DOI:10.16208/j.issn1000-7024.2013.06.023

基于 FFMPEG 解码的音视频同步实现

刘丽霞，边金松，张 琍，穆 森

（中国航天科工集团第二研究院 706 所，北京 100854）

摘 要：为实现音视频同步播放，针对音视频数据同时被采集，但编码和存储独立的情况，提出了将音频播放时钟作为同步时钟，采用时间戳技术实现历史音视频同步播放。该方法使用 FFMPEG 对历史音视频文件分别进行解码，将解码后计算得到的音频播放时钟作为同步时钟，控制视频播放速度同步到音频播放时钟上，保证了音视频数据流畅播放，同步无滞后，无延迟。通过实验设计，验证了提出的基于音频播放时钟的时间戳同步方法是有效的。

关键词：音频；视频；同步；时间戳；显示时间戳（PTS）

中图法分类号：TP311 **文献标识号**：A **文章编号**：1000-7024（2013）06-2087-06

Synchronization playing of audio and video based on FFMPEG

LIU Li-xia, BIAN Jin-song, ZHANG Li, MU Sen

(Institute 706, Second Academy of China Aerospace Science and Industry Corporation, Beijing 100854, China)

Abstract: To synchronize the playback of audio and video, based on the situation that the audio and video are acquired at the same time, but independent of being encoded or stored, a new method of timestamp is proposed. It is that audio playback clock is used as synchronous clock and FFMPEG is used to decode the audio and video data separately. Then the audio playback clock which is computed by the decoded timestamp is used as synchronous clock. Video playback speed is controlled by audio playback clock. The method can guarantee the audio and video to play smoothly, without lag and delay. Finally the experimental results prove the timestamp synchronization method proposed is effective.

Key words: audio; video; synchronization; timestamp; presentation time stamp (PTS)

0 引 言

音视频同步是同步采集的声音和视频图像信号在再现过程中保持同步的程度。

随着网络和高新技术的发展，音视频传输由同轴电缆发展到由网络传输，同步采集到的原始音视频数据在发送端需要压缩编码，经网络传输到接收端后，进行存储或实时播放。在接收端无论是播放实时音视频还是播放历史音视频，都需要对压缩的音视频数据进行解码，获得原始音视频数据。然后采用专业技术实现音视频同步播放。

目前，比较常用的音视频同步技术有：

时间戳技术：视频和音频打上播放时间戳，时间戳相同的音视频同时播放。时间戳本身就是同步信息，这种方法简单明了，具有广泛应用。

音视频数据复用技术：将音频数据和视频数据复用成一个数据流。这种复用方法不需要额外的同步时钟，但是复用算法比较复杂，对来源不同的音频数据和视频数据不太适合。

基于反馈的技术：当接收信息端检测到音视频不同步时，将信息反馈回信息发送端，通过反馈机制平衡发送端与接收端的同步。它适用于网络条件较佳的实时播放情况。

信道同步技术：音频数据和视频数据单独传输，另外同步信息也单独传输，接收端依靠接受的同步信息完成音视频同步。它可用于直连设备，能支持复杂的同步关系，但是同步信息易丢失。

经过比较各类音视频同步算法的优缺点及适用场合，本文采用时间戳技术实现对同时被采集，但独立编码、传输、存储的历史音视频数据的同步播放，与原有时间戳技术不同的是本文将音频播放时钟作为同步时钟，将视频同步到音频上，即音频通过声卡时钟自动播放，只控制视频播放到音频时钟上。这样，可以保证音视频数据流畅播放，无滞后，无延迟。另外，使用有效的内存读写管理算法来

收稿日期：2012-09-04；修订日期：2012-11-25

作者简介：刘丽霞（1985-），女，山西忻州人，硕士，工程师，研究方向为图像处理和识别；边金松（1976-），男，河北沧州人，工程师，研究方向为计算机技术及应用；张琍（1973-），女，北京人，高级工程师，研究方向为计算机技术及应用；穆森（1972-），男，北京人，研究员，研究方向为信息化技术应用。E-mail：lixia20082008@gmail.com

控制播放速度，可以使音视频播放更平滑流畅。

本文使用开源库 FFMPEG 对音频视频进行解码，FFMPEG 的高性能设计，以及其包含的经过了优化的解码算法，使得音视频解码速度相当快，且占用内存小。另一方面 FFMPEG 库对主流的音频、视频编码格式几乎都支持，用 FFMPEG 进行解码，具有通用性。

