

文章编号:1001-9081(2005)12-2798-02

基于单指令多数据技术的 H. 264 编码优化

朱林,冯燕

(西北工业大学电子信息学院,陕西西安 710072)

(zlhfc@yahoo.com.cn)

摘要:简要介绍了单指令多数据(Single-Instruction Multiple-Data, SIMD)指令系统,并用该技术对 H. 264 中的整数离散余弦变换(Discrete Cosine Transform, DCT)、量化、插值和运动估计等模块进行了优化,实验结果表明优化后程序的编码速率可达到 30fps 左右,比未优化时速度提高了 6~8 倍。

关键词:H. 264;单指令多数据;整数离散余弦变换;插值;绝对误差和

中图分类号:TN919.81 **文献标识码:**A

Optimization of H. 264 encoder based on SIMD technology

ZHU Lin, FENG Yan

(School of Electronic and Information, Northwest Polytechnical University, Xi'an Shaanxi 710072, China)

Abstract: The SIMD (Single-Instruction Multiple-Data) instruction system was introduced, and further Integer DCT (Discrete Cosine Transform), quant, interpolation and motion estimation of H. 264 were optimized with the SIMD technology. The experiment indicates that the encoding speed of program after optimization reaches about 30fps and the speed has been improved by 6~8 times.

Key words: H. 264; Single-Instruction Multiple-Data (SIMD); Integer DCT (Discrete Cosine Transform); interpolation; Sum of Absolute Difference (SAD)

0 引言

H. 264 是适应于低码率传输的新一代视频压缩标准,此标准被称为 ITU-T 的 H. 264 协议或 ISO/IEC 的 MPEG-4 高级视频编码部分(AVC)。H. 264 标准的编码框架与现有的视频编解码框架一致,都是基于块的混合编码方法,但它用了全新的编码技术,如帧内预测、可变块大小运动补偿、多参考帧预测、1/4 像素精度的运动估计、4×4 整数 DCT、去块滤波器等,使其性能远远高于其他标准,但 H. 264 的高效性是建立在其实现的高复杂度基础上的,比如解码器复杂度是 H. 263 解码器的 3 倍左右,而编码器的复杂度更是 H. 263 的 10 倍多。所以在保证图像质量的基础上,尽可能的加快运算速度,就成为当前研究的新热点。本文主要就是在使用优化算法的基础上,利用 SIMD 中的 MMX (Multi Media Extensions)、SSE2 (Streaming SIMD Extensions2)以及 SSE3 等三个指令集对程序中比较耗时的几个函数如 4×4 整数 DCT、量化、半像素插值和 SAD (Sum of Absolute Difference) 进行优化,对各种视频序列测验均表明优化后编码速度比未优化提高 6~8 倍。

1 SIMD 指令介绍

为了加快多媒体数据处理速度,Intel 推出了单指令多数据并行加速技术(SIMD),SIMD 技术的核心就是一条指令能同时完成多条相同的操作。

目前,IA-32 的 SIMD 指令包括 MMX、SSE、SSE2、SSE3 等几级,它们都是在原来的处理器指令集的基础上添加的扩展指令集,都是 SIMD 指令。其中 MMX 只能在整数上支持 SIMD,而 SSE 指令增加了单精度浮点数的 SIMD 支持,SSE2

则可以支持 64 位双精度浮点运算。MMX 中没有定义新的寄存器,SSE 定义了 8 个全新的 128 位寄存器,每个寄存器可以存放 128 位的整数或浮点数。SSE2 可以分为 MMX 和 SSE 两部分,它又在 SSE 基础上增加了对双精度浮点数的支持和一些新指令。最新的 SSE3 与 Intel Prescott 处理器一同发布不久,共 13 条指令,主要增强了视频解码、3D 图形优化和超线程性能。

