

◎工程与应用◎

基于多层次并行架构的 H.264 解码算法

王 宇, 陈耀武

WANG Yu, CHEN Yaowu

浙江大学 浙江省网络多媒体技术研究重点实验室, 杭州 310027

Network Multimedia Technology Research Key Laboratory, Zhejiang University, Hangzhou 310027, China

WANG Yu, CHEN Yaowu. Multi-layer parallel H.264 decoding algorithm. Computer Engineering and Applications, 2014, 50(8): 215-220.

Abstract: In the demand of real-time decoding of high-definition video stream, a multi-layer parallel algorithm is proposed. First, this algorithm distributes tasks to the processors according to the decoding function module and optimizes the load-balance between processors through the secondary partition of the function module. Then, in the loop-filter module, this algorithm processes the macro blocks parallelly according to the dependency relationship between the macro blocks. Experimental results indicate that this algorithm improves the efficiency and the parallelization of the decoder, realizes the real-time decoding of high-definition video stream on the TILEPro64 platform. This algorithm speeds up the decoding process by 10.01 at most, achieves a 80% performance increase of speedup comparing to the existing parallel algorithm.

Key words: H.264 real-time high-definition decoding; parallel decoding; parallel filter; many-core processor; TILEPro64 platform

摘 要: 针对高清图像视频的实时解码需求, 提出一种基于多层次并行流水架构的解码算法。该算法首先针对图像的宏块行实现基于功能模块的行级并行算法, 并通过功能模块的二次划分进行核间负载均衡的优化, 再针对解码过程中开销较大的滤波环节, 利用宏块之间的依赖关系进行多核并行处理, 对行级并行算法进行更深层次上的再优化设计。实验结果表明, 该算法可以在 TILEPro64 平台上实现 1 080P 全高清码流的实时解码, 实现了较高的并行加速比, 最高达到 10.01, 和已有的并行解码算法相比, 加速比提升 80%。

关键词: H.264 实时高清解码; 并行解码; 并行滤波; 众核处理器; TILEPro64 平台

文献标志码: A **中图分类号:** TP37 **doi:** 10.3778/j.issn.1002-8331.1212-0395

1 引言

H.264 是由 ITU-T 和 MPEG 组织联合开发的数字视频标准, 该标准较以前的标准能够大幅提高图像压缩率, 通过较低的码率提供较高的图像质量, 在相同图像质量下, H.264/AVC 标准所需的码率仅为 MPEG2 的 36%, H.263 的 51%, MPEG4 的 61%^[1]。随着近年来高清视频的广泛普及, H.264 标准也得到广泛的应用, 但是 H.264 标准的算法复杂度较高, H.264 解码的复杂度约是 MPEG-4 的 3 倍^[2]。为了满足高复杂度的计算性能需求, 以及同时克服在散热、功耗以及电路板外形尺寸等

因素上的种种限制, 使用多核处理器进行高清视频处理成为一种在高清视频应用领域中的最新潮流^[3-4]。

现有的多核并行的解码算法主要通过数据划分或功能划分的方式来进行多核并行处理。文献[5]提出了一种基于数据划分的多帧并行解码算法, 提高了解码效率, 但是解码时延较长。文献[6]提出一种以 slice 为处理单位的片级并行算法, 不同的核处理不同的 slice, 但是这种并行策略在处理单 slice 图像时性能表现较差。文献[7]提出一种基于宏块划分的多核并行算法, 通过减小数据划分的颗粒度来提高并行效率, 但是该算法在

基金项目: 国家自然科学基金仪器专项(No.40927001); 浙江省重点科技创新团队计划资助项目(No.2011R09021-02)。

作者简介: 王宇(1986—), 男, 硕士研究生, 主研方向: 多媒体视频编解码; 陈耀武, 男, 教授, 博士生导师, 主研方向: 多媒体视频编解码。E-mail: yuwang.zheda@gmail.com

