软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(Suppl.(2)):239−246

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向 CPU-GPU 平台的 HEVC 编码器优化*

罗法蕾[1,4], 王苫社[2], 马俊铖[2], 马思伟[2,3], 高 文[1,2,3]

[1](中国科学院 计算技术研究所,北京 100190)

[2](北京大学 数字媒体所&协同创新中心,北京 100871)

[3](北京大学 深圳研究生院,广东 深圳 518055)

[4](中国科学院大学,北京 100049)

通讯作者: 马思伟, E-mail: swma@pku.edu.cn

摘 要: 针对 CPU-GPU 平台提供了一种能显著降低高效视频编码(high efficiency video coding,简称 HEVC)复杂度的优化方案.根据编码器的复杂度分布及不同模块的特点,针对帧内预测、帧间预测以及环路滤波分别进行了优化.在帧内预测中,基于相邻编码单元(coding unit,简称 CU)之间的相关性,提出了一种 CU 的深度决策方法以及一种减少率失真优化(RDO)的模式数量的方法,降低了帧内编码的复杂度.在帧间预测中,提出将耗时最大的运动估计模块完善在图形处理单元(GPU)上,通过中央处理单元(CPU)和 GPU 的流水线工作获得了明显的加速,并基于预测残差的能量提出了一种编码单元提前终止划分的方法,有效降低了帧间编码复杂度.在环路滤波中,提出了一种 GPU 端的自适应样本点补偿(sample adaptive offset,简称 SAO)参数决策方法及去块滤波方法,有效分担了 CPU 端的复杂度.上述优化实现在 HM16.2 上,实验结果表明,提出的优化方案可以获得高达 68%的编码复杂度节省,而平均性能损失仅为 0.5%.

关键词: 高效视频编码; 图形处理单元; 帧内编码; 帧间编码; 环路滤波

中文引用格式: 罗法蕾,王苫社,马俊铖,马思伟,高文.面向 CPU-GPU 平台的 HEVC 编码器优化.软件学报,2015,26(Suppl.(2)):239−246. http://www.jos.org.cn/1000-9825/15034.htm

英文引用格式: Luo FL, Wang SS, Ma JC, Ma SW, Gao W. HEVC encoder optimization for CPU-GPU platform. Ruan Jian Xue Bao/Journal of Software, 2015,26(Suppl.(2)):239−246 (in Chinese). http://www.jos.org.cn/1000-9825/15034.htm

HEVC Encoder Optimization for CPU-GPU Platform

LUO Fa-Lei[1,4], WANG Shan-She[2], MA Jun-Cheng[2], MA Si-Wei[2,3], GAO Wen[1,2,3]

[1](Institute of Computing Technology, The Chinese Academy of Sciences, Beijing 100190, China)

[2](Institute of Digital Media & Cooperative Medianet Innovation Center, Peking University, Beijing 100871, China)

[3](Shenzhen Graduate School, Peking University, Shenzhen 518055, China)

[4](University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: This paper provides a comprehensive optimization strategy aiming at reducing the complexity of high efficiency video coding (HEVC) encoder with CPU-GPU cooperation. Based on the computational complexity distribution of HEVC encoder and characteristics of different modules and coding tools, intra coding, inter coding and in-loop filtering are collaboratively optimized. For intra coding, based on the correlation between neighboring coding units (CUs), depth range of CU is predicted and the number of candidates in intra mode candidate set for RDO (rate distortion optimization) is cut down, to avoid unnecessary computations. For inter coding, the most time consuming module, motion estimation (ME), is implemented with the collaboration of CPU and GPU in pipeline. Based on the energy of prediction residuals, an early termination scheme of CU splitting is proposed in this paper. For in-loop filtering, GPU based sample

* 基金项目: 国家高技术研究发展计划(863)(2015AA015903); 国家自然科学基金(61322106, 61272255); 深圳市孔雀计划; 北京市优秀博士学位论文导师奖资助项目(20128000103)

收稿时间: 2015-05-15; 定稿时间: 2015-10-12

adaptive offset (SAO) parameter decision scheme and GPU based deblocking scheme are proposed to further reduce the coding complexity on CPU. The overall optimization scheme is implemented on the HM 16.2 platform, and experiments demonstrate that the proposed optimization scheme can reduce over 68% of the coding complexity of HEVC encoder, with only 0.5% performance loss in average.

