

基于 TMS320DM642 的 H.264 六抽头滤波插值的优化*

·论文·

何苏勤, 延瑾瑜

(北京化工大学 DSP 与嵌入式系统研究室, 北京 100029)

【摘要】 在分析了 H.264 压缩编码算法中各个模块耗时情况的基础上, 结合 TMS320DM642 的硬件特性对耗时占总程序将近 10% 的六抽头滤波插值部分进行优化, 分别采用了使用内联函数、数据打包、改变垂直方向插值的数据结构、EDMA 乒乓缓冲等方法。实验结果证明, 在相同编译选项的情况下, 优化以后插值部分的程序运行速度提高了近一倍。

【关键词】 H.264; 六抽头滤波插值; TMS320DM642; 优化

【中图分类号】 TN919.8

【文献标识码】 A

Optimization of 6-tap Filter Interpolation in H.264 Based on TMS320DM642

HE Su-qin, YAN Jin-yu

(DSP and Embedded System Institute, Beijing University of Chemical University, Beijing 100029, China)

【Abstract】 The time-consuming of all modules in H.264 is analyzed, and the profile results show that the 6-tap filter interpolation takes almost 10% time in H.264 encoder. This procedure according to the hardware feature of the TMS320DM642 chip is optimized. The optimizing methods include using intrinsic, packed-data processing, changing the data-base of vertical interpolation procedure and using EDMA PING-PONG buffering. Experiment results show that at the same build options, the processing speed of sub-pixel interpolation is improved by twice.

【Key words】 H.264; 6-tap filter interpolation; TMS320DM642; optimization

1 引言

相对以往的视频编码标准(H.263, MPEG-4 等), H.264 采用了许多新的先进技术, 具有良好的网络亲和力、1/4 像素运动估计、4x4 整数 DCT 变换、多参考帧预测等^[1], 因此在相同的码率下用 H.264 标准进行编码能够获得更高的视频质量。但 H.264 视频压缩算法的优越性是以复杂度为代价的。TI 公司专为视频编码设计的 C64 系列芯片具有强大的功能, 包括两级 Cache、超长指令字结构、64 通道的 EDMA、专为视频处理设计的扩展指令集等^[2-3], 这些特性使得 H.264 的实时编码成为可能。

2 H.264 编码器的耗时分析

H.264 的开源代码主要有 3 种:

1) JM: 官方测试代码, H.264 的各种新特性实现很全面, 但结构复杂, 适合进行学术上算法的研究, 但不适合实时应用的开发;

2) X264: 网上自由组织联合开发, 兼容 H.264 标准码流, 注重实用, 与 JM 相比, 在不明显降低图像质量的基础上, 大大降低了计算复杂度;

3) T264: 中国视频编码自由组织联合开发, 速度较

快, 但是在与 JM 相同码率的条件下, PSNR(峰值信噪比) 下降较多, 平均下降 1~2 dB。

经过对这 3 种源代码进行测试, 本文选择 X264 作为开发源代码。

CCS3.1 是 TI 公司 TMS320 系列 DSP 的集成开发环境 (IDE), 具有代码剖析 (profile) 的功能, 可以准确统计应用程序各个模块的执行时间。

对 X264 在 baseline 档次作如下参数设置:

1) 运动估计方法和精度: 用菱形法先进行整像素的动态搜索, 再进行半像素和 1/4 像素精度的精细搜索;

2) 量化参数: $QP=26$ 。

CCS 编译选项设置: -pm, -op2, -o3, -mt, -mh。

-pm: 程序级优化, 所有的源文件都被编译成一个中间的文件, 使得编译器可以从整体对程序进行优化;

-op2: 通常与 -pm 一起使用, 表明没有源程序以外的因素可以改变变量值或调用函数;

-o3: 最高级别的文件级优化, 对循环进行多种优化, 包括软件流水、循环展开、SIMD;

-mt: 消除存储器相关性;

-mh: 消除冗余循环, 移除 prologs 和 epilogs, 降低寄

* 国家自然科学基金项目 (60473032)

存器压力。使用该选项可获得较好的代码长度和性能。

在进行了以上各项设置后,在 CCS3.1 开发环境下分别对 5 个标准测试视频序列 Foreman, Flower, Football, Tempete, Stefan 用移植好的 X264 进行编码,编 50 帧。用 CCS3.1 对 X264 源代码进行剖析,结果见表 1。表中平均耗时百分比指该函数在 5 个测试序列的编码过程中耗时百分比的平均值。

表 1 X264 主要函数耗时剖析结果

函数	函数功能	平均耗时 百分比/%
mc_hh(), mc_hv(), mc_hc()	六抽头滤波插值	9.192
get_ref()	获取运动估计所需的参考帧数据	8.971
block_residual_write_cavlc()	CAVLC 编码	8.746
x264_macroblock_cache_load()	读入周围宏块信息	5.424
bs_write_vlc()	VLC 编码	4.548
x264_pixel_sad_16x16()	SAD 计算	4.354
x264_macroblock_encode()	编码一个宏块	4.106
x264_macroblock_cache_save()	保存一个宏块编码完的数据	2.988

