

基于 Android 平台的手机视频优化编码^①

王作成, 摆玉龙, 李 昂

(西北师范大学 物理与电子工程学院, 兰州 730070)

摘 要: 视频编码技术是无线视频监控技术的关键. 提出了一种基于 Android 系统平台的视频压缩优化编码的设计. 介绍了 Android 系统及其上的应用开发, 阐述了传统 H.264 编码模式选择的思路, 重点描述了模式选择的优化方法. 利用 Android 平台结合 DSP 芯片实现 H.264 编码, 通过测试对比, 优化编码在保真的情况下, 一定程度上提高了算法的运算速度.

关键词: Android; H.264; 优化算法; DSP

Design and Implementation of Video Monitoring System Based on Android

WANG Zuo-Cheng, BAI Yu-Long, LI Ang

(College of Physics and Electronic Engineering, Northwest Normal University, Lanzhou 730070, China)

Abstract: Video coding technology is the key of wireless video monitoring technology. This paper presents a system that it based on the video compression Android platform optimization design of the code. It introduces the Android system and its application development. Moreover, this paper expounds the traditional H.264 coding modes selection thought, and the authors describe the modes selection optimization method. Use Android platform combined with DSP chip realize H.264 coding. Through contrast test, optimizing code in the fidelity and, to some extent, improve the speed of the algorithm.

Key words: Android; H.264; optimization algorithm; DSP

1 引言

随着流媒体和智能手机行业的不断发展, 基于无线网络和智能手机平台的无线监控将取代传统意义上的监控方式, 在众多领域得到广泛应用, 如银行、超市、远程教育领域以及各种工业场所等^[1].

由于 3G 网络所支持的带宽有限, 而视频监控要在有限的带宽下, 实时的传输清晰流畅的视频信号, 这就要求在监控的时候, 不仅仅要保证视频信号不失真, 而且还要进行最大可能的压缩. 为了能更有效的对视频信号进行压缩, 各种压缩算法及国际标准层出不穷. H.264 算法由于诸多优点脱颖而出, 它采用高精度、多模式预测技术, 同时差错消除工具保证了视频传输的正确性和有效性.^[2-4]但是, 由于智能手机内存一般偏低, 传统 H.264 算法复杂的运算严重影响到了视

频监控的实时性.

本文介绍了一种基于 Android 平台的视频压缩优化编码的设计, 利用手机自带摄像头以及 DSP 芯片, 通过快速选择的模式选择算法优化算法, 实现视频的快速压缩.

2 Android操作系统介绍

Android 是一种主要应用于手持设备的以 Linux 为基础的操作系统. 它是 Google 公司于 2007 年 11 月宣布的手机操作系统. Android 的 Java 应用更接近于底层系统, 效率更高, 而且成熟的虚拟机技术使得 Android 系统上的应用安全性更高. 与 Symbian、Wince、IOS 等厂商不同, 在 Android 操作系统上开发应用程序是免费的, 这也得到了众多开发人员的青睐.

^① 基金项目:国家自然科学基金(41061038)

收稿时间:2012-06-24;收到修改稿时间:2012-07-24

Android 系统平台由操作系统、中间件、用户界面和应用软件组成。Linux 内核层在硬件和软件之间, 提供系统底层服务。Android 包含一组核心库和 Dalvik 虚拟机, Dalvik 虚拟机执行 Dalvik 可执行文件.dex, 该格式文件针对手机运行环境的小内存特点进行了优化。该虚拟机是基于寄存器的, 所有的类经由 Java 编译器编译, 通过专门工具转化为.dex 格式由虚拟机执行^[5]。Android 开发者可以通过 Google 提供的 Android JNI 机制调用 C/C++开发的共享库, 实现 Java 和 C/C++联合编程。

3 传统模式选择算法简介

H.264 采用从 16×16 到 4×4 7 种块大小模式, 这样, H.264 相比 H.263 节省了大约 50% 的码率^[6]。

进行帧间预测编码时, H.264 分别对 7 种宏块模式进行遍历计算, 然后使用率失真优化模型计算当前宏块在各种模式下的率失真开销, 得到最佳模式, 从而提高了压缩效率, 基本原理如下:

在可选的宏块编码模式中, 遍历每种编码模式, 计算出在各种模式下的编码代价。式(1)为率失真优化模型公式:

$$D(A, C, M | Q, \lambda_M) = W(A, C, M | Q) + \lambda_M * O(A, C, M | Q) \quad (1)$$