Key words: high efficiency video coding; graphic processing unit; intra coding; inter coding; loop filtering

由国际标准化组织动态图像专家组(Moving Picture Experts Group,简称 MPEG)和视频编码专家组(Video Coding Expert Group,简称 VCEG)联合制定的新一代高效视频编码标准(high efficiency video coding,简称 HEVC)[1,19],相比于当今广泛应用的先进视频编码(advanced video coding,简称 AVC)[2]在相同主观质量下能够节省 50%以上的比特率[20].然而,由于采用了更加灵活的编码单元划分结构以及更多的预测模式,HEVC 编码器的复杂度明显上升,难以满足实际应用的需求.

HEVC 采用了一套相比于以往编码标准更加灵活的四叉树划分结构,同时提供了多种新型编码工具,这使得编码器的性能得到了显著的提升.在 HEVC 中,一帧图像首先被划分成大小相同的多个编码树单元(coding tree unit,简称 CTU),其中每个编码树单元又可以按照四叉树递归划分的方式划分,最小能划分为 8×8 的编码单元,如图 1(a)所示.每个编码单元根据不同的划分方式还可以划分成多个预测单元(prediction unit,简称 PU),图 1(b)显示了帧间预测单元的划分方式.对帧内预测单元而言,HEVC 中仅支持方形的预测块划分,同时为了降低编码器的复杂度,只有最小编码单元可以划分成 4 个预测单元.如图 1(c)所示,对于变换单元(transform unit,简称 TU),HEVC 同样支持四叉树递归划分的方式,以进一步减少残差信息的冗余,实现更高的压缩比.

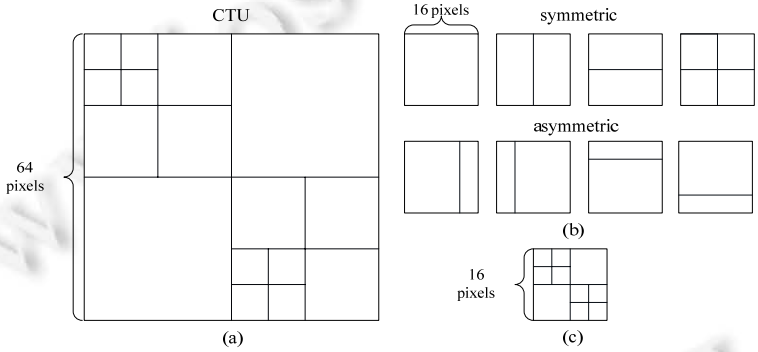


Fig.1 An example of a CTU, and its associated CU, PU and TU partitions

图 1 编码树单元划分的示例及对应的预测单元、变换单元树的划分结构

其中,图 1(a)为编码树单元划分成编码单元;图 1(b)为预测单元的不同划分模式;图 1(c)为变换单元的四叉树划分.

在当今主流的视频编码器中,编码模式或编码参数的选择过程是基于率失真优化准则的,即求解最小的率失真代价(J_{mode}),其计算公式如下:

$$J_{\text{mode}} = D + \lambda_{\text{mode}} \cdot R \tag{1}$$

其中,D 表示原始图像和对应编码模式下重构图像的失真,λ_{mode} 是拉格朗日乘积因子,用于均衡编码比特数量与失真对最优模式的影响,R 是编码模式与相关语法所耗费的比特数量.

HEVC 灵活的块划分方式与多样的灵活编码工具,使得上述编码过程需要遍历的模式数量组合大为增加,这给编码器增加了成倍的运算量,也给实际应用带来了较大的挑战.本文利用 GPU 的强大并行计算能力,实现了 HEVC 编码器中的运动估计模块,明显降低了编码复杂度.同时,针对帧内预测、帧间预测以及环路滤波的特性,分别进行了优化,大大节省了 HEVC 的编码时间.

1 相关工作

在降低 HEVC 的编码复杂度上,许多研究者针对帧内预测、帧间预测等编码模块均进行了深入的研究.

