

# 基于决策树的 HEVC 编码单元划分算法

岑跃峰<sup>1</sup> 王万良<sup>1</sup> 姚信威<sup>1</sup> 王超超<sup>1</sup> 潘铁强<sup>1,2</sup>

(浙江工业大学计算机科学与技术学院 杭州 310023)<sup>1</sup>

(衢州学院电气与信息工程学院 衢州 324000)<sup>2</sup>

**摘要** 为了有效降低高效视频编码标准 HEVC 编码过程中的计算复杂度,提出了一种基于决策树的编码单元划分算法。该算法将编码单元划分问题归为分类问题,提取编码单元的划分信息到决策树模型中进行学习,得到决策树分类器。利用分类器对满足分类条件的编码单元进行划分,跳过了率失真计算,从而降低了视频编码的计算复杂度。实验结果表明,本算法在保证视频质量的同时有效降低了编码计算复杂度。

**关键词** HEVC,计算复杂度,决策树,率失真,信噪比

**中图分类号** TP391.41 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.4.063

## Decision Tree Based Coding Unit Splitting Algorithm for HEVC

CEN Yue-feng<sup>1</sup> WANG Wan-liang<sup>1</sup> YAO Xin-wei<sup>1</sup> WANG Chao-chao<sup>1</sup> PAN Tie-qiang<sup>1,2</sup>

(College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China)<sup>1</sup>

(College of Electrical and Information Engineering, Quzhou University, Quzhou 324000, China)<sup>2</sup>

**Abstract** To reduce the computational complexity of high efficient video coding (HEVC), a decision tree based coding unit (CU) splitting algorithm was proposed. The splitting of the CU is seen as a classification problem. Furthermore, splitting information extracted from CUs is added to the decision tree model for principle learning. A decision tree based classifier is obtained after the principle learning. Then the classifier is used to determine the splitting of the CU if the classification condition is satisfied. Thus, the computation of rate distortion (RD) cost is skipped, and the encoding computational complexity is reduced. Experimental results demonstrate that the proposed mechanism achieves significant reduction of the encoding computational complexity, while still maintaining high video coding quality.

**Keywords** HEVC, Computational complexity, Decision tree, Rate distortion, PSNR

## 1 引言

随着数字技术和网络技术的快速发展,人们对视频质量提出了更高的要求,高清和超高清视频也随着人们的需求快速发展起来<sup>[1,2]</sup>,这使得网络中视频数据量不断增加。如何在有限的带宽下保证高清视频的传输,这对视频压缩技术提出了更高的要求,传统的视频压缩技术 H. 264/AVC 也因此面临着新的挑战<sup>[3]</sup>。

国际电信联盟 (ITU-T) 与 ISO/IEC 动态图像专家组 (MPEG) 共同成立了视频编码联合小组 (JCT-VC), 并制定了下一代高效视频编码标准 (High Efficient Video Coding, HEVC)<sup>[3]</sup>。与之前的 H. 264/AVC 视频压缩标准相比, HEVC 引入了一些新的技术,在保证视频质量的同时,可将编码效率提升一倍<sup>[4,5]</sup>。其主要技术包括四叉树结构的编码单元 (Coding Unit, CU) 技术、帧内预测技术、编码单元合并技术等。其中基于四叉树结构的 CU 技术对编码效率和视频质量的提升具有重要意义。但是,其复杂的结构也使 HEVC 编

码具有极高的计算复杂度,这对现有的编码设备提出了极大的挑战,而且编码时间长会影响视频实时传输。因此,如何降低 HEVC 编码计算复杂度是当前的研究热点。

近年来,降低 HEVC 编码计算复杂度的研究大致可以分为以下 3 类:快速 CU 划分决策、快速帧内预测模式决策和快速帧间预测模式决策。

文献[6-11]主要介绍了快速 CU 划分的方法。Shen<sup>[6]</sup>提出了自适应的 CU 深度区间机制,当视频帧划分到最大编码单元 (Largest Coding Unit, LCU) 时,提取相邻 LCU 深度来预测当前 CU 的深度区间,舍去区间外的率失真计算。文献[7,8]分别利用相邻 CU 的 Skip 模式和残差来提前终止 CU 的划分。但是,这些方法没有考虑其对视频质量造成的影响。文献[9]采用相邻 CU 的纹理特性来预测当前 CU 深度,同时结合了帧内预测模式决策,取得了较好的实验效果。文献[10,11]提出用机器学习的方法实现 CU 划分,分别采用贝叶斯和支持向量机方法,开始尝试使用机器学习的方法解决 CU 划分问题,舍去了率失真计算,降低了编码的计算复杂

