3): 422-435.
- [ 12 ] LEE J, KIM S, LIM K, et al. A fast CU size decision algorithm for HEVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2015, 25(3): 411-421.
- [ 13 ] SHEN L, ZHANG Z, LIU Z. 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.
- [ 14 ] TAN H L, CHI C K, RAHARDJA S. Fast coding quadtree decisions using prediction residuals statistics for high efficiency video coding ( HEVC ) [ J ]. IEEE Transactions on Broadcasting, 2016, 62(1): 128-133.
- [ 15 ] 李静敏,杨 静. 基于 HEVC 的快速帧间预测算法[J]. 电视技术, 2016, 40(7): 1-5.
- [ 16 ] JCT-VC of ISO/IEC and ITU-T. HM reference software 16.0 [ EB/OL ]. ( 2014-10-30 ). [ 2017-03-10 ]. [https://hevc.hhi.fraunhofer.de/Svn/svn\\_HEVCSoftware/tags/HM-16.0/](https://hevc.hhi.fraunhofer.de/Svn/svn_HEVCSoftware/tags/HM-16.0/).

编辑 金胡考