

# 基于 C/S 模式的机载多路视频同步回放技术

郝 朝, 聂瑶佳

(中国飞行试验研究院, 陕西 西安 710089)

**摘 要:**在飞行试验中,机载视频数据是试飞测试数据中的重要组成部分。目前机载记录视频数据为独立的 M2T 文件,现有视频播放软件只能对单路视频文件进行回放。为解决机载多路视频数据同步回放的问题,提出了一种基于 C/S 模式的多路视频同步回放技术方案,服务器端进行视频文件读取控制,客户端进行视频解码显示。首先介绍了 FFMPEG 的基本概念,然后论述了系统的总体设计方案,重点介绍了服务器端和客户端的设计与实现。该系统采用 C++ 语言进行设计,服务器采用 TCP/IP 协议与客户端进行数据交互。软件设计利用模块化和多线程的思路,提高处理效率,保证功能的可扩展性。目前该软件已成功应用于多个型号任务多路视频同步回放中,实际应用效果表明该系统工作稳定,能够有效提高型号试飞效率。

**关键词:**C/S;多路视频;同步回放;FFMPEG;SDL

**中图分类号:**TP311.1

**文献标识码:**A

**文章编号:**1673-629X(2019)04-0144-05

**doi:**10.3969/j.issn.1673-629X.2019.04.029

## Synch-playback Technology of Multi-channel Airborne Videos Based on C/S Mode

HAO Zhao, NIE Yao-jia

(Chinese Flight Test Establishment, Xi'an 710089, China)

**Abstract:**In flight test, airborne video data is an important part of flight test data. At present, airborne recording video data is a single M2T file. A single video file can be replayed at one time using current video player software. In order to solve the problem of the synch-playback of multi-channel airborne videos, we propose a synch-playback scheme of multi-channel videos based on C/S mode. The server performs video file read control, and the client performs video decoding and display. Firstly, the basic concept of FFMPEG is introduced. Then the overall design of the system is discussed, and the design and implementation of the server and the client are emphasized. The system is designed in C++ language, and the server uses TCP/IP protocol to interact with the client. The modular and multi-threaded ideas are used in the software design to improve processing efficiency and ensure scalability of function. At present, the software has been successfully applied to synch-playback of multi-channel videos of multiple types' task. The application shows that the system works steadily and can effectively improve the efficiency of flight test.

**Key words:**C/S; multi-channel videos; synch-playback; FFMPEG; SDL

## 0 引言

飞行试验是验证航空产品设计指标要求和检验航空产品质量改进提高航空产品性能进行航空新理论和航空新技术研究的重要手段。在飞行试验中,视频数据能够全面、直观、时效、客观地反映人机交互动作,是试飞工程师给出航空武器装备鉴定/定型试飞结论的重要依据<sup>[1]</sup>。目前机载各路视频数据以 TS 流<sup>[2-3]</sup>的形式记录为独立的 M2T 文件,飞行结束后,通过地面卸载设备将视频数据下载到磁盘阵列进行存储。课题

人员将关心的视频数据下载到本地,然后采用现有的视频播放软件(像暴风影音等)即可进行播放。但是现有的视频播放软件只能实现单路视频数据回放,在某些关键试飞科目中,课题人员需要对多个视频画面进行同步对比分析。如果采用多个视频软件分别对各路视频进行回放,无法准确定位且过程繁琐。

基于以上分析,提出了一种基于 C/S 模式<sup>[4-5]</sup>的多路视频数据同步回放方案,减少了视频数据下载到本地的环节。服务器端对多路视频数据进行逐帧读

收稿日期:2018-05-12

修回日期:2018-09-14

网络出版时间:2018-12-20

基金项目:国家国防科技工业国防基础科研项目计划重点项目(JCKY2016205B006)

作者简介:郝朝(1990-),男,工程师,硕士,研究方向为飞行试验软件开发和数据处理。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20181220.1049.044.html>

取,然后对视频数据进行重新组包以 TCP/IP 协议<sup>[6-8]</sup>发送给客户端。客户端接收到视频数据后解包,将每路视频数据送给解码器进行解码,解码后的 YUV<sup>[9-10]</sup>数据通过 SDL<sup>[11-12]</sup>进行显示,从而实现多路视频数据的同步回放。

