

数字视频录像机的研究与设计

谢丁峰,夏新军

(湖南科技大学 计算机科学与工程学院,湖南 湘潭 411201)

摘 要:文中设计实现了数字视频录像机,重点阐述了最新的视频压缩 H.264/AVC 中的 DCT 变换与数字视频录像机系统结构和关键模块设计。TI 公司的 TMS3206416 是专门面向多媒体应用而设计开发的 32 位定点 DSP 芯片,可以很好地满足视频信号处理的要求,文中利用 TMS3206416 的 64 位和 128 位的 SIMD 指令进行算法的优化并利用 128 位 SIMD (Single Instruction Multiple Data)指令并行优化了 DCT 模块。系统性能测试结果表明,与传统的监控系统相比,本系统设计合理,运行可靠,视频图像质量较好,具有低成本、高质量数字视频监控的优点。

关键词:数字视频录像机;离散余弦变换;视频图像

中图分类号: TN946

文献标识码: A

文章编号: 1673-629X(2010)09-0246-04

Research and Design of Digital Video Record

XIE Ding-feng, XIA Xin-jun

(Sch. of Computer Sci. and Eng., Hunan Univ. of Sci. and Tech., Xiangtan 411201, China)

Abstract: Designed and implemented a Linux-based digital VCR, emphasized DCT transform of latest video compression method H.264/AVC, system structure and key module design of digital VCR. The TMS3206416 is specifically designed 32-bit fixed-point DSP chip for the development of multimedia applications, with the ability of well meet the requirements of the video signal processing. In this paper, the TMS3206416 64-bit and 128-bit SIMD instruction was used for the algorithm optimization and 128-bit SIMD instruction also for the parallel optimization of DCT module. Compared with the traditional ones, practice has proved that the system design is reasonable and with good image quality.

Key words: digital video record; discrete cosine transform; video image

0 引言

视频监控在各行各业得到广泛的应用。DVR (Digital Video Record) 数字视频录像机已经成为视频监控领域一个新型产品。模拟视频的数字化包括不少技术问题^[1],如电视信号具有不同的制式而且采用复合的 YUV 信号方式,而计算机工作在 RGB 空间,电视机是隔行扫描,计算机显示器大多逐行扫描等。视频监控关键的技术是网络技术、图像视频压缩技术和视频数据实时传输技术。

1 系统结构设计

数字视频录像机总体结构主要包括数字视频录像

机终端和数字视频录像机服务器端。视频录像机终端采集的视频图像经过 DSP 处理器处理编码后进行网络传输,视频服务器端网络接收模块接收视频数据,经过视频解码模块对视频数据解码后,用户通过数字视频录像机应用程序远程观看视频图像。数字视频录像机客户端与服务器端的结构如图 1 所示。

服务器端主要是视频图像的显示、视频画面的分析,对画面视频图像中的识别和系统配置等功能。客户端主要的功能是采集现场视频画面图像,并对视频图像编码。客户端按照一定的协议对实时采集的数据进行编码后发送给服务器,服务器对收到的数据信息进行解码后,在应用程序上可以显示视频画面图像。同时,服务器端可以对采集数据进行分析 and 判断识别,并将指令按照协议编码后发送给客户机,客户端可以根据指令发出警报等功能。

2 硬件设计

本数字视频录像机的硬件结构如图 2 所示。主要包括 TMS320C64xx DSP 处理器视频压缩模块、视频

收稿日期:2010-01-27;修回日期:2010-04-26

基金项目:国家自然科学基金资助项目(50875082);湖南科技大学 2009 年研究生创新基金资助项目(S090116)

作者简介:谢丁峰(1978-),男,江西吉水人,硕士研究生,研究领域为计算机嵌入式系统、视频图像;夏新军,副教授,硕士生导师,研究领域为 VLSI 设计自动化、嵌入式系统。