1 音视频数据

1.1 基础知识

打包：为便于传输，对被压缩的视频流、音频流进行“打包”，就是将顺序、连续传输的数据流按一定的时间长度进行分割，分割的小段叫“包”。

PES（packetized elementary stream）：将视频流、音频流打包后，在每一个包前加包头，用于区别不同性质的流。PES 包头信息中加入 PTS（presentation time stamp）和 DTS（decode time stamp），用于音视频同步。视频 PES 一般一帧一个包，音频 PES 一般一个包的数据量不超过 64KB。

时间戳：视频流或音频流在形成包时，包头需要包含时间信息用于标识显示时间和解码时间。如 PTS 与 DTS。

PTS（presentation time stamp）：音视频显示的时间标签。

DTS（decode time stamp）：音视频解码的时间标签。

视频帧：一帧就是一副静止的图像，连续的帧就形成视频。帧率，是在 1 秒钟时间里传输的图片的帧数，通常用 FPS（frames per second）表示。帧率（FPS）愈多，所显示的动作就会愈流畅。

音频采样率：是指录音设备在一秒钟内对声音信号的采样次数，采样频率越高声音的还原就越真实越自然。在当今的主流采集卡上，采样频率一般共分为 22.05KHz、44.1KHz、48KHz 这 3 个等级，48KHz 是数字电视、DVD、DAT、电影和专业音频所用的采样率。

FFMPEG：FFMPEG 是一套可以用来记录、转换数字音频、视频，并能将其转化流的开源计算机程序。它是功能非常强大的多媒体处理工具，提供了录制、转换以及流化音视频的完整解决方案。它包含了目前领先的音/视频编码库 libavcodec 等。FFMPEG 是在 Linux 平台下开发的，但可以在 Windows 等其他环境下编译运行。可以实现跨平台应用。

PCM 码（pulse-code modulation 脉冲编码调制）：将采集到的时间连续，取值连续的声音信号先抽样，再对样值幅度量化，编码成二进制码。

YUV 空间：表示颜色空间，用来描述影像色彩及饱和度，用于指定像素的颜色。Y 表示明亮度（luminance），也就是灰阶值，U 和 V 表示的则是色度（chrominance）。图像的 YUV 为最初的颜色值，是未经压缩编码的数值。

1.2 音视频数据流向

图 1 显示了音视频数据从采集、传输，再到同步播放的流程。音视频数据在传输前，经过打包，并在每个包头中添加了 PTS 与 DTS 信息，用来标识解码时间和显示时间。在接收端，可以实时同步播放音视频，也可以分别存储音视频数据。基于时间戳的音视频同步播放，就是控制视频播放时钟与音频播放时钟的步调一致。

