

基于 TMS320DM642 平台的 4×4 DCT 变换的实现与优化

杨观华, 朱俊梅

(中南民族大学电子信息工程学院, 湖北 武汉 430073)

摘要: 针对离散余弦变换算法(DCT)的特点, 通过对 TI 公司 TMS320DM642 DSP 平台的软件优化试验数据进行分析, 实现了 4×4 DCT 快速变换算法的优化, 得到了优化后 4×4 DCT 快速变换算法的库函数, 该库函数可应用于现有的视频标准, 能有效提高视频编解码的效率。

关键词: 离散余弦变换; TMS320DM642; 算法优化; H.264

中图分类号: TN762 **文献标识码:** A

Implementation and Optimization of 4×4 DCT Based on TMS320DM642 Platform

YANG Guan-hua, ZHU Jun-mei

(College of Electronics and Information, SCUFN, Wuhan, Hubei 430074, China)

Abstract: Based on the characteristics of existing discrete cosine transform (DCT) methods in literature, and results obtained from the experimental data of software optimization based on TMS320DM642 platform of Texas Instruments (TI), an optimized subroutine of 4×4 DCT for this platform is proposed. An optimized library function is obtained for special platform, which can be widely used in the implementation of the existing video standards, such as H.264.

Key words: discrete cosine transform; TMS320DM642; algorithm optimization; H.264

0 引言

视频编解码标准一般采用离散余弦变换(DCT)进行变换编码, 以达到去处图像冗余信息进行压缩的目的。ITU 的视频编码专家组和 ISO/IEC 的运动图像专家组组成了联合视频组(JVT), 开发了一个新的视频编码标准——H.264。H.264 具有很好的编码压缩效率, 在相同的重建图像质量下, 它能够比 H.263 节约 50% 的比特率, 使其成为当前最有竞争力的视频编码标准之一。以往的视频编码标准采用的是 8×8 离散余弦变换, 计算量大而且容易带来解码数据失配。H.264 的一个新特点就是采用了 4×4 离散余弦变换, 可以克服上述问题^[1,2]。

较高的压缩效率以及图像质量往往是以提高算法的复杂度为代价的, 而且视频编解码系统的承载平台的资源往往十分有限, 特别对嵌入式产品而言。为此, 必须对其复杂的核心算法(例如 DCT)针对特定的平台资源进行软件优化实现, 做成库函数的形式, 为后来的系统集成提高资源的利用率。

1 DCT 在 H.264 编码器中的作用

H.264 编码器模型如图 1 所示^[3]。

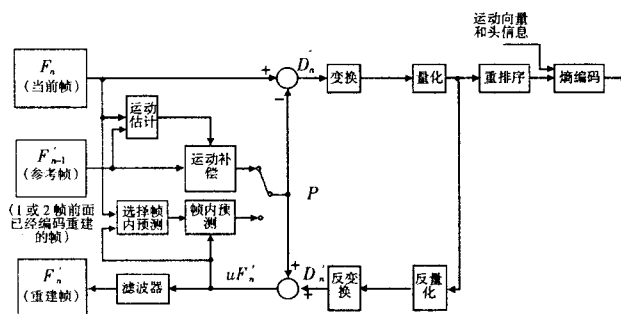


图 1 H.264 编码器模型

它包含了运动估计补偿、帧内预测编码、基于 DCT 的变换、量化、熵编码等模块。帧内预测以及运动估计补偿虽然已经大大减少图像的冗余, 但随后产生的图像残差 D 也包含相当可观的相关性较高的数据, 因此要对残差数据进一步去相关, 以达到数据压缩的目的, 这就是 DCT 变换在编码器中的作用。由于要对每一 4×4 残差块进行 DCT 变换, 对 DCT 变换模块的调用

十分频繁,如果不能提供一种高效的实现方法,该模块对整个编码系统产生的负荷是很重的,会影响编码的整体质量(比如实时性达不到要求)。为此,下面将针对 TMS320DM642 平台,通过对几种不同的算法在该平台上的实验数据进行比较,寻求一种较适合该平台的实现方法。

2 DCT 算法在 TMS320DM642 上的实现

正交变换编码算法已经广泛应用于图像编码中,其中 K-L 变换的性能最理想,但缺乏快速算法。DCT 的编码性能最接近于 K-L 变换^[3],其二维 DCT 计算公式如下:

$$Y_{xy} = C_x C_y \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \cos \frac{(2j+1)y\pi}{2N} \cos \frac{(2i+1)x\pi}{2N} \quad (1)$$

其中: $x=0, 1 \dots N-1; y=0, 1 \dots N-1; C_k = \sqrt{1/N} (k=0)$, $C_k = \sqrt{2/N} (k>0)$