采集模块、音频采集模块、网络通信模块等。

DSP 处理器模块:DSP 处理器采用 TMS320C6416 的定点 DSP,主要特点是体系结构上采用了 VelociTI 超长指令字(Very Long Instruction Word, VLIW)^[2],通过指令间的并行获取更高的性能,其系统性能和特点适应多媒体数字应用。该压缩模块主要采用 H.264 视频压缩算法完成数字视频数据编码压缩工作。

视频采集模块、音频采集模块:将模拟视频、音频数据转换为数字数据,输出格式为 YUV(4:2:0)数据。YUV 是在数字视频信号处理中最常用的一种颜色表示方式,“Y”表示亮度,也就是灰度值;“U”和“V”表示色度^[1]。

网络通信模块把 DSP 产生的压缩数据转换为网络数据包,同时接收网络中控制命令。

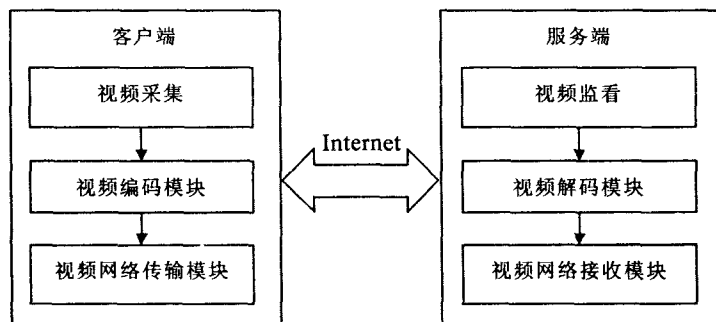


图1 数字视频录像机客户端与服务器端的结构

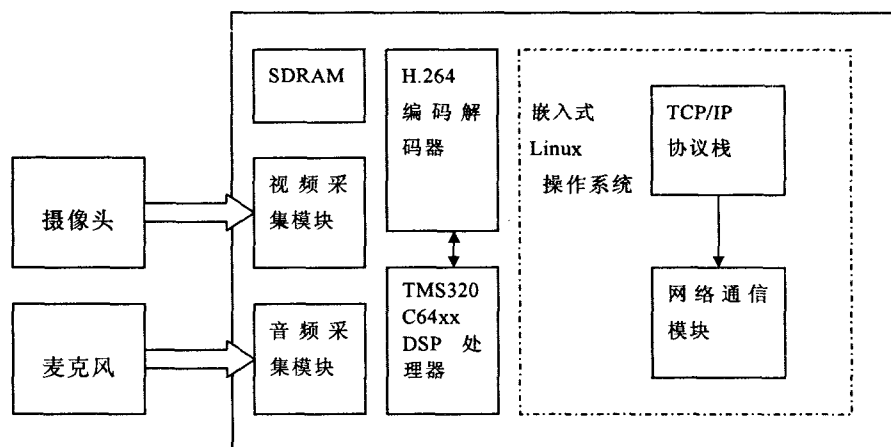


图2 系统硬件结构图

3 DSP 处理器模块中的 H.264 压缩算法

H.264 是新一代运动图像压缩标准,ITU-T 的 H.264 标准的制定是由 ISO/IEC 下属的运动图像专家组 MPEG(Moving Picture Experts Group)和 ITU 下属的视频编码专家组 VCEG(Video Coding Expert Group)共同成立的联合视频小组 JVT(Joint Video Team)负责完成的^[3]。H.264/AVC 成为近年来最为高效的一个视频压缩标准,其编码效率提高到以往标

准(MPEG-2 和 MPEG-4)的 2 倍^[4]。H264/AVC 中采用变块尺寸运动补偿、多参考帧选择、1/4 像素精度的运动矢量描述及多假设运动补偿与加权预测等先进技术。H.264 编码器移植到 DSP 端,到实现实时编码的要求,优化工作是十分繁重的。一般的优化顺序是先做 DSP 端的程序结构优化,再做 H.264 算法级优化,最后做代码级优化包括汇编语言的改写。下面是 DCT 和 SAD 的详细介绍与代码级优化。