在帧内预测中,一个预测块最多可以采用 35 种预测模式[3],这给帧内预测的优化带来了很大的灵活性与提升空间.在文献[4,5]中,Zhao 和 Jiang 分别提出了其基于梯度方向的帧内预测优化方法,减少了预测模式数量,从而降低了帧内编码的复杂度.此外,利用空间域相邻或者时间域相邻块进行优化决策也取得了不错的加速效果[6,21].

灵活的块划分结构和较为复杂的运动补偿模块,使得 HEVC 中最复杂的模块落在帧间预测中.因而针对帧间预测的优化也已有很多的研究工作,主要做法分为运动估计模块优化、减少编码单元的递归划分以及编码单元的快速模式选择这 3 种.文献[7]提出了一种自适应参考帧选择方法,由此提前终止了后续参考帧的搜索过程,有效降低了复杂度.文献[8,22]利用相邻编码单元之间的相似性提出了一种提前终止编码单元划分的方法,可以减少大量无用编码单元划分的遍历,从而大大降低了帧间编码的复杂度.文献[9]利用预测残差大小进行判定,实现提前终止后续编码单元划分的效果.此外,文献[10]提出了一种有效控制编码器复杂度的方法,能够适用于不同计算能力的设备.

近些年来,随着图形处理单元(graphic processing unit,简称 GPU)的迅速发展,利用 GPU 进行编码器加速的研究成果不断涌现,一种适合于编程的 GPU 语言——CUDA[11]也逐步获得了广泛应用.基于 GPU 的强大计算能力和运动估计模块的易并行性,出现了较多的针对运动估计模块的优化.文献[12]提出了一种针对可变块大小的 GPU 端运动估计方法,采用递进层次的绝对误差和(sum of absolute differences,简称 SAD)合并方式求得大块的 SAD 值,实现了较大幅度的加速.文献[13,14]也利用 GPU,用类似的方式完成运动估计,实现了编码速度的明显提升.

本文基于文献[15,16]中提出的 GPU 端运动估计方法,进一步完善了一种面向 CPU-GPU 平台的低复杂度编码器优化方案.该方案结合了帧内编码中的快速编码决策,帧间编码中运动估计的 GPU 优化和编码单元划分的提前终止,以及基于 GPU 的环路滤波模块的去块滤波和 SAO 参数的决策,以较低的性能损失换取了明显的编码端复杂度节省.

2 优化技术

基于对 HEVC 编码器复杂度的分析,本文提出了一种结合 CPU 和 GPU 的编码器优化方案,如图 2 所示.对于一帧待编码的图像,首先按照一定编码树单元个数分成多个 CTU 窗口,对每个 CTU 窗口,在 CPU 端完成当前窗口的编码决策的同时,GPU 端完成下一个窗口的运动估计过程.当一个窗口编码决策完成之后,在 GPU 端可以进行该窗口的滤波参数决策以及环路滤波过程,由此实现 CPU 和 GPU 之间的并行编码流水,从而大幅度缩短了编码时间.

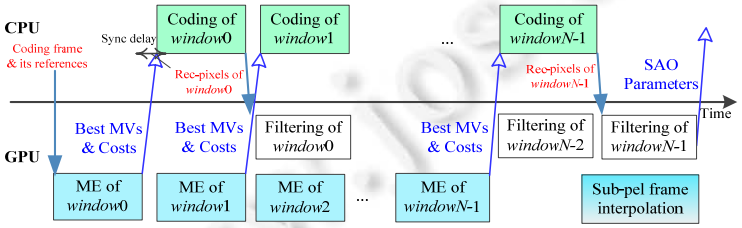


Fig.2 Framework of encoding one frame

图 2 编码一帧图像的流程

2.1 帧内编码优化

对自然视频序列而言,图像中的相邻块之间通常具有相似的纹理特征[17].基于这种特性,本文实现了一种简单、高效的编码树单元深度预测方法.

