

MPEG-4 算法在 DM642 上的实现与优化*

任克强 朱志敏

(江西理工大学美国德州仪器数字信号处理方案实验室 赣州 341000)

摘要 在 DM642 EVM 上实现了 MPEG-4 视频编解码算法,并对算法程序进行了优化。实验结果表明,经过优化的 MPEG-4 算法程序,在分辨率为 720×576 进行实时采集时,可以达到 28F/S 的实时要求。

关键词 DM642 MPEG-4 优化 VOP 视频编解码

中图分类号 TP311

Optimization and Implementation of MPEG-4 Algorithm Based on DM642

Ren Keqiang Zhu Zhiming

(Texas Instruments - Jiangxi University of Science and Technology DSPS Laboratory, Ganzhou 341000)

Abstract The MPEG-4 video encode/decode algorithm was implemented based on DM642 EVM, and optimized the program of MPEG-4 algorithm. Experimental results show that the optimized algorithm program could measure up real-time requirements of 28F/S, which was real-time capture in the resolution of 720×576 .

Key words DM642, MPEG-4, optimization, VOP, video encode/decode

Class Number TP311

1 引言

MPEG-4 是继 MPEG-1 和 MPEG-2 之后,国际运动图像专家组(MPEG)制订的又一个标准。MPEG-4 能够获得更高的视频压缩率,具有基于内容的交互能力,国内外许多公司都在研究和开发基于 MPEG-4 的应用产品,其中许多产品都是采用可编程媒体处理器来实现的。MPEG-4 视频编解码算法的运算量较大,要求处理器具有较高的运算速度和较强的数据处理能力。本文在 DM642 EVM 平台上,实现了 MPEG-4 简单框架类视频编解码算法,并对算法程序进行了综合优化。实验结果表明,经过优化的 MPEG-4 算法程序,在分辨率为 720×576 进行实时采集时,可以达到 28F/S 的实时要求。

2 MPEG-4 算法原理及实现

与传统视频编解码标准不同的是,MPEG-4 是基于内容的视频压缩方式。在 MPEG-4 中引入了视频对象 VO(Video Object)来实现基于内容的表示^[1],一个视频对象可以是视频场景中任意形状的一块区域,也可以是视频序列中的人物或具体的景物,它的存在时间可以任意长。视频对象平面 VOP(Video Object Plane)是某一具体时刻的视频对象。对输入的视频序列进行分解,可将其分割为多个 VO,对同一 VO 编码后形成 VOP 数据流。

一个 VOP 是对一个视频对象的时间采样,包括了视频对象的形状信息、运动参数和纹理数据。VOP 可以是相互独立的编解码,也可以通过运动补偿依靠其它 VOP 编解码。MPEG-4 的编解码

* 收稿日期:2008 年 5 月 30 日,修回日期:2008 年 7 月 9 日

基金项目:江西省教育厅科技项目资助(赣教技字[2007]207 号)。

作者简介:任克强,男,教授,硕士生导师,研究方向:嵌入式系统。朱志敏,男,硕士研究生,研究方向:嵌入式系统。

实现是以 VOP 内部 16×16 的宏块 MB (Macro block) 为单位进行编解码, 每个亮度宏块中包含有 4 个 8×8 的图像块, 一个 VOP 可划分成多个宏块。对 VOP 的编码就是针对某一时刻图像 VO 的形状、运动和纹理信息进行编码。

MPEG-4 采用层次化的数据结构来实现可分级编码。视频从基本层开始, 加上一定数量的增强层, 这样就可以根据不同的应用来有选择地进行视频编解码^[2]。MPEG-4 的分级描述如图 1 所示。

