

基于FFMPEG的视频水印系统设计

宋燕燕,秦 军

(中国传媒大学南广学院,江苏 南京 211172)

摘 要:数字媒体文件的复制与传输越来越方便,使得数字媒体版权保护面临空前挑战,而数字视频作为最主要的一种媒体形式,如何对大量的数字视频进行保护成为迫切需要解决的问题。数字版权保护包括数字水印、加密等关键技术。通过调用FFMPEG的解码库和SDL显示,在解码后的YUV像素数据的基础上实现视频、图片的水印叠加,重点研究视频水印系统的模块化设计。从系统功能需求出发,利用模块化的设计思想,将水印系统分为输入模块、处理模块和输出模块,为不同编码格式的视频水印算法提供一个通用的仿真平台。针对视频水印系统的高实时性要求,实现接收、解码、水印叠加及显示叠加后的YUV数据,提高视频水印的实时性,为视频的版权保护提供一种方法。

关键词:水印;编码;FFMPEG;视频;SDL

中图分类号:TP37

文献标识码:A

文章编号:1673-629X(2018)04-0144-04

doi:10.3969/j.issn.1673-629X.2018.04.030

Design of Video Watermarking System Based on FFMPEG

SONG Yan-yan, QIN Jun

(Nanguang College of Communication University of China, Nanjing 211172, China)

Abstract: The replication and transmission of digital media files is more and more convenient, which makes the digital media copyright protection under unprecedented challenges. The digital video as the main form of media, how to protect a large number of digital video is an urgent problem to be solved. The key technologies of digital copyright protection include digital watermarking and encryption and so on. Through the use of decodes of FFMPEG and SDL display, the watermark overlay of video and picture is implemented based on the decoded YUV pixel data. We focus on the modular design of video watermarking system. Starting from the requirements of system function, the watermarking system is divided into input module, processing module and output module. A common simulation platform is provided for the video watermarking algorithm for different encoding formats. For the high real-time requirements of the video watermarking system, the receiving, decoding, watermarking overlay and the display of the YUV data are implemented, which can improve the real time of video watermarking and provide a method for the copyright protection of video.

Key words: watermarking; coding; FFMPEG; video; SDL

0 引言

计算机与网络信息技术的快速发展,促使当今社会进入了网络信息时代,各个领域的数据和信息急剧增多。数字媒体的安全问题也极大地制约了当前信息化进程,数据的版权保护问题日益凸显,因此数字水印技术应运而生。数字水印技术是将一些标识信息直接嵌入在数字媒体元素中,且不影响原载体的使用价值。数字水印是保护信息安全、实现防伪溯源、版权保护的有效办法^[1]。当前的媒体形式中,最能吸引眼球的是视频文件,如何对视频文件进行有效保护已经成为十分迫切的研究课题。视频的格式有很多种,所以,相同

的水印算法在不同种视频中,并不一定兼容。文中设计的系统的意义在于,可以根据不同的编码过程在程序中加以调整,使得在同一个平台上进行多种格式视频水印的嵌入和提取,避免了给不同格式的视频单独调用编解码库构建仿真平台的麻烦。实现了一种基于FFMPEG的视频水印系统,使得各种编码格式的视频水印算法都能在同一个仿真平台通用。

1 相关技术

FFMPEG是一套可以用来记录、转换数字音频、视频,并能将其转化为流的开源计算机程序。它提供了

收稿日期:2017-05-13

修回日期:2017-09-12

网络出版时间:2017-12-05

基金项目:江苏省高校自然科学研究面上项目(15KJD520007);校级科研培育项目(2016KYPY054)

作者简介:宋燕燕(1978-),女,硕士,讲师,从事计算机应用的教学与研究。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20171205.1434.120.html>

录制、转换以及编解码功能的完整解决方案^[2-5],包含了非常先进的音频/视频编解码库 libavcodec。为了保证高可移植性和编解码质量,libavcodec 里很多编解码方式都是从头开发的^[6-9]。

FFMPEG 支持 MPEG,DivX,MPEG -4,AC3,DV,FLV 等 40 多种编码,以及 AVI、MPEG、OGG、Matroska、ASF 等 90 多种解码,涵盖了多种音视频格式。FFMPEG 也支持常见的图像色彩空间,并且在 libavswscale 中定义了色彩空间转换的相关函数,实现了各种色彩模式的相互转换^[10-12]。

SDL (simple direct-media layer) 是一套开放源代码的跨平台多媒体开发库,使用 C 语言编写。SDL 提供了数种控制图像、声音、输出、输入的函数,让开发者只要用相同或相似的代码就可以开发出跨多个平台 (Linux、Windows、Mac OS X 等) 的应用软件。目前 SDL 多用于开发游戏、模拟器、媒体播放器等多媒体应用领域^[13-15]。