收稿日期: 2013-01-04 **修回日期:** 2013-02-20 **文章编号:** 1002-8331(2014)08-0215-06

CNKI 网络优先出版: 2013-03-13, <http://www.cnki.net/kcms/detail/11.2127.TP.20130313.0957.026.html>

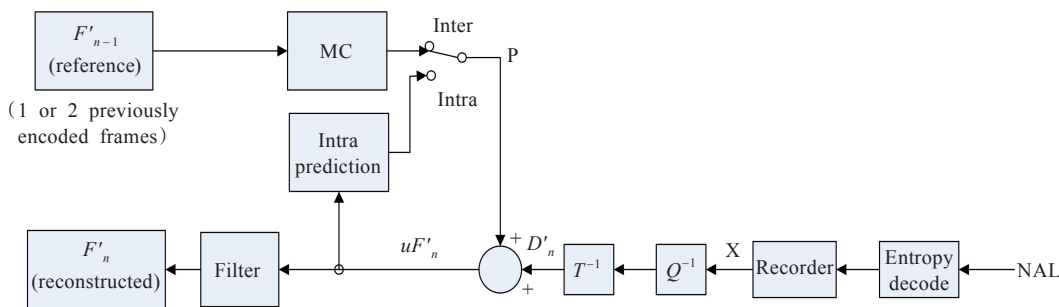


图1 解码流程图

并行时存在一定的数据访问冲突,影响解码效率。文献[8]利用三核处理器并行解码,基于功能模块的划分方式提高了解码效率,但是由于仅划分了三个功能模块进行并行处理,没有充分发掘出解码流程在功能和数据上的可并行性以及解决各处理器之间负载失衡的问题。文献[9]对于高分辨率高码率的码流,由于滤波解码开销较高,解码性能降低,扩展性不强。

本文提出一种基于多层次并行架构的 H.264 解码算法,首先利用 6 个处理器核来执行不同的解码模块,6 个核分别在相邻的 6 个宏块行上进行并行流水作业,实现基于功能模块的行级并行解码,并通过各功能模块的二次划分解决各核间负载均衡问题。然后,为了解决时间开销较大的环路滤波阶段的性能瓶颈,充分利用宏块之间的依赖性,针对一帧图像进行多核并行滤波。经实验得知,该算法可以有效提高解码效率和并行加速比,对于分辨率较高的 1 080P 全高清码流,解码性能提升显著,可以实现 30 f/s 的实时解码。

2 基于功能模块的行级并行解码算法

2.1 H.264 解码流程

H.264 解码器解码流程如图 1 所示,大体可以分为,熵解码(ECD),反量化(IQ),反离散余弦变换(IDCT),帧间预测(MC),帧内预测(IP),BS 值计算和环路滤波(LPF)这几个环节。

