基于 TMS320DM642 平台的 H.264 编码器优化与实现

The Optimization and Realization of H.264 Encoder Based on the TMS320DM642 Platform

孙乐美* 刘文臣**

SUN Le-mei LIU Wen-chen

摘 要 针对 TI 公司的多媒体处理芯片 TMS320DM642 硬件结构特点，设计了适应实时信号处理的高性能 H.264 编码器架构。采用了先进的快速变换和搜索算法，完成了对代码的逐步优化和调试。实验结果表明，该编码器能实现对各种运动幅度视频数据的快速编码，满足不同运动级别视频信号的实时处理。

关键词 补偿 预测 移植 超长指令字

Abstract Based on the architecture of multimedia chip TMS320DM642 invented by TI Semiconductor, a design of a H.264encoder with high performance for real time signal processing is presented in this paper, which chooses the advanced integer translation and searching algorithm, accomplishes the optimization and debugging for codes. The result of experiment indicates that the encoder can encode all kinds of video data rapidly and can fulfill the real time processing for all kinds of video signals.

Keywords Compensation Estimation Grafting VLIW

1 概述

H.264 协议是由 ITU－T 和 ISO/IEC 活动图像专家组(MPEG)组成的联合视频小组(JVT)共同制定的新一代视频编码标准，在 ITU－T 中称之为 H.264，ISO/IEC 称之为 MPEG－4 的第十部分：先进视频编码(AVC)。H.264 引入了一些新的技术，如多种模式的运动估计、小尺寸的 4×4 整数变换、更精确的帧内预测、统一的可变长编码等，使得编码性能比以往的 MPEG4、H.263[3]等视频编码标准有了明显的提高。

2 TMS320DM642 系统平台架构

TMS320DM642 是 TI 公司在其最高性能的 TMS320C64x 系列 DSP 基础上专门为多媒体应用而设计/开发的 DSP。TMS320DM642 采用 T I 第 2 代超长指令集结构(VelociTI.2)的 TMS320C64x DSP 内核：

二级 Cache(2×16K 字节 1 级程序和数据 Cache，256K 字节二级 Cache)

支持 8 个 8 位和 4 个 16 位并行 MAC 运算，有利于图像处理 64 通道 EDMA，方便高效数据传输主频 600MHz，4800MIPS 峰值处理能力，可实时处理 4 路 CIF 或 1 路 D1 64 位、133MHz 外部存储器接口(EMIF)TMS320DM642。

TMS320DM642 基本系统由 TMS320DM642 外扩的存储器和外设组成，而外扩的存储器和外设均通过 TMS320DM642 的外部存储器接口(EMIF)进行扩展。其次，TMS320DM642 基本系统所必须的外扩资源：SDRAM：4M × 64 位，存放程序和缓存数字视频/音频数据 Flash：4M×8 位，存放固化程序，以便进行 ROM 引导 UART2 ×8×8 位，扩展 2 个异步串口(RS232/RS422/RS485)板上寄存器：n×8 位，若干个 8 位状态/控制寄存器 ATA 硬盘接口：2×8×16 位，本地大容量存储接口。

采用 TMS320DM642 芯片、64MB DDR SDRAM、视频解码器 SAA7115H、视频编码器 SAA7121H、网络控制器、电子硬盘搭建系统平台，具体框图如图 1 所示

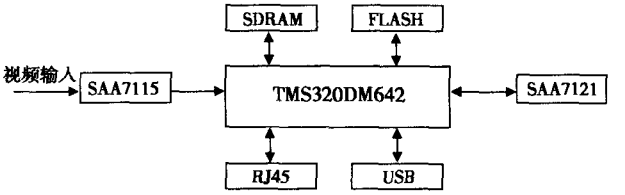


图 1

视频数据发送流程：PAL 视频信号经过视频解码器 SAA7115H 解码变成 CCIR656 标准 4:2:2 数字信号，通过 DMA 方式直接读入内存后由 TMS320DM642 完成数据的压缩，然后通过网络控制器后传到以太网或电子硬盘。

视频数据接收流程：来自以太网或电子硬盘的视频数据经过网络控制器后由 PNX1500 完成数据的解压，解压后的 CCIR656 标准数字信号再通过 DMA 方式传输到视频编码器 SAA7121H 进行编码，最终，编码后的 PAL 视频信号在液晶屏上得到显示。

3 基于 TMS320DM642 平台的 H.264 编码器优化

为实现对视频信号的实时处理，必须对 H.264 编码器进行大量的优化。按照自顶向下、模块化的设计原则，下面主要从“框架设计优化”、“算法选择优化”、“程序代码实现优化”三个层面对 H.264 编码器进行具体的优化和实现。