到稿日期:2015-03-19 返修日期:2015-06-30 本文受国家自然科学基金资助项目(61402414,61379123),“十二五”国家科技支撑计划课题资助项目(2012BAD10B01),浙江省教育厅科研项目资助项目(Y201431815),浙江省基金资助项目(LQ14F020005)资助。

岑跃峰(1986—),男,博士,主要研究方向为无线传感网、高效视频编码,E-mail:cyf\_zjut@hotmail.com;王万良(1957—),男,教授,博士生导师,主要研究方向为网络控制、人工智能,E-mail:wwl@zjut.edu.cn;姚信威(1986—),男,博士,副教授,主要研究方向为无线传感器网络,E-mail:xwyao@zjut.edu.cn;王超超(1992—),男,硕士,主要研究方向为高效视频编码;潘铁强(1974—),男,博士,主要研究方向为高效视频编码,E-mail:ptq7442@163.com。

度。但是在进行编码之前,基于支持向量机和贝叶斯决策的方法在对特征进行离线学习时计算仍然比较复杂。

文献[12-15]主要介绍了快速帧内预测模式决策的方法。Zhang等<sup>[12]</sup>提出对帧内预测模式先进行粗选,挑选最大可能的预测模式而不是遍历所有的预测模式,从而减少率失真计算。文献[13-15]提出将帧内预测模式分成若干预测模式候选集,根据相邻样本内的统计特性来选择合适的预测模式候选集。这些特征主要利用的是视频帧的空间相关性信息,包括相邻或上一层CU内的预测模式、边界的纹理特性等。通过利用这些统计特性缩小预测模式的范围,不再需要遍历所有的35种帧内预测模式,从而降低编码计算复杂度。

文献[16-19]介绍了快速帧内预测模式决策的方法,利用的信息与帧内预测模式决策相似。值得注意的是,CU的划分与帧内、帧间预测模式的决策是可以结合的,可进一步降低HEVC的编码计算复杂度。

本文提出了一种基于决策树的HEVC编码单元划分算法,即提取CU的划分信息并将其加载到决策树模型中进行学习,得到决策树分类器;在满足分类条件的基础上,通过分类器实现CU的划分决策,跳过了CU的率失真计算,有效降低了编码过程中的计算复杂度。本文还提供了本算法和原有的HEVC方法以及其它CU划分改进算法对不同分辨率以及场景的视频进行编码之后的实验比较结果,分析了视频质量以及计算复杂度等各项性能指标。

## 2 HEVC 编码单元划分

CU是HEVC编码的基本单元,其基于四叉树的组成结构相比于H.264/AVC具有重大的改进。H.264/AVC采用的是固定的 $16 \times 16$ 像素的编码单元,而HEVC采用了更加灵活有效的四叉树结构,编码单元尺寸可以为 $64 \times 64$ 像素、 $32 \times 32$ 像素、 $16 \times 16$ 像素和 $8 \times 8$ 像素,如图1所示。这也是HEVC作为新一代视频压缩标准,与之前的视频压缩标准相比最显著的改进技术之一。

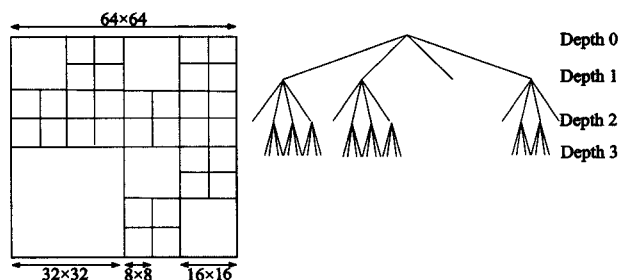


图1 四叉树结构的编码单元组成

四叉树结构的CU设计能够更好地提高视频的信噪比以及编码压缩效率。针对视频比较平滑或者背景比较单调的场景,例如草地,采用大尺寸的CU可以提高编码压缩效率;针对视频中物体变化较快或者事物存在突变的场景,例如比赛运动或者物体的边界,采用小尺寸的CU能够使视频中的物体更加精细,提高视频的信噪比。