2.2 6 核并行流水设计

根据上述解码流程,以解码功能模块作为各个核组任务划分的根据,将解码流程分配到 6 个处理器核进行并行处理,如图 2 所示。

基于图 2 所示的任务划分,采用一种区别于宏块并行的行级并行策略。这种策略有效避免了流水线中核

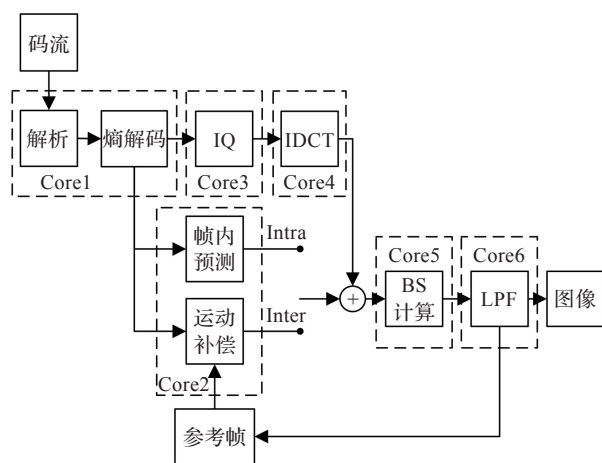


图2 解码流程的 6 核划分

间数据的访问冲突,提高了整体的解码速率。

6 核并行的流水线排布如图 3 所示。

第 1 个核负责解析码流,完成对第 1 行宏块熵解码后,启动第 2 个核进行运动补偿,然后继续执行下一行宏块的熵解码,直至一帧码流结束。

第 2 个核对已经完成熵解码的第 1 行宏块,进行运动补偿,结束后,启动第 3 个核进行反量化,然后继续执行下一行宏块的运动补偿,直至一帧码流结束。

第 3 个核对已经完成对第 1 行宏块进行反量化后,启动第 4 个核进行 IDCT 变换,然后继续执行下一行宏块的反量化,直至一帧码流结束。

第 4 个核对已经完成对第 1 行宏块进行 IDCT 变换后,启动第 5 个核进行 BS 计算,然后继续执行下一行宏块的 IDCT 变换,直至一帧码流结束。

第 5 个核对已经完成对第 1 行宏块进行 BS 计算后,启动第 6 个核进行去块滤波,然后继续执行下一行宏块的 BS 计算,直至一帧码流结束。

第 6 个核完成所有行的去块滤波后,解码一帧结束。

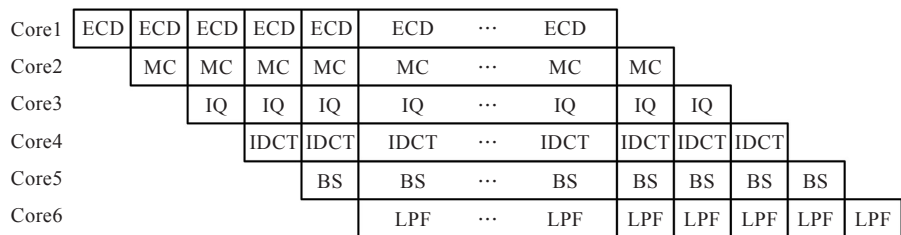


图3 6 核并行流水示意图

利用6个核的并行流水作业,可以有效提高解码过程的整体效率,避免了流水线中核间数据的访问冲突,提高了整体的解码速率。并且,这种以宏块行为单位的并行处理方式,可以很好地解决文献[6]中以slice为单位的并行算法遇到单 slice 图像时性能退化的问题。

但是,这种单纯以功能模块作为各核划分根据的并行算法也存在一定的缺陷,由于各功能模块间时间开销差距较大,其中 IQ,IDCT,BS 计算模块时间开销较小,而 MC 和环路滤波 LPF 模块时间开销较大,这样会造成各核间运算量负载不均衡,影响整体的解码效率。

2.3 核间负载均衡的优化

针对核间运算量不均衡情况,可以通过细化功能模块颗粒度,对各核的任务负载进行二次划分的方式解决。除了熵解码需要连续的数据、只能在单核上连续计算外,其他各个功能模块都能够被分解进行。功能模块二次划分可选方式见表1。

表1 功能模块二次划分可选方式	
模块	二次划分方式
MC	MC16×16,MC16×8,MC8×16,MC8×8 及
	MC_Y,MC_U,MC_V
IQ	IQ_Y,IQ_U,IQ_V
IDCT	IDCT_Y,IDCT_U,IDCT_V
LPF	LPF_Y,LPF_U,LPF_V

在分析各功能模块的相互关系和效率情况后,对模块划分做出优化调整,将MC模块和帧内预测模块都按数据划分成MC1/2、MC2/2、PredIntra1/2、PredIntra2/2并行执行;IQ和IDCT模块合并一个核执行;LPF模块拆分成LPF_Y和LPF_UV两部分,LPF_UV和BS计算模块合并一个核执行;LPF_Y单独执行。表2是六核功能模块任务的二次划分方式,功能模块经过负载均衡优化后的流水线如图4所示。

