

AVS - M 解码在 Symbian 平台上的实现

熊爱平^{1,3}, 周经野¹, 刘东华^{2,3,4}(1. 湘潭大学 信息工程学院, 湖南 湘潭 411105; 2. 国防科学技术大学, 湖南 长沙 410073;
3. 上海中科计算技术研究所, 上海 210213; 4. 中国科学院 计算所, 北京 100080)

摘要: AVS - M 是《先进音视频编码》系列标准 AVS 的第七部分, 是无线网络与手机等移动设备视频编解码的规范和标准, 高效快速的编解码器对 AVS - M 标准能否得到推广和应用有着极其重要的作用。介绍了 AVS - M 标准参考代码移植到 Symbian 平台的方法和要解决的问题, 并针对 Symbian 手机平台对 AVS - M 标准参考代码给出几种优化方法, 实现了 AVS - M 在 NOKIA 6670 上解码 QCIF 格式达到 15 帧/秒, 基本上能够在 Symbian 平台的流畅播放, 对推广 AVS - M 有很大的帮助。

关键词: AVS - M 标准; 视频; Symbian 平台; 解码

中图分类号: TN925.5

文献标识码: A

文章编号: 1673 - 629X(2008)02 - 0133 - 03

Implementation of AVS - M Decoder on Symbian Platform

XIONG Ai-ping^{1,3}, ZHOU Jing-ye¹, LIU Dong-hua^{2,3,4}

(1. College of Information, Xiangtan Univ., Xiangtan 411105, China;

2. National University of Defense Technology, Changsha 410073, China;

3. Shanghai Institute of Computing Technology, Shanghai 201203, China;

4. Institute of Computing Technology, Chinese Academy of Science, Beijing 100080, China)

Abstract: AVS - M coding standard is the seventh part of advance audio and video coding standard. It is a video coding standard for wireless network, handset and other mobile devices. One key for broad application of AVS - M is to implement efficient and rapid software AVS - M encoder. Introduces the AVS - M standard reference code transplanted to the Symbian platform and the method to solve the problem, and Symbian mobile platform against the AVS - M standard reference code gives several optimization method. Implementation of AVS - M in 6670, NOKIA decoding QCIF format to achieve 15 frames/s. Basically, achieve a further Symbian platform fluent broadcast, it is useful to promote AVS - M.

Key words: AVS - M; video; Symbian; decoder

0 引言

Symbian 是一种用于手机上应用程序开发和运行的开放式工业标准操作系统, 是新一代手机的基础, 它是一个实时性、多任务的 32 位操作系统, 具有功耗低、内存占用少等特点, 非常适合手机等移动设备, 经过不断完善, 可以支持 GPRS、蓝牙、SyncML, 以及 3G 技术。在目前手机操作系统的争夺中, Symbian 市场占有率达到了 70%。

AVS 标准^[1]是基于我国自主创新技术和国际公

开技术所构造的信源编解码标准, 是视音频编解码领域一批关键技术的集中体现。AVS 标准的制定和推广对减轻国内视音频相关产业的专利费负担以及提升核心竞争力具有极其重要的意义。AVS - M 是 AVS 的第七部分: 移动视频, 是车载终端、手机等移动设备视频编解码的规范。目前该标准的最终文档已报审并已公示, 有望成为下一代移动视音频编解码标准。

随着通信技术和多媒体技术的不断发展, 手持终端设备的发展必然要具备视频播放功能。目前, 很多中高端手机都支持 MPEG - 4, H. 264 等标准, 为了 AVS - M 的发展和推广, 设计了基于 Symbian 平台的 AVS - M 解码软件。

1 Symbian OS

Symbian OS 手机^[2]提供 3 种针对 C++ 程序员开

收稿日期: 2007 - 05 - 25

基金项目: 上海市科技发展基金创新基金创新攻关项目 (PKJ2006 - L04)

作者简介: 熊爱平 (1982 -), 女, 湖南长沙人, 硕士研究生, 研究方向为视频编解码; 周经野, 教授, 研究方向为自然语言处理。

发的用户界面: S80, S60, UIQ。可用的开发工具包括 Metrowerks 的 CodeWarrior, Borland 的 C++ Builder, 以及微软的 Visual Studio。Symbian 操作系统有如下一些主要特性:

- 1) 操作系统运行在 ROM。
- 2) 系统运行在电池驱动设备上。
- 3) 完全面向对象。
- 4) 栈和内存比较小。
- 5) 被设计成确保用户的数据不会丢失。
- 6) 简单的 UI 系统。

2 AVS-M 解码器改进和优化

AVS 工作组提供有参考解码软件, 但速度较慢, 在 PC 机器(奔腾 4 代 CPU 3.0 GHz, 内存 512MB) 上解码 80 帧/秒, 要将 AVS-M 移植到 Symbian 平台上就必须提供解码器的足够速度。下面首先找出速度瓶颈, 然后结合 Symbian 平台对解码器进行改进和优化。

2.1 主要优化思想

代码优化都是在以下方面进行改善: 更快的运行速度、有效的资源使用、更小的内存使用。一般情况下, 在优化之前, 找出解码器的瓶颈, 也就是哪些模块最费时, 根据代码的情况分重点进行测试。经过微软的测试工具 Intel VTune(TM) 测试, 结果见表 1。

表 1 试验数据

Function	Clockticks samples(52)	Clockticks %(52)	Clockticks events(52)	Size(52)
get_block	1 093	14.75	2 186 000 000	0x2946
Inv_Transform_B4	276	3.73	552 000 000	0x2b4
Recon_B4	206	2.78	412 000 000	0x2C7
Dequant_B4	106	1.43	212 000 000	0xF3
Get_residual	89	1.2	176 000 000	0x160
decode_golomb_word	71	0.96	142 000 000	0x108
readLumaCoeff_4x4_UVLC	65	0.88	130 000 000	0xC1D
write_frame	43	0.58	86 000 000	0x150
intrapred	41	0.55	82 000 000	0x1896
intrapred_chroma_4x4	25	0.34	50 000 000	0x8CF

结果说明, 在不考虑调试和动态分配内存所占时间的情况下, 插值计算、反变换、反量化、变长解码等几个模块占了解码的大部分时间。

2.2 具体优化措施

1) 程序结构优化。

由于参考解码器判断比较多, 程序比较复杂, 冗余也比较多, 优化后的解码器根据宏块类型(P-16*16, P-16*8, P-8*16, P-SKIP, P-8*8, I-4*4) 分情况讨论, 分别写成不同的函数, 取代了原来一个函

数解决所有的情况, 使程序在执行过程中少了很多判断和跳转语句, 简化了函数。虽然整体上增加了代码, 但提高了代码执行效率^[3]。

2) 避免频繁地分配和释放内存。

解码器频繁分配和释放内存也会严重影响解码器的速度。参考解码器图像头解析函数 Header() 之所以占有很多的解码时间, 就是因为每次调用该函数时需要分配一个 MAX_CODED_FRAME_SIZE 大小的 NAL 单元缓冲区, 读取其中的信息之后再销毁。每解码一帧需要调用该函数一次, 因此, 内存的频繁分配和释放占用了大量时间。在帧一级别申请和释放内存是一种极为不明智的方法。优化设计后的解码器将 NAL 单元缓冲区、参考图像缓冲区、当前图像缓冲区、残差数据缓冲区等的内存分配和释放统一放到解码器初始化函数 decoder_init() 当中, 对所有内存只申请和释放一次, 并在解码器销毁函数 decoder_destroy() 中进行内存释放。

3) 避免大量缓冲区的倒换^[4]。

在参考解码器中, 它为各个模块分配了很多临时的缓冲区, 各缓冲区需要大块的数据进行移动, 极大地影响了解码速度, 如图 1 所示。而优化后的解码器(如

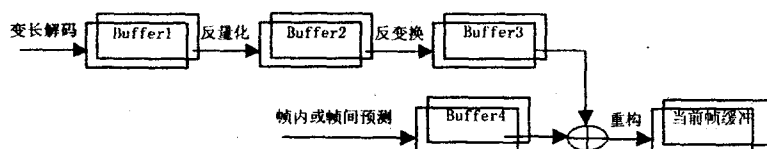


图 1 参考解码器的反量化、反变换、重构计算图 2 所示), 省去了 buffer1 和 buffer2, 把变长编码、反量化、反变换三个模块集中在一个缓冲区计算, 避免了大量数据的移动, 节省了时间。