3.1 框架设计优化

为了实现系统平台的高性能运行，必须从系统整体出发，根

*山东建筑大学 济南 250014

**西安工程大学 西安 710048

据其内部的各模块功能划分设计出综合性能最佳的系统架构。

针对 TMS320DM642 的处理特性，参考 H.264 编码器模型，把编码器分成以下几个模块：视频输入模块、4×4 整数变换模块、运动预测模块、图像重建模块、量化和反量化模块、Zigzag 存储模块、Huffman 编码模块等。为了充分利用 TMS320DM642 的并行处理能力，在整个编解码过程中应尽量减小中间运算环节，避免重复操作。如：整数变换、量化、反量化和整数逆变换是紧密相连的，在实现时就不必分开，可以看作一个函数对整数变换系数直接进行量化进而反量化、整数逆变换，从而避免了将整数变换后的系数存储而在量化时又要重新读取的问题。根据 I 帧、P 帧的不同特点，分别设计出其编码的功能模块，如图 2、图 3 所示

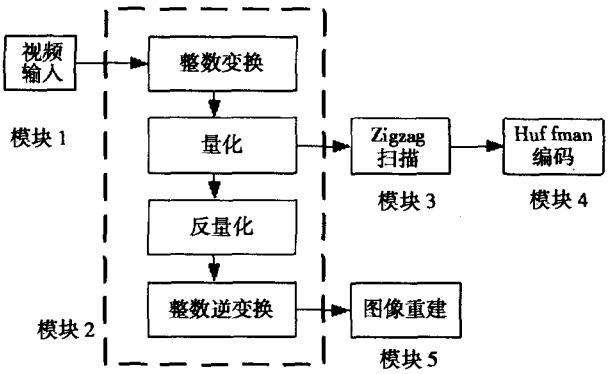


图 2 I 帧功能模块划分

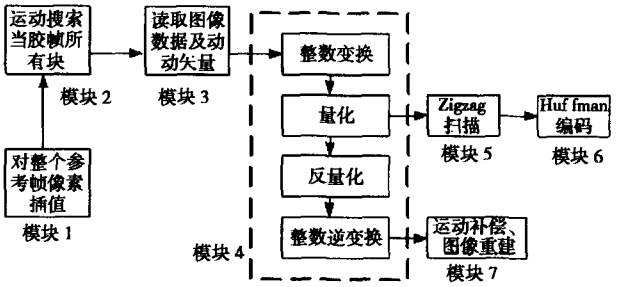


图 3 P 帧功能模块划分

3.2 算法选择优化

在 H.264 编码器中，变换算法、运动估计的搜索算法是耗时较多、计算量较大的模块，要提高编码器的性能就必须采用高效的快速变换和先进的搜索算法。下面主要就这两方面进行具体阐述。

3.2.1 4×4 整数变换

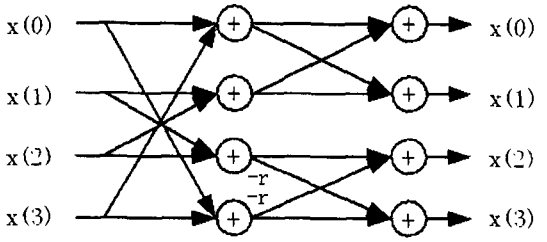
在以往的标准中，如 H.261、H.263、MPEG－1、MPEG－2、MPEG－4 都是采用 8×8 单位块的离散余弦变换(DCT)作为基本变换，而在 H.264 中采用了 4×4 的单位块，不但减小了计算量，而且也减小了运动物体边缘处的衔接误差。另外，以往标准对实数的 DCT，由于在解码端的浮点运算精度问题，会造成解码后的数据失配，进而引起漂移。为此，H.264 对变换矩阵进行了改造，采用整数 DCT 技术，可有效地减少计算量，同时也不损失图像的精确度。

其变换矩阵为：

$$Y=(C_fXC_f^T)\otimes E_f=$$

$$\left\{\begin{bmatrix}1&1&1&1\\2&1&-1&-2\\1&-1&-1&1\\1&-2&2&-1\end{bmatrix}X\begin{bmatrix}1&2&1&1\\1&1&-1&-2\\1&-1&-1&2\\1&-2&1&-1\end{bmatrix}\right\}\otimes\begin{bmatrix}a^2&\frac{ab}{2}&a^2&\frac{ab}{2}\\\frac{ab}{2}&\frac{b^2}{4}&\frac{ab}{2}&\frac{b^2}{4}\\a^2&\frac{ab}{2}&a^2&\frac{ab}{2}\\\frac{ab}{2}&\frac{b^2}{4}&\frac{ab}{2}&\frac{b^2}{4}\end{bmatrix}$$