表2 六核任务二次划分	
任务	功能模块
Task1	码流解析+熵解码
Task2	MC1/2 PredIntra1/2
Task3	MC2/2 PredIntra2/2
Task4	IQ+IDCT
Task5	LPF_UV+BS
Task6	LPF_Y

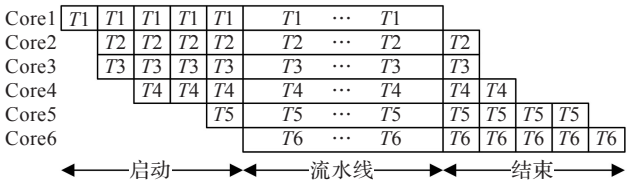


图4 功能模块负载均衡优化后的流水线

与文献[8]相比,通过细化功能模块的颗粒度及各核间负载均衡的优化,有效地提高了解码器的并行度,

更充分地利用多核处理器的运算资源。

经实验测量可知,环路滤波时间开销较大,成为提升整体解码效率的瓶颈。因此需要充分发掘出滤波过程中数据上的可并行性,来实现多核并行滤波,以此来提高整体的解码效率。

3 环路滤波模块的并行算法

3.1 环路滤波的机制

图像产生方块效应的原因有两个,一是在IDCT过程中,由于反量化过程恢复的变换系数引入的误差。二是在MC过程中。运动补偿块可能从不同帧的插值样点得到,其匹配不可能绝对准确,造成边界数据不连续^[10-11]。在H.264标准中,在亮度块和色度块中其需要处理的边缘及顺序如图5所示。

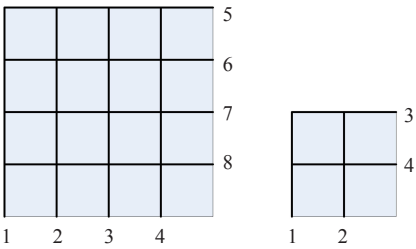


图5 亮度块和色度块需要滤波的边缘

根据标准,所有的边缘按照从上到下,从左到右进行处理,垂直边缘先进行处理。在一帧图像中,所有的宏块按照光栅扫描顺序进行滤波处理,当前宏块分别对其左边,上边,右上角的宏块具有依赖关系,如图6所示。当滤波当前宏块时,其依赖的其他宏块应该已经处理完毕。

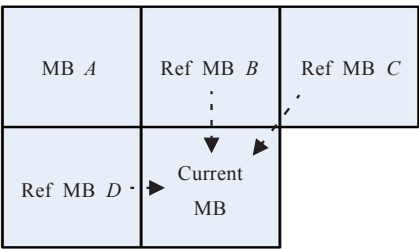


图6 宏块滤波时依赖关系

3.2 滤波过程的并行化设计

当按照光栅扫描顺序对图像进行滤波处理时,每一行的宏块是按照从左到右的顺序来逐个进行滤波处理,宏块数据之间并行效率较低,因此,可以考虑不使用光栅扫描顺序,而是充分利用宏块间的依赖特性来进行多宏块的并行处理,宏块的滤波顺序也因此有所改变,如图7所示。

图7中的数字代表了进行滤波处理的宏块顺序,图中每个宏块进行滤波时其所依赖的宏块都已经处理完毕,符合标准规定的依赖性要求,因此标有同样数字的宏块可以同时进行独立的并行滤波处理,这样可以大大



