

# 基于 Android 系统的 FFmpeg 多媒体 同步传输算法研究

胡 成,任平安,李文莉

(陕西师范大学 计算机科学学院,陕西 西安 710062)

**摘 要:**FFmpeg 是一个开源跨平台多媒体数据解决方案,常被移植到各种嵌入式系统中。将 FFmpeg 移植到 Android 系统中,能够增加 Android 系统对编解码格式标准的支持,但由于目前手机处理能力低,内存小等硬件配置因素,严重影响 FFmpeg 对音视频流的解码效率,导致解码出的音视频数据无法同步。通过研究基于时间戳的多媒体音视频同步算法模型,将其引入到 FFmpeg 中,并在 Android 平台进行算法实验。实验证明,基于时间戳多媒体音视频同步算法模型能够有效地保证多媒体数据的同步。

**关键词:**FFmpeg;多媒体;时间戳;音视频同步;Android

**中图分类号:**TP301.6

**文献标识码:**A

**文献编号:**1673-629X(2011)10-0085-03

## FFmpeg Multimedia System Based on Android Synchronous Transmission Algorithm

HU Cheng, REN Ping-an, LI Wen-li

(School of Computer & Science, Shaanxi Normal University, Xi'an 710062, China)

**Abstract:**FFmpeg is an open source multi-media data solution. It is independent to platform and often be ported to a variety of embedded systems. Porting FFmpeg to Android platform will increase the codec standards supported by Android. However, the low processing power, low memory and limited hardware configuration of the mobile phone, seriously affected the codec efficiency of FFmpeg. And finally led to audio and video data can not be synchronized. By studying the timestamp-based multimedia audio and video synchronization algorithm model and introduced it into FFmpeg, this paper validated the effectiveness of it on Android platform. The experiments showed that the timestamp-based model of multimedia audio and video synchronization algorithm can effectively ensure the synchronization of multimedia data.

**Key words:**FFmpeg; multimedia; timestamp; audio and video synchronization; Android

## 0 引言

FFmpeg 是一个开源跨平台多媒体数据解决方案,它包含非常先进的音视频编解码库 Libavcodec,支持超过 90 种编解码标准及 RTSP、HTTP、MMS 等多种网络传输协议。FFmpeg 通过 C 语言实现,常被移植到各种嵌入式系统中<sup>[1]</sup>。Android 系统是 Google 公司开发的基于 Linux 系统的开源手机操作平台,该平台由操作系统、中间件、用户界面和应用软件组成,号称是首个为移动终端打造的真正开放和完整的移动软件<sup>[2]</sup>。

将 FFmpeg 移植到 Android 系统中,能够增加 Android 系统对编解码格式标准的支持,提高 Android 系统对多媒体信息处理能力,改善用户体验。但由于目前手机处理能力低、内存小等硬件配置因素,严重影响 FFmpeg 对音视频流的解码效率,导致解码出的音视频数据无法同步。

## 1 基于时间戳的多媒体同步传输模型

多媒体数据同步传输从广义上包含内容同步、空间同步和时间同步。内容同步是指同一数据不同表示方式之间的同步。空间同步是指在某一时间点上不同媒体在显示设备上的位置关系。从狭义上,是各种媒体在时间上的同步关系<sup>[3]</sup>。在实际传输过程中,存在单个媒体流中相邻媒体单元所经历的时延抖动及音频和视频的相关媒体单元的时间差,即偏移。有媒体内偏移和媒体间偏移两种,相应地存在着偏移许可范

收稿日期:2011-03-11;修回日期:2011-06-22

基金项目:国家自然科学基金(61070189)

作者简介:胡 成(1983-),男,硕士研究生,主要研究方向为信息安全以及多媒体技术;任平安,副教授,硕士生导师,主要研究领域为信息安全。

围<sup>[4]</sup>,如音视频媒体内最大时延均为 0.25s,最大时延抖动均为 10ms;音视频媒体间偏移在不同条件下也有相关的范围,如视频动画在相关的条件下,允许偏移范围是 $\pm 120$  ms,而音频在背景音乐条件下,允许偏移范围是 $\pm 500$ ms,这些参数很大程度上依赖于具体的应用。基于时间戳的多媒体同步传输模型从狭义范围内保证各路媒体数据在时间上的同步<sup>[5-7]</sup>。