HEVC 中对于每个 CTU,其编码树划分深度由 0,1,2,3 组成,即其取值范围为 R=[0,3].统计表明,对于平滑或变化细微的区域,编码器会倾向于选择较小的编码树划分深度,这样耗费比较少的比特就可以达到较好的编码效果.而对于运动剧烈或者纹理复杂的区域,更大的编码树单元划分深度更容易被选中,以实现更准确的预测并减少编码残差耗费的比特.这意味着,对每个编码树单元都进行各个深度的递归遍历存在较大的计算冗余.本文中,将编码树单元的划分深度分为 3 个类别:$R0$=[0,3],$R1$=[0,2],$R2$=[1,3].其中,$R1$ 和 $R2$ 分别用于平滑和纹理复杂区域,而 $R0$ 则用于难以区分的区域.当前编码树单元的划分深度取值范围 DR_{curr} 可以通过如下等式求解.

$$DR_{curr}=\begin{cases}R1, & D_{\max}^{left}\leqslant 1 \text{ and } D_{\max}^{up}\leqslant 1\\ R2, & D_{\max}^{left}>1 \text{ and } D_{\max}^{up}>1\\ R0, & \text{others}\end{cases} \tag{2}$$

其中, $D_{\max}^{up}$ 和 $D_{\max}^{left}$ 分别表示当前编码树单元的上邻编码树单元和左邻编码树单元的最大划分深度.

此外,图像中相邻区域的纹理存在较大的相似性,该相似性可用于减少不必要的预测模式.在本文中,针对帧间编码帧也采用 Wang 等人提出的帧内预测模式快速选择方法[17],以进一步降低编码复杂度.

2.2 帧间编码优化

帧间预测依赖于对每一个预测块进行准确的运动搜索,以找到一个最佳匹配,而 HEVC 中灵活的块结构使得编码器需要遍历大量的块划分,因而复杂度大为增加.本文通过两个方面来有效降低帧间编码的复杂度,即 GPU 端的运动估计和 CPU 端的编码单元划分提前终止.

2.2.1 GPU 端的运动估计

如图 2 所示,在每一个 CTU 窗口编码前,先在 GPU 端完成运动估计.运动估计分为整像素搜索和分像素运动搜索两步.GPU 端分像素运动搜索与 HM(HEVC test model)的实现类似,区别在于 GPU 端可并行完成搜索过程.

在整像素搜索中,为实现任意预测单元的 SAD 的快速计算,本文采用了文献[16]中所述的一种快速查表的求解方法.该方法包括两个部分,即预测单元的模式索引表以及 SAD 查找表.两个表的示例如图 3 所示,图 3 左边为预测单元索引表,用于确定编码树单元中一个唯一的预测块的大小和位置.表中的每个元素为一个四元组,即(L,T,R,B),其单位 1 表示 4 个像素的位置偏移,表示了对应的预测块的起始坐标的左边界 x 轴位置、上边界 y 轴位置、右边界 x 轴位置以及下边界 y 轴位置.如图中的 17 号索引,对应了图中的阴影区域的两个 4×4 的小方块组成的预测单元.图 3 右边为 SAD 快速查找表,表中的每个圆圈代表了一个累加的 SAD 值,而方块表示对应位置的一个 4×4 块的 SAD 值.

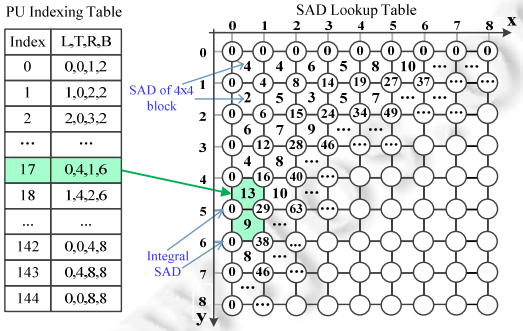


Fig.3 A fast SAD lookup table

图 3 SAD 快速查找表

利用上述索引表,可通过下式快速求得任意预测单元(L,T,R,B)的 SAD 值 $D_{PU}(L,T,R,B)$.

$$D_{PU}(L,T,R,B)=S(R,B)+S(L,T)-S(R,T)-S(L,B) \tag{3}$$

其中,$S(M,N)$表示从(0,0)位置到(M,N)位置的矩形区域内所有 4×4 块的 SAD 值累加和,其计算方式如下.