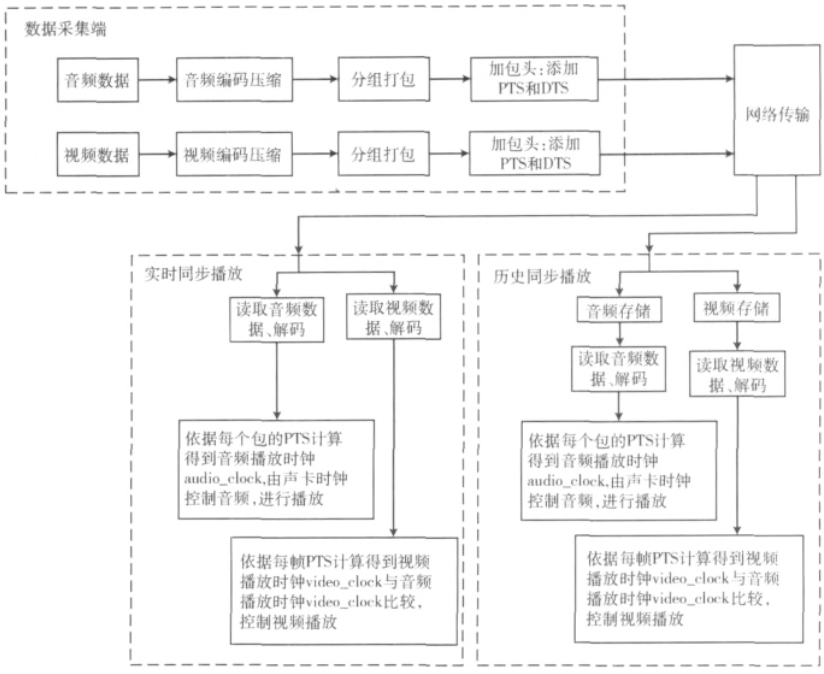

图 1 音视频数据流向

2　音视频同步测量

音视频同步好坏不仅和同步技术相关，还与人们的听觉和视觉特性有极大的关系。音视频同步用它们出现的时间差表示，单位为 ms，声音信号提前为正，迟后为负。图 2 是来自 ITU-R BT. 1359-1[1]的建议，它是对人类听觉、视觉觉察能力的研究成果。图中，不可察觉门限为-90ms 与 20ms 间，即声音落后视频 90ms 到超前视频 20ms 的范围内，人们几乎察觉不出音视频质量的变化；将主观评价降 0. 5 级时作为可察觉门限，它对应于-125ms 到 45ms；图中 A，A’称为可接受门限，对应于-185ms 到 90ms。这些数据明确指出，人们的感觉对声音落后于图像要比声音超前于图像更容易接受，这可能是由于人们对声音落后于图像的一般自然现象长期习惯的结果[2]。

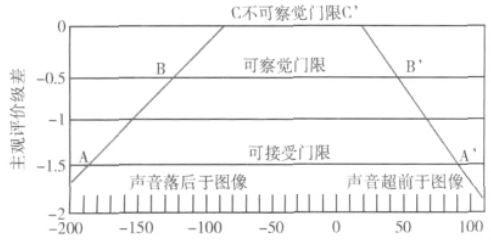

图 2　人们对音视频同步的主观评价

3　基于 FFMPEG 的音视频同步播放

3. 1　基于 FFMPEG 的流媒体解码流程

FFMPEG 解码音视频流程为：从媒体文件中读取出来的媒体流，首先由 FFMPEG 的格式解析器获取其格式，再由媒体格式解码器去除媒体流外层的媒体格式，然后送入 FFMPEG 的 codec 结构，并由其中的 decode 找到媒体流对应的解码器，从而进行解码，获得原始视频数据 YUV 或原始音频数据 PCM。利用 DirectX 媒体库把 YUV 图像显示出来，从而实现视频播放，将 PCM 码送入声卡缓冲区，完成声音播放。具体流程如图 3 所示。

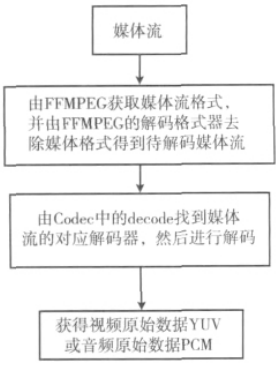

图 3　FFMPEG 解码流程

FFMPEG 编解码器 Codec 结构为：AVCodec。AVCodec 结构体中用于编码、解码的函数分别为：

int (* encode) (AVCodecContext *, uint8 _t * buf, int buf _size, void * data);

int (* decode) (AVCodecContext *, void * outdata, int * outdata _size, AVPacket * avpkt);

当用于解码时，codec 结构中的编码函数指针 encode 为空，同样，当用于编码时，codec 结构中的解码函数指针 decode 为空。

3. 2　同步原理

基于时间戳的音视频同步方法有：同步音频到视频，同步视频到音频，或者同步音视频到外部时钟。

本文所采用的同步机制为：将音频播放时钟作为同步时钟，控制视频播放时钟同步到音频播放时钟上，即视频的播放完全根据声卡最后的数据输出来同步。

3. 3　音视频同步播放模块

本文通过五类模块实现同步算法，分别是内存管理模块、解码模块、时间控制模块、音频播放模块、视频播放模块。

（1）内存管理模块

将传输来的媒体数据写到内存中，通过读操作为解码模块提供数据源。

（2）解码模块

包括音频解码模块和视频解码模块。

音频解码模块负责把从内存管理模块得到的音频数据利用 FFMPEG 进行解码，然后送入音频播放模块进行播放。

视频解码模块负责把从内存管理模块得到的视频数据利用 FFMPEG 解码，然后送入视频显示模块进行图像显示。

（3）时间控制模块

负责得到音视频解码时间（DTS）、显示时间（PTS）、音频播放时钟、视频播放时钟、同步时间，为音视频时间同步做准备。

（4）音频播放模块

负责把解码后的音频数据 PCM 送入音频驱动，使声音播放出来。

（5）视频显示模块

负责把解码后的视频数据 YUV，通过 DirectX 媒体库把图像显示出来。

各模块间的关联关系如图 4 所示，黑线框中是用到的相应模块。

3. 4　音视频同步播放流程

3. 4. 1　音频播放

音频播放流程为：内存管理模块将音频数据写到指定内存中，解码模块对音频数据进行解码，音频播放模块将

解码后的音频数据送入音频驱动，播放声音。

将音频播放时钟作为同步时钟，需要更精确的时间，来记录正在播放的音频位置。本文采用当前音频帧的 PTS 和下一帧的 PTS 计算获取更精细的播放时钟，假设由当前 PTS 获得的音频播放时钟为 audio _clock _pts，下一帧 PTS 换算得到的音频播放时钟为 audio _clock _next _pts，则当前音频播放时钟 audio _clock 计算如下