λ_M 为拉格朗日系数, D 为失真开销, Q 为宏块的量化参数, A 和 C 表示原图像和重建图像的像素值, O 表示在特定的参数和模式下当前宏块的编码输出比特数, W 表示失真度, M 表示可选的编码模式。式(2)为图像失真度的计算公式:

$$W(A, C, M | Q) = \sum_{x=1, y=1}^{16, 16} (A_r[x, y] - C_r[x, y, M | Q])^2 + \sum_{x=1, y=1}^{8, 8} (A_l[x, y] - C_l[x, y, M | Q])^2 + \sum_{x=1, y=1}^{8, 8} (A_v[x, y] - C_v[x, y, M | Q])^2 \quad (2)$$

率失真开销包括整数 DCT 变换/量化、整数反量化以及熵编码等。H.264 的这种模式选择算法对一些出现几率很小的模式坚持进行率失真开销的计算, 这样就增大了计算量, 如果能节省无用的计算时间, 就可以大大降低计算量。

3 模式选择算法的优化设计

3.1 优化算法的描述

传统的 H.264 的模式选择算法运用了大量的遍历, 运算量过大, 适当的减少遍历的模式或者提前终止遍历就可以提高编码的速度^[7-8]。如果能预先知道当前编码模块可用的几种可能的编码模式, 就能够减少遍历范围, 再设置阈值中断遍历, 就可以实现减少无效遍历所带来的大计算量的目的。

图 1 为优化模式选择算法的流程图, 1 和 2 是当前模块所相邻的两个不同方向的模块, 阈值 I 可以利用不同模块的代价得出。当前宏块按照某一方向的宏块编码模式进行编码, 当编码代价小于 I 时, 即终止其他模式遍历。否则, 比较不同方向的编码代价, 得到最优编码。

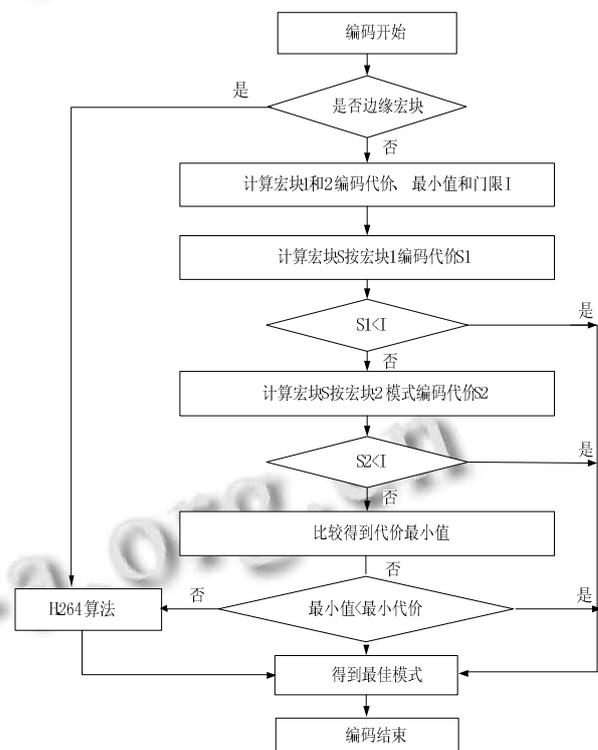


图 1 优化算法流程图

3.2 软件的设计

通过 Android 手机自带的摄像头采集图像信息, 系统截取视频每帧数据并传送至 DSP 芯片实现视频编码, 设计选用 Blackfin533 芯片进行编码。

Android 应用程序界面设计步骤: 建立 Android 项目, 编辑 Layout 目录中.xml 文件设计程序界面, 确定布局: 采用嵌套的 LinearLayout 和 RelativeLayout 进行布局, 调整高宽。在 ViewGroup 中添加 View 控件, 最

后完成整个 UI 设计.

在 DSP 芯片上进行 H.264 移植, 选择精简版的 h.264 编码器. micro_h264 是一种在 JM 编码模型的基础上更精简的软件模型, 它减少了 JM 中复杂的结构体, 进程更加清晰. 另一方面 Micro_h264 是一个标准 C 语言编写的编码器, 适合进行 DSP 移植. 移植的过程包括对代码的裁剪, 精简代码. 并在 Visual DSP++ 环境下编译通过, 能够实现正确的编码^[9,10].