图7 多宏块并行滤波的顺序示意图

地提高滤波过程的并行度和滤波效率。图8举例说明了一帧图像是如何进行并行滤波处理的。

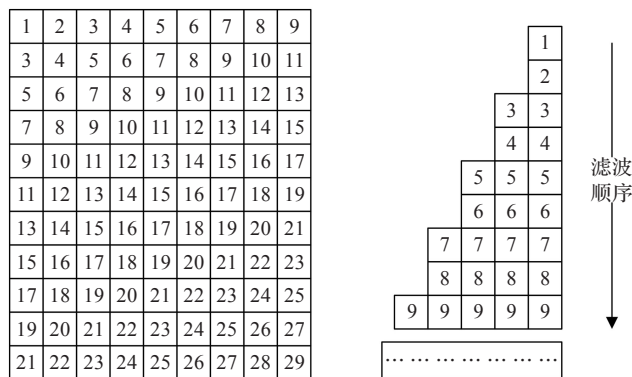


图8 宏块并行滤波处理的顺序

由图8可知,随着滤波的进行,能够并行处理的宏块逐渐增多,大大提高了图像在滤波环节的处理效率,对于某一特定分辨率的图像,其最大可并行处理的宏块数 MP 可通过如下公式计算出来:

$$MP = \begin{cases} \min(\text{ceil}(\frac{W}{2}), H), \min(\text{ceil}(\frac{W}{2}), H) \leq C \\ C, \min(\text{ceil}(\frac{W}{2}), H) > C \end{cases}$$

其中 C 是可以执行并行滤波的处理器核数, H 和 W 分别代表一帧图像中水平和垂直方向的宏块数。

虽然这种方法可以有效提高滤波环节的并行度和处理效率,但也存在一定的问题:(1)由图8可知,一帧图像在并行滤波的启动阶段,满足可并行处理的宏块数较少,且增长较慢,导致刚开始滤波时并行度较低。(2)由图6可知,每个宏块需要依赖其他三个相邻的宏块,所以当一处理器核在处理一个宏块时,需要从其他三个处理器核上获取其所依赖的宏块的数据信息,这就造成了一定的处理器核间通信的开销,影响解码器整体性能的进一步提升。

3.3 宏块级的并行滤波再优化

为了解决上述问题,在宏块级的滤波处理层次上通过对边缘滤波顺序的优化,改善滤波过程的整体效率。

图9表示了滤波时,当前宏块与相邻宏块之间的数据依赖关系及对应的滤波边缘。当前宏块上方相邻的四个 4×4 块为 $T1 \sim T4$, 左边相邻的四个 4×4 块为 $L1 \sim L4$, $V1 \sim V7$ 为垂直滤波边缘, $H1 \sim H6$ 为水平滤波边

缘,其中 $V1, V5, V7$ 和 $H1, H5$ 为宏块之间的滤波边缘。由宏块滤波时的数据依赖性可知,当 $T1 \sim T4$ 和 $L1 \sim L4$ 的经过相应的边缘滤波处理后,当前宏块便可以进行滤波处理。

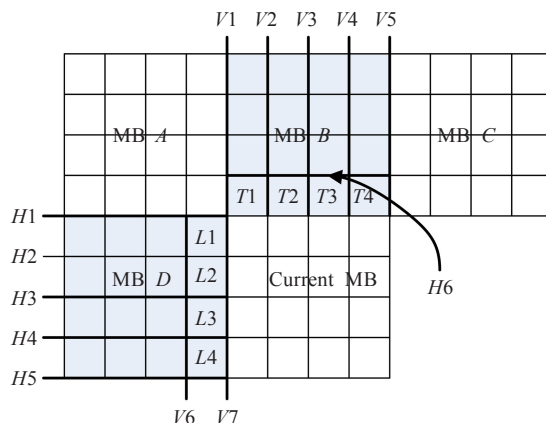


图9 宏块所依赖的相邻滤波边缘

当边缘 $V1 \sim V5$ 和 $H6$ 处理完毕后 $T1 \sim T4$ 中的数据便不再变化,当边缘 $H1 \sim H4$ 和 $V6$ 处理完毕后 $L1 \sim L4$ 中的数据不再变化,此时当前宏块滤波所依赖的数据都已准备完毕。虽然在后续的宏块滤波过程中,当滤波边缘 $H5$ 时, $L4$ 的数据仍会发生变化,但当前宏块已经处理完毕,不会对其造成影响。由于一般所采用的边缘滤波顺序为图5中所示的顺序,边缘 $V5$ 属于宏块 C , 因此当前宏块对于宏块 B, C, D 均有依赖关系。