3.1 离散余弦变换

图像变换是视频编码系统的核心组成部分,基于变换的方法也是目前图像编码和视频编码方法的主流^[2]。离散余弦信号变换方法,实际上是离散傅里叶变换的一种特例。在余弦变换中,变换核函数由于 FT 中复指数函数 $e^{-j2\pi ft}$ 变成余弦函数 $\cos 2\pi ft$ 。

一维 N 点离散余弦变换(DCT)表示为:

$$y_k = c_k \sum_{n=0}^{N-1} x_n \cos \frac{(2n+1)k\pi}{2N} \quad (1)$$

其中, x_n 是输入时域中的第 n 项, y_k 是输出频域序列中的第 k 项,系数 c_k 定义如下:

$$\text{当 } k=0 \text{ 时, } c_k = \sqrt{\frac{1}{N}},$$

$$\text{当 } k=1,2,3,\dots,N-1 \text{ 时, } c_k = \sqrt{\frac{2}{N}}。$$

二维 $N \times N$ 图像块的 DCT 先对图像块的每行进行一维 DCT,然后对经行变换的块每行再应用一维的 DCT。表示为:

$$y_{mn} = c_m c_n \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \cos \frac{(2j+1)n\pi}{2N} \cos \frac{(2i+1)m\pi}{2N} \quad (2)$$

其中, x_{ij} 是图像 X 中第 i 行第 j 列图像或残差值, y_{mn} 是变换结果矩阵相应频率点上的 DCT 系数^[5]。

H.264 的变换是以 DCT 为基础的,对于输入序列 X , 4×4

的 DCT 变换矩阵表示如下:

$$Y = A \times A$$

$$= \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & -b \\ c & -b & b & -c \end{bmatrix} \times \begin{bmatrix} a & b & a & c \\ a & c & -a & b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \quad (3)$$

这里 $a = 1/2$, $b = \sqrt{1/2} \cos(\pi/8)$, $c = \sqrt{1/2} \cos(3\pi/8)$, A 中的 a, b , 与 c 是实数,而图像块 X 中的元素是整数。H.264 对式(3)进行了改造,可变换为:

$$Y = (CXC^T) \otimes E =$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \quad (4)$$

其中, $d = c/b (\approx 0.414)$ 。符号“ \otimes ”表示 $(C \times C^T)$ 结果中的每个元素乘以矩阵 E 中对应位置上的系数值的运算。为了简化计算,取 d 为 0.5。这样变换都是整数运算,避免了浮点操作带来的四舍五入误差,使得变换准确匹配。此外,采用小尺寸的块有助于降低块效应和明显的人工处理痕迹,而且整数运算能减少运算量和运算复杂度。

3.2 DCT 变换的优化

二维 DCT 计算复杂度高,一个 8×8 大小的图像直接使用公式(4)进行 DCT 变换,共需要 4096 次乘法运算和 3584 次加法运算,极大地占用嵌入式系统 CPU 的资源。C. Loeffler, A. Ligtenberg 和 G. Moschytz 提出了一种有效的快速算法,成为 LIM 算法。DCT4 * 4 蝶形算法,将二维整数变换转换为两次一维的变换,首先对行进行操作,然后对列操作。TI 公司于 2002 年推出的一款专门为数字媒体应用而设计开发的 32 定点 DSP 芯片 TMS320C64xx^[6-11],即每次读取指令都是读取 256bit,也就是 8 条指令,称之为一个指令包,这样才有可能使得 8 个功能单元能够同时执行指令,达到高度的并行性。以下是并行操作的部分汇编代码:

```
.global dct_Y
.dct_Y:
    ; A4 --> input,
    ; B4 --> ref,
    ; A6 --> output,
    ; B6 --> shift,
    SUB2 .D1X A26, B26, A24
    || SUB2 .D2X B27, A27, B24
    || ADD2 .L1 A22, A25, A22 ; S03 = pIn[0] + pIn[12]
    || SUB2 .S1 A22, A25, A25 ; D03 = pIn[0] - pIn[12]
    :
    || ADD2 .D1 A23, A24, A23 ; S12 = pIn[4] + pIn[8]
    || SUB2 .L1 A23, A24, A24 ; D12 = pIn[4] - pIn[8]
    :
    | ADD2 .D1 A22, A23, A22 ; pOut[0] = S03 + S12
    || SUB2 .L1 A22, A23, A23 ; pOut[8] = S03 - S12
    || ADD2 .D2 B22, B23, B22
    || SUB2 .L2 B22, B23, B23
```

```
AND .D1 A26, A7, A26 ; (D03 << 1)
|| AND .L1 A27, A7, A27
|| AND .D2 B26, B7, B26
|| AND .L2 B27, B7, B27
ADD2 .D1 A26, A24, A24 ; pOut[4] = (D03 << 1) +
D12
|| SUB2 .L1 A25, A27, A25 ; pOut[12] = S03 - (D12 <
< 1)
; end column
:
```

