文章编号:1007-1423(2019)22-0047-04 DOI:10.3969/j.issn.1007-1423.2019.22.010

基于 FFmpeg 和 SDL 的视频流播放存储研究综述

邓正良

(广东粤电阳江海上风电有限公司,阳江 529500)

摘要:

随着网络带宽的增加和视频压缩技术的发展,多媒体在互联网上越来越受欢迎,基于互联网媒体流传输的需求也越来越高,视频和音频正在逐渐取代传统的信息传输方式。为了确保音视频流稳定传输并实现远端播放存储,使用高效的编解码与显示工具库实现该功能尤为重要。介绍 FFmpeg 与 SDL 的重要数据结构及 FFmpeg 的解码过程,综述目前主流的基于 FFmpeg 和 SDL 来实现视频流播放存储的方法。

关键词:

视频播放; 解码; FFmpeg; SDL

0 引言

在当今的互联网时代,随着网络视频的兴起与流媒体通信技术的不断提高、完善,多媒体数据传输成为了日常生活中不可或缺的一部分。网络视频传输依靠其快捷、即时和分享性等优势,已经成为当今环境下的主宰,越来越多的平台需要凭借网络视频传输实现其功能,例如视频直播、在线观影和周界安防监控等[1]。不同的视频服务提供商使用不同类型的传输协议、媒体容器格式以及音视频编解码标准。传输方式包括 HTTP、RTMP、RTSP 等多种媒体传输协议[2];媒体容器格式包括 FLV、MP4、AVI、TS、RMVB 等;视频编码标准[3]包括 H.264、H.265、MPEG4、MPEG2 等[4]。为了能够支持不同的传输协议、媒体容器格式以及音视频编解码标准,市场上一般常用 FFmpeg 和 SDL 作为主要的工具库实现视频播放的功能。FFmpeg[5]作为一个开放的多媒体框架,支持多种传输协议、媒体容器格式以及音视频编解码标准,能够通过其封装的数据结构来存储从多媒体数据中提取的信息,有效解决视频数据的编解码;SDL[6]全称为 Simple DirectMedia Layer,它作为一套开源应用工具开发库,依靠其简洁的接口能够支持音视频、事件、绘图等功能,应用于游戏研发、多媒体影像、图像处理等领域。基于 FFmpeg 和 SDL 工具库,能够实现高效地实现视频流播放存储。

1 FFmpeg解码技术

1.1 FFmpeg数据结构

FFmpeg 结构体功能概述[7]如下:

(1)AVFormatContext:作为输入、输出相关信息的一个容器,封装了格式上下文结构体,对相关视频文件的封装格式信息进行存储。

(2)AVPacket:在读媒体源文件和写输出文件时都需要用于存储每一帧压缩编码后的视频数据。

(3)AVFrame:存储每一帧解码后的采样数据。

(4)AVStream:音视频流与对该结构体对应。

(5)AVInputForma:封装格式与该结构体对应。

(6)AVCodecContext:它作为编码器关联结构,对视音视频编解码相关信息进行了存储。

(7)AVCodec:音视频编解码器与该结构体对应。

上述结构体[8]之间的关系如图 1 所示。

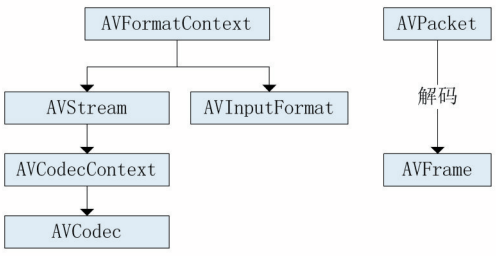

图1 FFmpeg结构体间的关系

1.2 FFmpeg常用函数

在利用 FFmpeg 进行解码的过程中，主要会用到以下相关函数：

（1）av_register_all()：全注册函数，该函数的作用是初始化工具库。

（2）avformat_open_input()：输入源打开函数，该函数的作用是打开输入视频文件。

（3）avformat_find_stream_info()：流找寻函数，该函数的作用是获取视频信息，视频数据包含了媒体数据中的流信息。

（4）av_find_decoder()：编解码器找寻函数，该函数的作用是查找解码器，包含了 id 和 encoder 这两个参数。

（5）avcodec_open2()：解码器打开函数，该函数的作用是打开解码器，初始化 AVCodecContext 结构。

（6）av_read_frame()：帧数读取函数，该函数的作用是将编码数据从输入数据中读取出来。

（7）avcodec_decode_video2()：视频解码函数，该函数的功能是解码压缩数据。解码视频流数据包结构体，调用上述提及的帧数读取函数获取流，并判断其是否为视频流，若为视频流则执行解码操作。

