

移动 Flash 播放器的设计与实现^①

Design and Implementation of Flash Player for Mobile Device

沈 静 何必仕 周丽 (杭州电子科技大学 计算机学院 浙江 杭州 310018)

摘 要: 针对国内外移动 Flash 播放器产品存在的问题和不足, 设计与实现了一款针对手持移动设备的 Flash 播放器。介绍了移动 Flash 播放器的关键技术、总体设计以及具体实现, 最后将移动 Flash 播放器移植到 Linux 智能手机进行测试分析, 实验结果数据证明了该软件能广泛应用于中低端智能手机。

关键词: Flash 播放器 手持移动设备 嵌入式软件 矢量图 渲染引擎 解码器

1 引言

移动 Flash 就是利用矢量图形技术进行制作的多媒体动画, 通过互联网提供下载、播放、转发。移动 Flash 主要用于开发移动动漫、用户界面等应用。此前, 移动动漫业务一直只限于以 SMS 为基础的文字和静态图像服务以及简单的 WAP 应用, 并没有实现真正意义上的多媒体动画业务。移动 Flash 的出现使在手机上实现真正的多媒体动画服务成为可能, 增强了用户对无线互联网的使用体验。2006 年以来我国移动运营商们先后启动了基于移动 Flash 的手机动漫业务, 并推出了自由享受多媒体动画、游戏、电影等各种娱乐项目体验, 手持移动设备嵌入式 Flash 播放器已经逐渐成为智能手机、PDA、MP4 等智能化终端系统的标准配置。

目前, 国内外出现了多款针对手持移动设备的嵌入式 Flash 播放产品, 国外以 Macromedia Flash Lite 为代表, 国内以广州飞思的基于 KJava 的 MOLASH 为代表。综观目前国内外移动 Flash 播放产品, Macromedia Flash Lite 已经成功运用到 Nokia, 索爱等手机上并得到多家知名厂商的支持, 但是 Flash Lite 对手机硬件要求高无法应用于中低端手机, 同时对于国内厂商来说没有自主知识产权、受 ADOBE 约束条件多、授权费高等因素影响了 Flash Lite 在国产手机中预装; 广州飞思软件 MOLASH 播放器, 虽然终端不用安装播放器, 但是需要将 Flash 先反编译为

KJava 程序, 因而 Flash 应用受到一定的限制。因此, 自主知识产权的移动 Flash 播放器的研究与开发是非常必要和迫切的。

2 关键技术

实现移动 Flash 播放器关键技术就是渲染引擎中的矢量图绘制。目前面向手持移动设备矢量图绘制存在以下几个问题:

(1) 矢量图绘制占用系统资源较多, 由于手持移动设备的 CPU 主频和内存都是有限的, 就会导致 Flash 文件不能流畅播放。

(2) 浮点运算。由于大部分嵌入式设备都没有除法和浮点运算部件, 浮点运算和除法都需要通过软件实现, 导致渲染引擎效率不高。

(3) 由于手持移动设备的屏幕大小和分辨率有限, PC 上的矢量图形库不适合应用到手持移动设备中。

针对这些问题, 我们在实现中提出了针对移动设备的策略和改进算法。

(1) 通过矢量图绘制流程分析, 对矢量图中路径光栅化和曲线分解算法进行了优化, 采用了基于活性边表的光栅化技术和基于曲率的曲线非均匀分解算法, 从而减少了曲线分解计算量, 节约的路径光栅化时间。

(2) 采用自动浮点转定点阈值预测算法, 同时在编码过程中将除法和浮点运算转化为移位运算, 从而降低了播放器对硬件资源的要求。

^① 基金项目: 国家自然科学基金项目(60873023)

收稿时间: 2008-07-16

对矢量图的反走样算法进行改进，采用了斜向扫描编码查表法实现反走样渲染，使得 2D 矢量图形库中反走样算法更适用于嵌入式系统。

3 总体设计

移动 Flash 播放器突破了嵌入式系统的限制，将精美、交互的 Flash 动漫放入到手机、PDA 等手持移动设备。播放器具有体积小、速度快、稳定性高、需要内存小、支持 Flash 新版本、跨平台、目标设备灵活应用等特点。

移动 Flash 播放器总体上有三大核心模块组成：

- (1)SWF 解码器：解析符合 MicroMedia 公司的 Flash 文件格式标准的 SWF 文件。
- (2)渲染引擎：路径绘制、字体和图形图像以及纹理等的渲染。
- (3)声音解析和播放：解析各种格式的声音文件,实现声音同步。

围绕着 SWF 文件解码器、渲染引擎、声音解析播放等主要研究内容提出了如图 1 所示系统总体设计架构。

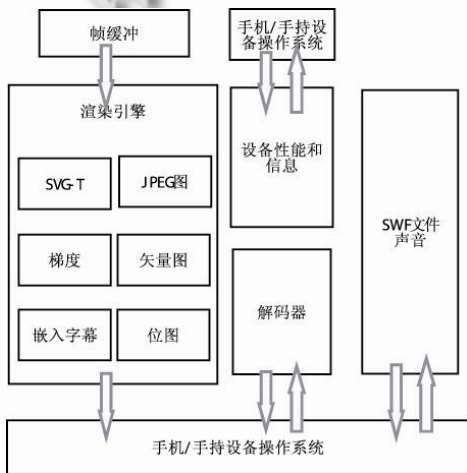


图 1 系统体系结构

4 播放器模块实现

4.1 SWF 文件解码器

Flash 文件由文件头 (Header) 和文件体 (Body) 组成。文件头定义了 Flash 文件的格式、版本、文件长度、帧大小、帧播放速率和帧总数。表 1 给出了文件头的完整格式。

