

文章编号:1004-9037(2012)06-0690-06

基于 TMS320DM8168 的高清视频编码技术与实现

姜忠兵 罗 钧 杨晓花 孟 凯 彭加新

(重庆大学光电技术及系统教育部重点实验室,重庆,400030)

**摘要:**针对高清视频数据量大、处理速度慢、在实际应用中局限性大的缺点,选用 TI 最新推出的针对高清视频应用的 TMS320DM8168 硬件平台,采用 H. 264 算法对高清视频进行实时压缩编码。通过对 H. 264 算法中计算量大的帧内模式选择算法、帧间模式选择算法、快速搜索算法的优化,提高 H. 264 算法的实时处理能力。整个系统充分利用 ARM 核和 DSP 核的强大处理能力,经过算法级和程序级的优化,CIF 的压缩帧速率能达到 75 帧,分辨率为 1 280×720 的高清视频压缩速率能达到 8 帧。  
**关键词:**高清视频;H. 264 算法;实时处理  
**中图分类号:**TN91                      **文献标识码:**A

Encoding Technology and Implementation for High-definition Video Based on TMS320DM8168

Jiang Zhongbing, Luo Jun, Yang Xiaohua, Meng Kai, Peng Jiaxin  
(Key Laboratory of Optoelectronic Technology & System of the Ministry of Education, Chongqing University, Chongqing, 400030, China)

**Abstract:** High definition video has a large amount of data which consumes much processing time. Aiming at this important constraint for applications, the H. 264 algorithm is chosen for high definition video compression in the TMS320DM8168 platform, which is the latest high definition application platform developed by TI. The real-time processing ability of H. 264 is improved by optimizing the inter prediction mode method, intra prediction mode method and fast searching algorithms. With the excellent performance of ARM and DSP as well as the optimization in algorithms and data structures, the compression rate reaches 75 frames per second for CIF videos while 8 frames per second for 1 280×720 videos.  
**Key words:** high-definition video; H. 264 algorithm; real-time processing

引 言

在所有信息资源中,视频信息的开发、利用具有重要的理论意义和应用价值,人类有70%的信息是通过视觉获取的。视频信息具有直观性、确切性、高效性、广泛性等优点,同时视频信号的内容也是丰富多彩的,所以视频信号包含的信息量大,传输网络所需要的带宽相对较宽。高清视频的优越性更是公认的,然而高清视频的信息量将会更大,那么要存储与传输高清视频信息就会付出更大的代价,这就使得它的应用局限性很大。因此如何高效地对其数据量进行压缩编码是一个关键问题。

MPEG 和 VCEG 联合开发一个比早期研发的 MPEG 和 H. 263 性能更好的视频压缩编码标准,即 H. 264<sup>[1]</sup>。H. 264 算法的主导思想与以前的视频编解码标准是一致的,但它同时运用大量不同的技术,使得其视频编码性能远远优于其他标准。H. 264 编码复杂度也是最高的,所以掌握 H. 264 视频压缩标准的编码原理及关键技术,通过算法优化提高编码质量和编码速度是一个关键问题。  
当前实现视频编码的主要途径有以下几种<sup>[2]</sup>:基于 PC 的软件实现、基于 DSP 的实现、基于 FPGA 开发 ASIC 的图像压缩处理的实现。基于 DSP 的视频编码具有良好的并行特性、高度可编程性、超强的数据吞吐能力和运算速度。因此,本文选用 TI 推

出的最新产品TMS320DM8168。TMS320DM8168是一款处理能力强和运算速度快的DSP+ARM平台<sup>[3-4]</sup>,将H.264算法优化并移植到该处理器平台下,在不降低编码质量的情况下提高程序运行效率,降低运算复杂度。

1 算法介绍

H.264算法是目前性能最优的视频压缩算法标准,它包含许多优异的性能,包括率失真优化、多参考帧、树状结构帧间预测、码率控制、去方块滤波等多种关键技术,它通过高复杂度算法来提高编码性能<sup>[5]</sup>。