audio _clock = audio _clock _next _pts - (M-N) / audio _play _bps

其中，M 为播放内存中存放的一帧音频数据的大小，N 为当前已播放数据量大小，audio _play _bps 为当前音频播放速率。

3.4.2 视频同步播放

图 4 给出了音视频同步的流程，其基本原理是视频的播放时钟 video _clock 要与当前同步时钟即音频播放时钟 audio _clock 相匹配。本文设定了阈值 threshold1，threshold2，当 audio _clock 与 video _clock 的差值 Diff 在 threshold1 与 threshold2 范围之内的，就认为基本同步，否则为失同步。当音视频同步时，播放当前视频帧，然后继续读取音频数据、视频数据，解码，更新当前 audio _clock 与 video _clock，计算差值 Diff，重复整个过程。

根据图 2 ITU-R BT. 1359-1 标准，即人们对音视频同步的主观评价建议，threshold1 与 threshod2 的值设置在可察觉门限内，本文采用 threshold1= -90ms，threshold2= 20ms。

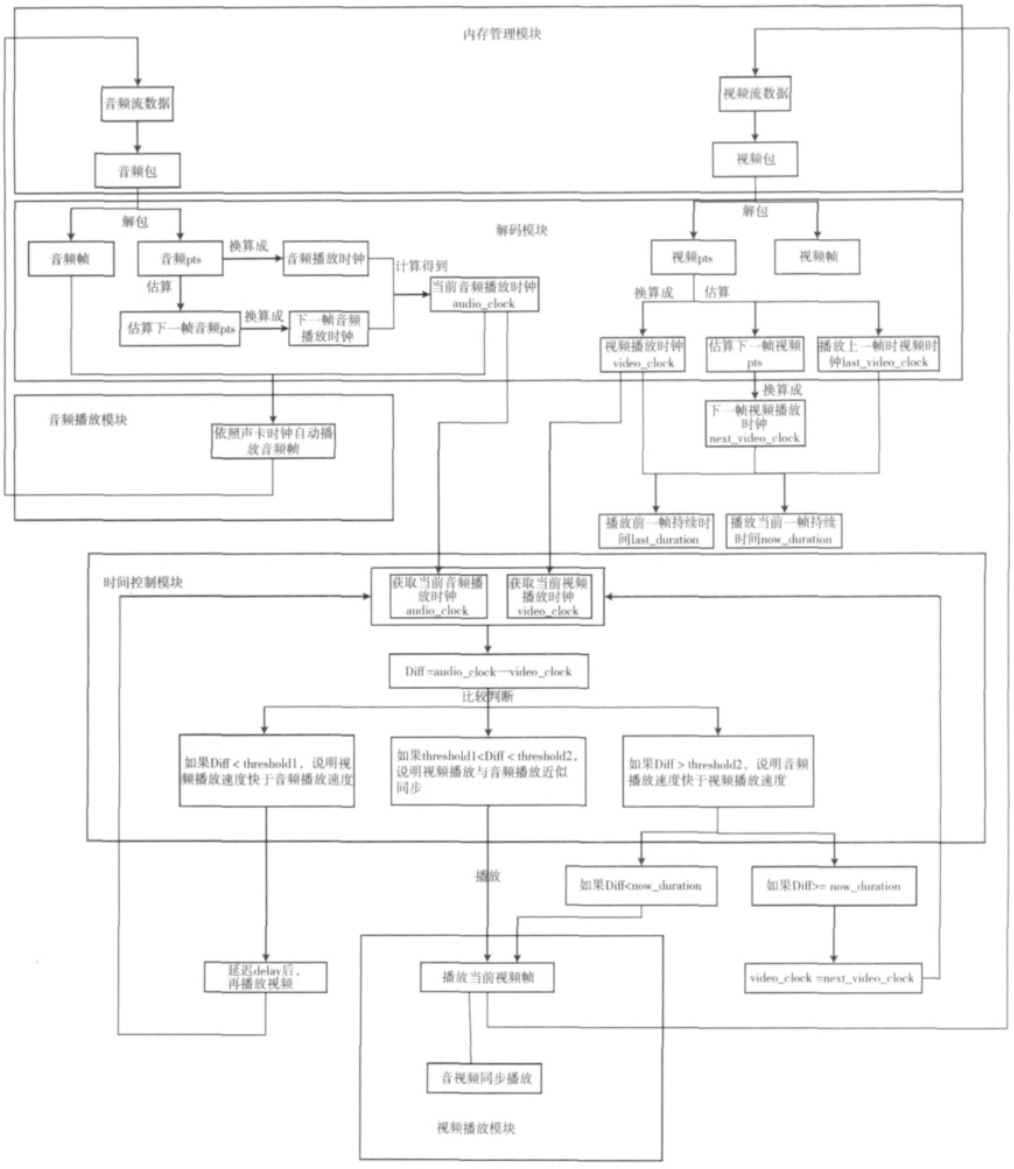

图 4 音视频同步播放流程

3.4.3 失同步处理

当音频播放时间 audio _clock 与视频播放时间 video _clock 之差值 Diff 落在 threshold1 与 threshold2 以外，说明音视频不同步。失同步有两种情况：

（1）音频播放速度快于视频图像显示速度

当 Diff＞threshold2 时，此时音频播放速度快于视频图像显示速度。如果 Diff 的绝对值小于播放当前一帧持续时间 now _duration，播放当前视频帧，继续读取音视频包，更新音频播放时间、视频播放时间，循环到计算 Diff 步骤；如果 Diff 的绝对值大于播放下一帧视频的持续时间 now _duration，跳过此视频帧不播，更新视频播放时钟即 video _clock = next _video _clock，然后读取视频包，解码，获得 next _video _clock，跳转到计算当前 audio _clock 与 video _clock 差值 Diff 的步骤，如图 4 所示。

（2）音频播放速度落后于视频图像显示速度

音频开始播放时，设置视频总延迟量 total _delay，初始化 total _delay = 0。当 Diff＜threshold1 时，此时音频播放速度落后于视频图像显示速度。视频播放需要延迟，初步设置延迟时间为 delay。