根据上述分析,采用新的边缘滤波顺序,如图10所示。

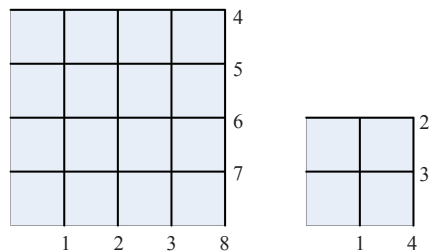


图10 新的边缘滤波顺序

根据新的边缘滤波顺序,边缘 $V5$ 属于宏块 B 而不是 C , 因此在宏块 B 滤波结束后, $T1 \sim T4$ 的数据便已准备完毕,满足当前宏块滤波的数据依赖关系,宏块 C 无法再通过边缘 $V5$ 来影响当前宏块的滤波过程,当前宏块对于宏块 C 的依赖关系也就不存在了,新的宏块之间的依赖关系如图11所示。

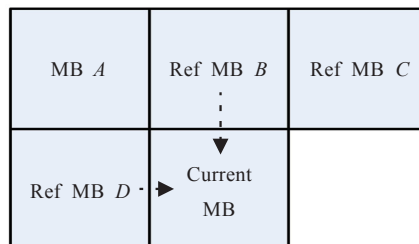


图11 新的宏块之间依赖关系

采用这种新的边缘滤波顺序,可以在保证数据正确性和依赖性的前提下,减小宏块之间的依赖关系,有效提高了滤波阶段的数据并行性,图像中可并行滤波的宏块顺序也发生了相应的改变,新的并行滤波顺序如图12所示。

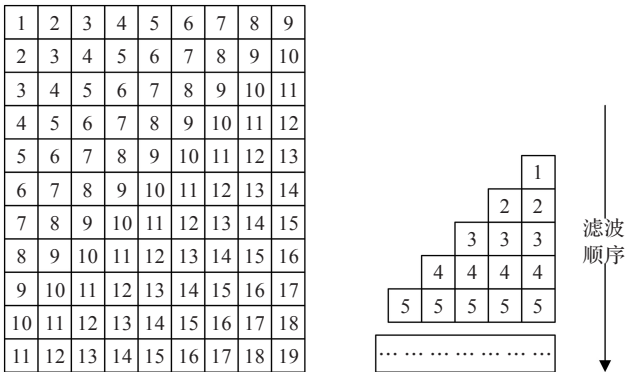


图12 经过优化的宏块并行滤波顺序

由图12可知,通过在宏块的滤波边缘上采用新的处理顺序,和图8相比,数据的可并行性进一步提高,在一帧图像的滤波开始阶段便有较好的并行性,且可并行处理的宏块数增长较快。同时,由于所依赖宏块数的减少,处理器之间的通信开销也随之降低,有效提高了滤波阶段的解码器性能。

3.4 算法在TilePro64平台上的具体实现

在TILERA公司的TilePro64平台(867 MHz主频、64核处理器)上,通过将上述基于功能模块的行级并行算法与并行滤波算法结合起来,实现多层次并行解码算法,整体的解码架构如图13所示。

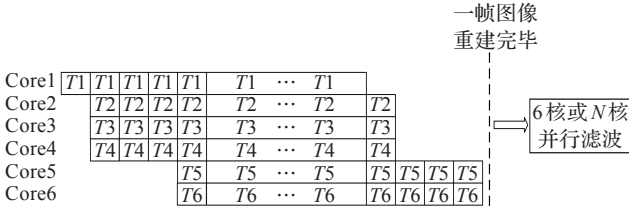


图13 算法在TilePro64平台上的解码架构

在TilePro64平台上利用6个处理器核进行并行解析和重建,由于解码架构与上文所提的行级并行算法不同,功能模块二次划分的策略也相应的有所改变,各核具体任务划分如表3所示。等一帧图像完全重建完毕后,再根据上述的滤波并行算法,用这6个处理器核共同对整帧图像进行并行滤波,将同一时刻可并行处理的宏块均分到6个核上进行处理,每个核滤波一个或多个宏块。