对于 C 语言进行优化, 主要从以下几个方面入手: 指针优化数组的寻址, 可以提高效率和缩短计算时间; 运用位运算代替普通运算, 因为编译器的位运算效率高于通常的加减乘除运算; 合理运用判断和循环; 利用 Blackfin533 芯片的硬件循环器实现零开销的循环判断, 提高循环指令的执行速度.

Blackfin533 芯片提供了视频指令 SAA 计算绝对差值求和函数 SAD(). 部分 SAA 的指令使用如下所示:

```
SAD+=ABS(*PSRC++ - *PREF++);
LSETUP(SAD_START,SAD_END) LC0=P1;
SAD_START:
R3 = [T1++];
SAA(R1:0,R3:2) || R1= [T0++] || R2= [T1++];
SAA(R1:0,R3:2)(R) || R0= [T0++] || R3= [T1++];
SAA(R1:0,R3:2) || R1= [T0++] || R2= [T1++];
```

4 实现结果

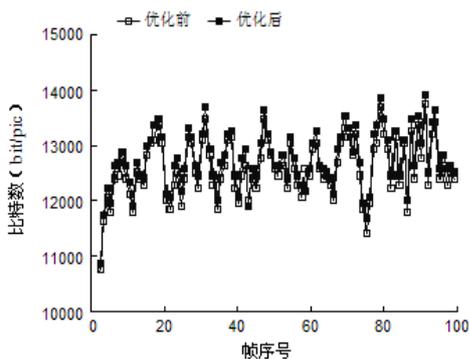


图 2 优化前后的每帧编码比特数示意图

图 2 为优化前后每帧编码比特数的对比, 优化后的比特数要高于优化前, 表明优化后的算法降低了压缩比. 图 3 为优化前后的峰值信噪比曲线的对比, 两条曲线基本吻合, 说明算法在优化前后的效果基本相同. 由此证明算法在优化后, 在保真的情况下, 提高

了算法的运算速度.

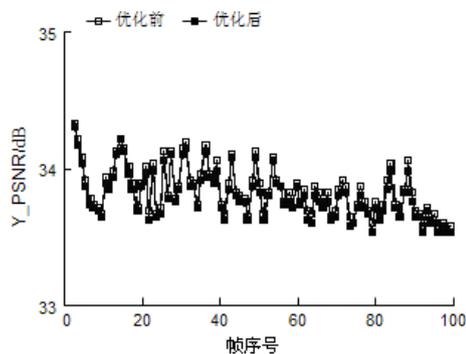


图 3 优化前后的峰值信噪比曲线

5 结语

利用优化模式选择的 H.264 算法和 DSP 编码移植实现了 Android 手机端的视频压缩编码. 优化后的算法经过实验室测试, 平均每帧的处理速度得到了一定幅度的提高. 进一步的测量失真度实验证明, 同样的图像序列在优化前后的效果明显, 在不失真的情况下, 完成了运算速度的提升.

随着视频监控技术的发展, 该优化编码技术在有限带宽情况下可以更好的应用到实时监控的编码工作中.

参考文献

- 曹晓芳,王超,李杰.一种基于 Android 智能手机的远程视频监控的设计.电子器件,2011,34(6):709-712.
- 张淑芳,李华.基于 H.264 的多参考帧快速选择算法.电子学报,2009,37(1):62-66.
- 张阳,崔慧娟,唐昆.H.264 视频编码中的快速失真与速率估计算法.清华大学学报(自然科学版),2009,49(7):977-980.
- 王欣欣,路中成,程欣欣.H.264/AVC 标准中帧内预测宏块模式选择的优化.计算机工程与应用, 2011,47(16):182-184.
- 杨明极,毕晶.基于 Android 视频客户端的设计.电视技术, 2012,36(3):43-47.
- 蒋志平,艾云峰,侯向宁.H.264 视频最佳宏块模式快速选择.计算机工程与应用,2011,47(9):171-174.
- 顾梅花,余宁梅,寇立康,等.H.264 快速模式选择算法中的提前终止策略.中国图像图形学报,2011,16(3):305-309.
- 熊江.H.264 中零块判决阈值的研究.西南大学学报(自然科学版),2009,31(5):172-177.
- 叶志辉,赵建明.一种基于 DSP 的 H.264 实时视频编码器软件架构.自动化与仪表,2008(2):43-46.
- 闫健恩,许海燕,林建秋.基于 ADSP-Blackfin533 的 H.264 视频编码器的实现.微计算机信息, 2006(5):27-29.