与以 C 语言为主的二维 DCT 编码相比,用汇编语言实现的二维 DCT 程序执行时可以大量采用并行处理,使得代码执行时间上得到了很大改善,优化了整个算法的效率。与 C 代码比较如表 1 所示。

表 1 DCT4 * 4 蝶形算法线性汇编
与 C 时钟周期比较

代码类型	未优化时钟周期	-03 优化时钟周期
DCT 程序 C 语言	1026	127
DCT 程序线性汇编	136	32

相比较未优化 C 程序,线性汇编采用优化选项提高了近 30 倍。

4 软件设计实现

本系统采用 Linux 嵌入式操作系统,由于 Linux 内核采用模块化的设计,很多模块可以独立地加载或卸载,所以小型化就是对 Linux 内核重新编译,在编译时仔细地选择嵌入式设备所需要的功能模块,同时删除不需要的功能^[12]。

监控系统不光有画面显示功能,还应该具备回放、报警、远程网络监控等功能。要实现这些功能,必须定义好一些模块接口的定义。下面是视频数据结构:

```
typedef struct VideoData
{
    int Size; //the buffer size
    int DataSize; //the total data size
    int VideoLen; //video length
    int AudioLen; //audio length
    int OtherPar; // other parameter
    int * AudioData; //audio data
    int * VideoData; //video data
} VIDEODATA, * PVIDEODATA;
```

编码视频函数 int Videocode (int handle, int * pVideo, PVIDEODATA pvidData), 参数含义: int handle[in], 视频编码器句柄; int * pVideo[in], 指向 int 类型的待编码视频图像数据指针; PVIDEODATA pvidData[out], 指向 PVIDEODATA 的指针返回编码数据及编码信息。解码视频函数 int DecodeVideo (int

handle, int Type, int * pVideoCode, int CodeLen), 参数含义: handle h[in], 视频解码器的句柄, int Type[in], 帧类型, int * pVideoCode[in], 待解码视频流指针, int CodeLen[in], 待解码视频流长度; 获得解码后的视频图像函数 int GetVideo(int handle, int * pOut, int rowspace) 参数含义: int handle[in], 视频解码器的句柄, int * pOut[out], 指向 BYTE 类型的视频图像缓冲区指针, int rowspace[in] 输出视频画面前后两行相隔的差距。

5 结束语

该设计采用了高效的 H. 264 编码技术和实时媒体传输技术, 能满足视频监控系统的的需求, 且可扩展性强。本系统采用特定嵌入式场合的专用 Linux 操作系统, 裁减后, 放在容量只有几百 k 字节或几兆字节的 Flash 芯片中。下一步将加入智能化模块, 使系统不断提高和完善。

参考文献:

- [1] 苏光大. 微机图象处理系统[M]. 北京: 清华大学出版社, 2000.
- [2] 郭宝龙, 倪伟, 闫允一. 通信中的视频信号处理[M]. 北京: 电子工业出版社, 2007.

