基于 DM6437 的 H.264 编码器的移植与优化

谭骧骏，苏扬
武汉理工大学电子信息系，武汉（430070）
E-mail: erictxj@163.com

摘　要:H.264 作为新一代视频编码标准，压缩性能优异，图像质量高，可满足各种不同速率、不同场合的视频应用。然而 H.264 的高编码效率是以增加复杂度为代价的，要在实际视频通信产品中应用就必须研究编码器的优化算法。本文以 X264 开源编码器为参考模型，详细介绍了 H.264 编码器在 DM6437 上的移植以及优化。

关键词： H.264；X264；DM6437；视频压缩编码；编码器
中图分类号： TN919.81

1 引言

H.264 标准不仅具有很高的编码效率，而且具有良好的网络适应性[1]。如何在现有技术水平和硬件条件下实现实时多媒体通信终端设备和产品一直是信号处理领域关注的话题。随着数字信号处理器（DSP）的高速发展，为我们实现实时高效的多媒体处理提供了可能性。其中 TI 公司的 TMS320C64x+系列产品，具有高主频、多流水线、高并行度以及专门的视频信号处理指令等优点[2]，使其成为视频处理领域优先选择的 DSP 芯片之一。因此，将 H.264 标准与 DSP 芯片相结合，实现高效的多媒体通信平台具有一定的工程意义和市场价值。

2 H.264 编码算法在 DM6437 上的实现

H.264 有三大开源编码器：JM、X264 和 T264。JM 是 H.264 的官方测试源码；X264 是由网上自由组织联合开发的编码器，输出码流与 H.264 标准兼容，与 JM 相比，更注重实用，在不明显降低编码性能的前提下，降低编码的计算复杂度[3]。本文选择 X264 编码器作为移植对象。

DM6437 芯片架构属于 C64x+系列与 PC 机的 x86 架构有许多差别，而且 X264 源代码是在 VC 环境下编译运行的，DSP 编译使用的 CCS 环境与其有很大的不同。移植工作主要从以下几个方面着手：

(1) 编译器设置

C6000 C/C++编译器提供了大量的编译选项，对 X264 工程的 build option 选项进行部分修改：首先，根据 DSP 芯片的体系架构将 Target Version 项设置为 C64x+，其对应的编译选项即为-mv6400+；其次，DM6437 芯片只支持小端模式，因此 Endianness 项为：little Endian。最后，设置处理器编译时查找文件路径，即 Inlude Search Path 为$(Proj_dir)。

(2) 编写头文件

因为 CCS 运行时支持的库与 VC 环境的不同，许多源代码中被调用的库函数将不被兼容。这些库主要包括：memory.h，sys/timeb.h，malloc.h，global.h，ratectl.h，configfile.h。这些 DSP 不支持的库函数，大部分对于编码过程不是必须的，仅完成一些数据统计等功能，因此可以在代码中屏蔽；但有一些是编码过程中必须的函数，只有进行手工重写进行代替。

(3) 存储器的分配

DM6437 片内存储器资源是一个两级的缓存结构，第一级包含程序缓冲区 L1P(32KB)和数据缓冲区 L1D(80KB)两个独立的高速缓存模块；第二级是程序/数据缓冲区 L2(128KB)，它不能与 DSP 内核直接交换数据[3]。这两级缓存结构又与开发板提供的 128MB 片外 SDRAM

一起形成一个三级存储器系统。同时 DM6437 还提供了 64 个独立的 EDMA3 控制器负责片内 L2 存储器与片外外存及其他外设之间的数据传输。

合理布置代码段的内存布局，把需要连续调用的函数存放在一片连续的内存空间中，并配合 Cache 大小，可以被 Cache 一次性读入，这样可以提高 Cache 的命中率。对 Cache 大小配置的原则是将尽量多的关键数据分配在片内，由于片内内存容量有限，所以在片内 RAM，即 L2 级 SRAM 中主要存放编码过程中频繁使用的一些关键数据，主要包括原始宏模块数据、重构宏块缓冲区、DCT 变换后的系数、量化后的系数等。片外 SDRAM 主要存放原始图像帧、当前编码重建帧和参考帧等。编码器主要存储空间的分配如表所示

