

一种嵌入式 SWF 解码器的设计与实现

田华健^{1,2}, 熊庆国^{1,2}

(1. 冶金自动化与检测技术教育部工程研究中心, 湖北 武汉 430081;

2. 武汉科技大学 信息科学与工程学院, 湖北 武汉 430081)

摘要:为了满足嵌入式多媒体平台的发展需要,设计开发了一种嵌入式 Flash 播放器,该播放器也是一个标准 SWF 格式的解码器。SWF 是一种成熟的技术,具有各方面性能的优越性,标准的开放性和稳定性,很适宜应用于各种多媒体环境,并且由于其优异的性能在各行各业得到广泛的应用。介绍了 SWF 格式,分析了播放器的系统架构,并详细介绍了核心模块和采用的一些关键技术。访播放器已在机顶盒 LSI 和 ST 平台实际运行,取得了令人满意的效果。

关键词:SWF 技术;SWF 解码器;核心模块;关键技术;机顶盒

中图分类号:TP391.4

文献标识码:A

文章编号:1673-629X(2009)05-0198-04

Design and Implementation of an Embedded SWF Decoder

TIAN Hua-jian^{1,2}, XIONG Qing-guo^{1,2}

(1. Ministry of Education Eng. Research Center of Metallurgical Automation and Measurement
Technology, Wuhan 430081, China;

2. School of Information Science and Technology,
Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract: In order to meet the developing need of the embedded multimedia platform, an embedded Flash player which also is a standard SWF format decoder, is designed and developed. SWF as a mature technology of many performance advantages, with the standard opening and stabilization, is suitable for the application in the multimedia circumstance and also widely applied in many fields because of its ascendant performance. This paper firstly introduces the SWF format, then analyzes the system frame and describes the key models and some technologies adopted in detail. The Flash player has run very well on the ST and LSI platforms of the STB and has shown the satisfying effect.

Key words: SWF technology; SWF decoder; kernel modules; key technology; STB

0 引言

近几年,消费类电子产品发展成为世界潮流^[1],作为消费类电子产品核心的嵌入式多媒体平台获得了巨大的发展空间,如 Web 可视电话、机顶盒、PDA、多媒体手机、导航设备等。各消费电子厂商纷纷以产品能支持丰富多彩的音视频播放,以及动漫和游戏作为卖点和产品发展方向。SWF 是互联网上流行的多媒体矢量交互格式,通常也称为 Flash。该技术以其高效的压缩技术、优秀的播放质量、网络传输要求低、很好的支持交互性等诸多优点在动漫和互联网方面取得巨大成功,成为目前计算机互联网领域最流行、最先进的流

式多媒体矢量交互技术。SWF 技术开放成熟,可用资源相当丰富,很适宜集成到嵌入式多媒体平台上,实现用户界面和操作方式的巨大进化,丰富娱乐节目。从 2003 年开始,各手机生产厂商和机顶盒厂商纷纷推出支持 Flash 播放的产品。但由于 SWF 解码器的设计相当复杂,综观目前国内外的嵌入式 SWF 解码器产品,大都不成熟,在移植性和功能实现等方面存在缺陷,针对以上问题,提出了一种嵌入式 SWF 解码器的设计与实现。

1 SWF 解码器的设计

SWF 文件格式由文件头和许多个 Tag(标记)组成,最后以一个特殊的 End Tag 结束文件,其基本结构如图 1^[2]所示。

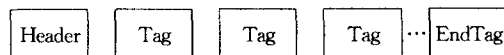


图 1 SWF 文件结构

收稿日期:2008-08-05

基金项目:科技部国家技术创新基金(05C26224201053)

作者简介:田华健(1980-),男,硕士研究生,研究方向为嵌入式系统的应用;熊庆国,教授,主要研究方向为嵌入式系统的应用、计算机控制技术。

SWF 文件头由文件标识(又称文件签名)及文件基本信息组成,文件头后是一系列 Tag 数据块,这些数据块形式一致而独立,一个数据块中的指针只能偏移指向本数据块,而不能偏移到另一个标记中。当解码器解析时如果发现有标记无法解析,可以跳过,不会影响到其他数据的解析呈现。