在多媒体流时间信息中,音频流通过采样率表示当前音频数据速度,视频流通过帧率表示当前视频帧数据时间。作为一种时间信息的重要补充,多媒体流时间信息中提供 DTS (Decoding Time Stamp) 和 PTS (Presentation Time Stamp) 用于各路多媒体流的同步。以 MPEG 编码标准为例,MPEG 的语法结构主要分三层:传输 (Transport stream) 层、PES (Packetized Elementary Stream) 层,和 ES (Elementary Stream) 层。MPEG 算法的特点是在空域和时域上都进行压缩编码。在三层码流结构中都带有表示时间的信息。ES 中有 Temporal-reference, PES 中有显示时间标签和节码时间标签,而传输层中有节目参考时钟<sup>[8-10]</sup>。图 1 是传输流、PES 流、ES 流之间关系示意图。

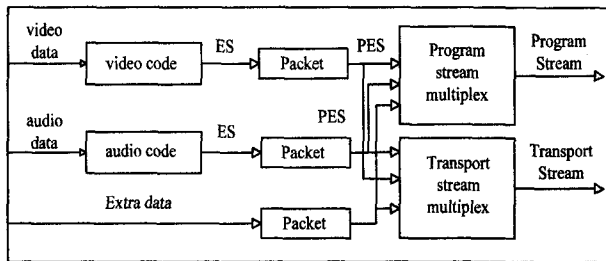


图 1 传输流、PES 流、ES 流之间关系示意图

针对传输的多媒体数据流,选择一个参考时钟,以参考时钟为基准对每个数据块打上时间戳,在播放显示多媒体数据时,读取数据块的时间戳,同时参考当前参考时钟上的时间来安排播放<sup>[11,12]</sup>。基于时间戳的多媒体同步传输算法模型如图 2 所示。

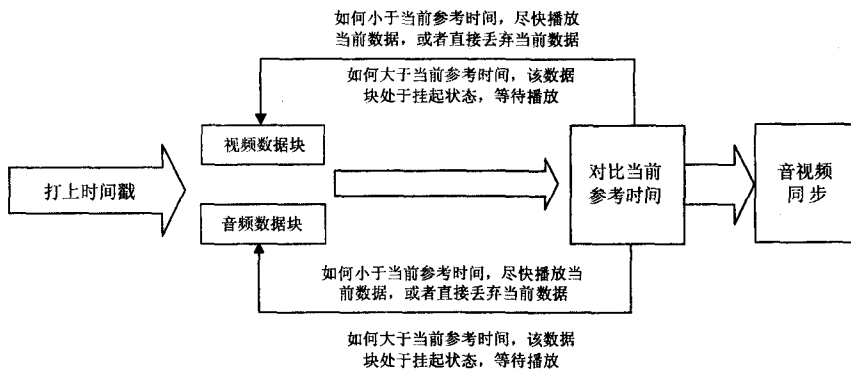


图 2 基于时间戳的多媒体同步传输算法模型

当对解码完的多媒体数据 (音频或视频) 进行显示播放时,提取多媒体数据块中的时间戳信息,对比当前的参考时间,如果多媒体数据块的时间戳小于当前

的参考时间,就尽快播放当前数据,甚至直接丢弃当前多媒体数据;如果多媒体数据块的时间戳大于参考时间,则该数据块就立即转入挂起状态,等待播放<sup>[13-15]</sup>。