delay = last _duration = video _clock -last _video _clock。

设 delay _Diff = audio _clock + total _delay + delay-video _clock。

如果 delay _Diff＜threshold1，说明 delay 时间较短，加倍延迟即 delay = 2 * delay，重新计算 delay _Diff，并与 threshold1，threshold2 进行比较，直到 delay _Diff 在 threshold1 与 threshold2 范围内。

如果 delay _Diff＞threshold2，延迟时间较长，需缩短延迟时间为 delay =delay/2，重新计算 delay _Diff，并与 threshold1，threshold2 进行比较，直到 delay _Diff 在 threshold1 与 threshold2 范围内。

如果 threshold1＜delay _Diff＜threshold2，视频播放延迟 delay 时间后，更新音频播放时钟 audio _clock，更新 total _delay=total _delay+delay。重新返回到图 4 计算 Diff 步骤。

3.5 数据内存读写控制

为了便于对内存中音视频数据读写速度的控制，使其平缓、流畅的播放，本文设计了一个循环缓冲器，来存放音视频数据。它的工作机制是：设置了两个指向循环缓冲器的指针，一个读指针一个写指针，循环缓冲区中只要有数据，读指针就工作，只要有空间，写指针就工作。但是两个指针都不可以互相超越。设置合适的阈值 thresh 来控制读写指针读写速度。

当读指针超越写指针或写指针与读指针的差在阈值 thresh 之内，说明内存缓冲区数据将为空。此时通知读线程停止读取操作，直到写指针超越读指针并且写指针与读指针的差大于阈值 thresh，重新开始读取数据。

当读指针相对量与写指针相对量的和等于或大于循环缓冲区大小，说明未被读到的数据将会被新的数据冲刷掉，此时该消息会通知写线程停止写操作，直到读指针相对量与写指针相对量的和小于循环缓冲区大小后，写操作重新被执行。

建立读写相互制约的循环缓冲区，不仅节省内存，还保证了播放的流畅性。

4 实验与分析

为了模拟以数据流的方式读取文件，本文采用的信息源是从网络下载得到的电影、MV 的音视频数据（带有字幕），经专业软件分离得到单独的音频和视频。实验选择四段长短不一的视频源。