2 H. 264 的优化设计

图 1 是 H. 264 的编码器框架,树型运动补偿、高精度运动估计、整数 DCT 等新技术的采用需要进行大量的重复性计算,增加了运算复杂度。本文对 DCT、量化以及在运动估计和运动补偿中用到的插值和 SAD 这三个重复计算次数最频繁模块使用 SIMD 技术进行了优化。

2.1 整数 DCT 和量化

H. 264 为了在反变换端能得到与源数据相同的输出,避免 DCT 中浮点运算带来的失真现象,变换核矩阵采用了基本单位是 4×4 的整数变换,这也是为了和运动补偿中的最小的 4×4 块相适应。本文采用了蝶形快速算法的思想来实现 DCT,这样只需要使用加法、减法和移位指令就能实现矩阵乘法,见图 1,其中 X0~X3 分别代表源矩阵经转置后的 4 行像素。

因为扩展寄存器只能一行一行地对数据进行读取和运算,所以首先对输入的源矩阵进行转置,对转置后的矩阵利用蝶形快速算法的思想进行一次行变换,然后对行变换后的矩阵再进行一次转置,最后再进行一次行变换。转置可以用交错指令 punpckhwd、punpcklwd、punpckhdq 和 punpckldq 完成,行变换可以用 paddw(组加)、psubw(组减)和 psllw(组逻辑左

收稿日期:2005-06-30;修订日期:2005-09-04

作者简介:朱林(1980-),男,山东济宁人,硕士研究生,主要研究方向:视频压缩、流媒体技术;冯燕(1963-),女,陕西西安人,教授,主要研究方向:数据压缩、图像处理。

移)实现,见图 3,反变换时使用 psraw(组逻辑右移)来实现乘 1/2。而对 DC 系数采用了哈达玛变换,这时只需要用 paddw 和 psubw 就能实现。

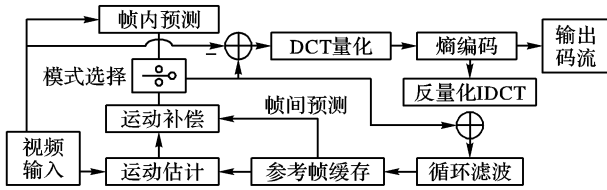


图 1 H.264 编码器框架

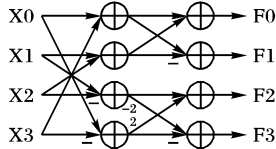


图 2 一维蝶形行变换

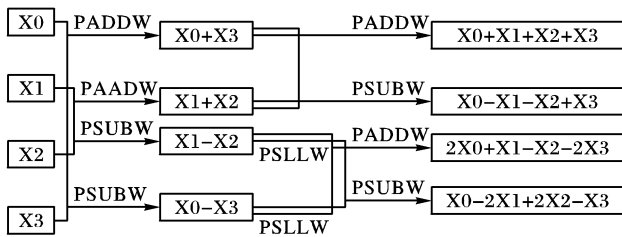


图 3 行变换过程

整数 DCT 后的尺度调整在量化过程中完成,这样减少了乘法次数,降低了算法复杂度。式(1)是量化公式, $Y_{i,j}$ 是 DCT 变换系数, q_{ount} 是伸缩矩阵, f 是修正参数,对于帧内编码宏块 $f = 2^{q_{\text{bits}}}/3$,对于帧间编码宏块 $f = 2^{q_{\text{bits}}}/6, q_{\text{bits}} = 15 + \text{floor}(QP/6)$ 。通过 SSE2 中的 pmullw, pmulhw, paddd 和 psrad 就可以实现量化过程,其中取整和取余可以使用 cdq 扩展为 eax 存商, edx 存余数来实现。

$$|Z_{i,j}| = \text{round}(|Y_{i,j}| \times q_{\text{ount}} + f) \gg q_{\text{bits}} \quad (1)$$