在编码过程中,视频帧首先将会被分隔成许多 $64 \times 64$ 像素的编码单元,即LCU。接着每个LCU将被等分成4个 $32 \times 32$ 像素的CU,每个 $32 \times 32$ 像素的CU又会被划分成4个 $16 \times 16$ 像素的CU。依此类推,直至被分到 $8 \times 8$ 像素的CU为止。定义在四叉树结构中CU的深度根据其尺寸从 $64 \times 64$ 像素到 $8 \times 8$ 像素依次为深度0到深度3。

为了得到最佳的CU尺寸,HEVC将遍历所有的CU尺

寸,然后通过率失真(Rate Distortion cost, RD cost)计算来求解,采用的方法是拉格朗日乘数法<sup>[6]</sup>,具体公式如下:

$$J = B + \lambda \cdot R \quad (1)$$

其中, $J$ 是最佳编码单元, $B$ 是用于编码单元CU编码所需要的位, $\lambda$ 是拉格朗日系数, $R$ 是当前CU与匹配CU的残差。

在CU划分的过程中,帧内与帧间预测模式也将确定下来。每个LCU的划分共需要执行85次率失真计算,最终根据率失真最小来得到最佳CU尺寸。此计算过程具有极大的计算复杂度。研究表明,率失真计算占了整个编码时间的40%<sup>[21]</sup>。

实际上,没有必要穷尽所有可能的CU尺寸,视频帧内存在着空间相关性,相邻CU之间的划分信息也存在相关性。因此,本文将CU划分归为分类问题,充分利用相邻CU的划分情况,通过决策树算法来辅助决策当前CU的划分。

## 3 基于决策树的CU划分算法

决策树是一种简单、实用的分类方法,给定一群样本,每个样本都有一组属性和一个类别,通过学习得到一个分类器。分类器能够对新出现的对象给出正确的分类。在本文中,通过将视频帧内CU的分布输入到决策树模型中进行学习,得到分类器并将其加载到编码过程中,以实现CU的划分。

### 3.1 决策树建模

ID3算法是决策树学习方法中最具影响和最典型的算法。它引入了信息论中的熵来计算信息增益,作为判断能力的度量。

设 $S$ 是进行CU划分训练的 $s$ 个训练数据样本的集合。根据划分需求,最终将样本数据分为 $m$ 个不同的类 $C_i, i=1, 2, \dots, m$ 。本文中 $m=2$ ,因为根据需求训练样本最终被分为在当前编码深度CU进行划分和不进行划分两类。 $s_i$ 是类 $C_i$ 中的样本数。在给定的训练样本中,其分类所需的期望信息如下:

$$I(s_1, s_2) = - \sum_{i=1}^2 p_i \log_2 p_i \quad (2)$$

其中, $p_i$ 是抽取样本中属于 $C_i$ 的概率,通常用 $s_i/s$ 表示。

采用ID3决策树建模需要计算样本集合中不同属性的期望信息。由于视频帧内存在较强的空间相关性,其CU划分也存在较强的相关性,因此,本文将多个相邻CU的划分信息作为数据样本属性。采用同一视频帧内当前(Current)CU的左(Left)、左上(L-U)、上(Up)、右上(R-U)的CU作为相邻CU,如图2所示。将以上相邻CU的划分信息分别定义为样本属性A、属性B、属性C、属性D。

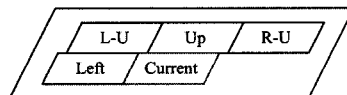


图2 相邻编码单元

属性A具有 $v$ 个不同值 $\{a_1, a_2, \dots, a_v\}$ ,将 $S$ 划分为 $v$ 个不同的子集 $\{S_1, S_2, \dots, S_v\}$ , $S_j$ 中样本在属性A上具有相同的值 $a_j, j=1, 2, \dots, v$ 。在本文中,各个属性中的 $v$ 的值均为2,即该相邻CU在当前CU深度只有划分和不划分两种情况。 $s_{i,j}$ 是子集 $S_j$ 中类 $C_i$ 的样本数,由A划分成子集的期望信息如下:

$$E(A) = \sum_{j=1}^2 \frac{s_{1,j} + s_{2,j}}{s} I(s_{1,j}, s_{2,j}) \quad (3)$$