表 2.1 内存分配

图像数据项目	占用内存(Byte)	存放位置
原始图像帧	38016	片外 SDRAM
当前编码重建帧	38016	片外 SDRAM
参考帧	38016	片外 SDRAM
原始宏块数据	384	片内 L2 SRAM
重构宏块缓冲区	384	片内 L2 SRAM
DCT 变换后系数	384	片内 L2 SRAM
量化后系数	384	片内 L2 SRAM
整像素参考窗	5184	片内 L2 SRAM

3 X264 基于 DSP 的优化

3.1 整数变换的优化

整数 DCT 变换的输出可以表示为：$Y=(CXC^T)\otimes E_f$，其中 X 为输入，C 是变换矩阵 $C=\begin{bmatrix}1 & 1 & 1 & 1\\ 2 & 1 & -1 & -2\\ 1 & -1 & -1 & 1\\ 1 & -2 & 2 & -1\end{bmatrix}$。运算“$\otimes$”对每个矩阵元素只进行一次乘法，H.264 将 DCT 中“$\otimes E_f$”运算的乘法融合到后面的量化过程中，实际的输出为：$W=CXC^T$。它可以看作将此矩阵乘法改造成两次一维整数 DCT 变换，例如先对残差的每行进行一维整数 DCT，然后对经行变换的每列再应用一维整数 DCT，而一维整数 DCT 可以采用蝶形快速算法。在 C 代码中的实现是按下面的公式进行的：

$$CXC^T=C(CX^T)^T=((CX^T)C^T)^T=((XC^T)^TC^T)^T \qquad (3\text{-}1)$$

其中 X 为 4×4 当前块和 4×4 参考块之间的残差数据。使用 C 实现的变换和量化模块并未利用 DSP 的结构特性，需要利用线性汇编语言和 DSP 特性进行改写[4]。

每个像素点用 8bit 二进制数表示，用 LDW 一次读取一行数据即 4 个数据，而原始数据和参考数据不相关，可以同时读取[5]。然后通过 UNPKHU4 和 UNPKHLU4 指令将所得到的一行数据拆包成两个 32bit 的数据，再通过 SUB2 求的残差。

得到残差数据后，通过 PACK2 和 PACKH2 指令将残差数据打包，然后通过 ADD2 和 SUB2 指令完成矩阵列变换，其中乘以 2 使用 ADD2 指令完成。最后利用 STDW 指令将所

得结果存储到对应存储空间。

有了4×4的整数 DCT，就可以利用它来做8×8的 DCT，再利用8×8块做的 DCT 做16×16的 DCT。本文将8×8块划分为 4 个4×4块后对各个块做整数 DCT。

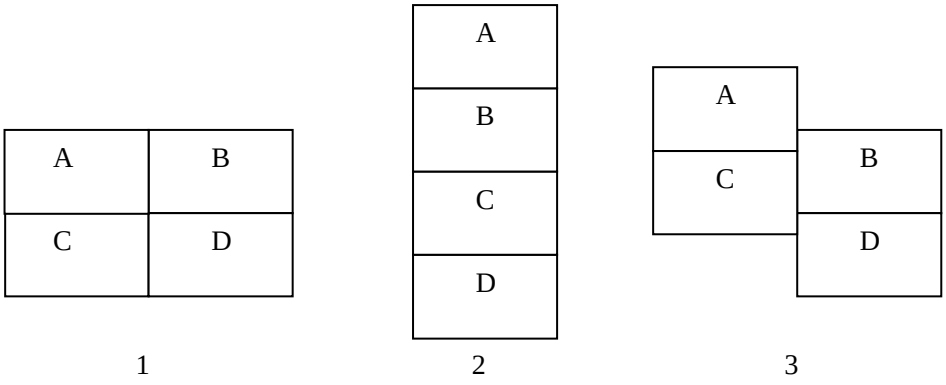


图 3.1 8×8块 DCT 编排图

A，B，C，D 为4×4块的整数 DCT，1 表示8×8块的原始数据，2 表示各个块依顺序执行，3 表示利用流水线方式执行。2 中的顺序执行至少是各小块变换时间的 4 倍，而 3 中8×8使用流水线执行的 DCT 只需要4×4块 DCT 时间的两倍多，利用 DM6437 的的多功能单元结构以及采用“流水线”技术，可以使运算速度明显的加快。