2 系统方案

视频水印系统主要分为三个模块:输入模块、处理模块和输出模块,各模块之间通过调用各个接口进行交互,使得功能更加完善,使用起来更加契合^[11]。

1. 输入模块。

(1) 内存管理。

内存管理的主要功能就是将选中的视频文件信息写入内存,并且通过一系列的操作将视频数据提供给 FFMPEG 的解码器。

(2) 解码。

解码主要是对视频解码,文中对音频信息不作处理。视频解码主要的内容就是之前存入内存的数据信息通过 FFMPEG 的解码器进行解码,解码完成后送给 SDL。并且利用 SDL 进行 YUV 数据显示。

(3) 渲染纹理。

负责把利用 FFMPEG 解码获得的 YUV 数据渲染到纹理中,再拷贝到渲染器,实现视频成功显示。

2. 处理模块。

利用 FFMPEG 中的过滤器的各种方法,给视频加“特效”,而文中的特效就是图片水印。FFMPEG 的 libavfilter 类库提供了大量的视音频过滤器,种类繁多,有很多现成的 filter 可以直接使用,也给大家研究视频播放器带来了许多便捷。文中是在 FFMPEG 的 libavfilter 类库的基础上完成一个水印叠加器,利用 Visual Studio 2015 平台加以实现。

可以将一张透明背景的图片作为水印叠加到一个视频文件上。需要注意的是,其叠加工作是在解码后的 YUV 像素数据的基础上完成的。程序支持使用

SDL 显示叠加后的 YUV 数据,也可以将叠加后的 YUV 输出成文件。

3. 输出模块。

利用 FFMPEG 中封装好的代码,实现视频的输出。系统方案如图 1 所示。

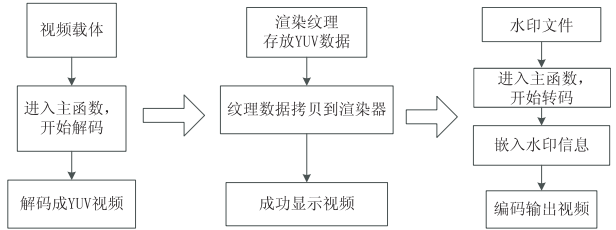


图 1 系统方案

3 视频流解码

每个视频都有自己的封装格式,而所谓封装格式,也就是视频的类型。比如:MP4、TS 等。也就是说,在每个视频的封装格式中,都包含此视频的各种信息,比如视频流和音频流等。对于有封装格式的视频,需要先从中提取出视频流,然后通过解码器解码。例如解码 TS 格式的视频文件,就是“TS→H. 264 码流→YUV”^[4]。视频流解码流程如图 2 所示。

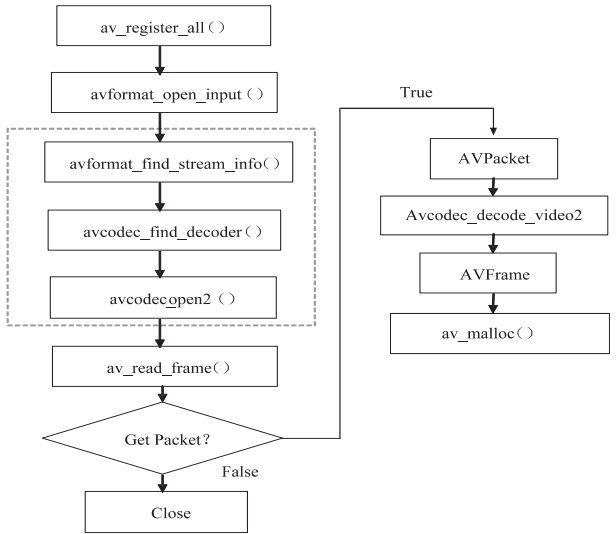


图 2 视频流解码流程

(1) 首先设定函数 av_register_all() 进行初始化,再设定 avformat_open_input() 方法打开文件,获取视频文件的封装格式信息。其中需要判断打开是否成功,如果不成功则显示“无法打开视频文件”。

(2) 使用 for 循环,判断流信息中哪个是视频流信息,哪个是音频流信息;找到视频流后,寻找解码器;找到解码器后,使用 avcodec_open2() 打开解码器。

(3) 定义方法 av_read_frame() 获取封装格式 AVPacker 信息,再使用 Avcodec_decode_video2() 方法解码 AVPacker 信息。若没有获取到 AVPacker 信息,则停止运行。