为了提高整数变换的运算速度，可将矩阵乘法运算改造成两次一维整数 DCT 变换，再对每次一维整数 DCT 变换采用蝶型快速算法，如图 4 所示。



r=1:hadamard 变换；r=2：整数 DCT 变换

图 4 一维快速变换算法

3.2.2 运动估计

运动估计是图像压缩算法中最为耗时的模块，因此必须采用高性能的搜索方法。通过比较当前的快速算法：三步搜索法、二维对数法、钻石搜索法、六边形搜索法，本文采用效率最高的六边形搜索法。六边形搜索法(如图 5 所示)在搜索时采用了六边形搜索模式和小菱形搜索模式相结合的方式。六边形搜索模式有 7 个搜索点(中心点及周围按菱形分布的 6 个围绕点)，小菱形搜索模式有 5 个点(中心点及与中心点垂直、水平相邻的 4 个点)。六边形方式搜索时，先以预测到的搜索中心为中心，进行六边形搜索，计算 7 个点，如果 7 个点中最优点不在六边形的中心，则将六边形的中心移至该点，重复六边形搜索，直到最优点处于其中心为止。然后在该点切换到小菱形搜索模式搜索，共搜索 5 个点后得到最终的搜索结果作为运动估计的最优匹配点。

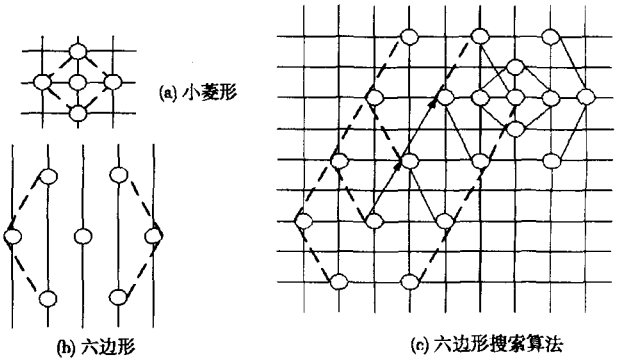


图 5 小菱形、六边形、六边形搜索算法

3.3 程序代码实现优化

3.3.1 优化 C 代码

(1)使用 intrinsics(内联函数)。C6000 编译器提供了许多内联函数，这些函数直接与 C6000 的汇编指令映射，可快速优化 C 代码。优化过程中用到的内联函数包括：_abs( )，_amem4( )

, _amem4_const ( ) , _pack2 ( ) , _packl4 ( ) , _amemd8 ( ) , amemd8_const ( ) , _min2 ( ) , _max2 ( ) , _dotpu4 ( )等,调用这些内联函数大大减少了函数执行时间。

(2)宽内存数据访问和数据打包处理。C6000 访问存储器很费时,为了提高 DSP 的内存数据吞吐率,可用单指令访问内存中连续的多个短字长数据,当程序要对一连串短数据进行操作时,把数据进行打包处理,这样可以减少对内存的访问,打包操作的指令在一个指令周期内可并行处理一组数据,从而提高运行速度。

(3)软件流水

软件流水是一种用于安排循环内指令运行方式,使循环的多次迭代能并行执行的一种技术。编译 C 代码时,使用 - o3 选项,编译器能从程序中收集信息,尝试对程序循环实现软件流水。在 C 代码中,用户不用直接参与软件流水的实现,只要向编译优化器提供信息,使之能较好地编排软件流水就行。因此为改善软件流水.优化过程中主要利用特定的关键字和指令向编译器提供优化信息。如:使用关键字 restrict 消除数据间的相关性;使用并行代码,尽可能把长的有依赖的代码链分解成几个可以在流水线执行单元中并行执行的没有依赖的代码链;用宏指令 #pragma MUST_ ITERATE 告诉编译器有关循环迭代次数的信息等。编译器会根据这些信息安排指令并行,进行软件流水。

(4)某些函数实现过程优化

从代码剖析的结果看,有些函数完成的功能很简单,但调用次数颇多,在这些函数上运行耗费的总时间不少,但它们的运行效率低,优化的余地比较大。比如: T264_median (求 3 个数的中值)修改了其实现过程,简化了求中值时的比较步骤,同时将它改成内联函数,减少函数调用的开销。array_non _zero_count 函数将 for 循环里取数组元素的部分,改成用指针访问且每取完一个值指针自动指向下一个内存单元,这样节省了大量的寻址时间。