其中, $\frac{s_{1,j} + s_{2,j}}{s}$ 为第 $j$ 个子集的权值,即所有子集中属性取 $a_j$

值的样本数之和除以  $S$  集中的样本总数。 $E(A)$  值越小,其子集划分越好。给定子集  $S_j$  的期望信息如下:

$$I(s_{1,j}, s_{2,j}) = - \sum_{i=1}^2 p_{i,j} \log_2 p_{i,j} \quad (4)$$

其中,  $p_{i,j} = s_{i,j} / s_j$  是  $S_j$  中样本属于  $C_i$  的概率。

属性  $A$  上分支获取的信息增益为:

$$Gain(A) = I(s_1, s_2) - E(A) \quad (5)$$

$Gain(A)$  为根据属性  $A$  取值进行样本集合划分所获得的熵的减少量。

同理,属性  $B$ 、属性  $C$ 、属性  $D$  分别通过计算可以得到该属性的信息增益,信息增益值最大的被选为测试属性,创建一个节点。引出的分支检查样本是否属于同一类,如果属于同一类,根据算法在该分支端点创造树叶节点;否则,该分支计算其他属性的信息增益,继续向下划分分支。

在建立决策树模型之后,不失一般性地从 HEVC 提供的不同分辨率和不同场景的测试视频中分别随机选取 5 帧视频帧,视频序列如第 4 节中表 1 所列;提取视频帧内每一个 LCU 内的 CU 深度分布信息;根据 CU 深度分布信息获取在当前深度的 CU 划分与相邻 CU 划分的相关性,将其作为样本数据输入到决策树模型进行学习。最终得到能对 CU 划分进行决策的分类器。

### 3.2 CU 划分方法

对决策树模型进行训练之后,得到的决策树分类器就可加载到视频编码中去,对符合判断条件的 CU 进行划分,具体步骤如图 3 所示。

首先,视频帧将被划分成许多的 LCU,对应的 CU 深度  $Depth$  为 0。接着获取当前 CU 的相邻 CU 划分信息。在决策树分类器对 CU 划分决策之前需检验是否存在判断属性,即是否存在 3.1 节中提到的 4 个相邻 CU 划分信息,这是因为处于视频帧边界的 CU 存在相邻 CU 划分信息获取不完整的情况。

如果存在判断属性,则加载决策树分类器来判断是否需要进一步划分。如果决策树得出的结果是需要进一步划分 (Split) 且当前  $Depth$  小于 3,则将当前 CU 划分成 4 个相同的 CU,舍去当前  $Depth$  的率失真计算,  $Depth$  值加 1,并进入下一轮 CU 的划分判断;如果决策树模型判断当前 CU 无需进一步划分或者  $Depth$  的值等于 3,则直接返回最终编码单元。

如果获取判断属性失败,这种情况主要针对的是处于视频帧的边界上的 CU,则仍需要执行当前  $Depth$  的率失真计算,如果  $Depth$  小于 3,则划分成 4 个相同的 CU,  $Depth$  值加 1,并进入下一轮 CU 的划分判断。如果  $Depth$  等于 3,则返回最终编码 CU。