1.1 基本H.264算法

H.264是在以前的视频压缩技术基础上发展起来的,它的编码器结构与以往的编码器结构的基本组成是一样的,采用的仍然是变换和预测的混合编码法。H.264编码器主要由5部分组成:预测编码、变换编码、量化、熵编码以及环路滤波,基本框架如图1所示。图中 $D_n$ 为残差值, $X$ 为量化系数, $D'_n$ 表示经反量化,反变化的残差值, $uF_n$ 表示未经滤波的帧,T表示整数变换,Q表示量化。

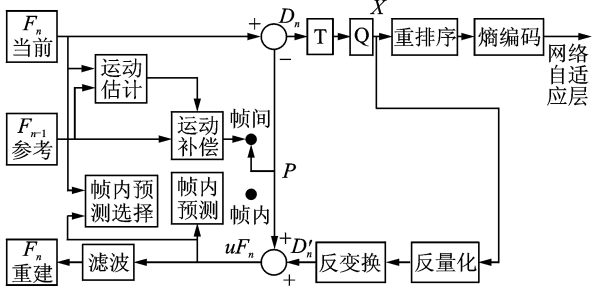


图1 H.264 编码器结构

H.264编码器的不同主要体现在各个功能模块的细节上,这也是H.264有优异编码性能的关键所在。H.264视频压缩的关键技术包括:基于块的多模式帧内预测、树状结构帧间预测、SP/SI帧技术、整数变换、CAVLC和CABAC熵编码、码率控制、去方块滤波等。帧内预测编码充分利用相邻宏块间的相关性,而不只是利用宏块内部像素之间的相关性,提高编码效率;帧间预测的特点是块尺寸范围更广、亚像素运动矢量及多参考帧的运用等;SP/SI帧技术解决视频流应用中终端用户可用带宽不断变化、不同内容节目拼接时数据量的激增、快进快退以及错误恢复等问题;整数变换在编解码器中使用相同精度,避免以往标准中使用的通用

$8\times8$ 离散余弦变换逆变换经常出现的失配问题,所以没有“反变换误差”;码率控制体现在通过选取合适的编码参数,控制单位时间内的编码视频流的数据量,使产生的比特流达到不同的应用要求;引入去方块效应滤波器,解决图像边界的视觉不连续性问题,提高视频编码性能。

1.2 H.264 算法优化

相对于传统的视频编码标准而言,H.264算法有很大的优势。但H.264在提高编码效率、增加编码灵活性的同时,也增加了编码的运算复杂度,其编码的实时性要求难以保证,因此有必要对H.264编码器的算法进行优化。编码器算法的优化主要包括帧内预测模式快速判决、帧间预测优化等。

1.2.1 帧内预测模式快速判决

H.264采用 $16\times16$ 和 $4\times4$ 两种亮度预测块,对于 $16\times16$ 子块有4种预测模式,对于 $4\times4$ 子块有9种预测模式<sup>[6]</sup>。传统的H.264算法是对两种子块的13种模式进行全搜索,然后基于RDO理论从中选取最优模式编码,大大增加了编码器的复杂度。本文首先利用相邻(左方A,上方B,左上方C)已编码块的模式预测当前编码块的模式,如果都是 $16\times16$ 则选取 $16\times16$ 预测模式,如果都是 $4\times4$ 则选取 $4\times4$ 预测模式;否则判断边缘相邻像素差之和D,如果D小于阈值TH则选取 $16\times16$ ,否则选取 $4\times4$ 。算法框图如图2所示。