每个标记数据块都是以 Tag ID 和数据块长度开始的,Tag ID 在整个 SWF 的标记定义中是唯一的。依据其长度,Tag 数据块有短数据块(62 个字节以下)和长数据块(63 个字节以上,最大 4G 字节)两种。

根据 SWF 文件的格式,SWF 解码器的设计采用边解析边解码的方式。

1.1 模块结构设计

整个解码器的程序设计采用模块化的思想,其功能模块结构图如图 2 所示。

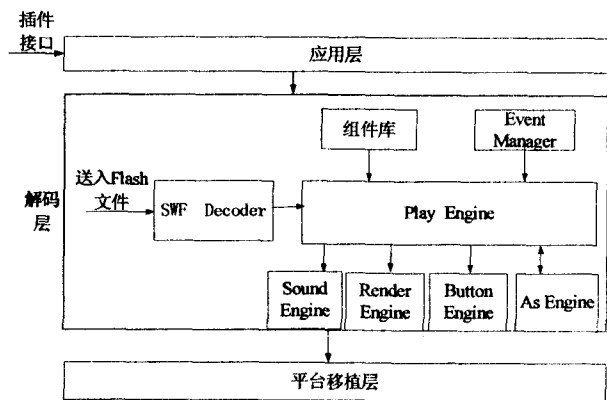


图 2 模块结构图

图 2 中各模块功能为:

(1) 插件接口:该接口被嵌入式浏览器所采用。
(2) 应用层:用来控制播放器的初始化和控制播放流程,如实现播放、停止,实现画面尺寸的变化等功能。

(3) 解码层:

* SWF Decoder:SWF 标签解析模块。主要负责 Flash 文件的解析。

* 组件库:主要包括字库、JPEG 解码库、ZLIB 解码库、MP3 解码库等信息。

* Event Manager:消息管理模块。用于接收或处理用户消息(遥控器按键消息等)和系统消息(定时器消息)。

* Play Engine:播放执行模块。

* Sound Engine:声音处理模块。SWF 支持 AD-PCM、MP3 等音频格式。该模块将从解析模块得到的编码后的音频数据,解码成 PCM 音频格式,调用底层音频播放接口,进行声音播放。

* Render Engine:渲染引擎。负责画面、字体等

的渲染、填充。

* Button Engine:实现 Button 的解析、排序,并实现 Button 的交互。

* As Engine:解析并执行 ActionScript 脚本。

(4) 平台移植层:封装了定时器操作、信号量、线程等与平台相关的操作,便于移植。

1.2 核心模块的实现

1.2.1 标签解析模块

SWF 的文件结构如图 1 所示。其文件由文件头和文件体组成。文件头定义了文件的大小、文件播放时画面的显示尺寸、帧的个数、帧速、是否被压缩等信息。文件体则是由一系列的标签组成。整个解析框架的实现包括:

(1) SWF 文件头解析,根据文件头信息配置动画播放环境;

(2) 正确解析每个 Tag,建立标签字典;

(3) 根据控制标签建立显示表列,渲染帧画面,执行动作;

(4) 正确结束解析过程,回收资源。

1.2.2 播放执行模块

SWF 中的标签分为两种:定义标签和控制标签。

定义标签定义对象的内容,如形状、声音等。SWF 在解析时会把所有的定义标签放到一个称为字典的元素列表里,并被赋予唯一的 ID,以供控制标签调用。

控制标签则对定义标签进行操作。由控制型标签 PlaceObject 选择需要显示的标签,这些标签从字典中取出放置到显示列表中去,它们被称为层,而显示列表是将在下一帧显示的元素列表集,当 CurFrame^[2] 标签到来时,则对显示列表里的每个元素解码。不同元素的解码则调用相应的解码模块。