1 FFMPEG 简介

FFMPEG<sup>[13-14]</sup>是一个开源且跨平台的音视频流方案,是自由软件中最完备的一套多媒体支持库,具备高可移植性和编解码质量,为音视频转换、解码以及流化提供了完整的解决方案。FFMPEG 包括 libavcodec、libavdevice、libavfilter、libavformat、libavutil、libpostproc、libswresample、libswscale 共 8 类库文件。libavcodec 包含全部 FFMPEG 音频/视频编解码库,相关数据结构包括 AVFormatContext、AVCodecContext、AVCodec、AVPacket、AVFrame 与 AVPicture 等。libavformat 实现了流媒体协议(udp、rtsp、rtmp、rtsp 等),媒体容器(mp4、AVI、FLV 等)和基本的 I/O 访问。libavutil 是一个实用程序库,包含了安全的移动字符串函数、随机数生成器、数据结构、附加数学功能、加密和多媒体相关的功能。libavfilter 是一个通用的音视频后处理库。libavdevice 提供用于从采集和渲染到常见多媒体输入/输出设备的通用框架,并支持多个输入和输出设备。libswresample 实现音频的重采样和混音。libswscale 实现颜色格式的转换和缩放。

2 总体设计

系统采用 C/S 设计模式,服务器端以 TCP/IP 协

议与客户端进行数据交互,保证连接与数据传输的可靠性。服务器在 2333 端口进行监听。客户端发送请求建立连接实现数据交互。系统功能结构如图 1 所示。服务器分为命令解析模块、文件读取模块、组包模块和数据发送模块。命令解析模块主要实现接收客户端命令并解析,完成暂停、继续和进度跳转功能。文件读取模块实现采用 FFMPEG 对视频文件进行逐帧读取。组包模块实现对各路视频数据进行重新组包以便客户端进行分路解码。客户端分为解包模块、视频解码模块、图像显示模块和人机交互模块。解包模块实现接收服务器数据并解析,得到一帧视频数据并送给相应的解码器。视频解码模块实现采用 FFMPEG 对一帧视频数据进行解码,得到 YUV 数据。图像显示模块实现采用 SDL 对 YUV 图像数据进行显示。人机交互模块主要实现响应界面操作,将暂停、继续、进度跳转等指令发送给服务器。

- 服务器运行流程如下:
- (1)网络初始化,监听客户端连接请求;
  - (2)当有客户端发送请求时,将该客户端的信息加入到客户端列表中,针对该客户端开辟单独的线程进行数据交互;
  - (3)当接收到客户端发送的文件信息时,首先初始化 FFMPEG,然后打开对应的各路文件,将视频配置信息发送给客户端;
  - (4)循环逐帧读取视频数据并组包,发送给对应的 SOCKET;
  - (5)当有客户端断开连接时,将该客户端信息从客户端列表中剔除,并关闭对应的线程。

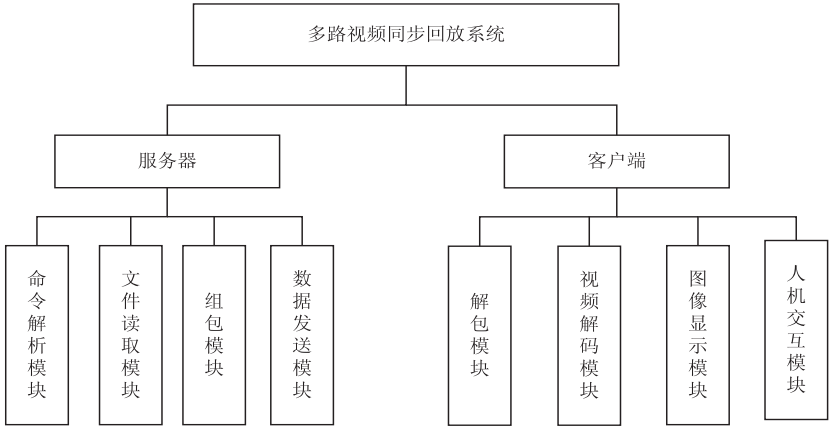


图 1 系统功能结构

客户端运行流程如下:

- (1)网络初始化,初始化 SDL;
- (2)向服务器发送请求,建立数据连接;
- (3)将需要回放的飞机号、日期、飞行架次等发送给服务器;
- (4)接收服务器数据并解包,如果为视频配置信

息,则对每路 FFMPEG 进行初始化,否则得到一帧数据和当前时间;