## 2 FFmpeg 音视频同步算法实现

### 2.1 FFmpeg 多媒体解码流程

FFmpeg 提供了一系列多媒体数据处理函数,利用这些函数可以完成音视频文件的解码播放功能。FFmpeg 对音视频数据的解码流程如图 3 所示。

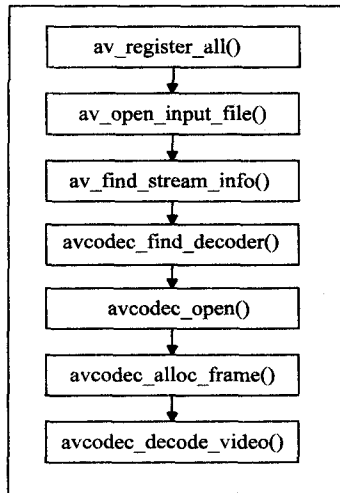


图 3 FFmpeg 音视频数据的解码流程

首先通过 `av_register_all()` 函数完成 FFmpeg 系统支持的编解码格式注册,然后通过 `av_open_input_file()` 函数打开本地或者处于网络服务器端多媒体文件。打开多媒体文件后,利用 `av_find_stream_info()` 找到多媒体文件中的视频流、音频流对应的流索引 ID。`avcodec_find_decoder()` 与 `avcodec_open()` 使用视频流、音频流 ID,完成视频解码器与音频解码器的初始化工作,`avcodec_alloc_frame()` 会为解出的每帧视频开辟一个缓存,`avcodec_decode_video()` 开始真正的解码工作<sup>[16]</sup>。

### 2.2 FFmpeg 音视频同步算法实现

FFmpeg 提供了 DTS 和 PTS 参数,其中 DTS 表示解码时间戳,PTS 表示显示时间戳。当通过解包函数 `av_read_frame()` 从多媒体数据流中获得一个数据包时,PTS 和 DTS 的信息也会保存在包中,但 PTS 并不是解码出来的原始帧的 PTS,因此无法利用该 PTS 进行多媒体数据同步。

当通过 `avcodec_decode_video()` 函数对解出的数据包解帧时,`avcodec_decode_video()` 将调用一个函数

来为每一帧申请一个缓冲。因此在 FFmpeg 解帧时,利用一个函数来保存一个包的时间戳,来进行时间戳的校正。基于 FFmpeg 的音视频同步法伪代码如图 4 所示。

```
ref_clock = get_audio_clock(); //根据音频播放时间获得参考时间
diff = vp->pts - ref_clock; //当前帧的播放时间戳与参考时间的差值
sync_threshold = (delay > AV_SYNC_THRESHOLD)? delay: AV_SYNC_THRESHOLD;
if (fabs(diff) < AV_NOSYNC_THRESHOLD) {
    if (diff <= -sync_threshold) //视频落后于参考时钟(音频) {
        diff = 0; //不需要延时
    } else if (diff >= sync_threshold) //视频领先于参考时钟(频) {
        diff = 2 * delay; //增加延时
    }
}
schedule_refresh(is, (int)(diff * 1000 + 0.5)); // 延时控制
video_display(is); //显示视频图像
```

图 4 FFmpeg 的音视频同步算法伪代码

### 2.3 FFmpeg 的移植

操作系统: Windows xp

编译环境: Cygwin 1.7.6

开发平台: Android 2.2

NDK 版本: Android-ndk-r4b

Cygwin 是 Windows 操作系统下的类 Linux 环境,是 FFmpeg 项目的编译环境。Android-ndk 是 Android 的 c/c++ 程序集成开发平台,利用 NDK 可以将 c/c++ 程序编译成 Android 系统能够执行的动态库文件(.so)。

编译移植步骤如下:

a) 首先进行配置。在 ffmpeg 下创建一个 config.sh 配置文件

b) 在 cygwin 中执行 config.sh 文件

\$ chmod +x config.sh /\* 改变 config.sh

文件的执行权限 \*/

\$ ./config.sh

c) 编写 makefile 文件。Android 的 makefile 文件不同于一般的情况下 Linux 中的 makefile 文件,因此需要重新编写 FFmpeg 的 makefile 文件<sup>[10]</sup>。例如在 ffmpeg\libavcodec 下,Android.mk 内容如下:

```
LOCAL_PATH := $(call my-dir)
include $(CLEAR_VARS)
include $(LOCAL_PATH)/../av.mk
LOCAL_SRC_FILES := $(FFFILES)
LOCAL_C_INCLUDES := \
    $(LOCAL_PATH) \
    .....
LOCAL_SHARED_LIBRARIES := libavutil
LOCAL_MODULE := $(FFNAME)
include $(BUILD_SHARED_LIBRARY)
```