文件体由一个个标签 (Tag) 组成，在文件体最后有一个结束标签 (End Tag)。

图 2 给出了 Flash 文件格式的完整结构。

标签可以分为两类：定义标签 (Definition Tags)

表 1 文件头格式

名称	类型	说明
标志位	UI8	“F” 压缩, “C” 非压缩
标志位	UI8	通常是 “W”
标志位	UI8	通常是 “S”
版本	UI8	Flash 格式版本
文件长度	UI32	文件长度 (bytes)
屏幕大小	RECT	播放时屏幕大小, 具体结构和长度根据数据变化
帧频率	UI6	帧播放频率, 通常位 12
帧数	UI6	文件包含帧的总数

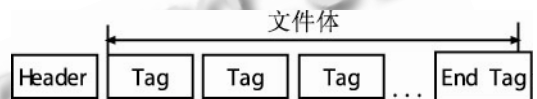


图 2 Flash 文件格式的完整结构

和控制标签(Control Tags)。定义标签主要是用来定义 SWF 电影的内容,包括形状(Shapes)、文本(Text)、图(BitMaps)声音等。每个标签都有唯一的一个 ID 来对应它自己的字典(Dictionary)。控制标签,控制标签主要是用来创建和操作字典中 characters 实例的渲染以及控制 Flash 播放流程。

每个标签都包括一个标签头(Tag Header)和一个标签体(Tag Body)。标签头包括两个部分：标签类型和标签长度，通过解析可以立即知道标签的类型和长度，如果无法识别这个标签类型，我们可以根据标签的长度直接跳过这个标签，从而保证了版本的兼容性，即使出现新版本格式的 Flash 文件播放器还是能够解析整个文件进行播放。

解码器主要有以下几个功能：

(1)提供一个 Zip 解压缩包，能够对数据进行正确的解压缩。

(2)正确分析 Flash 文件中的标签并根据标签的类型和数据生成响应的解码标签。

(3)维护一个全局的解码标签链表，用于保存帧内所有解码标签。

整个 SWF 文件解码流程如图 3。

解码器实现分为以下几个步骤：

(1)读取文件流。

(2)获得 Flash 文件头信息，如果是 “CWS” ，则用 Zlib 包进行解压缩，如果是 “FWS” 则直接跳到标签解码部分。

(3)根据不同的标签类型进行解码并构建帧链表为帧的渲染做准备。

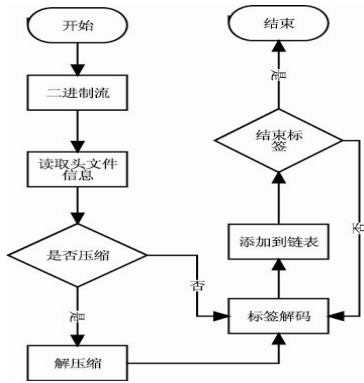


图 3 解码流程

4.2 渲染引擎实现

Flash 图像的组成主要通过 Path、Bitmap、JPEG、Font、Text、Shape 等标签组成。矢量图的生成过程就是一个数据解码的过程，也是 Flash 动画播放时所需花时间最多的部分，因此渲染引擎性能很大部分就决定了整个播放器的性能。本文主要从路径绘制，Bitmap 和 JPEG 图片处理，字体和文本处理三个方面来阐述渲染引擎的实现。

4.2.1 路径绘制

矢量图都是以路径的形式来表现的，矢量图通过一条或多条路径来描述几何物体，路径是矢量图的核心，体现了矢量图的特征。因此路径的绘制是实现矢量图渲染的关键。从功能管线的定义中，我们可以归纳出路径绘制的流程，如图 1 所示。路径的染色方式是与路径联系在一起的，首先要变换到用户坐标系，将路径和染色方式绑定到一起，再把用户坐标系转化到屏幕坐标系。其中如果路径的绘制方式为“描绘”路径的话，在进行坐标变换之前要先根据设置生成路径的轮廓，然后将生成的形状作为填充路径来处理。

经过光栅化后透明度信息就确定下来了，全屏的透明度信息经过剪切和蒙板处理之后用颜色信息进行渲染，最后与屏幕上已有的内容进行叠加和混合处理。如果用户定义了图像的话，还要在混合之前先根据图像绘制模式插入图像数据，最后经过混合处理之后输出到帧缓冲。