- (5)将一帧视频图像送给 FFMPEG 进行解码,得到 YUV 数据;
- (6)采用 SDL 对 YUV 图像数据进行显示。

系统运行流程如图 2 所示。

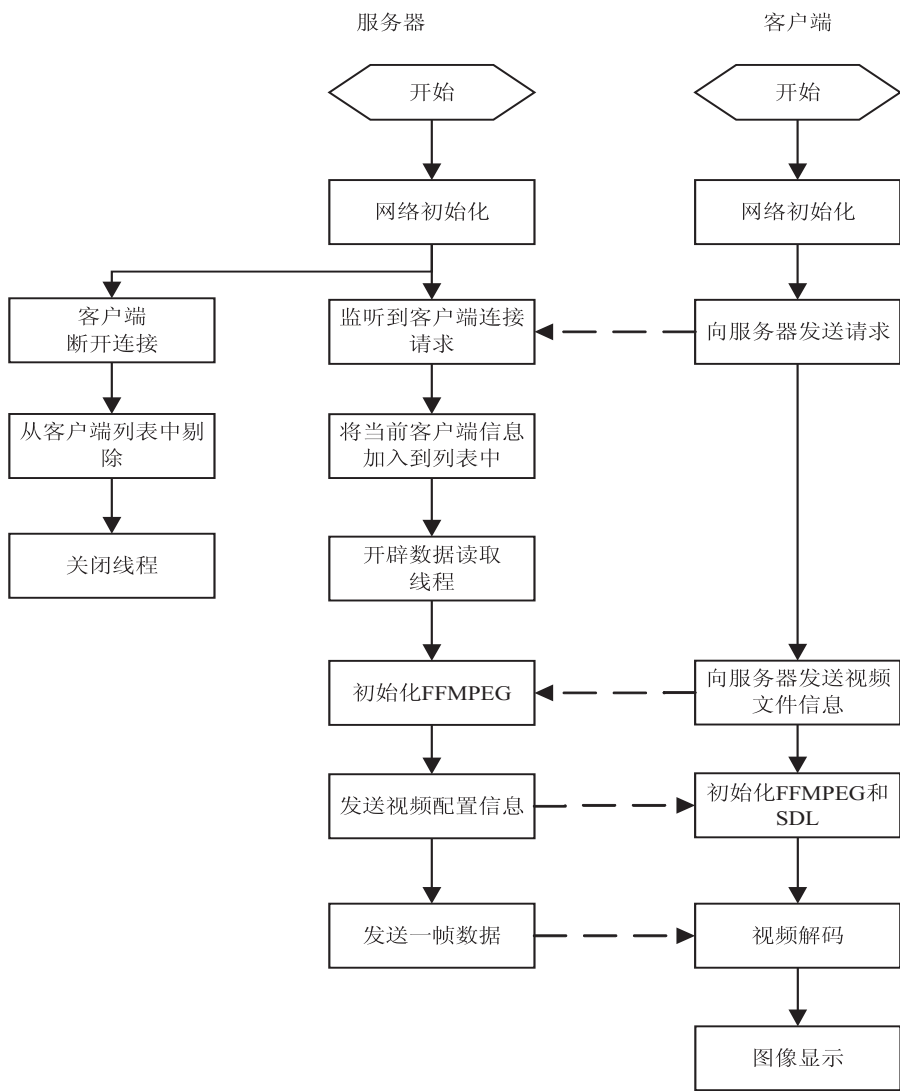


图 2 系统运行流程

3 软件实现

3.1 服务器端设计

针对多路视频同步回放需求,定义了结构体用来表示客户端的相关信息:

```
struct Client
{
    SOCKET s; //数据交互套接字
    char cmd[10]; //命令
    HANDLE hThread; //线程句柄
    CString strFile[4]; //视频数据路径,最多四路
    CFFMPEG [4]; //每一路对应的解码器
}
```

TCP 初始化流程:采用 WSASStartup 的初始化 Winsock;创建套接字;采用 bind 绑定套接字;调用 listen 进行监听。采用非阻塞的异步套接字 WSAAsyncSelect()实现网络通信,对网络事件采用基于消息的异步存取策略,能够方便地处理网络通信。当接收到数据时,会触发 FD\_READ 消息;当有客户端请求时,

会触发 FD\_ACCEPT 消息,生成唯一的 SOCKET 用来进行数据传输,当客户端断开连接时,会触发 FD\_CLOSE 消息,通过对客户端列表进行遍历查找,如果找到与当前客户端对应的 SOCKET,则从客户端列表中剔除。

服务器与客户端数据交互协议包括视频文件配置与视频数据发送两种。视频文件配置信息发送协议定义如下:【同步字】【第几路】【总秒数】【图像宽】【图像高】【编码格式】。同步字为“FEGH1234”,其他各字段均为 2 个字节,总共为 18 个字节。视频数据发送协议定义如下:【第几路】【当前秒数】【视频数据】。多路视频同步播放可选择帧同步和时间戳同步两种策略,帧同步就是每次依次读取多路视频的一帧数据发送给客户端进行解码显示,时间戳同步就是以第一路视频时间戳为基准,其他几路采用 av\_seek\_frame 实现相应时间戳的视频帧定位。