3.2 零块判决算法

低码率视频中具有缓慢变化或者静止背景的序列，在帧间编码时，经过运动估计的残差块信号绝对值通常很小。这样的残差信号经过变换、量化后的系数常常为零，即存在大量的全零块[6]。如果可以在变换和量化前检测到这种全零块，就可以省去变化、量化的过程，降低编码的计算复杂性，提高编码效率。

在 H.264 预测帧中，每个宏块(MB)都要将所有预测模式预运行一遍，然后选择编码效率最佳的预测模式，选择依据是在每种预测模式下，运动补偿后得到帧间 SAD(绝对差值和)最小[7]。

如果预测模式是$P4\times4$，那么宏块中每个4×4块的 SAD 值为$\sum_{n=0}^{3}\sum_{m=0}^{3}\left|X\left(m,n\right)\right|$，$X\left(m,n\right)$表示运动补偿后的帧间预测差值[8]。另外，整数变换后$X_{4\times4}$的直流分量为$\sum_{n=0}^{3}\sum_{m=0}^{3}X\left(m,n\right)$[9]，那么直流系数的量化值为：

$$x_q\left(0,0\right)=sign\left\{\sum_{n=0}^{3}\sum_{m=0}^{3}X\left(m,n\right)\right\}\left[\left(\left|\sum_{n=0}^{3}\sum_{m=0}^{3}X\left(m,n\right)\right|A\left(Q_M,0,0\right)+f2^{17+Q_E}\right)>>\left(17+Q_E\right)\right] \quad (3\text{-}2)$$

在预先判决零块算法中，如果$x_q\left(0,0\right)$为 0，则假定$x_q\left(i,j\right)(i,j=0,1,2,3)$所有值均为零，即整个块的变换系数量化后全为零[10]。由式(3-2)可知，$P4\times4$模式下判决4×4块在整数变化系数量化后全为零的条件是：

$$\left|\sum_{n=0}^{3}\sum_{m=0}^{3}X(m,n)\right| < (1-f)\cdot 2^{17+Q_E} / A(Q_M,0,0) \qquad (3\text{-}3)$$

另外，由于$\left|\sum_{n=0}^{3}\sum_{m=0}^{3}X(m,n)\right| < \sum_{n=0}^{3}\sum_{m=0}^{3}\left|X(m,n)\right|$，那么式（3-3）全零块判决条件可用下式代替：

$$\sum_{n=0}^{3}\sum_{m=0}^{3}\left|X(m,n)\right| < (1-f)\cdot 2^{17+Q_E} / A(Q_M,0,0) = T_{4\times4} \qquad (3\text{-}4)$$

由式(3-4)可以看出，不等式左边项即为4×4块运动补偿后 SAD 值。因此，本文在$P4\times4$预测模式选择过程中，可对4×4块进行全零块判决。步骤如下：

（1）计算4×4块位移矢量为零的 SAD 值。首先，判断 SAD 是否满足式（3-4），如果 SAD 小于判决门限$T_{4\times4}$，则终止当前块的位移估值，记录该块的位置$L_{x,y}$，然后执行第二步；否则继续完成位移估值，得到最小的 SAD 值，并计算4×4块的帧间预测差值后，再执行第二步。

（2）完成4×4块整数变化和量化。首先判断当前块是否属于$L_{x,y}$，如果4×4块是全零块，则对该块不再进行整数变换和量化，直接对4×4块取全零值；否则，对该帧间预测差值进行整数变化和量化。