$$S(M,N)=\sum_{i=0}^{M}\sum_{j=0}^{N}D_{4\times4}(i,j) \tag{4}$$

$$D_{4\times4}(i,j)=\sum_{k=4i}^{4i+3}\sum_{l=4j}^{4j+3}\left|ref_{k,l}-org_{k,l}\right| \tag{5}$$

其中,$D_{4\times4}(i,j)$表示编码树单元中坐标为(i,j)位置的 4×4 块的 SAD 值,$ref_{k,l}$和 $org_{k,l}$分别为参考块和当前块中(k,l)处的像素值.通过上述方式,任意预测单元的 SAD 值可以在复杂度 $O(1)$的时间内取得,有效减少了以往方式中多层归并带来的延时.在文献[16]的基础上,本文在 GPU 端的分像素运动估计部分也采用 SATD 进行决策,有效降低了运动估计模块的编码性能损失.

2.2.2 编码单元划分提前终止

在视频编码中,预测残差能够有效反映预测的准确性.对于时间上变化缓慢或空间上平滑过度的场景,预测残差通常较小,此时选择较大的编码单元能够节省较多码率.而较小的编码单元更加适合于具有复杂纹理的区域以及运动变化剧烈的区域.因此,如果一个编码单元在当前划分大小时残差较小,就可以提前终止编码单元的继续划分,以减少冗余计算,而残差越大表明当前预测越不准确,需要继续划分才能实现更为准确的预测[9].基于此,本文提出了一种利用当前编码单元的预测残差能量来终止编码单元继续划分的方法.给定一个 $M\times N$ 大小的编码单元,其残差的能量 E 定义为

$$E=\frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}r_{i,j}^{2} \tag{6}$$

其中,$r_{i,j}$是编码单元内点(i,j)的预测误差.

由于不同编码序列具有不同的内容特性,采用固定的阈值 E_{thres}存在较大的误差.基于此,本文提出了一种鲁棒的阈值计算方式,即采用第 K 小个残差能量值作为判决编码单元是否继续划分的阈值 E_{thres}.在本文中,K 的取值为 0.7 倍的块数量.利用该阈值进行提前终止划分决策的流程为,前 2 个 P 帧的残差能量统计结果用于训练 E_{thres},而后续帧则利用该阈值进行编码单元的提前终止划分决策.对后续帧编码时,编码一个 CU 的主要流程为

(1) 检查当前 CU 的 SKIP 模式和 $2N\times2N$ 的划分模式,并计算对应的预测残差能量 E;

(2) 完成当前 CU 的剩余率失真优化决策过程;

(3) 若当前 CU 的阈值 E 小于 E_{thres},则终止当前 CU 划分,否则,继续尝试当前 CU 划分的情况.

2.3 环路滤波优化

基于上述帧内优化和帧间优化,编码器的复杂度大为降低,此时原先一些耗时较小的模块,如环路滤波也开始凸显出了相对较高的复杂度.HEVC 中的环路滤波包括两个方面,即去块效应滤波和样本点自适应补偿(sample adaptive offset,简称 SAO),针对两种滤波器的不同特点,本文利用 GPU 分别进行了优化,有效降低了各自的复杂度.

2.3.1 去块滤波的优化

HEVC 的去块滤波主要包含两个步骤,即边界强度的计算和边界的滤波处理.边界强度的计算过程中涉及到变换块、预测块边界的检查,以及边界两侧预测信息的检查,具有较多的分支,不便于在 GPU 端实现.而在边界的滤波处理过程中,对图像中的所有 8×8 块边界均进行类似的操作,易于在 GPU 端并行实现.

由于去块滤波时最多影响或依赖于边界左右或上下各 3 个点,因此上一行编码树单元的滤波结果不影响下一行编码树单元的滤波决策.这样,当 CTU 窗口为一个 CTU 行时,基于 CTU 窗口的依次滤波可保证滤波的准确性.在这种行级流水的框架下,CPU 端首先完成滤波边界强度的计算,而后 GPU 端完成块边界滤波过程.通过该方式,GPU 端有效地分担了 CPU 端的工作量,降低了 CPU 端的编码复杂度.