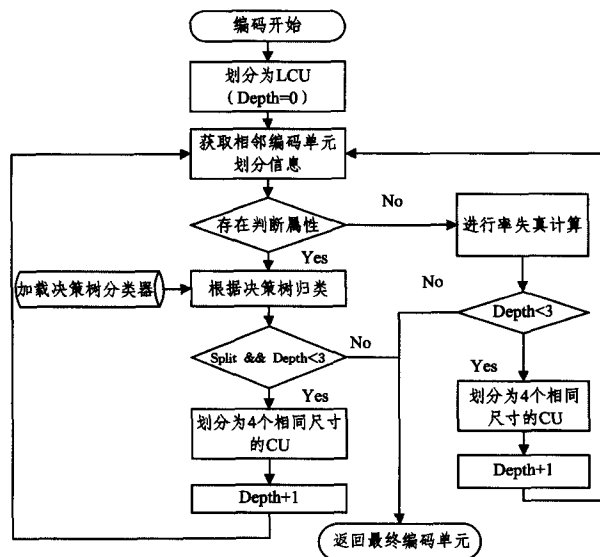


图 3 算法流程图

## 4 实验结果与分析

采用 Visual Studio 作为编程工具,在 Windows 8 环境下进行实验,计算机采用 Intel Ci5-3210M 双核 CPU, 2.5GHz 主频, 4GB 内存。实验采用的视频序列由视频编码联合组 JCT-VC 推荐,视频序列包含了不同分辨率和不同场景,如表 1 所列。实验设置编码环境为“intra-main”,每个视频序列在 4 种视频质量参数 QP 值(分别为 22、27、32、37)下进行实验。编码帧速率设置为 50fps。为了更好地分析本文算法的性能,提供了另外两种 CU 划分改进算法以及本文算法分别与 HEVC 提供的测试程序 HM 进行比较的结果。

表 1 给出了本文算法以及另外两种 CU 划分改进算法(即 ADRD 和 ES-SCUME<sup>[6]</sup>)分别对不同分辨率和不同场景的视频进行编码与 HM 编码比较的实验结果。其中计算复杂度通过编码时间来反映。 $\Delta T$  表示与 HM 相比算法编码时间变化的百分比,从而反映出与 HM 编码相比算法计算复杂度的变化。同时,信噪比是反映图像质量高低的一个重要指标;比特率是衡量视频压缩性能的一个重要指标。BDPSNR 和 BDBitrate 是反映两个视频之间信噪比差异和比特率差异的重要指标<sup>[22]</sup>,常用于视频质量比较。通过这两个指标评价算法的引入对视频质量造成的影响。分别将相应算法与 HM 编码之后的信噪比和比特率进行比较,得到的统计结果如表 1 所列。

表 1 实验统计结果

视频类型	序列名称	分辨率	ADRD			ES-SCUME			本文算法		
			BDBitrate (%)	BDPSNR (dB)	$\Delta T$ (%)	BDBitrate (%)	BDPSNR (dB)	$\Delta T$ (%)	BDBitrate (%)	BDPSNR (dB)	$\Delta T$ (%)
A	PeopleOnStreet	2560×1600	/	/	/	/	/	/	2.12	-0.12	-38
A	Traffic	2560×1600	/	/	/	/	/	/	1.67	-0.09	-35
B	BasketballDrive	1920×1080	1.11	-0.02	-36	0.12	0.00	-20	1.96	-0.06	-33
B	Cactus	1920×1080	0.84	-0.02	-34	0.17	0.00	-27	0.94	-0.04	-35
C	BQMall	832×480	2.13	-0.08	-26	0.64	-0.02	-34	0.55	-0.04	-32
C	RaceHorses	832×480	1.10	-0.04	-20	0.53	-0.02	-17	0.49	-0.03	-33
D	BasketballPass	416×240	1.93	-0.08	-19	0.50	-0.02	-26	0.82	-0.05	-30
D	RaceHorses	416×240	0.51	-0.02	-14	0.54	-0.02	-15	0.33	-0.03	-28
E	Vidyo1	1280×720	1.20	0.04	-42	0.03	0.00	-42	0.72	-0.03	-34
E	Vidyo3	1280×720	1.11	-0.03	-39	0.09	0.00	-36	2.70	-0.11	-33
平均值			1.24	-0.03	-29	0.33	-0.01	-27	1.23	-0.06	-33