2.2 半像素插值

H.264 中运动向量的位移精度采用 1/4 像素精度进行预测,即在整数像素搜索得到的最优值周围再进行精度更高的 1/2 像素点和 1/4 像素点的搜索。步骤就是首先通过水平或垂直方向的相临近的整数像素点值使用 6-tap 滤波器 (1, -5, 20, 20, -5, 1) 计算出中间结果 tmp,然后用 tmp 除以 32 舍入取整(0 到 255 之内)作为 1/2 像素点的值,如式(2)。式中以要求的半像素点 b 为中心水平或垂直方向取临近的 E、F、G、H、I、J 六个整像素点。利用整像素点和半像素点线性插值获得 1/4 像素值。

$$b = \text{round}((E - 5F + 20G + 20H - 5I + J + 16)/32) \quad (2)$$

由于 movq 只能存储连续的像素,所以需要利用 punpcklwb

和 punpckhbw 把全 0 寄存器插到连续的像素中间,用 movq2dq 得到放大图像的效果,给插值留下空间。6-tap 滤波器可以用 pmullw, paddw 和 psraw 来实现 8 个像素的同时插值。

2.3 SAD 计算

H.264 中使用了可变块大小树状运动补偿技术,选择运动补偿块的大小和形状比以前的标准更灵活,亮度运动补偿块可以从 16×16 取到 4×4 共 7 种。这 7 种帧间预测模式使计算量变得非常大,为了简化计算,可以根据各种块模式 SAD 的统计特性预先做模式判决,减少不必要的块模式的预测, SAD 就是运动估计时的失真判断准则,过程就是取源宏块和预测宏块对应元素绝对差的和。

在优化过程中为了降低 128bit 内存访问的冲突,本文用 SSE3 指令中的 lddqu 指令代替 movdqu, lddqu 是一个比较快速的未对齐字节读取指令,它的做法是在读取一未对齐 16 字节时,会把它变成一个已对齐的 32 字节读取,然后从 32 字节的块中提取出正确的 16 字节数据,这样做的目的是避免超高速缓存线冲突的情形。SSE2 指令集中的 psadbw 指令,可以同时完成 SAD,需要注意的是在对 16 字节同时操作时,它是把低 8 字节和高 8 字节的绝对差分别求和,得到两个字,所以需要 psrldq 指令和 paddd 指令把高 8 字节加到低 8 字节,便于在通用 32 位寄存器上存储,见图 3。鉴于对小块数据采用 4 行凑成一行的办法会增加过多的指令时钟,我们对 4×4 小块每行分别计算 SAD。

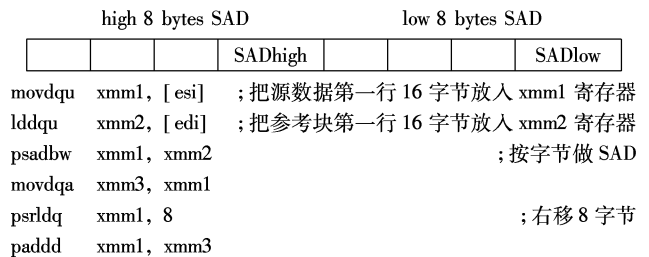


图 4 psadbw 存储方式及应用程序

3 优化效果

实验在 P4 2.6G, 512M 内存的环境下进行,采用 JVT 的 JM8.6 作为实验平台,程序中使用菱形快速搜索以及帧间快速模式选择,未使用 B 帧预测,帧内编码只使用 4×4 块预测模式,在上述平台基础上利用 MMX、SSE2 和 SSE3 指令集分别对 DCT、量化、插值和 SAD 进行了指令优化。实验使用 Intel VTune(TM) Performance Analyzer 7.0 对 QCIF (176 × 144) 和 CIF (352 × 288) 两种序列标准进行测试,分别测出优化前后三个模块消耗的时钟周期(表 1)。表 2 是通过亮度 PSNR 和编码速率两个方面对优化前后的程序做以比较。