总时间计算方法为: pFormatCtx→duration/AV\_ TIME\_BASE。当前时间以第一路视频为基准,计算方

法为:当前视频数据包的DTS减去该视频文件的起始时间的差经过转换就是当前视频帧的对应秒数,即:  
 $T_n = (\text{packet} \rightarrow \text{dts} - \text{pFormatCtx} \rightarrow \text{streams}[\text{videoIndex}] \rightarrow \text{start\_time}) * \text{pFormatCtx} \rightarrow \text{streams}[\text{videoIndex}] \rightarrow \text{time\_base.num} / \text{pFormatCtx} \rightarrow \text{streams}[\text{videoIndex}] \rightarrow \text{time\_base.den}$ 。  
 接收到客户端发送的进度跳转为 $T_d$ ,则采用`av_seek_frame`进行帧跳转时,需要进行时间转换,计算公式为: $T_r = \text{pFormatCtx} \rightarrow \text{streams}[\text{videoIndex}] \rightarrow \text{start\_time} + \text{av\_rescale}(T_d, \text{pFormatCtx} \rightarrow \text{streams}[\text{videoIndex}] \rightarrow \text{time\_base.den}, \text{pFormatCtx} \rightarrow \text{streams}[\text{videoIndex}] \rightarrow \text{time\_base.num})$ 。

为了充分利用系统资源,服务器采用多线程并发<sup>[15-17]</sup>技术。主线程与客户端进行指令交互,各个客户端独立的线程实现多路视频数据的读取与发送。主线程控制子线程的挂起与恢复。服务器接收到客户端指令后完成解析,当有客户端发出进度跳转的指令时,采用`av_seek_frame`将各路文件指针移动到对应时间点的视频帧;当有客户端发出暂停或继续的指令时,采用`SuspendThread`或`ResumeThread`实现挂起或者恢复对应的线程。

### 3.2 客户端设计

客户端TCP初始化流程为:采用`WSAStartup`的初始化Winsock;创建套接字;采用`connect`进行连接;客户端与服务器交互的命令包括:(1)视频文件信息,向服务器发送“FILE#飞机#日期#架次”字符串;(2)暂停,向服务器发送“Stop”指令;(3)继续,向服务器发送“GoOn”指令;(4)进度跳转,向服务器发送“MOVE#时间”字符串。

为了避免视频解码与显示的耗时导致视频数据包接收丢失,客户端采用多线程并发技术。客户端以用户UI界面操作为主线程,数据接收、视频解码与图像显示为子线程,主线程进行暂停、拖动进度条等操作。数据接收子线程接收到数据后,数据包存入FIFO队列。视频解码线程从队列中提取一帧视频进行解码,解码完成后向图像显示子线程发送消息。

FFMPEG解码视频文件的流程如下:

(1)初始化FFMPEG。

`av_register_all()`

`avformat_network_init()`;

(2)打开对应文件。

`avformat_open_input(&pFormatCtx, filepath, NULL, NULL)`;

(3)找到音视频流信息。

`avformat_find_stream_info(pFormatCtx, NULL)`;

(4)找到对应格式的视频解码器。

`pCodec = avcodec_find_decoder(pCodecCtx →`

`codec_id)`;

(5)读取一帧数据。

`av_read_frame(pFormatCtx, packet)`;

(6)打开对应格式的视频解码器。

`avcodec_open2(pCodecCtx, pCodec, NULL)`;

(7)对一帧视频数据采用相应的解码器进行解码。

`avcodec_decode_video2(pCodecCtx, pFrame, &got_picture, packet)`;

在服务器端,接收到客户端发送的视频文件信息后,调用步骤1~4完成视频数据文件读取,初始化完成后,循环调用步骤5即可实现依次读取一帧图像。在客户端执行步骤1完成初始化,然后接收服务器发送的视频文件配置信息,对解码所需的`pCodecCtx`进行初始化,然后调用步骤6打开解码器。循环接收到一帧视频数据后,调用步骤7即可实现视频解码。

SDL(simple directmedia layer)是一套基于C语言的跨平台多媒体开发库,提供了多种控制音视频输入与输出的函数。SDL显示YUV数据流程如下:

(1)初始化SDL。

`SDL_Init(SDL_INIT_VIDEO | SDL_INIT_TIMER)`;