实验结果显示本文算法与 HM 相比,计算复杂度显著降低,平均下降了 33%,均高于另外两种 CU 划分改进算法。特别地,对于高分辨率的视频序列如“Peopleonstreet”和“Traffic”,计算复杂度降低了 35%以上。进一步发现本文算法针对不同分辨率的视频序列的计算复杂度的降低存在如下现象:视频的分辨率越高,算法对其计算的复杂度降低越明显。例如,对分辨率为 2560×1600 像素的视频序列“PeopleOnStreet”算法复杂度降低了 38%,而对分辨率为 416×240 像素的视频序列“BasketballPass”算法复杂度只降低了 28%。这是因为在本文算法中需要通过相邻 CU 的划分信息来辅助决策当前 CU 的划分,在高分辨率的视频帧内,能获取相邻 CU 的 CU 占有的比例更高,更多的 CU 存在判断属性,从而跳过了率失真计算,因此能更大程度地降低计算复杂度。同时观察发现,本文算法对同一分辨率下不同场景的视频进行编码时与 HM 编码相比降低的计算复杂度并不明显。因此,本文算法能够适用于不同场景的视频。与另外两种 CU 划分改进算法相比,本文算法能够更加显著地降低计算复杂度,而且对不同的视频具有较好的适用性。

本文同时提供了本文算法以及另外两种 CU 划分改进算法分别与 HM 编码在信噪比和比特率的变化上的比较。实验显示与 HM 相比,本文算法的信噪比平均降低了 0.06dB。这是因为利用本算法实现划分的 CU 由于跳过了率失真计算,返回的 CU 并不一定是最佳尺寸,因此会对视频信噪比产生负面影响。而另外两种 CU 划分改进算法分别平均降低了 0.03dB 和 0.01dB。同时,与 HM 相比,本文算法平均额外引入了 1.23%的比特率。由于本文算法实现 CU 划分时舍去了率失真计算,就舍去了最小化比特率的限制条件,因此本文算法以及另外两种 CU 划分改进算法的比特率都略有增加。在实际应用中,相比于在计算复杂度的降低上取得的效果,这样的比特率增加与信噪比降低是可以忽略的。为了更好地展现本文算法对视频信噪比和比特率的影响,图 4 和图 5 给出了本文算法和 HM 分别在不同的 QP 值下对分辨率为 2560×1600 像素的视频序列“PeopleOnStreet”进行编码的比特率和信噪比。从图中可以观察到两者的差异很小,从而更加说明本文所提算法对视频质量的影响较小。

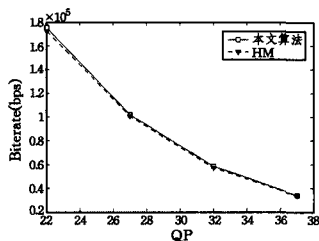


图 4 比特率变化情况

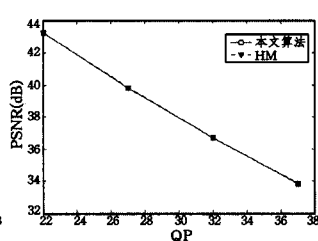


图 5 信噪比变化情况

**结束语** 本文提出了一种基于决策树的 HEVC 编码单元划分算法,利用视频帧内的空间相关性,提取编码单元的划分信息,将其加载到决策树模型中学习之后得到分类器,跳过了当前编码深度的率失真计算,在保证视频质量的同时,计算复杂度平均降低了 33%。

在接下来的工作中将结合快速编码单元划分和快速预测模式决策,在保证视频质量的同时,进一步降低 HEVC 的计算复杂度。

## 参考文献

[1] Ohm J R, Sullivan G J. High efficiency video coding: the next

frontier in video compression[J]. IEEE Signal Processing Magazine, 2013, 30(1):152-158

[2] Kang J W, Lou C C, Kim S H, et al. Efficient HD video coding with joint first-order-residual (FOR) and second-order-residual (SOR) coding technique[J]. Journal of Visual Communication and Image Representation, 2013, 24(1): 1-11

[3] Shen Yan-fei, Li Jin-tao, Zhu Zhen-min, et al. High Efficiency Video Coding[J]. Chinese Journal of Computers, 2013, 36(11): 2340-2355 (in Chinese)  
沈燕飞, 李锦涛, 朱珍民, 等. 高效视频编码[J]. 计算机学报, 2013, 36(11): 2340-2355