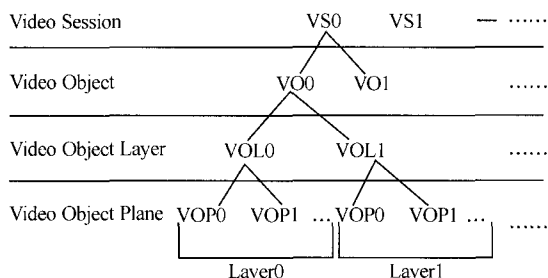


图 1 MPEG-4 的分级层结构

本文采用 MPEG-4 SP 简单框架类, 在 DM642 EVM 平台上实现了 MPEG-4 视频编解码算法。图 2 和图 3 所示为简单框架类的 I 帧和 P 帧编码和解码流程。

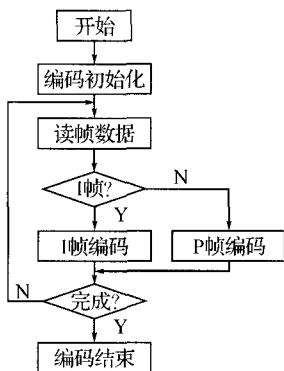


图 2 编码流程

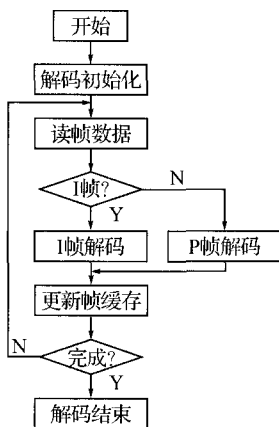


图 3 解码流程

3 算法程序的优化

DM642 EVM 采用 TI 公司的专用多媒体处理芯片 TMS320DM642 (简称“DM642”) 处理器^[3]。DM642 基于 C64x 内核, 600MHz 的时钟频率, 采用超长指令字 (VLIW) 结构; 包括两条数据路径, 它们之间可以进行数据的交叉存取; DM642 片上内存采用二级缓存 Cache 结构, L1 由 16KB 的数据 Cache L1D 和 16KB 的程序 Cache L1P 组成, 256KB 的 L2 可配置成 SRAM 或 Cache, 大幅度提高了程序的运行性能; 片内 64 位的 EMIF (External Memory Interface) 接口可以与 SDRAM、Flash

等存储器件无缝连接, 极大地方便了大量数据的搬运; DM642 包括了 3 个专用的视频端口 (VP0~VP2), 用于接收、处理视频数据, 提高了整个系统的性能; DM642 自带的 EMAC 口以及从 EMIF 口扩展出来的 ATA 口, 还为处理完成后产生的海量数据提供了存储通道。

MPEG-4 视频编解码算法程序直接在 DM642 上运行, 效率较低, 远远不能达到实时的要求。因此还需要结合 DM642 的结构特点进行综合优化^[4]。

1) 编译选项参数的选择

CCS 提供了功能较强的编译选项, 从 -o0 到 -o3 共四级优化; -pm 选项从程序角度来考虑, 把整个程序作为一个模块来处理。-o3 和 -pm 选项一起使用, 性能优化显著提高的同时, 代码大小也得到了优化。

2) 存储空间的优化和 Cache 的使用

将一些占据较多空间的数据或不常使用的程序存放于片外存储区, 启用 L2 Cache, 调用 C6000 芯片支持库 CSL (Chip Support Library) 中的 CACHE_setL2Mode 库函数, 将 L2 设置为 192KB SRAM 和 64KB Cache 的模式, 再调用 CACHE_enable Caching 库函数, 将存有数据的片外存储区设置为 Cache 可缓存模式。这样, L2 Cache 就可以缓存该片外存储区中的程序或数据。

3) C 程序级优化

使用 CCS 中的 profiler 工具, 对 C 代码进行评估, 找出运算量最大的程序段, 这些程序段的优化对提高算法程序的性能有显著影响^[5]。

(1) 使用 C6000 特有的关键字和内联函数来改写 C 代码。使用关键字 restrict 可消除数据间的相关性以提高代码并行执行能力, 而使用内联函数 intrinsics (如 _nassert(), _mpy2()) 可快速优化 C 代码, 提高代码在 DSP 中的执行效率。