编写 FFmpeg 工程下其它的源文件的 makefile 方法类似。

d) 在 cygwin 中执行 make。具体命令为:

\$ make

最终,FFmpeg 工程被编译到 Android 系统中。

## 3 实验结果

为验证基于时间戳的多媒体同步传输模型在 FFmpeg 应用的效果,选择三种不同码率的多媒体文件进行测试,测试环境如下:

移动终端: HTC G3

CPU 主频: 528MHz

内存: 288 MB

移动终端操作系统: Android 2.2

表 1 是分别采用多媒体同步传输算法与不使用同步传输算法的多媒体播放器播放不同码率的 flv 视频格式的对比。从表 1 可以看出,采用音视频同步方法,可以有效的缓解高码率文件播放时带来的音视频不同步现象,达到了预期的目的。

表 1 音视频同步对比表

视频文件	含音视频同步	不含音视频同步
500kbps FLV 文件	很流畅,无音视频不同步	很流畅,无音视频不同步
800kbps FLV 文件	流畅,无音视频不同步	流畅,部分画面出现音视频不同步
1200kbps FLV 文件	不流畅,出现跳帧现象,音视频基本同步	不流畅,音视频长时间不同步

## 4 结束语

随着多媒体技术的广泛运用,用户对多媒体应用程序的要求也越来越高。文中提出了一种适用于 FFmpeg 的音视频同步方法,并在 Android 2.2 软件平台上进行了系统测试,具有良好的效果,为其他移动终端嵌入式多媒体应用软件的开发给出了一些参考。

### 参考文献:

- [1] 吴张顺,张 询. 基于 FFmpeg 的视频编码存储研究与实现[J]. 杭州电子科技大学学报,2006,26(3):30-34.
- [2] 高焕堂. Google Android 应用框架原理与程序设计 36 技[M]. 台北: 广悦文化,2008:23-25.
- [3] 李笑佳,周 兵,李晓强,等. 视频重演中的同步技术[J]. 计算机工程,2005,31(2):64-66.
- [4] 张维明,吴玲达,老松杨. 多媒体信息系统[M]. 北京: 电子工业出版社,2002.
- [5] 吴 炜,常义林. 一种 MPEG 2 媒体同步控制算法[J]. 系统工程与电子技术,2005,27(1):173-177.
- [6] 许 延,常义林,刘增基. 多媒体同步技术研究[J]. 西安电子科技大学学报:自然科学版,2000,27(4):504-509.

(下转第 91 页)

成。并通过纹理映射模拟烟花粒子的形状和颜色,非常逼真。效果图4是烟花在空中爆炸时渲染,帧刷新率达到了163fps,画面渲染比较流畅,图5是渲染爆炸后粒子下落和消失时的渲染。两幅图基本对粒子系统周期进行了较好的渲染<sup>[12]</sup>。

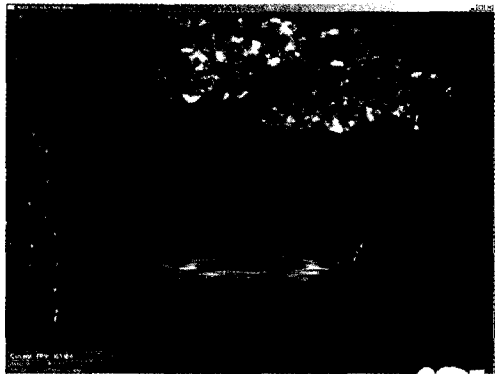


图4 烟花爆炸效果(1)

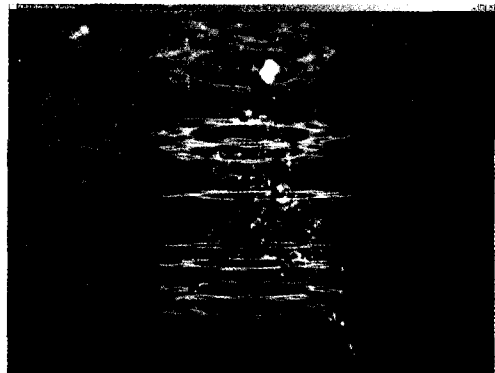


图5 烟花爆炸效果(2)