2.3.2 GPU 端的 SAO 决策

在参考软件 HM 中,SAO 的决策采用了一种基于滤波前重构图像失真统计的快速参数决策方法.该方法的主要过程是首先统计每类样本点的数量 N_i和失真值 E_i,然后根据公式(8)的率失真代价的变化值ΔJ求得最优解.

$$\Delta D = \sum_{i=1}^{4}\left(N_i h_i^2 - 2h_i E_i\right) \tag{7}$$

$$\Delta J = \Delta D + \lambda R \tag{8}$$

其中,h_i为给定第 i 个样本点类别的 SAO 补偿值,λ为拉格朗日因子,R 为编码相关参数耗费的比特数量.

SAO 的参数决策可以分为两个步骤,即样本点类别信息的统计和率失真优化决策.其中,对每个编码树单元的样本点分类信息的统计可以同时进行,另外,行之间也可以同时进行率失真优化决策,由此可充分利用 GPU 的并发处理能力,较快地完成整个参数的决策过程.

在样本点类别信息的统计过程中,一个线程块内的线程数量 N_T 主要受制于如下不等式:

$$N_T \times S_T < S_{\mathrm{MAX}} \tag{9}$$

其中,S_T是每个线程所需的最小共享缓存字节数,S_{MAX}是一个线程块的共享缓存上限.单个线程所需的共享缓存大小 S_T至少为 256 字节,因为单个像素点可能属于 32 个类别中的任意一种.这样受制于硬件设备的共享缓存限制,N_T 远远达不到编码单元内的像素点数量.基于此,本文中提出的线程分配方式为,每个编码树单元会分配一个线程块,而线程块中的每个线程完成一列像素的分类计算,既分担了所有点统计的计算量,也避免了计算资源的浪费.

在此基础上,利用 GPU 的多线程并发计算能力,本文提出如下决策方法:对每一帧图像的率失真优化决策过程遍历图像级 SAO 开关参数的多种组合状态,并利用基于表格的熵编码比特估计模型进行编码 SAO 参数的估计,最后从多种组合中选取总的率失真代价最小者作为最终的编码参数.通过该方法,可节省CPU端的编码复杂度,也改善了原有 SAO 决策的编码性能.

3 实验结果

将本文提出的编码器优化方案在 HEVC 参考软件 HM 16.2 上进行实验,并在一台配备有 1 颗 Tesla K40 的工作站上完成测试.测试中采用的编码配置为低延时 P 帧配置,使用 HEVC 通用测试序列,最大 CTU 的大小设置为 32×32.评估本文算法有效性采用复杂度节省ΔT和编码性能损失两个指标.前者的计算公式为

$$\Delta T = \frac{T_{anchor} - T_{pro}}{T_{anchor}} \tag{10}$$

其中,T_{anchor} 为参考软件的编码时间,T_{pro} 为本文所优化的编码器的编码时间.在衡量编码性能损失方面,采用 BD-rate[18]为参考度量标准.

表 1 给出了本文方案相比于参考软件 HM 16.2 的编码性能.可以看到,本文所实现的方法能够以 0.52%的性能损失,换取平均 68%以上的编码时间节省,有效地降低了编码器的复杂度.特别地,类别 A 和 E 的序列在亮度分量的平均损失都在 0.1%以内,而时间节省均超过了 70%.