(2) 使用整型访问短型数据; 使用 32 位整型一次访问 2 个 16 位短型数据, 可减少对内存的访问次数, 将程序读取数据的效率提高一倍; 使用能同时对 2 个寄存器对应高低 16 位进行操作的內联函数 (如 _add2(), _mpy2() 等), 可大大提高代码的执行效率。

(3) 采用循环展开的方法, 将多循环变为少循环甚至单循环, 减少循环嵌套, 消除冗余循环, 可以提高指令并行执行的程度。

(4) DSP 没有专门的硬件除法运算单元, 除法都用连续减法实现, 运算量比较大, 所以要尽量减

少除法运算,不能减少的除法用移位运算来实现,可减少运算耗时。

4)调用 IMGLIB 库函数

DM642 中为一些常用图像处理算法提供了执

行效率很高的库函数,封装为 IMGLIB。这些函数是优化过的,其代码效率高,可直接应用到程序中。算法程序中部分函数优化前后的对比情况表 1 所示,可以看出优化后其性能有明显的提高。

表 1 部分函数优化前后性能比较

函数名	优化前(cycles)	优化后(cycles)	提高的倍数
transfer_16×16copy	403	33	12.2
transfer_16to8copy	550	41	13.4
transfer_16×16mpy	633	68	9.3
dequent_intra	1123	265	4.3
dequent_inter	1179	223	5.2

5)利用 EDMA 传递数据

DM642 支持 EDMA 功能,通过 EDMA 通道提前把外设或片外存储器中的数据直接搬运到片上内存,这样对 CPU 来说,所访问的数据总是在片内的,没有阻塞的情况发生,减少 CPU 的等待时间。

在视频编码部分,通过摄像头采集一帧图像,使用 EDMA 通道把图像传输到片外内存,同时 CPU 对前一帧图像的编码。对宏块编解码时使用两对片上缓存,一对缓存用做要编解码的宏块,另一对缓存用做宏块码流。使用 EDMA 通道把宏块读入/输出到片上内存,同时使 CPU 总处于忙状态,可提高编码和解码的效率。这种乒乓式的缓存结构如图 4 所示。

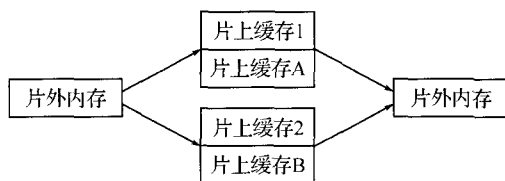


图 4 乒乓式片上缓存

6)汇编程序级优化

线性汇编语言是 TI 6000 系列 DSP 独有的一种编程语言,介于高级语言和低级语言之间。与标准汇编语言不同的是,在编写线性汇编程序时不必考虑指令的延时、寄存器的使用和功能单元的分配,可完全按照高级语言的方式编写,汇编优化器将根据代码情况自动确定这些信息。本文将代码中运算量大、调用频率高的关键部分用线性汇编进行编写,如量化、SAD 等模块。

7)软件流水

软件流水是一种对循环中的指令进行调度优化的技术,利用软件流水可生成非常紧凑的循环代码。当编译时采用 -o2 或 -o3 级别的优化选项时,编译器将对程序中的循环进行软件流水。通过软件流水的优化,可以大大提高循环代码的效率,

极大地实现指令的并行性。

4 算法测试结果

通过以上对算法程序的综合优化后,本文对 CIF 等格式的视频序列进行了测试。将测试文件从 PC 传到 DM642 EVM 开发板上,由 DM642 对测试文件进行编解码。算法程序优化前的编码为 0.12 帧/秒左右,解码为 0.8 帧/秒;算法程序优化后的编码可达 28 帧/秒,解码达 80.5 帧/秒。优化后的 MPEG-4 编解码器性能见表 2。