3.3.2 线性汇编级优化

线性汇编是 C6000 特有的一种编程语言,介于高级语言和汇编语言之间。在线性汇编代码中不需给出汇编代码必须指出的所有信息,如使用的寄存器,指令使用的功能单元等。汇编优化器会根据代码的情况确定这些信息,并安排软件流水。

本课题将部分耗时的函数用线性汇编进行了改写,包括计算 SAD,DCT/ IDCT,量化,插值等,下面以计算 16x16 亮度块的 SAD 为例来说明线性汇编的实现过程。计算 SAD 值是 H. 264 编码过程中一个很重要的步骤,无论是帧内模式选择还是帧间预测中的运动搜索都需要频繁调用该模块,计算 SAD 的公式为:

$$SAD = \sum_{M}\sum_{N} |S(x,y) - S(x',y')|$$

求 2 个 16x16 亮度块的差的绝对值之和时,考虑到 DM642 有数据打包处理指令:LDDW:一次从内存中读出 64 位,即 8 个相邻的亮度值 SUBABS4:一次计算 4 个亮度值的差的绝对值 DTOPU4:求 2 个 4 字节向量的点积亮度值都是不大于 128 的无符号单字节数据,每对亮度值的差的绝对值仍是 8bits 单字节数据,这样 SUBABS4 计算出来的结果与 0x01010101 点积得到的刚好是 4 对亮度值的差的绝对值之和。为了能够更好的运用流水线机制和充分利用 DM642 的通用寄存器组,采用连续读取两组数据的方法进行处理。线性汇编代码经汇编优化器优化后,该函数运算速度提升了 4 倍多。

4 实验结果表

本课题对优化前后的编码器进行了大量实验,实验前先固定一组编码参数,本文主要是研究 Baseline 档次下的编码,修改 T264 的编码配置文件(enconfig. txt)把编码器的主要参数设置为:测试序列为标准 QC IF 序列,运动搜索的最大范围为 ±16,一共编 100 帧, I 帧间隔为 10,不编 B 帧,即 IPPPP. . . 编码模式,参考帧个数为 1,量化参数 QP 为 26,不启用环路滤波,帧间编码时允许使用所有的 7 种分块方式, P 宏块允许帧内预测, 1 /4 象素运动估计,使用钻石搜索算法,使用 SAD 作为块匹配准则,不使用 CABAC 熵编码方案。

表 1 编码器的实验结果

测试序列	特点	条件	编码速度	平均 PSNR		
			(fps)	y	u	v
Foman	运动剧烈	优化前	2.97	36.71	39.99	41.95
	有场景切换	优化后	27.68	36.64	39.97	41.90
Mother and	背景简单,基本不变	优化前	3.34	37.68	41.00	40.89
daugher	画面人物运动幅度不大	优化后	29.59	37.64	40.92	40.89
News	背景变化	优化前	3.39	38.09	40.69	41.25
	画面人物运动幅度不大	优化后	30.08	38.04	40.67	41.23
Container	背景简单	优化前	3.46	37.58	42.04	41.74
	画面景物缓慢运动	优化后	30.40	37.56	42.04	41.74

5 结论

本文基于 TI 公司的多媒体处理芯片 TMS320DM642 开发平台,设计了适应实时信号处理的 H.264 视频编码器架构,采用了先进的快速变换和高性能的搜索算法,完成了对代码的逐步优化和调试,完全满足在各种环境下对视频信号的实时处理要求。

参考文献:

[1] ITU - T Recommendation H.264. Advanced video coding for generic audiovisual services, 2003 - 05.

[2] 毕厚杰. 新一代视频压缩编码标准——H.264/AVC. 人民邮电出版社, 2005 - 05.

[3] ITU - T Recommendation. H.263. Video coding for low bit rate communication, 1998 - 02.

[4] Iain E G Richardson 著, 欧阳合, 韩军, 译. 视频编解码器设计——开发图像与视频压缩系统. 长沙: 国防科技大学出版社, 2005 - 01.

[5] 沈兰荪, 卓立, 田栋, 等. 视频编码与低速率传输北京: 电子工业出版社, 2001 - 12.

[6] 江思敏,刘畅. TMS320C6000DSP 应用开发教程, 北京:机械工业出版社,2005.

(收稿日期:2007 - 06 - 19)