4.2.2 Bitmap 和 JPEG 图片处理

Flash 文件中位图主要有两种图片格式：Bitmap

和 JPEG。对 Bitmap 图片处理过程如下：

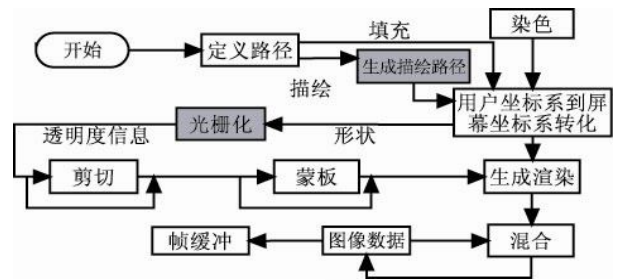


图 4 路径绘制流程

(1)首先根据标签类型判断数据格式是 DefineBitsLossless 还是 DefineBiteLossless2。

(2)分别读取 Bitmap Format、Bitmap Width、Bitmap Height 和 Bitmap Color Table Size 数据，再对 ZlibBitmapData 数据进行 ZLIB 解压缩，还原出数据。

(3)将数据装入到数据字典中去。

(4)遇到控制标签时从数据字典中取出数据，并加入到帧图像中。

对 JPEG 图片处理原方法和 Bitmap 图片处理方法一样。

4.2.3 字体和文本处理

Flash 支持两种文本：glyph text 和 device text。Glyph text 与设备无关，将自己的显示特征嵌入到 Flash 文件中去，而 device text 与设备相关，其显示要依靠设备。

对字体的处理需要借助 Shape 处理来实现，首先读取字体标签的 ID，然后读取 shape 所需要的元素并将这些元素存入到字典中去，然后调用 shape 的方法将其组合起来并存入到字典的 Font 结构中去。

对于文本的处理主要对 TextRecord 的处理，对 TextRecord，首先读取前 8 个字节判断其字体的属性，根据 StyleFlagsHasFont 判断 Text 采用的 Font，然后通过 ID 号在字典中查找，找到其属性后存入 DisplayList 中。根据 StyleFlagsHasColor 判断文本颜色，由于 Text 是通过 Shape 的方法绘制的，因此颜色设置调用设置 RGBA 函数对文字进行着色。

4.3 声音解析和播放

SWF 文件提供了一个小而高效声音模型。SWF 文件声音格式主要有两种：(1)事件声音：事件声音播放主要是针对一些事件比如点击鼠标或是 Flash 播放到

特定帧。事件声音在播放之前一定要下载到本地同时可以进行多次的播放。(2)流格式声音：流格式声音可以随播放时间轴进行同步下载和播放，因此声音文件包需要被存储在每个帧里面。目前最新版本的 Flash 文件可以支持以下三种声音格式：ADPCM，MP3，Nellymoser。

对于 Event 声音，由于声音数据较小，不存在跨帧播放问题，因此处理比较简单只需要读出数据调用响应的解码程序然后写入到声音设备中就可以了。对于流格式声音处理首先根据不同的声音格式进行解析，然后把解析到的数据写入到声音设备中去。流格式声音处理的难点是在保持声音和图像同步，当声音和图像不同步时，我们采用跳帧的方法来保持声音和图像一致。

5 性能分析

移动 Flash 播放器已经成功移植到夏新 E600 手机上并能流畅地播放 Flash 动画。播放器效果如图 5。表 2 给出了移动 Flash 播放器的技术参数，从播放器对 CPU、内存、屏幕等要求可以得出移动 Flash 播放器能很好地应用到中低端手机中实现 Flash 播放。



图 5 夏新 E600 播放效果

6 结束语

通过对移动 Flash 播放器的实现让我们对移动

表 2 技术参数

项目	参数
CPU	ARM7 30Mhz 以上
内存	350K
内存分配	动态内存调用
帧率	12 帧/秒
色彩	4,096 色(12 bit), 最佳效果 65000 色 (16bit) 以上
LCD 尺寸	大于或等于 128 X 128 像素
Flash 版本支持	Flash7.0
存储空间	大于 1M
设备音效	MIDI/AMR
声音格式	ADPCM, MP3

Flash 的文件结构、文件解析以及渲染引擎实现过程有了更深入的认识。随着 3G 时代的到来，移动 Flash 市场前景将非常值得期待，移动 Flash 技术研究也将成为一个非常重要的研究方向。在下一步工作中，我们将进一步完善解码器使其支持最新版本的 Flash 文件，同时优化渲染引擎图形库算法，使其更适合于手持移动设备应用需求。

参考文献

- 1 Adobe Macromedia Flash(SWF)File Format Specification.2005-10.<http://www.macromedia.com/support/documentation/doctypes.html>.
- 2 胡海波.移动 Flash 技术的发展分析.通信世界,2006,A(38):25-27.
- 3 李善平,刘文峰,王焕龙.Linux 与嵌入式系统.北京:清华大学出版社,2006.
- 4 Pulli K. New APIs for Mobile graphics. Nokia Research Center and MIT CSAIL. SPIE-IS & T/Vol.607401-2.
- 5 Worth C. An Insider's Guide to Cairo. 2005-07.http://cworth.org/~cworth/papers/cairo_ddc200.
- 6 Lee SY, Choi BU. Vector Graphic Reference Implementation for Embedded System. Technologies for E-Learning and Digital Entertainment 2000.