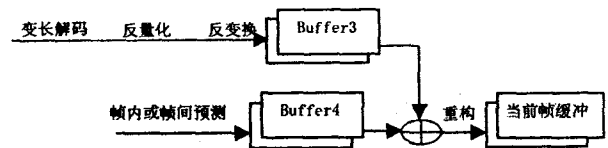


图 2 优化设计解码器的反量化、反变换、重构计算

4) 插值模块的优化。

AVS-M 采用水平 8 抽头滤波器(-1, 4, -12, 41, 41, -12, 4, -1) 和 4 抽头滤波器(-1, 5, 5, -1) 进行二分之一像素的水平和垂直插值, 并采用线性滤波器进行四分之一像素插值。参考解码器把 8 抽头滤波器和 4 抽头滤波器的系数分别封装在两个数组中, 通过 FOR 循环实现参考帧的数据与系数数组相乘, 而优化后的解码器则是直接展开循环把 FOR 循环去掉, 同时展开乘法运算表达式把系数数组也去掉, 直接将系数与参考帧数据相乘, 明显提高了运算速度^[3]。

5) 全零个数判断的优化。

全零个数的判断在优化后的解码器中,是个非常耗时的模块。每个哥伦布码的解析,首先是全零个数的判断。由于该模块调用非常频繁,同时参数解码器使用循环和移位一位一位的判别来实现,直接导致该模块执行效率低下,耗时严重。优化后的该模块,采用逐步缩小步长的方法实现。若判断是一个字节,直接判别该字节是否全为零。若为零,读取下一个字节继续判断,否则逐步缩小步长,如判别 4 位、判别 2 位,直接精确到 1 位。由于没有使用循环,同时使用可变步长,该方法的效果是原来方法的 3.5 倍^[5]。

3 移植的方法及问题

3.1 移植方法

Symbian OS 中使用 C++ 和其它环境中的 C++ 没有很大的区别,可以直接把 C 代码移植到 Symbian 平台,移植过程中需要注意的是, Symbian 不支持全局变量和静态成员变量,如果程序中有全局变量或者静态成员变量,在模拟器上编译的时候不会报错,因为模拟器使用基于 VC 的编译器,而 Symbian 目标设备使用的是 gcc 的编译器,所以在真机上不能执行。

具体的移植方法如下:

1) 把解码器的函数集成到一个类 AvsmEngine(名字任意),成为 class AvsmEngine 的成员函数,所有的全局变量成为类 AvsmEngine 的私有成员变量,把解码器中的静态成员变量 static 改成 const,避免出现静态成员变量。

2) 在 MMP 文件(MMP 文件可以作为任何建立应用程序选项、IDE 和开发平台的出发点)里加入需要调用的相应 LIB 库,加入要调用的库文件的路径。

3) 程序编译和运行,编译包括以下几步:

①创建 abld 文件,在当前工程目录下执行命令 bldmake bldfiles,在当前路径下生成 abld.bat 文件;

②运行 abld build wins udeb 生成必要的文件,并且把工程添加到模拟器上,如果运行成功,则可以打开模拟器运行此工程。运行 abld build armi urel,则可生成在真机上运行的所需要的文件。

4) 运行命令 makesis 文件名.pkg,生成 SIS 文件。把 SIS 文件安装在真机上就可以运行。

3.2 移植过程中注意的问题

在移植的过程中最重要的就是内存不足的错误,如今台式机的物理内存最少有 256MB,映射到 20GB 或更大的虚拟内存,同时用户可能要频繁地重启计算机,内存不足的现象很少出现,所以用户可能会漠视内存和资源的管理。从 C++ 代码移植到 Symbian OS 手机,手机只有 4MB 的 RAM,一般情况下也不会超过

16MB,相对台式机,资源显得相当少,而且手机没有虚拟内存。

在移植的过程中,必须考虑一些关键问题^[2]:

1) 及时释放资源,因为系统不可能承受某个程序连续占用内存却不释放内存。特别是在 AVS-M 解码器中,涉及到了很多的空间分配,比如说有两个参考帧的数组,还有一个数组存放当前帧,占用了很大的空间,因此在解码结束的时候一定要释放掉。

2) 发生内存不足情况时,终止某些操作,决不能丢失任何数据,返回到一个可接受的一致状态。

3) 处理栈空间的不足。普通 Symbian 的栈空间只有 8k,因此在移植的时候要注意栈的分配,在 AVS-M 熵编码时,对量化系数采用自适应变长编码,所以在解码的时候只需要查询二维变长码表,这包括了 18 个变长码表,其中 7 个用于帧内亮度量化系数编码,7 个用于帧间亮度系数编码,4 个用于帧内帧间色度量化系数编码,把 18 个码表都定义为 const 类型,既节省了栈空间,也避免了每次调用 read_chroma_coeff_4x4_UVLC_V(把色度系数的变长码转换成实际的系数)等函数时重复定义数组和赋值,提高了速度。参考解码器在每次变长码解码时,都要构造 VLC_INTRA_dec, VLC_INTER_dec 两个三维数组,由于数组很大,每次都要给数组赋值,占用了很多时间和栈空间,考虑到这两个数组是固定的,所以把这两个数组变成常量,节省了很多时间。

4) 为了提高播放速度,使用了直接写屏的方法,通过 iOffScrnBmp->DataAddress()得到系统的显存地址,把解码后的数据转换成 RGB 后直接放入此显存地址即可,大大提高了图像显示速度。

4 效果分析

移植和优化后的 AVS-M 解码软件编译后,打包成可以在真机上运行的 SIS 文件,安装在 NOKIA 6670 (AMR9 123MHz 的 CPU,内存为 8MB,操作系统为 Symbian 7.0s,第 2.1 版 NOKIA 60 系列用户界面)上,实现了对运行程度不同的标准视频序列的正常解码显示,平均速度达到了 15 帧/秒。

5 结束语

Symbian 系统因其良好的性能,以及诺基亚等厂商的支持,已经成为智能手机的主流操作系统之一,并随着智能手机的推广占有了越来越大的市场,目前基于 Symbian 系统的应用开发也日益广泛,尤其伴随着手机视频等第五媒体的兴起,基于 Symbian 的视频业

(下转第 151 页)

了水印的不可见性。

图 4 和图 5 是水印抵抗 JPEG 压缩和抵抗高斯噪声的稳健性图。图 6 是从经过常见攻击后的水印后载体 Lena 图像中提取的水印图像,其中对水印后载体的攻击分别为:(a) 表示水印后载体不经过任何攻击,它的 PSNR 值为无穷大,所以它也是原始水印;(b)、(c)、(d) 表示对水印后载体进行 JPEG 压缩,压缩因子分别为 75、50、30;(e) 表示对水印后载体中值滤波,采用 3×3 滤波模板;(f) 表示对水印后载体加入高斯噪声,高斯噪声均值为 0、方差为 0.001;(g) 表示色彩调整,把原始色彩 $[0, 1]$ 调整到 $[0.1, 0.9]$ 上;(h) 表示剪切水印后载体 25%。