由表 1 可以看出,在整个 H.264 编码过程中,六抽头滤波插值过程(mc_hv(), mc_hh(), mc_hc()),包括水平、垂直、对角 3 个方向插值,耗时 9.192%,成为实时编码的瓶颈之一。

3 X264 的六抽头滤波插值过程简介

H.264 算法中需要对重建图像进行插值,以便进行 1/2 和 1/4 像素运动估计,插值位置如图 1 所示。

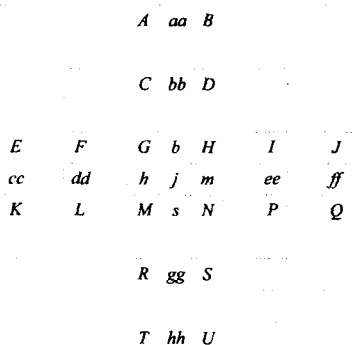


图 1 H.264 中亮度六抽头滤波插值位置

1/2 像素数据是由整像素数据经过六抽头 FIR 滤波器插值而成,插值系数为 $(1/32, -5/32, 5/8, -5/32, 1/32)$,图 1 中半像素点 b 的值可以从 6 个水平方向的整像素点数据 E, F, G, H, I, J 得出

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

类似的, h 点的值可以从 6 个垂直方向的整像素点数据 A, C, G, M, R, T 得出。水平方向和垂直方向的像素

点都完成插值以后,对角方向的像素点的值可由 6 个水平方向或垂直方向已完成插值的半像素点经过六抽头滤波得出,例如, j 点的值可由 cc, dd, h, m, ee 和 ff 或由 aa, bb, b, s, gg 和 hh 得出。对角方向的值由水平方向半像素点和垂直方向半像素点插值的结果一致,并且此时的水平或垂直方向的半像素点的值应该是插值以后没有经过四舍五入的值^[1]。

六抽头插值滤波复杂度较高,但是得出的值与整像素点的数据较为切合,因此可取得更好的运动估计性能。

X264 中以宏块为单位进行 3 个方向的半像素插值,对角方向的插值点采用水平方向半像素点经过垂直方向插值得到^[1]。

4 X264 中六抽头滤波插值的优化

4.1 使用内联函数和数据打包

TI 公司提供的 C6000 系列 DSP 编译器提供内联函数(intrinsics)以便进行应用程序的优化^[2-4]。

在 X264 的插值过程中,经过插值的数据可能超出了 $(0, 255)$ 的范围,因此需要调用 clip1 函数来进行限幅,clip1 函数的作用是将小于 0 的数据赋值为 0,将大于 255 的数据赋值为 255,其余值不变。本文将 clip1 函数改为内联函数 _min2 和 _max2 的组合。在进行六抽头滤波器插值时采用 _dotpsu4 内联函数,同时将插值系数设定为 2 个常数来参与 _dotpsu4 的运算,在同时访问内存中的多个数据时采用 _mem4() 等内联函数。

4.2 使用 QDMA 调整垂直方向插值数据结构

进行上述内联函数优化后,水平和对角方向插值速度的改进比垂直方向明显得多,原因在于进行垂直方向插值时垂直方向的 6 个整像素点在内存中的存放不连续,无法使用 _dotpsu4 等内联函数,笔者提出使用快速 DMA(QDMA)改变数据结构的方法来解决该问题^[5-6]。

QDMA 通常应用于整块数据的搬移,其操作是与 CPU 工作同步进行的,因此可以隐藏数据搬移的时间。本文使用 QDMA 将需要插值的整像素数据在进行垂直方向插值之前重排数据结构,即进行行列互换,使得可以参加六抽头滤波的 6 个数据在内存中连续存放,这样便可以采用与水平插值相同的优化方法,方便使用 _dotpsu4 内联函数,从而达到数据打包运算,提高程序运行速度的目的。

4.3 使用 EDMA 乒乓缓冲进行内存调度

TMS320DM642 具有两级片内存储器结构,包括 L1P (一级程序 Cache, 16 Kbyte)、L1D (一级数据 Cache, 16 Kbyte) 和 L2 (程序和数据共用, 256 Kbyte)。其中 L2 可配置为 SRAM、Cache 或者 SRAM 与 Cache 的组合。DM642 片上有限的内存资源不可能存储所有的程序和

数据,因此必须外扩 SDRAM。与片上 SRAM 相比,片外 SDRAM 的存取速度要慢得多,必须采用增强 DMA (EDMA)来将数据提前搬运到片上 SRAM。

EDMA 乒乓缓冲的基本思想为在 SRAM 开辟 4 个缓冲区,分别为 2 个输入数据缓冲区和 2 个输出数据缓冲区,EDMA 和 CPU 并行工作时,EDMA 在传输一组输入输出缓冲区的数据的同时,CPU 对另外一组缓冲区的数据进行运算。