其物理意义是将图像的重要信息集中到图像左上角的低频系数中,高频系数的幅值往往比较低,可以忽略掉,仅由少数的低频系数就可以较好的恢复原图像,具有很好的压缩性能。直接实现二维 DCT 变换需要 $N^2 \times N^2$ 次乘法,显然这是十分惊人的运算量,但是它快速算法,因此得以广泛应用于图像编码。

2.1 几种二维 4 × 4DCT 快速算法的比较

文献[4]提出了一种蝶型运算,简化的 4 点一维 DCT 变换如图 2 所示:

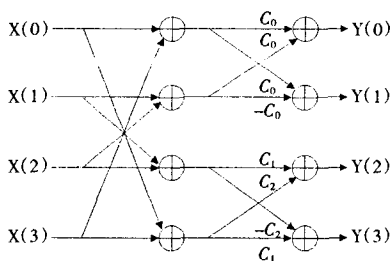


图 2 一维 DCT 变换的蝶型算法(N=4)

其中: $X(n)$ 是输入向量, $Y(n)$ 是变换系数, $C_0 = \cos(\pi/4)$, $C_1 = \cos(3\pi/8)$, $C_2 = \cos(\pi/8)$ 。

4 × 4 的 DCT 变换可以分解为两次一维的 4 点 DCT 变换,首先对 4 × 4 矩阵进行列变换,然后对获得的中间矩阵进行行变换,公式如下:

$$Y_{xy} = C_x \sum_{i=0}^3 \left[C_y \sum_{j=0}^3 X_{ij} \cos \frac{(2j+1)y\pi}{8} \right] \cos \frac{(2i+1)x\pi}{8} \quad (2)$$

利用蝶型算法,每次 4 点一维 DCT 变换需要 8 次

乘法,4 × 4 的二维 DCT 变换共需要 64 次乘法。相对于直接实现公式(1)所需的 256 次乘法而言,性能已经得到了很大的提高。

文献[5]提出了一种基于 DCT 的整数变换,它是一种 DCT 的变种,可以获得与 DCT 相似的性能,又可以在变换核 $C_f X C_f^T$ 中避免对一般平台来说消耗资源很大的浮点运算,公式如下:

$$Y = C_f X C_f^T \otimes E_f =$$

$$\left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} [X] \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

其中: $a = \frac{1}{2}, b = \sqrt{\frac{2}{5}}, d = \frac{1}{2}$ (3)

若将后面涉及浮点数运算的点乘 E_f 的运算放到量化模块中完成,可以减小对运算精度的影响。

以上的算法都比较适合软件实现,还有一些文献提出了其他一些基于硬件实现的算法^[6,7],主要是基于并行的乘法器阵列以及分布式算术运算。

2.2 软件优化策略

TMS320DM642 是由 TI 公司推出的具有 VelociTI.2™ 结构的一款高性能数字媒体芯片,在 720M 时钟频率的情况下处理能力可以达到 5760MIPS。C64X™ DSP 核拥有 64 个 32 比特字长的通用目的寄存器和 8 个高度独立的功能单元(2 个乘法器和 6 个算术逻辑单元(ALUs))。在 8 个功能单元中的 VelociTI.2™ 扩展包括针对视频图像处理的新指令以及对 VelociTI 并行结构的扩充。DM642 每时钟周期可以处理 4 条 16 比特的乘加指令(MACs),每秒可以处理 2880 百万次 MACs,或每时钟周期可以处理 8 条 8 比特的 MACs,每秒可以处理 5760 百万次 MACs。

针对前述几种不同的 DCT 算法,本文采用了以下几种针对 DSP 平台的软件优化策略,可以有效地提高算法具体实现在 DSP 平台上的运行速度。

(1)查表法 公式(3)的矩阵 E 涉及到浮点乘法运算,而且对于每一个 4 × 4 的变换均要使用一次,调用十分频繁。因此,事先计算好矩阵 E 的值,以表的方式出现在程序中,可以提高程序的效率。矩阵 E 的值: [0.2500, 0.1581, 0.2500, 0.1581; 0.1581, 0.1000, 0.1581, 0.1000; 0.2500, 0.1581, 0.2500, 0.1581; 0.1581, 0.1000, 0.1581, 0.1000]

(2)运用移位运算代替乘法运算

(3) 编译器选项的设置 选项控制着编译器的操作,与代码性能、优化、长度相关^[8]。避免使用用于调试的选项 `-g/-s/-ss/-o1/-o0` 等,因为它们有可能降低性能,增加代码长度。

(4) 使用数据打包处理 DCT 变换属于计算密集型运算,在代码中使用打包数据处理可充分利用内在并行机制,同时又保持了代码的密度,提高了执行效率。例如将两个半字节打包到一个通用寄存器中。