表3 六核任务分配

任务	功能模块
Task1	码流解析+熵解码
Task2	MC1/3 PredIntra1/3
Task3	MC2/3 PredIntra2/3
Task4	MC3/3 PredIntra3/3
Task5	IQ1/2+IDCT1/2+BS1/2
Task6	IQ2/2+IDCT2/2+BS2/2

当处理运算量较大的码流时,可以充分利用TilePro64平台的众核特性,为滤波环节分配更多的处理器资源,利用更多的处理器核进行并行滤波处理,这使该算法具有较好的可扩展性。

4 实验结果和分析

在TilePro64平台上实现上述多层次并行解码算法,并利用平台自带的线性汇编函数对其进行优化后,解码不同分辨率和不同码率的测试码流,将其与TILEPro64平台上单核版本的解码器解码结果进行比较,测试结果如表4,表5所示。

由表4,表5可知,相比于单核版本的解码器,这种多层次并行解码算法的解码效率和并行程度得到有效的提高,算法优化效果显著。

文献[5]在配置六核进行并行解码时,加速比最高为5.56,而本文算法对于码率为8 Mb/s的1 080P全清码流,加速比最高可达到10.01,相比于文献[5],并行加

表4 不同分辨率和码率下的单核解码效率 (f·s⁻¹)

码率/(Mb·s ⁻¹)	720像素×576像素	1 280像素×720像素	1 920像素×1 080像素
1	20.13	9.51	4.68
2	18.88	8.78	4.14
4	17.94	7.98	3.65
6	16.66	7.42	3.42
8	15.78	6.79	3.26

表5 不同分辨率和码率下的6核并行算法解码效率 (f·s⁻¹)

码率/(Mb·s ⁻¹)	720像素×576像素	1 280像素×720像素	1 920像素×1 080像素
1	154.27	76.52	40.46
2	146.21	73.66	37.86
4	139.17	71.07	35.64
6	131.96	66.03	33.87
8	126.02	61.27	32.63

速比提高了80%,另外,对于高分辨率和高码率码流,文献[12]和文献[13]的并行算法会因为滤波环节的计算复杂度增加导致解码性能下降,对于1 080P的测试码流解码帧率只能达到23 f/s左右,而本算法由于对滤波环节进行了深层次的并行优化,解码1 080P码流的帧率可达到32 f/s,可实现全高清码流的实时解码。

图14比较了不同分辨率和码流情况下,该算法并行加速比的变化情况。文献[14]和文献[15]会由于测试码流码率和分辨率的增高,运算复杂度增加而导致并行加速性能的下降,其中文献[15]在高分辨率高码率码流测试条件下并行加速比衰减30%~33%,而本算法随着测试码流分辨率和码率的增大,多层次并行处理的优势充分体现出来,并行加速比相应地有所提高。