Table 1 Coding performance of the proposed scheme

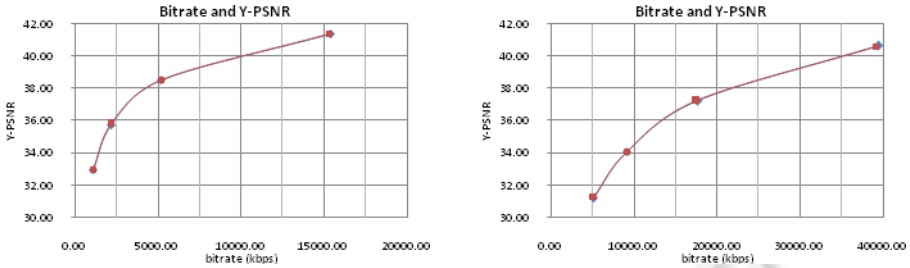
表 1 本文所提出方法的编码性能

序列类别	分辨率	BD-rate (piecewise cubic)			复杂度节省ΔT
		Y	*U*	*V*	
类别 *A*	2560×1600	0.03%	0.36%	0.92%	71.59%
类别 *B*	1920×1080	0.38%	1.44%	1.76%	73.44%
类别 *C*	832×480	0.45%	1.60%	1.42%	65.94%
类别 *D*	416×240	1.37%	0.56%	0.99%	62.17%
类别 *E*	1280×720	0.07%	3.50%	3.13%	70.46%
平均		0.52%	1.51%	1.65%	68.57%

采用本文所提出的编码器的编码性能与参考软件的编码性能的 RD 曲线对比如图 4 所示,可以看到,在任意码率下本文所提出的方法其编码性能都和参考软件非常接近.

本文方法和相关 GPU 优化方案[13,14]的对比结果见表 2.与以往方法相比,本文提出的一种基于查表的 SAD 快速求取方法应用于 GPU 端的运动估计模块,更适用于 HEVC 灵活的块划分结构.通过使用 GPU 完成运动估计,大大消除了该模块的耗时.在此基础上,GPU 也被进一步扩展应用于环路滤波中,进一步减少了 CPU 端的编

码复杂度,并通过遍历更多 SAO 开关状态的组合提升了编码性能.结合 GPU 端的优化,本文针对帧内预测和帧间预测的特点进行了改进,在较低的性能损失下显著地降低了总体编码复杂度,具有更广阔的应用前景.



(a) Traffic (b) PeopleOnStreet

Fig.4 Comparison of the proposed scheme with HM

图 4 本文方案和参考软件的编码性能对比

Table 2 Coding performance comparison of the proposed scheme with other optimization schemes

表 2 本文方法和相关优化方法性能对比

类别	本文方法		文献[13]		文献[14]	
	Y	Δ*T*	*Y*	Δ*T*	*Y*	Δ*T*
类别 *A*	0.0%	71.6%	1.3%	49.2%	–	–
类别 *B*	0.4%	73.4%	1.0%	50.5%	0.5%	54.4%
类别 *C*	0.5%	65.9%	–	–	0.8%	55.2%
类别 *D*	1.4%	62.2%	–	–	–	–
类别 *E*	0.1%	70.5%	–	–	–	–
平均	0.5%	68.6%	–	–	–	–

4 结 论

本文在 HEVC 编码标准灵活块划分结构下,利用 GPU 的并行计算能力,提出了一种快速的基于查表的 SAD 求取方法,完善了一套有效的并行运动估计方案.本文也利用 GPU 完善环路滤波模块,在进一步降低 CPU 端复杂度的同时提升了编码性能.在 GPU 端对编码复杂度有效分担的基础上,本文结合帧内编码、帧间编码的快速编码技术,大大降低了总体编码复杂度.与以往方法相比,本文方法在编码效率损失很小的情况下,大大降低了编码复杂度,对在通用 CPU-GPU 架构下实现高性能 HEVC 实时编码器有重要的意义.

References:

[1] Sullivan GJ, Ohm JR, Han WJ, Wiegand T. Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. on Circuits System for Video Technology, 2012,22(12):1649−1668.

[2] Ohm JR, Sullivan GJ, Schwarz H, Tan TK, Wiegand T. Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC). IEEE Trans. on Circuits and Systems for Video Technology, 2012,22(12):1669−1684.

[3] Lainema J, Bossen F, Han WJ, Min J. Intra coding of the HEVC standard. IEEE Trans. on Circuits and Systems for Video Technology, 2012,22(12):1792−1801.

[4] Zhao L, Zhang L, Ma S, Zhao D. Fast mode decision algorithm for intra prediction in HEVC. In: Proc. of the IEEE Visual Communications and Image Processing (VCIP). 2011. 1−4.

[5] Jiang W, Ma H, Chen Y. Gradient based fast mode decision algorithm for intra prediction in HEVC. In: Proc. of the Consumer Electronics, Communications and Networks (CECNet). Three Gorges, 2012. 1836−1840.