与已有的预判断零算法不同，基于$P4\times4$预测模式的预先判决零块算法，可以大大减少4×4块位移估值时 SAD 的计算次数，同时还减少了整数变化和量化的运算，且对$P4\times4$预测模式的编码性能和效率基本没有影响。

3.3 实验结果

表 3.1 统计的是整数变换、量化、反变换、和反量化函数的结果。Tc 表示为完成相应功能的 C 代码消耗时间；Tsa 表示完成优化后代码消耗时间。

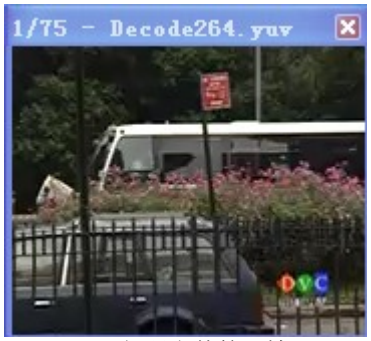
表 3.1 部分函数耗时统计

函数	Tc(cpu 周期)	Tsa(cpu 周期)
$add4\times4\_idct$	232	36
$add8\times8\_idct$	983	113
$dct4\times4dc$	89	32
$idct4\times4dc$	80	30
$sub4\times4\_dct$	96	35
$sub8\times8\_dct$	418	80

运行移植并优化后的代码压缩大小为 2.71MB 的视频文件（75 帧）BUS_176x144.yuv，输出.264 文件大小仅为 182KB，压缩率十分理想。

用解码工具解码压缩文件与源视频文件对比：

（a）原始文件第 1 帧



（b）解码文件第 1 帧

（c）原始文件第 23 帧

（d）解码文件第 23 帧

图 3.2 视频文件

4 结论

在 X264 源代码的基础上进行裁减和优化，H264 编码器在 DM6437 上移植成功。实验结果表明，优化后的代码效率有了明显的提高，编码压缩后的视频文件也保持了较好的主观质量。

参考文献

[1] 李宾,高平.H.264 编码系统的特点及其应用前景[J].数字电视与数字视频,2003,6(2):19-21.
[2] 任丽香,马淑芬,李方慧.TMS320C6000 系列 DSPs 的原理与应用[M].北京:电子工业出版社,230-260.
[3] 毕厚杰.新一代视频压缩编码标准—H.264/AVC[M].北京:人民邮电出版社,2005.97-131.
[4] 欧阳合,韩军.视频编解码器设计—开发图像与视频压缩系统[M].长沙:国防科技大学出版社,2005.100-150.
[5] MINQIANG JIANG,NAM LING. Low-Delay Rate Control for real-time H.264/AVC Video Coding[J].IEEE Transaction on Multimedia,2006,8(3):467-477.
[6] 尚书林,杜清秀,卢汉清.一种低复杂度 H.264 宏块级码率控制算法[J].计算机学报,2006,29(6):914-919.
[7] 倪伟,郭宝龙.H.264 变换编码和量化算法的研究[J].计算机工程与应用,2006,3:33-36.
[8] 马思伟,高文.一种面向 H.264/AVC 的码率控制算法[J].电子学报,2004,32(12):2024-2027
[9] 吴乐南.数据压缩的原理与应用[M].北京:电子工业出版社,1995.240-280.
[10]L.M.Po,W.C.Ma.A novel four-step search algorithm for fast block motion estimation[J].IEEE Transactions on CSVT,1999,6(3):314-317.

Transplantation and Optimization of H.264 Encoder Based on DM6437

Tan Xiangjun, Su Yang
Schoool of Information Engineering,Wuhan University Of Technology,Wuhan (430070)

Abstract

H.264,as new video encoding standard, which has high compression performance and image quality, can meet the need of the video applications with different rates on different occasions.However,the high encoding efficiency is based on high complexity.Research on optimization algorithm of H.264 codec must be done to apply in video communications products. This paper take X264 encoder open source as a reference model, introduce the transplantation and optimization of H.264 encoder based on DM6437.

Keywords: H.264;X264;DM6437;Video compression encoding;Encoder