(上接第 245 页)

入(名称、病原、发病机理等), 可实现相关疾病信息的查询, 并生成与数据库字段对应的 XML 格式文件。

5 结束语

文中通过对传统关系型数据库和 XML 格式数据特点的比较, 在 J2EE 平台上借助成熟的开发框架, 将现有的农业动物疾病关系数据库映射为 XML 形式的结构化文档, 在此基础上, 可以对 XML 文档做更精确的信息转换和查询^[12], 实现关系数据库全部内容映射为 XML 结构文档, 为今后该方向专家系统和语义网的建立打下良好基础。

参考文献:

- [1] Ronald B. XML and Databases[DB/OL]. 2009-05-11. <http://www.rpbouret.com/xmldbms>.
- [2] Widom J. Data Management for XML: Research Directions [DB/OL]. 2009-05-11. <http://www-db.stanford.edu/~widom>.
- [3] 周法国, 王映龙. 非结构化信息抽取关键技术研究探讨

- [3] JVT of ISO/IEC MPEG and the ITU-T JTC: ISO/IEC 14496-10:2005 Information technology - Coding of audio - visual objects - Part 10: Advanced Video Coding[S]. 2005.
- [4] Ostermann J, Bormans J, List P, et al. Video coding with H. 264/AVC: tools, performance, and complexity[J]. IEEE Circuit and Systems magazine, 2004, 41: 7-28.
- [5] 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.
- [6] Texas Instruments. TMS320C64x DSP Library Programmer's Reference[M]. [s.l.]: [s.n.], 2004.
- [7] Texas Instruments. TMS320C6000 CPU and Instruction Set Reference Guide[M]. [s.l.]: [s.n.], 2004.
- [8] Texas Instruments. TMS320C6000 Assembly Language Tools User's Guide[M]. [s.l.]: [s.n.], 2004.
- [9] 彭启踪, 管庆. DSP 集成开发环境——CCS 及 DSP/BIOS 的原理与应用[M]. 北京: 电子工业出版社, 2004.
- [10] Texas Instruments Incorporated. TMS320C6000 系列 DSP 编程工具与指南[M]. 田黎育, 何佩琨, 朱梦宇 编译. 北京: 清华大学出版社, 2006.
- [11] 魏本杰, 刘明业, 章晓莉. 二维 DCT 算法及其优化的 VLSI 设计[J]. 计算机工程, 2006, 32(2): 16-18.
- [12] 陈明, 梁兴东, 吴一戎. 基于 H. 264 的嵌入式无线视频监控系统的[J]. 微计算机信息, 2008, 32(5-2): 10-12.

- [J]. 计算机工程与应用, 2009(14): 1-6.
- [4] 李志辉. XML Schema 语义约束在关系数据库中的实现[J]. 计算机与现代化, 2009(10): 33-37.
- [5] Kong L B, Tang S W, Yang D Q, et al. Querying Techniques for XML Data[J]. Journal of Software, 2007, 18(6): 1400-1418.
- [6] 周爱武, 李孙长, 程博, 等. XML 数据库的研究与应用[J]. 计算机技术与发展, 2009, 19(9): 218-221.
- [7] 强保华, 潘家志, 余建桥. 从关系数据库中生成 XML 数据源的研究[J]. 计算机科学, 2002, 29(5): 70-71.
- [8] He Z Y, Li J Z, Wang C K. A data model for XML database[J]. Journal of Software, 2006, 17(4): 759-769.
- [9] Goldman R, McHugh J, Widom J. From Semistructured data to XML[C]//Proc of the 2nd Workshop on Web and Databases. [s.l.]: [s.n.], 1999: 25-30.
- [10] 薛星云. 探索 XML 模式与数据库模式之间的映射[J]. 福建电脑, 2009(9): 176-176.
- [11] 肖辉辉, 段艳明, 兰小机. 基于 Hibernate 的 XML 数据存储方法[J]. 计算机系统应用, 2009(10): 189-192.
- [12] 周健, 孙丽艳. 面向对象 XML 的存储模式的研究[J]. 计算机技术与发展, 2009, 19(3): 114-117.