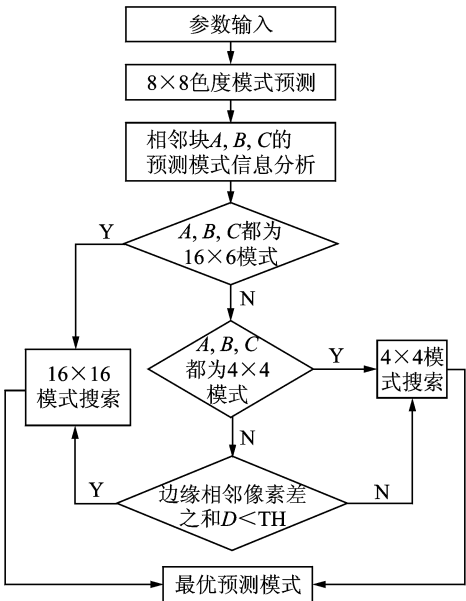


图2 帧内模式选择算法优化框图

1.2.2 帧间预测优化

帧间预测的计算复杂度在整个视频数据压缩编码系统中最大,占整个系统计算量的50%以上<sup>[7]</sup>,所

以研究帧间预测算法优化对提高视频编码效率有着重大的意义。帧间算法优化在本文里主要体现在帧间模式的快速选择和快速搜索算法的优化。

### (1) 帧间模式的快速选择

H. 264 使用不同大小尺寸的预测块进行运动估计,使得运动估计的运动矢量更加接近于物体的实际运动,提高了编码精度。一般而言,小尺寸的块分割可以提高细节处理能力和视频的主观质量,但用来表示分割类型和运动矢量的数据量就会增加,所以选取合适的预测模式对于提高视频编码效率很重要,已经有许多专家进行研究并取得良好的改进效果<sup>[8-9]</sup>。

H. 264 编码器的帧间预测宏块可以划分为:  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$ , 而  $8 \times 8$  模式还可以有 4 种分割形式:  $8 \times 8$ ,  $4 \times 8$ ,  $8 \times 4$ ,  $4 \times 4$ 。传统的帧间预测是对这 7 种模式按照  $16 \times 16$  到  $4 \times 4$  进行运动估计和运动补偿,然后根据率失真理论得到最优的预测模式。如果可以提前限制预测模式的范围,那么就可以大大减少搜索时间与计算量。本文就是基于这种思想,采用了一种新的帧间模式判决方法,这种方法主要是通过全零块预判决、运动矢量的时空相关性等特点来限制预测模式范围,从而达到快速选取最优模式的目的。如果编码块的残差经过变换、量化后的系数全部为零,那么对于这种全零块的变换、量化、反变换和反量化都是冗余操作,如果可以提前检测出全零块,就可以减少相应的计算负荷,提高编码效率。首先定义一个判别阈值 TH,如果绝对差值和小于这个阈值,就可以认为该块是全零块,采取 SKIP 编码模式;如果大于 TH,则计算 SKIP 模式代价函数  $J$ ,如果  $J$  小于  $16 \times 16$  模式的  $J$  则继续使用 SKIP 模式;否则分析邻近已编码块的分块与 MV 信息,判断当前编码块的运动情况,如果判断出运动剧烈则在小分割尺寸里搜索最优模式,如果运动平缓则在大分割尺寸里搜索。

### (2) 快速搜索算法

在匹配运动估计中的经典算法有全局搜索算法、菱形搜索算法、三步搜索算法、混合搜索算法等<sup>[10]</sup>。H. 264 采用匹配误差函数来反映整个运动估计的预测特性,如果能减少这个函数的计算量,那么就可以大大降低算法的计算复杂度。基于这种思想,本文采用子集匹配与搜索范围动态调整的方法来优化搜索算法。由于帧内像素的相关性,在计算匹配准则函数时,可以将块中的像素组成集合来计算,与逐点计算相比,可以大大减少计算量;同时,基于当前编码块的运动剧烈程度,可以动态选

取搜索范围来进行搜索,如果运动剧烈则选取大的搜索范围,运动平缓则选取小的搜索范围,这样也可以减少不必要的计算,在合适的区域里找到最优匹配点,得到当前编码块的运动矢量。