[4] 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

[5] Li Yuan, He Xiao-hai, Zhong Guo-yun, et al. A Fast Interframe Prediction Unit Mode Decision Algorithm for High Efficiency Video Coding Based on Temporal Correlation [J]. Journal of Electronics & Information Technology, 2013, 35(10): 2365-2379 (in Chinese)  
李元, 何小海, 钟国韵, 等. 一种基于时域相关性的高性能视频编码快速帧间预测单元模式判决算法[J]. 电子与信息学报, 2013, 35(10): 2365-2379

[6] 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

[7] Kim J, Jeong S, Cho S, et al. Adaptive Coding Unit Early Termination Algorithm for HEVC[C]// IEEE International Conference on Consumer Electronics. 2012: 261-262

[8] Chiang P T, Chang T S. Fast Zero Block Detection and Early CU Termination for HEVC Video Coding[C]// IEEE International Symposium on Circuits and Systems, 2013: 1640-1643

[9] Shen L, Zhang Z, Liu Z. Effective CU Size Decision for HEVC Intracoding[J]. IEEE Transactions on Image Processing, 2014, 23(10): 4232-4241

[10] Shen X, Yu L, Chen J. Fast Coding Unit Size Selection for HEVC based on Bayesian Decision Rule[J]. Picture Coding Symposium, 2012, 8355(3): 453-456

[11] Shen X, Yu L. CU splitting early termination based on weighted SVM[J]. Image and Video Processing, 2013, 2013(1): 1-11

[12] Zhang H, Ma Z. 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

[13] Wang L, Siu W. Novel Adaptive Algorithm for Intra Prediction with Compromised Modes Skipping and Signaling Processes in HEVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2013, 23(10): 1686-1694

[14] MI, Jo H, Sim D. Fast intra mode decision for HEVC intra coding[C]// The 18th IEEE International Symposium on Consumer Electronics (ISCE). 2014

[15] Shen L, Zhang Z, An P. Fast CU Size Decision and Mode Decision Algorithm for HEVC Intra Coding[J]. IEEE Transactions on Consumer Electronics, 2013, 59(1): 207-213

[16] Shen L, Zhang Z, Liu Z. Adaptive Inter-Mode Decision for HEVC Jointly Utilizing Inter-Level and Spatiotemporal Correlation[J]. IEEE Transaction on Circuits and Systems for Video Technology, 2014, 24(10): 1709-1722

[17] Xu Q, Hu D. A new inter prediction mode decision method based on temporal correlation for HEVC encoder[C]// IEEE Work-

[18] Vanne J, Viitanen M, Hamalainen T. Efficient Mode Decision Schemes for HEVC Inter Prediction[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2014, 24(9): 1579-1593

[19] Chen Y, Mukherjee D, Han J. Joint Inter-Intra Prediction Based on Mode-Variant and Edge-Directed Weighting Approaches in Video Coding [C] // International Conference on Acoustics

[20] Shen L, Liu Z, Zhang X, et al. Zhang. An Effective CU Size Decision Method for HEVC Encoders [J]. IEEE Transactions on Multimedia, 2013, 15(2): 465-470

[21] 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

[22] Bjontegaard G. Calculation of average PSNR differences between RD-curves: VCEG-M33[S]. Austin, March 2001

(上接第 302 页)

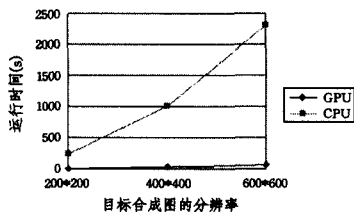


图 7 GPU 与 CPU 合成不同分辨率的纹理图的耗时比较

**结束语** 本文提出了一种改进的 Image quilting 方法,在样本中查找待合成的块时增加了纹理结构信息的引导,使得从样图中选取的块更为合理,改善了纹理合成的效果。但是 Chamfer 距离计算复杂,为了解决该方法带来的计算效率问题,本文使用了 GPU 并行加速,将合成时间控制在可接受的范围内,提高了纹理合成的质量,同时也保证了纹理合成效率。未来将进一步考虑如何在合成时根据样图纹理的特点更好地结合结构和颜色的信息,比如对于结构性较强的纹理,则在匹配时更多地考虑结构相似性;而对于结构性不那么强的纹理,则适当地添加结构信息。

### 参 考 文 献

[1] Wei L Y, Lefebvre S, Kwatra V, et al. State of the Art in Example-based Texture Synthesis[R]. Eurographics 2009 State of the Art Reports(STARs), 2009