## 6 结束语

文中利用 OGRE 图形引擎的粒子系统原理,对烟花进行了渲染,脚本和程序结合让渲染更加逼真。一方面充分对不规则模糊物体进行了很好的渲染,节省了成本,另一方面也具有较高性能的图形模拟表现,让

渲染画面清楚流畅。

由于烟花种类繁多,实现全部的烟花渲染是不可能的,这是文中不足之处。但是如果能够对粒子系统与场景管理进行很好的设置,将可以渲染出种类繁多的烟花渲染。

## 参考文献:

- [1] Admin. OGRE 图象引擎介绍[EB/OL]. 2010-06. <http://www.azure.com.cn/default.asp>.
- [2] 白建军,朱亚军,梁 辉,等. OpenGL 三维图形设计与制作[M]. 北京:清华大学出版社,1998:378-391.
- [3] 鲁 萌,刘建波. 三维地形显示中数据缓存与调度算法研究[J]. 微计算机信息,2010,4(1):210-212.
- [4] 杨子华,刘宏芳. 基于粒子系统模型的自然景物生成技术应用研究[J]. 计算技术与自动化,1998,17(3):20-23.
- [5] 王宏伟,刘 越,王涌天. 面向对象的通用粒子系统设计[J]. 系统仿真学报,2006,18(1):46-48.
- [6] Reeves W T, Blau R. Approximate and probabilistic algorithms for shading and rendering structured particle system[J]. Computer Graphics, 1985, 19(3):313-322.
- [7] 张尚华. 烟花燃放效果的仿真研究[D]. 广州:中山大学计算机与信息学院,2006.
- [8] Steven P, Reiss. An Engine for the 3D Visualization of Program Information[J]. Journal of Visual Languages & Computing, 1995, 6(3):127-129.
- [9] 张 璞,陶丽娜. 基于 OpenGL 的岩石楔形体边坡三维分析系统[J]. 计算机技术与发展,2009,19(3):53-56.
- [10] 张巧芳,李光耀. 基于单幅图像的三维浏览图生成算法[J]. 计算机技术与发展,2010,20(1):43-47.
- [11] Brundy A L, Saltzman D H, Emerson D, et al. Sonographic features associated with cleft palate[J]. Clin Ultrasound, 1986, 14: 486-489.
- [12] Monni G, Ibba RM, Olla G, et al. Colour Doppler Ultrasound and prenatal diagnosis of cleft palate[J]. Clin Ultrasound, 1995, 23:189-192.

(上接第87页)

- [7] 王少燕. 多媒体通信中的音视频同步问题研究与实现[D]. 西安:西安电子科技大学,2003.
- [8] 鲁 萍,马光四. 多媒体数据流实时传输速率研究及应用[D]. 西安:西安建筑科技大学,2005.
- [9] Moving Pictures Expert Group. MPEG-2 TextModel 5. DocISO/IEC JTC1/SC29/WG11/N0400[S]. 1993.
- [10] Fibush D. Timing and Synchronization Using MPEG-2 Transport Streams[J]. SMPTE Journal, 1996(7):395-405.
- [11] 求是科技. Vc++ 音视频编解码技术及实践[M]. 北京:人民邮电出版社,2006.
- [12] 陈 芳,沈晓军,陈 洁. 多媒体同步技术的研究[J]. 北京工业大学学报,1996,22(4):110-114.
- [13] 程文青,陈云鹤,徐 晶. 一种适用于嵌入式媒体播放器的音视频同步方法[J]. 计算机与数字工程,2007,35(2):161-163.
- [14] 吴 进,贺 辉,洪 辉. 多媒体数据流实时传输技术的研究[J]. 通信技术,2009,42(1):342-344.
- [15] 刘 芳. 基于时间轴模型的音视频同步的研究与实现[D]. 南京:暨南大学,2008.
- [16] ZHANG Wei-ming, WU Ling-da, LAO Song-yang. Multimedia Information System[M]. Beijing:Publishing House of Electronic Industry, 2002.