## 2 算法在 TMS320DM8168 平台的实现

### 2.1 平台的选择

TMS320DM8168 视频 SoC 可提供业界最佳的性能,将高清多通道系统的所有捕获、压缩、显示以及控制功能完美整合于单芯片之上,不断满足用户对高集成度、高清视频日益增长的需求。TMS320DM8168 有出色的视频处理能力,对于视频通信应用而言,可同时支持 3 个 60 帧/s 的 1 080 P 通道,由于编解码器时延低于 50 ms,因此解决了片外失效问题。本文选用处理能力强和运算速度快的 TMS320DM8168 作为高清视频编码实现的硬件平台。

TMS320DM8168 有强劲的硬件支持。TMS320DM8168 高度集成 1.2 GHz ARM Cortex™-A8 RISC 处理器,主要负责外围设备控制功能,包括数据读取、存储、控制等功能;1.0 GHz C674x VLIW DSP 处理器,负责视频压缩数据处理,高速数据处理功能;高清视频处理子系统,包括 2 个 165 MHz 视频采集通道和 2 个 165 MHz 的视频显示通道;高清视频编码协处理器,硬核编码协处理器,在高清应用中优势明显。此外还集成针对各种应用而优化的多种高带宽外设,包括 PCI Express Gen2、SATA 2.0、双千兆位以太网和双 DDR2/DDR3 接口,不仅可提供各种不同的网络连接选择和存储选项,还能在片上及片外实现快速的数据传输。它还带有扩展 I/O 子卡,包括用于 DVI 输入、分量视频输入、复合视频输入、HDMI TX、串行端口、以太网和音频输出的连接器。

### 2.2 实现难点与优化方案

将 H. 264 算法实现并优化到 TMS320DM8168 平台的 DSP 端和 ARM 端,需要做几个相关工作: x264 在 TMS320DM8168 平台的移植、基于 TMS320DM8168 平台的 x264 程序优化、TMS320DM8168 的存储器优化等。

#### 2.2.1 算法实现难点

(1) 本文采用的高清视频图像大小为 720 P (1 280 × 720),采样格式为 4 : 2 : 0,因此每帧图像占用的空间为  $1\,280 \times 720 \times 1.5 = 1\,382\,400\text{ B} = 1.3\text{ MB}$ ,根据 x264 的要求,程序最少需要同时保

留 2 个参考帧的数据,总共需要的存放图像的空间最少为 2.6 MB,仅数据部分即远大于 TMS320DM8168 所提供的片内内存大小(64 KB + 512 KB),因此很大部分帧数据都需要存放在片外存储器,降低系统的编码速度。

(2) x264 程序代码中有大量的 for 循环,函数多,函数嵌套复杂,调用造成系统运行缓慢,降低编码实时性。

(3) x264 代码数量大,所涉及的堆和栈占用空间大,在程序应用过程中会降低执行速度,甚至造成 DSP 死机现象。

2.2.2 系统软件框架

TMS320DM8168 是 TI 最新推出的双核系统 ARM+DSP,像以往的双核系统一样,ARM 和 DSP 之间的通信通过 TI 提供的 Codec Engine 软件框架进行。Codec Engine 是介于应用程序和具体算法之间的软件模块,其中的 VISA API 通过 stub 和 skeleton 访问 Engine SPI 最终调用算法。ARM 和 DSP 的共享缓冲内存是通过 CMEM 模块在 DDR 中分配的,缓冲内存地址连续且与 DSP 核 Cache 相关。

ARM 内核运行基于 Linux(Ubuntu10.04)操作系统的应用程序,所用的外围设备都由 ARM 负责控制。通过读取存储在 SD 卡或者通过网络文件系统(Network file system, NFS)的方式读取存储在 PC 上的 720 P 测试视频文件,并将该视频数据送到 DSP 和 ARM 的共享缓冲内存,通知 DSP 执行 x264 视频压缩算法。DSP 接收到该信号后,从共享内存中读取视频压缩数据,进行视频压缩编码,将编码后的数据放回共享内存中,并由 ARM 端的应用程序进行存储。