表 1 各模块优化结果

		DCT 和量化		插值		SAD	
		未优化	SIMD 优化	未优化	SIMD 优化	未优化	SIMD 优化
Foreman. QCIF298 帧	Clockticks(M)	3679.2	548.6	4113.4	629.4	7029.7	1125.6
Claire. CIF 167 帧	Clockticks(M)	3003.2	449.8	2033.2	343.2	9525.4	1083.8
Bus. QCIF 148 帧	Clockticks(M)	1173.2	296	1500.5	314	4256.3	517.6

表 2 总体优化结果

	Foreman. QCIF		Claire. CIF		Bus. QCIF	
	未优化	SIMD 优化	未优化	SIMD 优化	未优化	SIMD 优化
PSNR(Y) (dB)	34.8846	34.5142	34.3694	34.9840	32.066	31.095
编码速率(fps)	5.11	33.69	3.16	27.71	4.55	30.38

中可以看出,本文提出的算法比较稳定,缓冲区充满的状态一直控制在 50%~60%之间,且码率波动范围约在 $\pm 5\%$ 左右,明显比 JM 程序的范围小很多;而 JVT-H017 实现的算法在三个序列中缓冲区都有上溢和下溢现象发生,且码率波动范围较大。在图像的主观质量方面,本算法的重建图像没有出现马赛克现象,而 JVT-H017 实现的算法在遇到场景变换时出现了马赛克现象,影响了图像的主观质量。

表 1 码率控制算法的测试条件

序列	测试条件				
	GOP 结构	GOP 长度	序列长度	RDO	设定码率
Jet. yuv	IPPP...	15	550	OFF	6000000bps
Night. yuv	IPPP...	15	600	OFF	6000000bps
Raven. yuv	IPPP...	15	600	OFF	6000000bps