1.3 FFmpeg解码模块

FFmpeg 解码包含了主要的三个解码模块[9]，分别为解码协议模块、解码格式模块和解码视频和音频模块。

（1）解码协议模块

媒体流输入协议数据的协议类型和状态（例如 HTTP、RTMP、RTSP 等类型的传输协议）均被存储于 AVIOContext、URLProtocol 和 URLContext 中，但由于上述协议中存在对媒体流进行控制的信令数据，该数据会对输入协议数据产生影响，所以要将其抹去。解码协议模块的功能是对输入数据进行过滤，剔除协议信令数据，将包含协议信息的输入数据转换为媒体容器格式数据。

（2）解码格式模块

剔除协议信令数据并完成协议解码的媒体容器格式数据包含了音视频两部分数据，解码格式模块的作用就是对媒体容器格式数据进行拆分，独立出音频编码数据和视频编码数据这两部分，并将解码后的两部分数据分别放入特定格式的文件中。

该模块主要与 AVFormatContext 和 AVInputFormat 结构体相关，AVFormatContext 存储包含在媒体容器格式中的信息；AVInputFormat 存储输入媒体流媒体容器格式的类型。

（3）解码音视频模块

经由格式解码拆分独立的音视频两部分数据，仍需要进行进一步操作才能得到未压缩的数据。解码视频和音频模块的作用就是将视频编码数据和音频编码数据分别解码为未经压缩的视频和音频原始数据。经过解码处理后的视频数据被转换为未压缩的像素数据，而音频数据被转换为未压缩的音频采样数据。

该模块主要与 AVStream、AVCodecContext 和 AVCodec 结构体相关，AVStream 存储了音频和视频流信息；AVStream 存储了 AVCodecContext，用于保存音视频流解码信息；AVCodecContext 存储了 AVCodec，用于保存音视频频解码器信息。

2 SDL显示技术

2.1 SDL数据结构

（1）SDL_Window：该结构表示“窗口”。

（2）SDL_Renderer：该结构表示“渲染器”。

（3）SDL_Texture：该结构表示“纹理”。

（4）SDL_Rect：该结构表示“矩形结构”。

基于 SDL 数据结构，视频显示原理如图 2 所示。

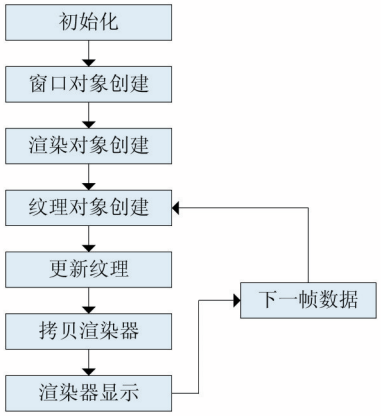

图2 视频显示原理

2.2 SDL显示常用函数

SDL 主要通过以下八个函数实现显示操作：

（1）SDL_Init()：系统初始化函数，该函数对整个系统进行初始化操作。

（2）SDL_CreateWindow()：窗口创建函数，该函数

创建了窗口 SDL_Window 结构体变量。它的主要作用是分别是为 SDL_Window 结构体分配内存、设置窗口的宽高与坐标位置以及创建窗口。

(3)SDL_CreateRenderer():渲染器创建函数,该函数创建了渲染器结构体变量。

(4)SDL_CreateTexture():纹理创建函数,该函数创建了纹理 SDL_Texture 结构体变量。它的主要作用分别是检查输入参数合理性、为 SDL_Texture 纹理结构分配内存以及创建纹理。

(5)SDL_UpdateTexture():纹理更新函数,该函数主要提供四部分功能,分别是设置相关纹理数据、检查输入参数合理性、处理特殊格式以及更新纹理。

(6)SDL_RendererCopy():渲染拷贝函数,该函数的功能是把纹理数据复制给渲染器,主要完成检查输入参数合理性以及复制纹理给渲染对象这两部分操作。

(7)SDL_RendererPresent():渲染显示函数,该函数的功能是显示已复制给渲染对象的图像。

(8)SDL_Quit():系统退出函数,该函数的作用是退出 SDL 系统,释放相关占用资源。

3 视频流播放存储

基于 FFmpeg 和 SDL 的视频流播放存储的实现方式如图 3 所示[10]。

视频流播放的实现步骤主要分为取流、解码和显示三部分组成。

(1)视频流播放

①读取视频流地址的配置文件信息,根据配置信息从流地址中读取实时视频的数据。

②通过 FFmpeg 初始化,获取媒体源后,寻找视频流的相关信息,寻找解码器,并打开解码器,读取视频帧,判断是否获取到数据包,存储压缩编码的数据,解码每一帧视频数据后,再存储每一帧解码后的数据,回到解码器进行下一帧的处理,直到无法获取到数据包为止。