SWF 中的定义标签、字典与显示列表关系如图 3 所示^[2]。

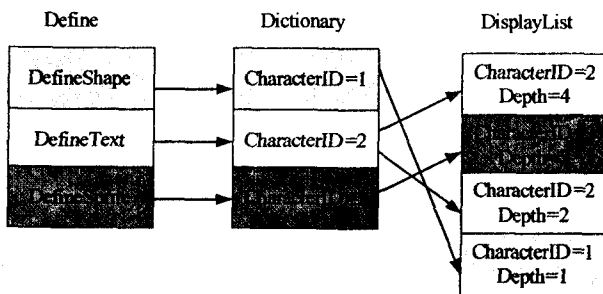


图 3 SWF 中的定义、字典与显示列表关系

1.2.3 AS 解析引擎

ActionScript 是 Flash 内置的脚本编程语言,用它为动画编程,可以实现各种动画特效、对影片的良好控制、强大的人机交互以及与网络服务器的交互功能。

虽然 ActionScript 给 Flash 文档带来很多好处,但如何正确解析并执行 AS 给解码器的设计带来了很大的难度。因为 Flash 在形成文档时,AS 脚本已被编译成一系列的字节码,这些字节码以一系列的形式的形式保存在文档里。所以在运行脚本时首先要将字节码反编译成脚本,实现类似于 JVM 的功能。事实上,Adobe 的 Flash 播放器就是内置一个 AVM^[2](actionscript virtual machine)来解析并运行 actionscript。

该模块的程序流程图如图 4 所示。

(1) 从 Flash 文档中解析动作脚本。

(2) 判断该动作是否是对对象进行操作。如果不是,则直接实现该动作的解码;如果是,则创建对象,进行 AS 链接,实现 AS 规范的内置函数^[3]以及对对象跟 ActionScript 执行的绑定。AS 链接提供两种绑定方式:一种为给定 AS 类名跟 AS 类对应实例的绑定;而另一种为给定类属性跟 AS 类对应属性的绑定。

(3) AS 执行 (AS 实现引擎):负责 AS 规范内建对象以及函数的实例化,并对动作进行解码。

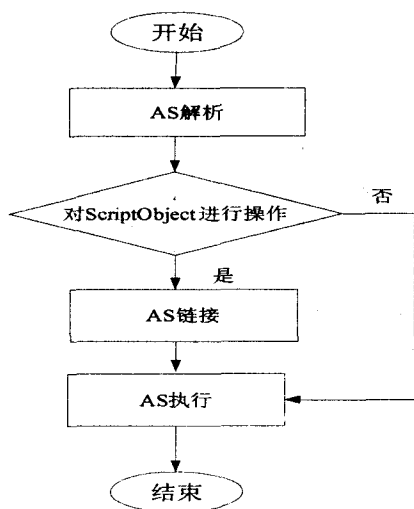


图 4 AS 解析引擎

1.2.4 渲染引擎

该模块主要负责 Shape 图形的解析、解码,Button 图形的解码,并把解码后的位图图形填充到帧缓冲区中。该模块功能模块图如图 5 所示,具体由以下子模块实现。

(1) Shape 解析模块。该模块从文档中解析出图形的描述信息,如图形的填充方式,图形的位置、大小等,并解析出包围每个图形的矢量线段,把这些线段的信息保存在一个链表里,活性边表填充时会用到该链表。

(2) Button 解析模块。Button 解析时会得到该按钮的图形信息和触发的动作。在渲染引擎里要把该按钮的图形画出来,AS 引擎会对其动作进行解码。

(3) Render Engine。该模块会对图形信息进行解码。图形分为色块和位图。如为色块图形,则调用 Raster Engine 对其进行光栅化处理,得到位图信息。如为位图图形,则判断其压缩方式,为 JPEG 位图时则调用 JPEG 解码库解码;如为 PNG 位图则调用 ZLIB 解码库解码。图形解码后通过矩阵的逆运算把图形的矢量坐标转换到显示区域^[4]。