整个系统的软件框架如图 3 所示。

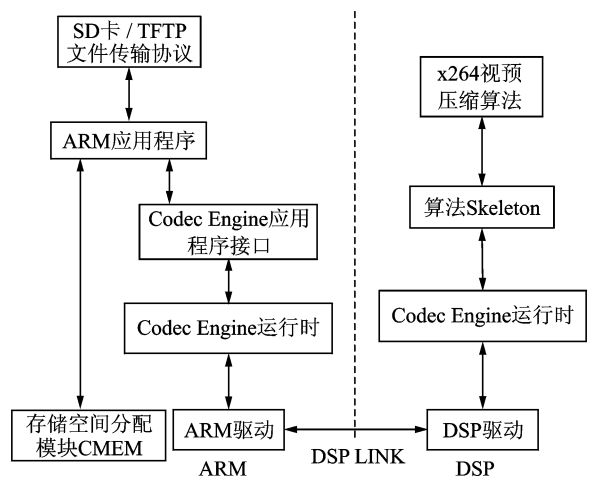


图 3 系统软件框架图

要使 x264 算法能够在 Codec Engine 的框架内运行,必须对 x264 算法进行 XDM 封装。XDM 是 TI 为数字图像处理应用而提供的一套名为 VISA 的 API 集合,其分为 Video, Image, Speech, Audio 等。为降低开发难度,本文选用 Codec Engine 提供的 examples 中的 VIDENC1\_COPY 作为底层算法库进行开发,在 algAlloc 中进行存储空间分配,在 VIDENC1\_TI\_Process() 函数中封装视频压缩算法,在 VIDENC1\_TI\_Control 中对异常情况进行处理,在 algFree() 中存储空间进行释放。同时,通过扩展 XDM 接口的入口参数,增加 param 等参数,灵活配置 process() 函数的入口参数,修改 package.bld 文件,将所有的资源文件添加 package.bld 文件中用于编译,实现对 x264 算法的 XDM 封装。

2.2.3 x264 在平台上的移植

在将 x264 移植到 TMS320DM8168 过程中,需要处理以下几个问题:冗余代码的去除、x264 主文件的重写、数据类型和头文件的修改。

(1) 冗余代码的去除

x264 源代码包含多种处理器平台的汇编代码和大量的与 TMS320DM8168 平台无关的处理代码,应当对代码进行精简处理:

- ① 针对 Baseline 档次的编码级别,在该档次中没有用到 B 帧,所以删除所有与 B 帧相关的语句及函数,如: x264\_mb\_analyse\_inter\_b8x8() 等。
- ② 因为菱形算法的速度最快,所以采用优化过的菱形算法作为搜索算法,并删除其他的算法。
- ③ 删除与命令行输入的相关代码,所有参数均在 x264\_param\_default(&param) 函数中设置。
- ④ 删除原代码中用来记录编码信息的函数,如 x264\_log() 函数等,和编码完全无关,去除这些函数有利于加快编码速度。
- ⑤ 原 x264 代码中为 X86 平台提供了很多优化加速代码,这些代码在 TMS320DM8168 平台上不适用,所以删除这些代码,如 x264\_cpu\_detect() 等函数。

(2) x264 主文件的重写

针对 TMS320DM8168 平台,按照编码流程,需要重写主文件。重写后的主文件主要流程如图 4 所示。

(3) 数据类型和头文件的修改

TMS320DM8168 的 DSP 编译器和 VC 环境下运行的数据类型并不是完全相同的,因此需要对数据类型做相应的修改。例如,VC 中的 \_int64 数据类型在 DSP 编译器中并不存在,需要相应修改为