(2)创建SDL显示窗口。

`screen = SDL_CreateWindowForm(m_hwnd)`;

(3)基于窗口创建渲染器。

`sdlRenderer = SDL_CreateRenderer(screen, -1, 0)`;

(4)创建纹理用于显示YUV数据。

`sdlTexture = SDL_CreateTexture(sdlRenderer, SDL_PIXELFORMAT_IYUV, 1, width, height)`;

(5)设置纹理的像素数据。

`SDL_UpdateTexture(sdlTexture, NULL, pFrameYUV → data[0], pFrameYUV → linesize[0])`;

(6)将纹理数据复制给渲染目标。

`SDL_RenderCopy(sdlRenderer, sdlTexture, NULL, NULL)`;

(7)显示画面。

`SDL_RenderPresent(sdlRenderer)`;

客户端在接收到服务器发送的视频配置信息后,执行步骤1~4实现SDL初始化显示设置,调用步骤5~7即可实现对YUV图像的显示。

## 4 应用效果

实际应用效果如图3所示。选择飞机、日期与架次,点击开始按钮,与服务器建立连接,接收服务器视频数据。点击暂停/继续按钮、拖动进度条,向服务器发送指令,实现播放进度控制。应用效果表明该系统



能够实现机载多路视频的同步回放。



图 3 实际应用效果

## 5 结束语

为了解决试飞机载多路视频数据同步回放的问题,设计了基于 C/S 模式的多路视频同步回放系统。服务器端与客户端以 TCP/IP 协议进行数据交互,服务器端采用 FFMPEG 进行多路视频文件读取与组包发送,客户端接收服务器一帧视频数据,采用 FFMPEG 进行解码并通过 SDL 进行显示。该技术已经应用于多个型号的飞行试验多路视频同步回放中,实现暂停、继续和控制进度的功能,正确性和可靠性得到验证,满足了多路视频同步回放需求,为保障型号试飞的高效顺利进行发挥重要作用。

### 参考文献:

- [1] 张杰,邹强,晏晖.机载多路视频 PCM 遥测传输技术[J].计算机与数字工程,2013,41(5):805-807.
- [2] 朱明海.流媒体服务器 TS 流封装的实现及流控的研究[D].北京:北京邮电大学,2011.
- [3] 李叶.基于 TS 流的数字电视播出系统的研究与实现[D].长沙:中南大学,2014.
- [4] 杨赞国,高敬惠.基于 C/S 模式的网络信息管理系统设计与实现[J].微计算机信息,2005,21(7-3):27-29.
- [5] 李云云.浅析 B/S 和 C/S 体系结构[J].科学之友,2011(2):6-8.
- [6] YOUM B J, PARK J. TCP/IP protocol over IEEE-1394 network for real-time control applications[J]. IFAC Proceedings Volumes, 2005, 38(1): 37-42.

- [7] 于宏亮. TCP 与 UDP 的原理及其在网络编程中的区别[J].计算机与网络,2007(22):186.
- [8] 罗亚非.基于 TCP 的 Socket 多线程通信[J].电脑知识与技术,2009,5(3):563-565.
- [9] MUKHERJEE J, LANGB M K, MITRA S K. Demosaicing of images obtained from single-chip imaging sensors in YUV color space[J]. Pattern Recognition Letters, 2005, 26(7):985-997.
- [10] 李亚南,杨亮,李波.一种基于 YUV 颜色空间的匹配跟踪算法[J].机电产品开发与创新,2017,30(4):89-91.
- [11] 孟成,陈亚. FFmpeg+SDL 实时播放摄像机视频设计[J].产业与科技论坛,2017,16(17):57-58.
- [12] 李科,李璐,兰时勇.基于 FFmpeg 和 SDL 实现多路实时流变换及播放[J].计算机技术与发展,2014,24(4):65-68.
- [13] 辛长春,姜小平,吕乃光.基于 FFmpeg 的远程视频监控系统编解码[J].电子技术,2013(1):3-5.
- [14] ZENG Hao, FANG Yuan. Implementation of video transcoding client based on FFMPEG[J]. Advanced Materials Research, 2013, 756-759:1748-1752.
- [15] COURBIN P, LUPU I, GOOSSENS J. Scheduling of hard real-time multi-phase multi-thread (MPMT) periodic tasks[J]. Real-Time Systems, 2013, 49(2):239-266.
- [16] 郝文化. Windows 多线程编程技术与实例[M].北京:中国水利水电出版社,2005.
- [17] 郝朝,药红红,山寿.基于 OSG 的飞行器实时监控软件设计[J].计算机技术与发展,2017,27(4):21-24.