2.3 DCT 软件优化的效果

特殊的指令集和芯片结构对于不同的算法有不同的效果,下面有 3 个利用不同实现方式的子程序。

- (1) `dct_butterfly`: 运用图 2 所示的蝶型运算算法;
- (2) `dct_shift`: 运用移位指令,完全避免乘法运算;
- (3) `dct_muladd`: 运用乘加运算。

实验结果表明, `dct_shift`、`dct_muladd` 拥有与 `dct_butterfly` 基本相似的效果,如图 4 和图 5 所示。能量集中在直流分量和少数低频分量上,它们之间的差别是相当小的,主要集中在 0 附近,如图 6 所示。但代码大小以及速度都得到了优化,其中 `dct_muladd` 最优,比较结果见表 1。

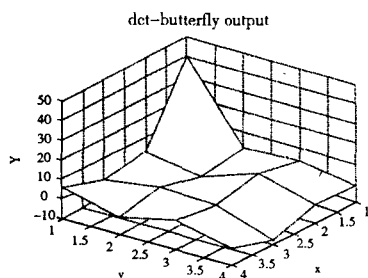


图 4 `dct_butterfly` 的输出

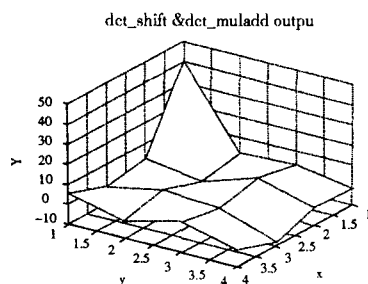


图 5 `dct_shift` 和 `dct_muladd` 的输出

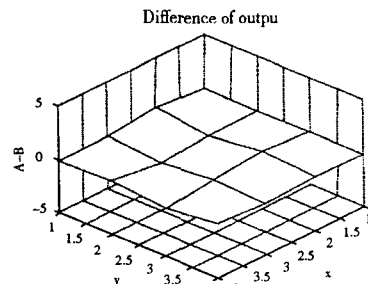


图 6 输出的差别

表 1 代码大小和时钟周期比较

子程序名	代码大小	运行时钟周期
<code>dct_butterfly</code>	1564 Byte	12578
<code>dct_shift</code>	1804 Byte	517
<code>dct_muladd</code>	668 Byte	215

由表 1 可以看出,利用蝶型运算的子程序由于涉及浮点数运算,浮点数乘法很占用芯片资源,显然是不利于在定点 DSP 上实现的。移位运算虽然完全避免了浮点数乘法,速度得到了可观的提升,但是代码偏大,对于片内程序空间有限的 DSP 而言,这是不利的。充分发掘了芯片性能的 `dct_muladd` 获得了相对最优的代码尺寸以及运行时间,在特定的平台上,这种实现方法是比较好的。

2.4 利用文档管理器建库

TI 公司提供的针对 C64x 系列 DSP 的图像处理库只提供了 8×8 的 DCT 变换^[9],缺少 4×4 的快速 DCT 变换,因此建立一个可以方便调用的 H.264 子函数库,为以后的科研以及工程提供高效的支持是很必要的。利用 TI 公司的建库工具 `ar6x` 可以方便地建立自己的子函数库。

命令行输入: `ar6x-as`h264C64x` dct_muladd.obj`

文档管理器响应如下:

```
==> new archive 'h264C64x.lib'
==> symbol defined 'dct_muladd.'
==> building archive 'h264C64x.lib'
```

这样就建立了一个名为 `h264C64x.lib` 的子函数库,可以方便地调用了。

3 结语

本文对几种不同的 DCT 算法进行了分析比较,在 TMS320DM642 平台上优化实现,并通过实验数据分析得出了一种较优化的实现方式。通过在特定的硬件平台上利用特殊指令集,保证了 H.264 标准中 4×4 离散余弦变换模块的快速实现。同时,通过建立 H.264 的子函数库,为后续的科研工作提供了保障。

参考文献:

- [1] Joint Video Team of ITU-T, ISO/IEC JTC 1. ISO/IEC 14496-10, ITU-T Rec. H.264. Advanced Video Coding for Generic Audiovisual Services[S]. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, 2005.
- [2] T. Wiegand, G. Sullivan, G. Bjontegaard, A. Luthra. Overview of the H.264 /AVC Video Coding Standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003,13 (7): 560-576.