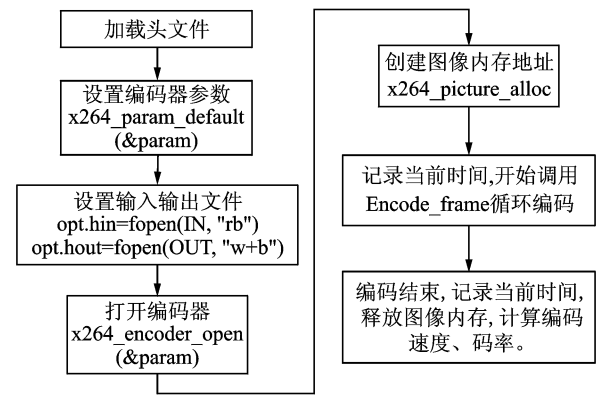


图 4 重写 x264 主文件

long 类型;源代码中调用函数 x264\_mdate()来记录时间,该函数所用到的头文件在 TMS320DM8168 编译器中无效,通过加载 time.h 头文件中的 clock()函数来代替 x264\_mdate(),并修改相应的计算公式。

2.2.4 x264 在 TMS320DM8168 平台的优化

x264 在 TMS320DM8168 平台的优化主要是根据 TMS320DM8168 平台的具体特点对代码数据结构优化、循环的优化、代码功能的精简以及代码的并行处理机制的实现。主要包括内联函数的使用、大循环的拆分、代码的线性汇编重写等。

(1)内联函数的使用。函数在调用的过程中,会发生寄存器的出栈与入栈。过度频繁的栈操作会浪费大量的时钟周期,影响程序的整体性能。在一些调用频繁的子函数前添加“inline”关键字,编译器在编译的过程中,会将这些子函数编译进主函数体中,从而成为一个整体,不会发生频繁的函数调用。

(2)大循环的拆分。计算机程序运行时,指令的访问并不是杂乱无序的,而是在一段时间内会集中访问部分指令,这就是程序的局部性原理。计算机中广泛采用的 cache 技术就是根据程序的局部性原理设计的,目的是减少存储器访问时间。将大循环进行拆分,可以增大 cache 的命中率,对代码的整体性能提升有很大帮助。

(3)代码的线性汇编重写。高清视频编码中包含大量的计算过程,主要有运动估计中的块匹配、离散余弦变换、绝对差值和的计算等。将这些模块对应的 C 语言代码用 TMS320DM8168 平台上的汇编重写,能够有效地提高运算效率,改善程序的整体性能。

3 实验结果与分析

本文的实验工作主要进行了 ARM+DSP 的算

法编码方案的性能测试。

该方案即是通过 Codec Engine 实现视频压缩算法在 TMS320DM8168 上的实现,其中 ARM 端主要负责视频数据的读取,DSP 端负载 x264 算法的视频编码部分的实现。应用 H. 264 算法对 Akiyo, Mobile, Park\_joy 序列作编码,该方案的测试结果如表 1 所示,编码图像质量如图 5 所示。

表 1 DSP+ARM 方案测试结果

测试序列	Akiyo	Mobile	Park_joy
视频分辨率	352×288	720×576	1 280×720
编码速率/(帧·s <sup>-1</sup> )	75.57	21.98	8.32
平均 PSNR/dB	44.430	35.230	28.343
码率/(kb·s <sup>-1</sup> )	224.14	3 506.35	5 933.94