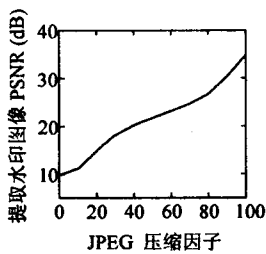


图 4 水印抗 JPEG 压缩稳健性

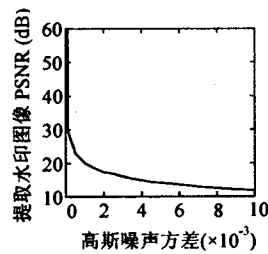


图 5 水印抗高斯噪声稳健性

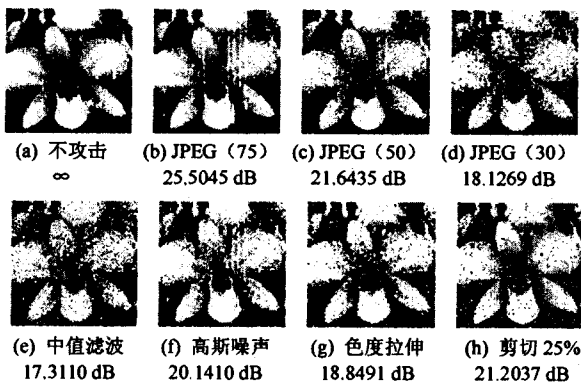


图 6 仿真攻击后提取水印效果图及其 PSNR 值
本算法在彩色载体图像中嵌入灰度图像作为水

印,没有采用常见的灰度水印图像位平面分解后归结为二值水印问题的方法^[5],而是利用混沌动力系统把灰度水印图像灰度值加密和拓展后嵌入载体图像中,显著提高了水印容量。与文献^[5]相比,文中算法嵌入水印容量和稳健性均优于前者。这得益于提出的水印跳变算法,它能有效利用载体的视觉冗余。

4 结论与展望

探讨了一种水印跳变数字水印算法。先把水印值拓展到若干水印子通道中,根据载体系数选择水印拓展值代替载体系数,赋予水印后载体系数多重意义。提取水印时对水印后载体系数解拓展得到水印信息。水印嵌入提取过程就像水印值跳动和变化的过程。提出利用水印正负拓展方法削弱邻道干扰的方法。所提水印方案对各种变换域均适用。算法能在彩色载体图像中嵌入灰度图像作为水印,仿真实验结果表明,利用该水印方案嵌入的水印不可见性好、鲁棒性强。

参考文献:

- [1] Cox I J, Kilian J, Leighton F T, et al. Secure Spread Spectrum Watermarking for Multimedia[J]. IEEE Transactions on Image Processing, 1997, 6(12): 1673 - 1687.
- [2] 谷利民, 方艳梅, 黄继武. 基于叠加嵌入的码分多址数字水印信道性能分析[J]. 计算机学报, 2005, 28(2): 269 - 273.
- [3] 谷利民, 黄继武. 数字水印信道中的分组码编码策略分析[J]. 软件学报, 2004, 15(9): 1413 - 1422.
- [4] Wang Pengfei, Qin Feng, Wang Xiaolin, et al. A novel blind watermarking scheme based on watermark hopping technique [C]//International Conference on Signal Processing ICSP 2006. Beijing: IEEE Press, 2006: 1257 - 1260.
- [5] Jiang Mingyan, Xu Guoping, Yuan Dongfeng. A novel blind watermarking algorithm based on multiband wavelet transform [C]//International Conference on Signal Processing, ICSP 2004. Beijing: IEEE Press, 2004: 857 - 860.

(上接第 135 页)

务将为移动运营提供稳定、有效的增值服务。

AVS-M 作为 AVS 系列标准的一部分,具有完整的自主知识产权,因其编码技术先进和专门针对手机等小屏幕设备选择适用的技术制定,因而有明显的针对性,由于自身存在的优势,AVS-M 必将得到广泛的应用,在 3G 的产业增值中占有一席之地。AVS-M 能够在 Symbian 上能够运行说明 AVS-M 参考软件具有良好的可移植性,而且优化后的软解码效果也能满足实际需求,进一步验证了 AVS-M 在手持设备上较为适用的。

参考文献:

- [1] 黄铁军. AVS 标准的背景·进展与产业应用展望[J]. 信息技术和标准化, 2003(9): 4 - 8.
- [2] Harrison R. Symbian os C++ 手机应用开发[M]. 北京: 人民邮电出版社, 2005.
- [3] 高文, 黄铁军. 信源编码标准 AVS 及其在数字电视中的应用[J]. 数字电视与数字视频, 2003(11): 4 - 6.
- [4] 易峰, 虞露. 帧内预测模式编码的改进[P]. 北京: AVS 工作组, AVS-M1456. 2004 - 06.
- [5] 国标 GB/T 20090.2 - 2006. 信息技术先进音视频编码第 2 部分: 视频[S]. 北京: AVS 工作组, 2006