[6] Shen L, Zhang Z, An P. Fast CU size decision and mode decision algorithm for HEVC intra coding. IEEE Trans. on Consumer Electronics, 2013,59(1):207−213.

[7] Wang S, Ma S, Wang S, Zhao D, Gao W. Fast multi reference frame motion estimation for high efficiency video coding. In: Proc. of the 20th Int'l Conf. on Image Processing (ICIP). Melbourne, 2013. 2005−2009.

[8] Hsu WJ, Hang HM. Fast coding unit decision algorithm for HEVC. In: Proc. of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. (APSIPA). Kaohsiung, 2013. 1–5.

[9] Yu Q, Zhang X, Wang S, Ma S. Early termination of coding unit splitting for HEVC. In: Proc. of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conf. (APSIPA). Hollywood, 2012. 1–4.

[10] Correa G, Assuncao P, Agostini L, Cruz L. Complexity control of high efficiency video encoders for power-constrained devices. IEEE Trans. on Consumer Electronics, 2011,57(4):1866–1874.

[11] NVIDIA. CUDA C Programming Guide. Version 6.5, 2014. http://docs.nvidia.com/cuda/cuda-c-programming-guide

[12] Wang XW, Song L, Chen M, Yang JJ. Paralleling variable block size motion estimation of HEVC on CPU plus GPU platform. In: Proc. of the IEEE Int'l Conf. on Multimedia and Expo Workshops (ICMEW). San Jose, 2013, 1–5.

[13] Kim S, Lee D, Sohn C, Oh S. Fast motion estimation for HEVC with adaptive range decision on CPU and GPU. In: Proc. of the 2nd IEEE China Summit and Int'l Conf. on Signal and Information Processing (ChinaSIP). Xi'an, 2014. 349–353.

[14] Radicke S, Hahn J, Grecos C, Wang Q. A highly-parallel approach on motion estimation for high efficiency video coding (HEVC). In: Proc. of the IEEE Int'l Conf. on Consumer Electronics (ICCE). Las Vegas, 2014. 187–188.

[15] Ma JC, Luo FL, Wang SS, Ma SW. Flexible CTU-level parallel motion estimation by CPU and GPU pipeline for HEVC. In: Proc. of the IEEE Int'l Conf. on Visual Communications and Image Processing (VCIP). Valletta, 2014. 14–17.

[16] Luo FL, Ma SW, Ma JC, Qi HG, Su L, Gao W. Multiple layer parallel motion estimation on GPU for high efficiency video coding (HEVC). In: Proc. of the IEEE Int'l Symp. on Circuits and Systems (ISCAS). Lisbon, 2015. 1122–1125.

[17] Wang Y, Fan XP, Zhao L, Ma SW. A fast intra coding algorithm for HEVC. In: Proc. of the 2014 IEEE Int'l Conf. on Image Processing (ICIP). Paris, 2014. 4117–4121.

[18] Bjøntegaard G. Improvement of BD-PSNR model. VCEG-AI11, Berlin, 2008.

[19] 马思伟.新一代高效视频编码标准 HEVC 的技术架构.中国多媒体通信,2013,(7):20–21.

[20] 沈燕飞,李锦涛,朱珍民,张勇东.高效视频编码.计算机学报,2013,36(11):2340–2355.

[21] 雷海军,杨忠旺,陈骁,袁梅冷.一种快速 HEVC 编码单元决策算法.计算机工程,2014,(3):270–273.

[22] 王超超,王万良,岑跃峰,姚信威,姚晓敏.HEVC 快速编码深度选择算法.计算机工程与应用,2014.

罗法蕾(1992－),男,湖北监利人,博士生,CCF 学生会员,主要研究领域为视频编码与优化.

马思伟(1979－),男,博士,教授,CCF 会员,主要研究领域为视频编码与处理.

王苫社(1981－),男,博士,主要研究领域为视频编码与优化.

高文(1956－),男,博士,教授,博士生导师,中国工程院院士,CCF 会士,主要研究领域为人工智能应用,多媒体技术.

马俊铖(1990－),男,硕士,主要研究领域为视频编码与应用.