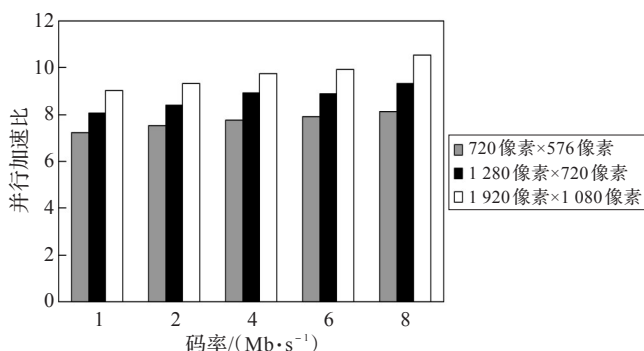


图14 不同分辨率和码率的并行加速比

5 结束语

本文提出了一种多层次并行解码算法,首先设计出基于功能模块的行级并行算法,并通过功能模块的二次划分优化了处理器核间的负载均衡问题,再针对开销较大的滤波环节,利用宏块之间滤波时的依赖关系,提出新的边缘滤波顺序,进行多核并行处理,有效解决了滤波时的性能瓶颈,提高了整体的解码效率。通过在TILEPro64平台进行实验测试,本文所提出算法解码效率高,具有较高的并行加速比,可扩展性较好,适用于各种分辨率规格的码流,尤其对于高码率的1 080P全高清码流可以达到30 f/s以上的实时解码的要求。

参考文献:

- [1] Schaefer R, Wiegand T, Schwarz H. The emerging H.264/AVC standard[EB/OL]. (2003-05-10). http://www.ebu.ch/trev_293-schaefer.pdf.
- [2] Ravasi M. JVT D153r1-L-2002 A computational complexity comparison of MPEG4 and JVT codes[S]. 2002.
- [3] Lee J Y, Lee J J, Park S M. Multi-core platform for an

efficient H.264 and VC-1 video decoding based on macro-block row-level parallelism[J]. IET Circuits, Devices & Systems, 2010, 4(2): 147-158.

- [4] Peng Xiulian, Xu Jizheng, Zhou You. Highly parallel line-based image coding for many cores[J]. IEEE Transactions on Image Processing, 2012, 21(1): 196-206.
- [5] Azevedo A. Parallel H.264 decoding on an embedded multicore processor[EB/OL]. (2009-03-11). http://www.docin.com/p-90474_766.html.
- [6] Chong J, Satish N, Catanzaro B. Efficient parallelization of H.264 decoding with macro block level scheduling[C]// Proceedings of the 2007 IEEE International Conference on Multimedia and Expo, 2007: 1874-1877.
- [7] Nishihara K, Hatabu A, Moriyoshi T. Parallelization of H.264 video decoder for embedded multicore processor[C]// IEEE International Conference on Multimedia and Expo, 2008: 329-332.
- [8] 郭倩, 陈耀武. 基于功能模块的H.264并行解码算法[J]. 计算机工程, 2010, 36(23): 231-233.
- [9] Wei Y C, Zhang R X, Lin R S. A parallel computing algorithm for H.264/AVC decoder[C]// Proceedings-1st International Conference on Robot Vision and Signal Processing, 2011: 332-335.
- [10] 毕厚杰, 王健. 新一代视频压缩编码标准-H.264/AVC[M]. 2版. 北京: 人民邮电出版社, 2009: 151-153.
- [11] Yang Xiangqiu, Liu Leibo, Yin Shouyi. Mapping deblocking algorithm of H.264 decoder onto a reconfigurable array architecture[C]// 2011 International Conference on Consumer Electronics, Communications and Networks, 2011: 4166-4169.
- [12] Kim M, Song J, Kim D H. Hybrid partitioned H.264 full high definition decoder on embedded quad-core[J]. IEEE Transactions on Consumer Electronics, 2012, 58(3): 1038-1044.
- [13] Sohn K H, Baik H, Kim J T, et al. Novel approaches to parallel H.264 decoder on symmetric multicore systems[C]// IEEE International Conference on Acoustics, Speech and Signal Processing, 2009: 2017-2020.
- [14] Wang S W, Yang S S, Chen H M. A multi-core architecture based parallel framework for H.264/AVC deblocking filters[J]. Journal of Signal Processing Systems, 2009, 57(2): 195-211.
- [15] Chi C C, Juurlink B, Meenderinck C. Evaluation of parallel H.264 decoding strategies for the cell broadband engine[C]// Proceedings of the International Conference on Supercomputing, 2010: 105-114.