(4)若获取到了 AVPacker 信息,则使用 av_malloc()方法将 YUV 格式信息从内存中取出,填充到 pAV-FrameYUV 中。

4 水印添加

视频文件经过解码、渲染等一系列工序之后,要将水印文件添加到视频文件上。FFMPEG 自带的类库中含有大量的过滤器,这些过滤器可以将一些文件信息转码。

AVFilter 的基本处理流程如下:

- (1) avfilter_register_all():注册。
- avfilter_graph_alloc():分配一个 AVFilterGraph。
- (2) 创建 AVFilterContext。
- avfilter_graph_create_filter():创建 filter。
- (3) 重复以上两步,创建一个 source,一个 sink。

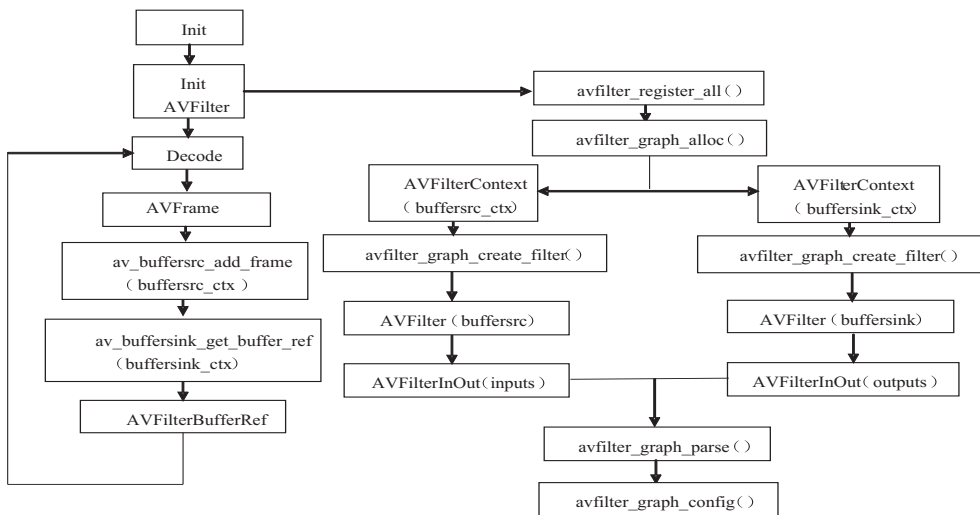


图 3 视频水印添加流程

5 运行结果

基于 FFMPEG 的视频水印系统已完成,系统运行环境:Windows 2007 系统;平台:Visual Studio 2015。

首先在 Visual Studio 中运行系统,如图 4 所示,可以看到左部的图片是默认图片,点击“打开文件”,选择一个视频文件进行添加,而文中选择了一部后缀名为 .ts 的视频进行添加,视频添加后左部默认图片会变成视频文件的首帧画面。

点击“播放”按钮,视频可以正常播放,而点击“暂停”按钮,视频文件会停在当前帧,点击“停止”按钮,视频会停止播放并且恢复到默认图片。

实现一张水印图片,点击“选择水印”按钮,选中该图片,选中后会呈现于界面上,用户可以查看是否选择正确。

点击“加水印”按钮,系统会自动将图片水印加入视频文件,并且添加完毕会自动导出到项目文

avfilter_graph_parse_ptr():将一串通过字符串描述的 Graph 添加到 AVFilterGraph 中。

avfilter_graph_config():检查 FilterGraph 的配置。

av_buffersrc_add_frame():向 FilterGraph 中加入一个 AVFrame。

av_buffersink_get_frame():从 AVFilterGraph 中取出一个 AVFrame。

(4)重复以上两步,加入一个 AVFrame 处理 AV-Frame,取出 AVFrame。

以上就是 AVFilter 的处理流程,其他特效通过 avfilter_graph_parse_ptr(filter_graph, filters_descr, &inputs, &outputs, NULL)中 filters_descr 这个字符串来决定,调用 FFMPEG 提供的方法和传入了符合格式的过滤字符串。

具体流程如图 3 所示。

件,导出的视频格式类型是 .yuv。并且在添加完成后,会有弹框提示,提醒用户已经完成水印添加,并且已经导出。



图 4 运行结果

6 结束语

从数字媒体元素版权保护需求出发探讨了视频水印系统设计方案,利用 FFMPEG 编解码库解决视频水

印算法中的编解码问题,实现 H. 264 的视频解码功能,包括 ts、mp4 等视频格式。提出了 FFMPEG 转码图形用户界面的具体实现方法,介绍了 FFMPEG 转码加密模块的流程、函数调用关系、具体加密算法的实现和视频播放模块的具体实现。利用模块化思想,为不同编码格式的视频水印算法提供一个通用平台,对视频的版权保护起到了积极作用,具有一定的现实意义和参考价值。

参考文献:

[1] CHANG X, WANG W, ZHAO J, et al. A survey of digital video watermarking [C]//Seventh international conference on natural computation. [s. l.]:[s. n.], 2011:61-65.