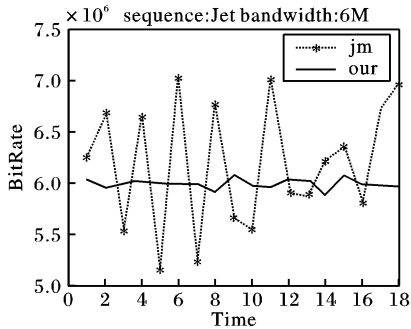


图 1 Jet 序列码率变化曲线

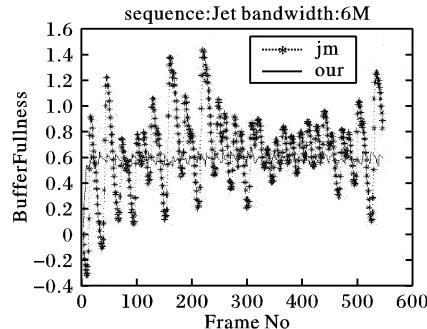


图 2 Jet 序列缓冲区充满度变化曲线

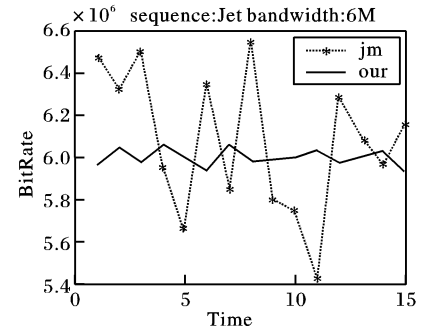


图 3 Night 序列码率变化曲线

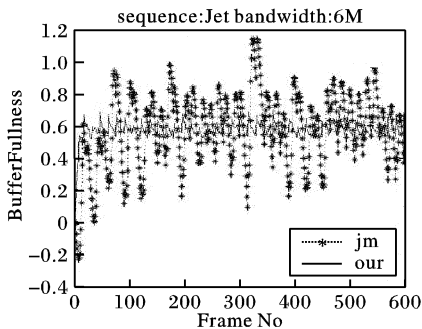


图 4 Night 缓冲区充满度变化曲线

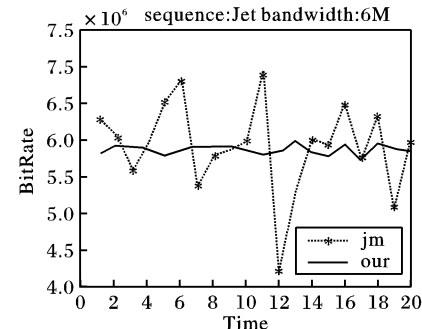


图 5 Raven 序列码率变化曲线

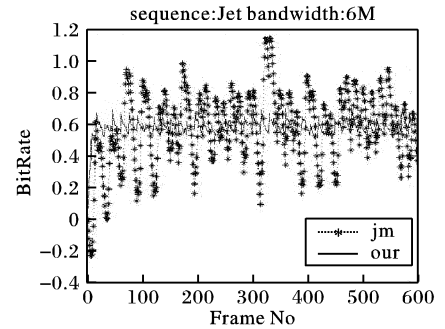


图 6 Raven 序列缓冲区充满度变化曲线

4 结语

H. 264/AVC 是最新的国际视频编码标准,与其他编码标准一样,对于编码器并未作出规定,码率控制作为编码器中的关键部分在实现上也有了很大的灵活性,所以不少学者和机构都提出了各自的码率控制策略。本文在阐述 H. 264 的参考程序中采用的 JVT-H017 提案中的算法的基础上,提出了一种简单有效的帧一级的码率控制算法。以高清图像作为实验序列,控制效果较 JVT-H017 有显著的提高,在其他尺寸的图像序列(如:QCIF、CIF 等)的实验也有同样的效果。实验证明该算法实现简单,控制比较稳定,图像质量得到了保证。

参考文献:

- [1] WIEGAND T, SULLIVAN GJ, BJONTEGAARD G, *et al.* Overview of the H. 264/AVC Video Coding Standard[J]. IEEE transactions on circuits and systems for video technology, 2003, 13(7): 560-576.
- [2] 贺军. 一种快速有效的帧级码率控制算法[J]. 电视技术, 2003, (9).
- [3] MA SW. JVT-H017. doc: Proposed Draft of Adaptive Rate Control. 8th Meeting[R]. Geneva, 2003.
- [4] VETRO A, SUN HF, WANG Y. MPEG-4 Rate Control for Multiple Video Objects[J]. IEEE transactions on circuits and systems for video technology, 1999, 9(1).

(上接第 2799 页)

从表 1 的测试结果可以得出,优化后这三个模块所消耗的时钟周期减少为优化前的 12%~20%。表 2 的整体优化结果表明经过 SIMD 技术优化后,编码速率比未优化时提高了 6~8 倍,但在图像场景改变时也会出现峰值信噪比 PSNR 下降的情况,最大降幅仅为 0.3dB。

经过各种测试,说明本文采用 SIMD 技术对 H. 264 进行优化,编码效率得到了很大的提高,并且保证了图像质量,完全满足 H. 264 的实时要求。

参考文献:

- [1] WIEGAND T. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H. 264 | ISO/IEC 14496-10 AVC) [S]. JVT-G050, 2003.

- [2] IA-32 Intel Architecture software Developer's Manual Volume1: Basic Architecture[S]. 2001.
- [3] ZHOU Z, MING-TING SUN, YUH-FENG HSU. Fast Variable Block-size Motion Estimation Algorithms based on Merge and Split Procedures for H. 264/MPEG-4 AVC[A]. Proceedings of IEEE International Symposium on Circuits and Systems[C]. 2004. 725-728.
- [4] 魏芳,李学明. H. 264 中变换与量化的 SIMD 优化[J]. 计算机工程与应用, 2004, (17): 24-27.
- [5] 张颖,王兴国,刘济林. 基于单指令多数据流技术的视频信息处理优化[J]. 数字电视与数字视频, 2003, (11): 40-43.
- [6] 李蕊,陈浩,姜显明. SSE 在多媒体数据处理中的应用[J]. 计算机应用, 2002, 22(7): 99-103.