③通过 SDL 创建窗口,渲染器和纹理,将一帧帧解码后的视频贴图显示,实现视频的播放。

(2)视频流存储

视频流存储的实现步骤主要分为创建、取流、读帧和存储四部分组成。

①指定扩展名,输入存储文件的名称,创建文件进行保存,若视频文件名称已存在,提示是否覆盖。

②读取视频流地址的配置文件信息,根据配置信息从流地址中读取实时视频的数据。

③通过 FFmpeg 初始化用于输出结构体,为视频流创建 Stream 通道。获取媒体源后,寻找视频流的相关信息,寻找并打开解码器。地址的打开方式选择读写状态,调用头函数创建函数完成视频文件头的写入。循环读取并存储每一帧数据,判断是否获取到数据包。

④通过写视频文件尾,关闭上下文结构体,释放输出结构体,完成视频存储。

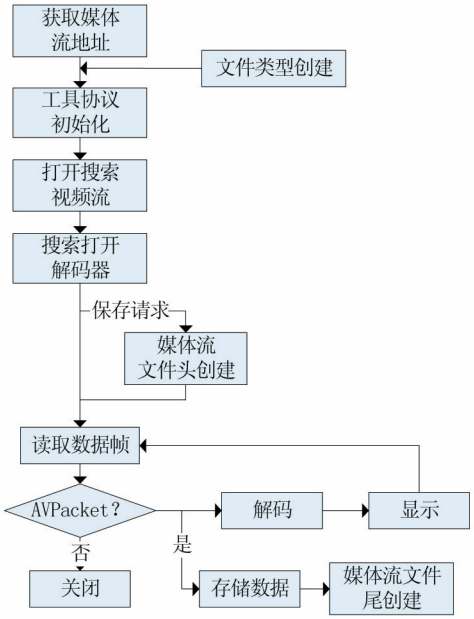

图3 视频流播放存储过程

4 结语

本文介绍了基于 FFmpeg 和 SDL 工具库,实现视频流播放的方式,阐述 FFmpeg 与 SDL 的常用函数及各结构体间的关系,并给出视频流播放存储的实现过程。在实际应用中,可以根据支持的不同种传输协议、媒体容器格式和音视频编解码标准,调整码率、帧数,实现不同需求的实时视频流播放存储,提高在不同情况下的媒体流播放效果。

参考文献：

[1]林国庆. 浅析互联网视频技术的发展与应用[J]. 电脑知识与技术，2019(07).

[2]全鸣. 流媒体传输方式及相关技术[J]. 科技风，2016(02).

[3]毕厚杰. 新一代视频压缩编码标准——H. 264/AVC[M]. 人民邮电出版社，2009.

[4]Jang Wei. Research on Video Transcoding Techniques from H. 264 to HEVC[D]. Hangzhou: Zhejiang University，2013:6-10.

[5]FFmpeg[EB/OL]. http://ffmpeg. org.

[6]SDL[EB/OL]. http://www. libsdl. org.

[7]ZENG Hao，ZHANG Zhi-yong，SHI Lul-in. Research and Implementation of Video Codec Based on FFmpeg[C]. 2016 International Conference on Network and Information Systems for Computers，2016.

[8]辛长春，娄小平，吕乃光. 基于 FFmpeg 的远程视频监控系统编解码[J]. 电子技术，2013. 01.

[9]何圆圆，何凯. 基于 FFmpeg 的 H. 264 视频解码器的研究与实现[J]. 电脑知识与技术，2012. 35.

[10]刘文华. 基于 FFmpeg 和 SDL 的多路视频播放器设计与实现[J]. 漳州职业技术学院学报，2016. 01.

作者简介：

邓正良(1985-)，男，本科，研究方向为电气自动化控制方向，E-mail: dengzl1573@126. com

收稿日期：2019-05-30　　修稿日期：2019-06-11

Summary of Research on Video Streaming Playback and Storage Based on FFmpeg and SDL

DENG Zheng-liang

(Guangdong Yude Yangjiang Offshore Wind Power Co.，Ltd.，Yangjiang 529500)

Abstract:

With the increase of network bandwidth and the development of video compression technology, multimedia is becoming more and more popular on the Internet. The demand for Internet-based media streaming is also increasing. Video and audio are gradually replacing traditional information transmission methods. In order to ensure stable transmission of audio and video streams and remote playback storage, it is especially important to use an efficient codec and display tool library. Introduces the important data structure of FFMPEG and SDL and the decoding process of FFmpeg, summarizes the current mainstream methods of video stream playback and storage based on FFmpeg and SDL.

Keywords:

Video Playback; Decoding; FFmpeg; SDL