在 X264 插值运算中,需要进行 3 个方向的插值,由于垂直方向插值的整像素数据结构进行了调整,因此需要独立于水平方向和对角方向进行操作。水平和对角方向插值过程中输出缓冲区只需开 2 个 16×16 大小的缓冲区用来存放插值完成的水平和对角方向的半像素数据即可,另外,因为边界的插值需要用到相邻宏块的 3 个数据,考虑到传输数据为 8 byte 的倍数可以发挥 EDMA 的最好性能,输入缓冲区需要开辟两组 (16+4×2)×(16+4×2) 大小的区域。垂直方向的插值过程需要在 SRAM 中开辟 2 组输入输出缓冲区,为了节省内存资源,这 2 组缓冲区可以与水平和对角方向插值过程的 4 个缓冲区进行复用,见图 2。

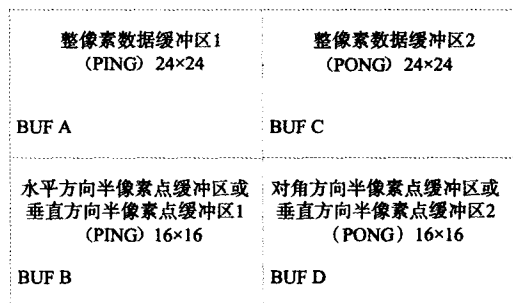


图 2 EDMA 乒乓缓冲示意图

在开始进行水平方向插值之前先用 EDMA 将 24×24 大小的整像素数据搬到 BUF A 中,然后进行水平方向插值,插值结果存在 BUF B 中,同时将以下一宏块为中心的 24×24 大小的整像素数据搬到 BUF C;插值完成以后进行对角方向的插值,插值结果存在 BUF D 中,同时将插值完成的水平方向半像素数据通过 EDMA 写回 SDRAM;在进行下一宏块的水平插值的同时将对角方向的半像素数据写回 SDRAM,同时将以下一宏块为中心的 24×24 大小的整像素数据搬到 BUF A 中,如此循环。

垂直方向插值的 EDMA 传输过程与上述类似。

5 实验结果

对 X264 的六抽头滤波插值过程通过上述几种方法优化以后,在相同的编译选项情况下,优化效果明显,如表 2 所示。

由上表数据可以看出,在经过上述方法优化以后,

表 2 优化前后耗时周期数比较

函数	Incl. Average		Excl. Average	
	优化前	优化后	优化前	优化后
x264_frame_filter()	4 302 092	2 500 131	16 407	43 614
mc_hh()	2 522	1 299	2 522	1 299
mc_hv()	3 199	1 299	3 199	1 299
mc_hc()	3 994	2 415	3 994	2 415

注:Incl.Average 和 Excl.Average 分别表示包含调用别的函数和不包含调用别的函数花费的 CPU 周期数;x264_frame_filter() 函数为六抽头滤波插值的主函数,调用 mc_hh(),mc_hv(),mc_hc()等函数。

mc_hh(),mc_hv()和 mc_hc()等六抽头滤波插值函数速度都有大幅度的提高,虽然 x264_frame_filter()函数的 Excl.Average 有一些增加,但是对于整个插值滤波函数的耗时节省了 42%的时间,速度提高了将近一倍。另外,本文中采用的优化方法对编码的质量(PSNR)和码率都没有影响。

6 小结

笔者在精确分析了 X264 源码中各个耗时模块在总程序中所占比例的基础上,确定了实时编码的瓶颈之一为六抽头滤波插值过程,进而根据 TMS320DM642 的硬件特点对其进行多种方法的优化,取得了明显的效果。对别的费时模块也可进行类似的优化,另外还可以在必要时将 C 代码改写为线性汇编代码,来进一步提高编码效率。本文针对编码效率较高的 X264 提出的优化方法对于嵌入式视频编码器的开发具有一定的应用价值。

参考文献:

- [1] ITU-T. ITU-T Recommendation H.264 Advanced video coding for generic audiovisual services[S]. 2005.
- [2] TI Company. TMS320C6000 Programmer's Guide[EB/OL].[2009-03-10].<http://focus.ti.com/lit/ug/spru198i/spru198i.pdf>.
- [3] TI Company. TMS320C6000 DSP Enhanced Direct Memory Access (EDMA) Controller Reference Guide [EB/OL].[2009-03-10].<http://focus.ti.com/lit/ug/spru234c/spru234c.pdf>.
- [4] 侯金亭. H.264/AVC 编码器子像素插值快速实现方法[J]. 计算机工程与应用,2006,33:94-96.
- [5] 陈晨航,卢官明. 基于 TI DM642 的一种 H.264 插值快速实现方法[J]. 山西电子技术,2007(3):8-9.
- [6] 千宗良,王凯,朱秀昌. 基于 DSP 平台的 H.264 运动补偿解码优化[J]. 南京邮电大学学报:自然科学版,2007(4):80-84.

作者简介:

何苏勤(1957-),女,教授,硕士生导师,主要研究方向为嵌入式系统应用研究,DSP 技术应用研究;

延瑾瑜(1984-),女,硕士生,主研视频压缩编码的 DSP 实现。

责任编辑:任健男

收稿日期:2009-06-25