(下转第 63 页)

3 结束语

UML 是一种建模语言,是一种标准的表示,而不是一种方法或方法学。方法是一种把人的思考和行动结构化的明确方式,方法需要定义软件开发的步骤,告诉人们做什么、如何做、什么时候做以及为什么要这么做,而 UML 只定义了一些图以及它们的意义,它的思想与方法无关。因此,我们会看到人们将用各种方法来使用 UML,而无论方法如何变化,它们的基础是 UML 的图,这就是 UML 的最终用途,为不同领域的人们提供统一的交流标准。

蜂窝客户管理系统利用统一建模语言 UML 对现有用户通过地域、消费结构差异进行了区域划分,在街道办事处管理的社区、聚合类商区、大中专院校及其他集中大客户单位所构成的相对独立的信息区域的基础

上,完成了信息单元的建立工作,作为蜂窝信息系统的重要组成部分很好地完成了系统目标。

参考文献:

- [1] Craig Larman. Applying UML and Patterns[M].李洋,郑龚译.北京:机械工业出版社,2006.
- [2] Rumbaugh J, Jacobson I, Booch G. UML 参考手册[M].UMLChina 译.北京:机械工业出版社,2005.
- [3] 刘晓华. J2EE 企业级应用开发[M].北京:电子工业出版社,2005.
- [4] Jiawei Han. OLAP Mining: An Integration of OLAP with Data Mining[M]. Canada: Intelligent Database System Research Laboratory, School of Computing Science, Simon Fraser University, British Columbia, 1997.
- [5] Robert Ian Oliver. Visual Basic .NET 企业应用程序设计[M].李祥青,徐泓,王抒译.北京:清华大学出版社,2003.
- [6] 段云峰,吴唯宗,李剑威,韩洁. 数据仓库及其在电信领域中的应用[M].北京:电子工业出版社,2003.
- [7] 常绪刚. 数据仓库中的 ETL 和 Metadata[C].北京:2003 年首届中国软件技术大会,2003.
- [8] 何荣勤. CRM 原理设计实践[M].北京:电子工业出版社,2003.

(上接第 46 页)

- [3] N. Ahmed, T. Natarjan, K. R. Rao. Discrete Cosine Transform [J]. IEEE Transactions on Computer, 1974,23:90-93.
- [4] W-H Chen, C. H. Smith, S. C. Fralick. A Fast Computational Algorithm for the Discrete Cosine Transform[J]. IEEE Transactions on Communications, 1977, 25(9):1004-1009.
- [5] H. Malvar, A. Hallapuro, M. Karczewicz. Low Complexity Transform and Quantization in H.264/AVC [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003,13(7): 598-603.
- [6] C. V. Schimpfle, P. Rieder, J. A. Nossek. A Power Efficient Implementation of the Discrete Cosine Transform[J]. Signals, Sys-

tems & Computers, 1997,1:729-733.

- [7] R. H. Turner, T Courtney, R Woods. Implementation of Fixed DSP Functions Using the Reduced Coefficient Multiplier[C]. IEEE Conf. on Acoustics, Speech and Signal Processing, 2001,2:881-884.
- [8] Texas Instruments Incorporated. TMS320C6000 Programmer's Guide (Rev. D)[M]. Texas Instruments Incorporated, 2006.
- [9] Texas Instruments Incorporated. TMS320C64x Image/Video Processing Library Programmer's Reference (Rev. B)[M]. Texas Instruments Incorporated, 2003.

(上接第 56 页)

5 结语

该系统基本实现了预期的目标,实现了企业范围内的信息共享以及员工与客户的自动化管理,使企业能更好地为客户提供服务。如果加以完善,则可以成为具有整合交叉功能的、对全部客户联系点进行管理的理想工具。

参考文献:

- [1] 马纲,李洪心,杨兴凯.客户关系管理[M].大连:东北财经大学出版社,2005.
- [2] 樊建. ASP.NET+ADO.NET 项目开发实例[M].北京:清华大学出版社,2004.
- [3] 张蓓. ASP.NET 通用模块及典型系统开发[M].北京:人民邮电出版社,2006.

- [4] 杨鲲鹏,孟凡琦. ASP.NET+SQL Server 动态网站开发[M].北京:电子工业出版社,2005.
- [5] 孙少波,吴晨,叶堯. ASP.NET 数据库项目案例导航[M].北京:清华大学出版社,2004.
- [6] 李勇平,陈峰波. ASP.NET(C#)基础教程[M].北京:清华大学出版社,2006.
- [7] 张文仲. ASP.NET 网络开发技术[M].北京:人民邮电出版社,2006.
- [8] 苏贵洋,黄穗,何莉. ASP.NET 网络编程从基础到实践[M].北京:电子工业出版社,2005.
- [9] 启明工作室. ASP.NET+SQL Server 网络应用系统开发与实例[M].北京:人民邮电出版社,2005.
- [10] 杜思强. ASP.NET 与数据库设计[M].北京:中国铁道出版社,2002.