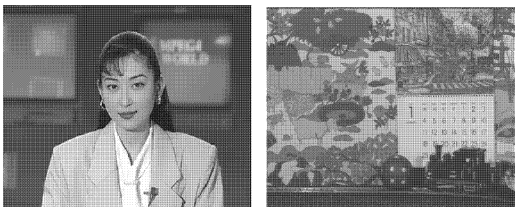


图 5 编码图像质量示意图

4 结束语

本文在全面分析 H. 264 算法的基础上,在 TMS320DM8168 平台上实现对视频文件的编码显示,主要完成了以下工作:(1)在分析基本 H. 264 算法的基础上,实现了对帧内选择模式和帧间选择模式的优化,对编码效率的提高起了基础研究作用;(2)改写优化过后的 H. 264 代码,通过 Codec Engine 机制移植到 TMS320DM8168 的 DSP 端;(3)程序级优化措施,包括内联函数的使用,大循环的拆分,线性汇编的编写、存储空间分配合理等;(4)将优化 H. 264 算法移植到 TMS320DM8168 上,利用 TMS320DM8168 强大的视频编码能力对高清视频进行压缩处理。通过算法的优化和程序的

优化,依赖于TMS320DM8168强大的视频处理能力,本文实现了对视频文件的压缩编码并显示,对高清视频实时编码技术的发展和产业应用具有重要的意义。

#### 参考文献:

[1] Wiegand T, Sullivan J, Bjntegaard G. et al. Overview of the H. 264/AVC video coding standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003,13(7):560-576.

[2] 刘仕翔. 基于DM642 DSP 的x264 编码器研究及实现[D]. 成都:西南交通大学电气工程学院,2010.

Liu Shixiang. The research and realization of H. 264 based on DM642 DSP[D]. Chengdu: School of Electrical Engineering, Southwest Jiaotong University, 2010.

[3] Texas Instruments. TMS320DM816x davinci digital media processors [EB/OL]. 2011-06-06. <http://www.ti.com>.

[4] Texas Instruments. TMS320DM816x davinci digital media processors technical reference manual [EB/OL]. 2011-04-07. <http://www.ti.com>.

[5] 卢光辉. P 帧快速模式选择算法[J]. 电子科技大学学报,2010,39(3):397-401.

Lu Guanghui. Fast mode checking algorithm for P-frame[J]. Journal of University of Electronic Science and Technology of China,2010,39(3):397-401.

[6] 王凤琴,黄敏,邓璐娟,等. 基于H. 264/AVC 的快速帧内预测模式决策方法[J]. 数据采集与处理,2008,23(6):676-701.

Wang Fengqin, Huang Min, Deng Lujuan, et al. Fast intra mode decision algorithm based on H. 264/

AVC[J]. Journal of Data Acquisition & Processing, 2008,23(6):676-701.

[7] 吴晓军,白世军,卢文涛. 基于H. 264 视频编码的运动估计算法优化[J]. 电子学报,2009,37(11):2541-2545.

Wu Xiaojun, Bai Shijun, Lu Wentao. Optimization on motion estimation algorithm based on H. 264[J]. Acta Electronica Sinica, 2009,37(11):2541-2545.

[8] Du Anmu, Chun Jiang, Zhou Donghui. A fast inter mode selection algorithm based on the statistical features for the H. 264[J]. Journal of Convergence Information Technology, 2012,7(10):372-380.

[9] Hsieh J L, Yang S H, Yang C C. Fast inter-mode selection for H. 264 encoders based on coded block patterns and interblock correlation [J]. Journal of Information Science and Engineering, 2011,27(6):1901-1917.

[10] 高燕. H. 264 实时视频压缩编码器关键算法研究及DSP 实现[D]. 重庆:重庆大学通信工程学院,2010.

Gao Yan. The key algorithm research and realization of H. 264 video compression encoding system based on DSP[D]. Chongqing: Institute of Communication Engineering of Chongqing University,2010.

**作者简介:**姜忠兵(1986-),男,硕士研究生,研究方向:信息处理、高清视频编码,E-mail:jiangzhongbing20@126.com;罗钧(1963-),男,教授,研究方向:数码影像科技、嵌入式系统、精密仪器及机械、测试计量技术与仪器;杨晓花(1988-),女,硕士研究生,研究方向:测试与控制技术、视频编码;孟凯(1987-),男,硕士研究生,研究方向:仪器科学与技术、视频编码;彭加新(1987-),男,硕士研究生,研究方向:信息获取与处理。