表 2 优化后 MPEG-4 编解码器性能

序列	格式	平均 PSNR(/dB)	帧率(帧/秒)
foreman	CIF(352×288)	38.9	45
Mobile	4SIF(704×480)	35.8	22

从 CCD 摄像头获得的图像大小为 720×576pixel,格式为 YUV4:2:2,图 5 和图 6 分别是编码前和解码后得到的图像,其画面无失真现象,图像质量并没有明显下降。

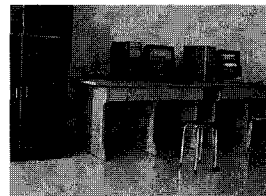


图 5 编码前的图像

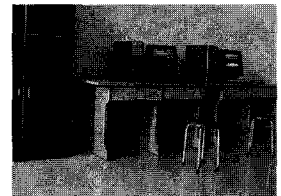


图 6 解码后的图像

5 结语

本文探讨了 MPEG-4 视频编解码算法在 DM642 上的实现与优化,实现了 MPEG-4 简单框架类视频编解码算法,重点介绍了编解码算法程序的优化方法。实验结果表明,所设计的编解码算法具有较高的易实现性和实用性,性能测试获得了满意的效果。在此基础上,还可进一步对 MPEG-4

(下转第 22 页)

0;0 1 0 1 0;1 0 1 0 1],采用腐蚀算子和膨胀算子进行边缘提取。图 3 给出了采用不同的结构元素得到的边缘检测结果比较。由图 3 看出,结构元的抗噪能力 $S4 > S3 > S2 > S1$,若要形态边缘检测算子具有较好的去噪能力,则形态边缘检测算子的结构元尺度必须大于或等于噪声点的尺度。



(a)加噪原图像 (b)Sobel 算子检测结果 (c)形态学算子检测结果

图 2 经典算子与形态学算子边缘检测效果比较(加噪)



(d)S3 腐蚀检测结果 (e)S4 腐蚀检测结果 (f)S1 膨胀检测结果

图 3 不同的结构元素的边缘检测性能比较(加噪)

4 结语

研究了一种基于多尺度数学形态学的边缘提取方法。实验比较结果表明:基于数学形态学的边缘提取方法对于含噪图像的边缘提取有很好的表现;利用多尺度结构元素形态学方法能够有效地改进含噪图像的边缘提取效果。

参考文献

- [1]崔屹. 图象处理与分析—数学形态学方法及应用[M]. 北京:科学出版社, 2000
- [2]段毅华,王雨点. 基于分形和数学形态学的图像边缘检测方法[J]. 计算机工程与应用, 2004, 40(14):63~64
- [3]付永庆,王咏胜. 一种基于数学形态学的灰度图像边缘检测算法[J]. 哈尔滨工程大学学报, 2005, 26(5):685~687
- [4]赵怀勋,郑敏,李志强. 一种基于数学形态学的含噪、低对比度图像的边缘检测方法[J]. 计算机与数字工程, 2006, 34(7):11~13
- [5]范立南,韩晓微,王忠石,徐心和. 基于多结构元的噪声污染灰度图像边缘检测研究[J]. 武汉大学学报(工学版), 2003, 36(3):86~90

(上接第 9 页)

高级框架类视频编解码算法的实现和优化方法进行深入研究,以满足更高的应用要求。

参考文献

- [1]钟玉琢. 基于对象的多媒体数据压缩编码标准—MPEG4 及其校验模型[M]. 北京:科学出版社,2000
- [2]李晓飞,陈来春等. MPEG 标准及其应用[M]. 北

京:北京邮电大学出版社,2002

- [3]李方慧,王飞等. TMS320C6000 系列 DSPs 原理与应用[M]. 北京:电子工业出版社,2003

[4]TMS320C64x Optimizing C/C++ Compiler User's Guide, Texas Instruments Incorporated, 2001

- [5]TMS320C6000 Programmer's Guide. Texas Instruments Incorporated, 2000