[2] 任 严,韩 臻,刘 丽. 基于 FFMPEG 的视频转换与发布系统[J]. 计算机工程与设计, 2007, 28 (20): 4962-4963.

[3] 王 彤. 基于 FFmpeg 的 H. 264 解码器实现[D]. 大连:大连理工大学, 2011.

[4] CHENG Yun, LIU Qingtang, ZHAO Chengling, et al. Design and implementation of mediaplayer based on FFmpeg [J]. Software Engineering and Knowledge Engineering, 2012, 2: 867-874.

[5] ZENG Hao, FANG Yuan. Implementation of video transcoding client based on FFMPEG [J]. Advanced Materials Research, 2013, 756-759:1748-1752.

(上接第 143 页)

[6] 刘延华,陈国龙,吴瑞芬. 基于云模型和 AHP 的网络信息系统可生存性评估[J]. 通信学报, 2014, 35(8): 107-115.

[7] 杨姗媛,朱建明. 基于 TOPSIS 的信息安全风险评估应用研究[J]. 现代管理科学, 2014(2): 24-26.

[8] KHANMOHAMMADI K, HOUMB S H. Business process-based information security risk assessment [C]//Fourth international conference on network and system security. Washington, DC, USA: IEEE Computer Society, 2010: 199-206.

[9] FIRESMITH D G. Common concepts underlying safety, security, and survivability engineering [R]. [s. l.]:[s. n.], 2003.

[10] KARABACAK B, SOGUKPINAR I. ISRA M; information security risk analysis method [J]. Computers & Security, 2005, 24(2): 147-159.

[6] 辛长春,姜小平,吕乃光. 基于 FFmpeg 的远程视频监控系系统编解码[J]. 电子技术, 2013(1): 3-5.

[7] 张国庆. 基于 FFmpeg 的视频转码与保护系统的设计与实现[D]. 武汉:华中师范大学, 2011.

[8] 李芳芳,苏凯雄. 基于 FFmpeg 的 H. 264 格式转换器的设计与实现[J]. 电视技术, 2016, 40(7): 32-35.

[9] 胡 聪,周 甜,唐璐丹. 基于 FFMPEG 的跨平台视频编解码研究[J]. 武汉理工大学学报. 2011, 33(11): 139-142.

[10] 吴 岳,施惠娟. 基于 FFMPEG 的视频水印系统[J]. 电子设计工程, 2013, 21(23): 185-187.

[11] 胡 成,任平安,李文莉. 基于 Android 系统的 FFmpeg 多媒体同步传输算法研究[J]. 计算机技术与发展, 2011, 21(10): 85-87.

[12] LI Chengbo, JIANG Hong, WILFORD P, et al. Video coding using compressive sensing for wireless communications [C]//WCNC 2011. [s. l.]: IEEE, 2011: 2077-2082.

[13] 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.

[14] 汪俊杰,王志明. 基于 SDL 的 H. 264 流媒体播放系统[J]. 计算机系统应用, 2013, 22(12): 51-54.

[15] DUAN H, NG B P, CHONG M S S, et al. Applications of the SRV constraint in broadband pattern synthesis [J]. Signal Processing, 2008, 88(4): 1035-1045.

[11] SENDI A S, JABBARIFAR M, SHAJARI M, et al. FEMRA: fuzzy expert model for risk assessment [C]//Fifth international conference on internet monitoring and protection. Washington, DC, USA: IEEE Computer Society, 2010: 48-53.

[12] RHINOW F, CLEAR M. Scargos: towards automatic vulnerability distribution [C]//International conference on security and cryptography. [s. l.]: IEEE, 2015: 369-376.

[13] Forum of Incident Response and Security Teams (first.org). Common vulnerability scoring system CVSS v3.0 specification [S/OL]. 2015. <https://www.first.org/cvss/cvss-v30-specificationv1.7.pdf>.

[14] 肖 晖,张玉清. Nessus 插件开发及实例[J]. 计算机工程, 2007, 33(2): 241-243.

[15] 温 涛. 安全漏洞危害评估研究暨标准漏洞库的设计与实现[D]. 西安:西安电子科技大学, 2016.