实验结果由 10 人打分进行主观评测。评测结果分为 3 个等级：不同步（0 分），同步但有偏差（1 分），同步（2 分）。将 10 人给出的累加分数作为最终评测得分（满分为 20 分）。同步评测标准为观看影片流畅及音频与视频间无滞后、无超前情况，如：视频中台词与人的口型一致，见表 1。

表 1 实验结果

信息来源	总时间(s)	音频格式	视频格式	评测标准	评测分数(满分 20 分)	满意度(评测分数/满分)
MV：美丽的神话（带歌词）	290	Mp3	avi	字幕(图像)与音频是否一致	20	100%
电影：BATTLESHIP（Part 1）（超级战舰＜上＞）	3853	Mp3	MPEG4	字幕(图像)与音频是否一致	19	95%
脱口秀：艾伦杜兰大学毕业演讲（双语字幕）	634	wav	avi	字幕(图像)与音频是否一致	20	100%

实验中的视频格式有 AVI、MPEG4，音频格式有 WAV、MP3，FFMPEG 会自动识别音视频流，并判断是哪种格式，FFMPEG 支持多种格式的音视频解码，无需统一转码，保证了实验的高效性。

从得分情况看，无论是长影片还是短影片，本文采用的同步算法给人们以很好的感官效果。

5 结束语

针对采集同步，编码、传输独立的音视频数据，在接收端，本文提出采用时间戳技术，将音频播放时钟作为同步时钟，通过控制视频播放速度实现音视频同步播放。为此，详细讨论了音视频数据流向、音视频同步测量标准，同步原理及失同步处理过程，分析了内存管理技术难点，并提出了数据内存读写控制策略。依据音视频同步测量标准，结合 FFMPEG 成熟的解码技术，设计了相应的测试实验，结果表明提出的音视频同步技术是有效的。

参考文献：

[1] ITU-R BT. 1359-1. Relative timing of sound and vision for broadcasting [S]. 1998.

[2] XU Kangxing. The time difference and measurement between audio and video [C] //the 10th National Consumer Electronics Technology Annual Conference and Digital TV Seminar, 2008: 213-217 (in Chinese). [徐康兴. 音频与视频信号时间差及其测量 [C] //2008 年第十届全国消费电子技术年会暨数字电视研讨会，2008：213-217.]

[3] CHEN Zhiyin, XIE Weiduo, WANG Lei. Research and implementation of audio and video real-time synchronization transmission [J]. Huaqiao University periodical (Natural Science), 2011, 32 (4): 389-392 (in Chinese). [陈志颖，谢维多，王磊. 音视频实时同步传输技术研究与实现 [J]. 华侨大学学报(自然科学版)，2011，32 (4)：389-392.]

[4] LIU Lixia. Study and comparison of image texture features [D]. Beijing: Beijing University of Post and Telecommunication, 2011 (in Chinese). [刘丽霞. 图像纹理特征研究和比较 [D]. 北京：北京邮电大学，2011.]

[5] LIU Lixia, ZHANG Honggang, FENG Aiping, et al. Simplified local binary pattern descriptor for character recognition of vehicle license plate [C] // Seventh International Conference on Computer Graphics, Imaging and Visualization, 2010: 157-161.

[6] ZHANG Honggang, LIU Lixia, FENG Aiping, et al. A new combination feature of S-LBP and main color descriptor for video frame retrieval [C] // IEEE IC-NIDC, 2010.

[7] DONG Chunbing. Research and implementation of audio and video synchronization [D]. Changchun: JiLin University, 2007 (in Chinese). [董春兵. 音视频同步的研究与实现 [D]. 长春：吉林大学，2007.]

[8] QI Chengming. Research and implementation of audio and video synchronization problems [D]. Harbin: Harbin Institute of Technology, 2009 (in Chinese). [齐成明. 音视频同步问题的研究与实现 [D]. 哈尔滨：哈尔滨工业大学，2009.]

[9] YANG Bei. Design and implementation of audio and video synchronization in streaming media [D]. Wuhan: Huazhong University of Science and Technology, 2008 (in Chinese). [杨蓓. 流媒体系统中音视频同步机制的设计与实现 [D]. 武汉：华中科技大学，2008.]

[10] AN Cui. Design and implementation of multimedia distance learning system [D]. Chengdu: University of Electronic Science and Technology of China. 2010 (in Chinese). [安翠. 多媒体远程教学系统的设计与实现 [D]. 成都：电子科技大学，2010.]