[2] Efros A A, Freeman W T. Image quilting for texture synthesis and transfer[C]//Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. ACM, 2001: 341-346

[3] Efros A A, Leung T K. Texture synthesis by non-parametric sampling[C]//Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999. IEEE, 1999, 2: 1033-1038

[4] Wei L Y, Levoy M. Fast texture synthesis using tree-structured vector quantization[C]//Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., 2000: 479-488

[5] Ashikhmin M. Synthesizing natural textures[C]//Proceedings of the 2001 Symposium on Interactive 3D Graphics. ACM, 2001: 217-226

[6] Zelinka S, Garland M. Towards real-time texture synthesis with the jump map[C]//Proceedings of the 13th Eurographics Workshop on Rendering. Eurographics Association, 2002: 99-104

[7] Lefebvre S, Hoppe H. Parallel controllable texture synthesis [J]. ACM Transactions on Graphics (TOG), 2005, 24(3): 777-786

[8] Guo B, Shum H, Xu Y Q. Chaos mosaic: Fast and memory efficient texture synthesis[R]. Microsoft Research Paper MSR-TR-

2000-32, 2000

[9] Zhu Wen-hao, Wei Bao-gang. The Technology of Sampled-Based Texture Synthesis[J]. Journal of Image and Graphics, 2008, 13(11): 2063-2069 (in Chinese)

朱文浩, 魏宝刚. 基于样本的纹理合成技术综述[J]. 中国图象图形学报, 2008, 13(11): 2063-2069

[10] Liang L, Liu C, Xu Y Q, et al. Real-time texture synthesis by patch-based sampling [J]. ACM Transactions on Graphics (ToG), 2001, 20(3): 127-150

[11] Nealen A, Alexa M. Hybrid texture synthesis[C]//Proceedings of the 14th Eurographics Workshop on Rendering. Eurographics Association, 2003: 97-105

[12] Kwatra V, Schödl A, Essa I, et al. Graphcut textures: image and video synthesis using graph cuts [J]. ACM Transactions on Graphics (TOG), 2003, 22(3): 277-286

[13] Wu Q, Yu Y. Feature matching and deformation for texture synthesis[J]. ACM Transactions on Graphics (TOG), 2004, 23(3): 364-367

[14] Lee T Y, Yan C R. Feature-based texture synthesis[M]//Computational Science and Its Applications-ICCSA 2005. Springer Berlin Heidelberg, 2005: 1043-1049

[15] Lefebvre S, Hoppe H. Appearance-space texture synthesis [J]. ACM Transactions on Graphics (TOG), 2006, 25(3): 541-548

[16] Shen Jian-wei, Wang Shuo-zhong. Structural Matching in Texture Synthesis Based on Image Quilting[J]. Journal of Shanghai University (Natural Science Edition), 2010, 16(1) (in Chinese)

沈剑伟, 王朔中. 拼接法纹理合成中的结构特征匹配[J]. 上海大学学报 (自然科学版), 2010, 16(1)

[17] Zhang Jun, Zhu Wei, Huang Wei-qiang. Novel Structure Adaptive Algorithm for Texture Synthesis[J]. Journal of Chinese Computer Systems, 2011, 32(2): 351-355 (in Chinese)

张军, 朱为, 黄伟强. 一种新的结构自适应纹理合成算法[J]. 小型微型计算机系统, 2011, 32(2): 351-355

[18] Gui Y, Chen M, Xie Z, et al. Texture Synthesis based on Feature Description [J]. Journal of Advanced Mechanical Design, Systems, and Manufacturing, 2012, 6(3): 376-388

[19] Barrow H G, Tenenbaum J M, Bolles R C, et al. Parametric correspondence and Chamfer matching: two new techniques for image matching [C]//Proceedings of the 5th international joint conference on Artificial intelligence-Volume 2. San Francisco, CA, USA; Morgan Kaufmann Publishers Inc., 1977: 659-663

[20] Borgefors G. Hierarchical Chamfer matching: a parametric edge matching algorithm[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1988, 10(6): 849-865

[21] Farber R. CUDA application design and development[M]. Morgan Kaufmann, 2011