(4) 位图填充模块。将解码后的位图按像素放到帧缓冲区里,等待拷贝。

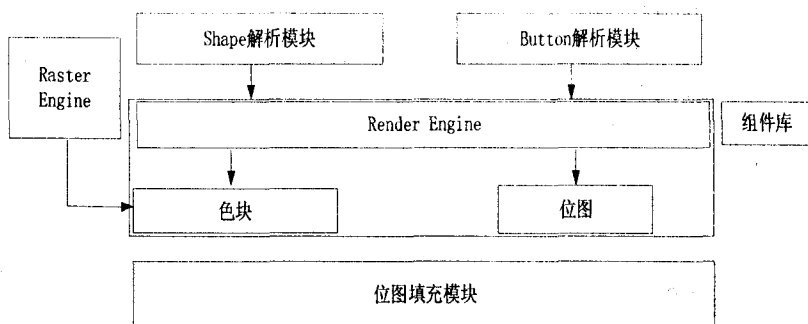


图 5 渲染引擎模块

2 解码器采用的关键技术

2.1 帧间复用技术

就像一般运动图像一样,Flash 动画的帧间动作也是连续的,这样,在上一帧的基础上只需要更新帧缓存部分区域,就可以得到新的一帧图像,大大减少了硬件资源的消耗。特别是在嵌入式平台上解码 SWF 文件,无论怎么优化解码过程、图形图像算法,渲染一帧画面仍然需要大量的计算。减少需要渲染的部分对降低硬件消耗效果更明显,所以一个高效的帧间复用方案是必须的。

该帧间复用方案基于 SWF 文件这样一个特点:需要渲染的标签都定义有一个能完全包围该对象的矩形,且该矩形的长宽是符合条件的矩形的长宽的下界。当该标签被加入显示表中时,矩形乘以 PlaceObject 标签所带的矩阵就得到该标签的矢量图形在屏幕上显示的位置。这样就知道了一帧中所有对象的显示范围。前后两帧有变动的标签由 PlaceObject 来设定,这样,就能确定哪些对象发生了改变。

根据以上分析,该帧间复用具体算法如下:

(1) 判断当前帧的 PlaceObject 标签设定的层的显示区域是否和文件头定义的电影区域相交。如果不相交,则忽略该层对象,不对其解码;否则,将相交区域加

入数组 SRECT devDirtyRect[] 中。这里有两种情况:

(a) 该层在前一帧出现过, 当前帧中它移动了位置, 这时需要将其所占的新旧两个区域加入数组中;

(b) 该层在前一帧中没有, 则需要将其在当前帧中所占区域加入数组;

(2) 将 devDirtyRect 数组中的矩形划分为互不相交的矩形, 要求划分后单个矩形尽可能的大, 这样 devDirtyRect 中各矩形互不脚叠;

(3) 遍历显示表中所有的层, 如果层与 devDirtyRect 中的矩形相交, 则记录相交区域, 并重新渲染层在相交矩形中的部分。

因为通常需要更新的部分很小, 通过采用这种帧间复用方案可以极大地提高解码效率。

2.2 填充算法的优化

图形的填充速度对 Flash 一帧的图形渲染速度有着决定性影响。在嵌入式平台, 计算量大的填充算法会占用大量的 CPU 资源, 严重影响 Flash 的播放速度和可视效果。本系统利用了经典的有序边表填充^[5,6]算法的思想, 结合 SWF 格式的特点, 设计了一种适合嵌入式平台的填充算法, 使解码器能够很流畅地播放 Flash。

该填充算法的算法如下:

(1) 解码器在对每个图形进行解码时会得到描述矢量边在显示区域的位置等信息。把位置信息和扫描线对应起来建立边表, 此边表把与该扫描线相交的所有矢量边链接起来。

(2) 遍历显示列表中的每个对象, 若扫描线与该对象相交, 则对于每条扫描线建立边表。

(3) 遍历每条扫描线, 把与该扫描线相交的每个边表链接起来, 形成一个活性边表。此链表包含与该扫描线相交的一帧 Flash 图像所有对象的边线段的斜率、近似斜率、与扫描线交点的横座标以其该边的纵座标的最大值。

(4) 对所有的活性边表按其中每个边与扫描线交点的大小排序。

(5) 填充每条活性边表。

2.3 通过矩阵运算来实现画面无失真的放大与缩小

画面中每个对象位置的移动及形状的改变是通过矩阵运算实现的。

如果一个点的原来的坐标为 (x, y) , 变换后的坐标为 (x', y') , 则计算公式如下^[2]:

$$x' = x * ScaleX + y * RotateSkew1 + TranslateX$$

$$y' = x * RotateSkew0 + y * ScaleY + TranslateY$$

每个 Flash 的显示区域在 Flash 文档中都有定义,

这样每个定义标签都会按照 Flash 文档中的定义在固定的位置和显示区域改变其位置和形状。通过用算法改变矩阵参数的值, 则可以改变显示对象最终的位置和形状。又因为 Flash 图形的变化是通过解码图形数组元素来实现的。解码 Style change record^[2] 时可以得到图形的位置、图形的填充方式等信息。一个图形可能由一系列的 Straight edge record^[2] 或 Curved edge record^[2] 所包围, 通过解码这两种图形数组元素, 可以得到包围图形数组的边的信息: 如是直线, 可以得到线段的起始点和终点相对于图形位置的偏移; 如果是曲线, 则可以得到曲线的控制点和锚点相对于图形位置的偏移。正是这些信息决定了图形的位置和形状。通过矩阵运算则可以改变这些信息, 又因为图形填充的信息是对图形的描述, 而不是像素, 所以图形的形状改变时图形不会失真。

3 结束语

在引入数字化技术后^[7], 多元化的数据广播服务逐渐发展起来, 数字电视终端产品——机顶盒也向着综合多媒体应用平台的方向发展。利用目前互联网中最成熟流行的 SWF 格式和 Flash 技术, 设计实现了数字电视的终端载体——机顶盒上的 SWF 解码器。文中在详细分析系统架构和核心模块的基础上, 又介绍了该解码器的一些核心技术。该解码器具有体积小、速度快、跨平台和稳定性高等优点, 已成功移植到机顶盒 LSI 和 ST 平台^[8], 取得令人满意的效果。

参考文献:

- [1] 赵海. 嵌入式 Internet——21 世纪的一场信息技术革命 [M]. 北京: 清华大学出版社, 2001: 91-96.
- [2] Macromedia Flash(SWF) File Format Specification Version 7 [EB/OL]. 2005-10. <http://www.macromedia.com/support/documentation/docTypes.html>.
- [3] SpiderMonkey (JavaScript C) Engine [EB/OL]. 2005-12. <http://www.mozilla.org/js/spidermonkey/>.
- [4] 郭春煊, 毛志刚, 谢懋. 矩阵求逆运算的 VLSI 实现 [J]. 计算机技术与发展, 2008, 18(5): 219-223.
- [5] 杨钦, 徐永安, 翟红英. 计算机图形学 [M]. 北京: 清华大学出版社, 2005: 254-260.
- [6] 郭文平, 龙帮强. 扫描线种子填充算法的改进 [J]. 天津工业大学学报, 2008, 18(4): 48-51.
- [7] 王艳荣. 数字电视机顶盒技术综述 [J]. 广播电视信息, 2003(4): 43-46.
- [8] 王秀丽, 麦继平. $\mu C/OS-II$ 操作系统在 ARM 处理器上的移植 [J]. 计算机技术与发展, 2007, 17(1): 213-215.