# [总结]FFMPEG视音频编解码零基础学习方法



在CSDN上的这一段日子,接触到了很多同行业的人,尤其是使用FFMPEG 进行视音频编解码的人,有的已经是有多年经验的"大神",有的是刚开扩 学者。在和大家探讨的过程中,我忽然发现了一个问题:在"大神"和初学者之间好像有一个不可逾越的鸿沟。"大神"们水平高超,探讨着深奥的问题; 还停留在入门阶段。究竟是什么原因造成的这种"两极分化"呢?最后,我发现了问题的关键:FFMPEG难度比较大,却没有一个循序渐进,由简单到约 程。现在网上的有关FFMPEG的教程多半难度比较大,不太适合刚接触FFMPEG的人学习;而且很多的例子程序编译通不过,极大地打消了学习的积 己在刚开始学习FFMPEG的时候也遇到了很大的困难。为了帮助更多的人快速成为"大神",我想总结一个学习FFMPEG的方法,方便大家循序渐进的: FFMPEG.

PS: 有不少人不清楚 "FFmpeg" 应该怎么读。它读作 "ef ef em peg"



#### 0. 背景知识

本章主要介绍一下FFMPEG都用在了哪里(在这里仅列几个我所知的,其实远比这个多)。说白了就是为了说明: FFMPEG是非常重要的。

使用FFMPEG作为内核视频播放器:

Mplayer, ffplay, 射手播放器,暴风影音, KMPlayer, QQ影音... 使用FFMPEG作为内核的Directshow Filter:

ffdshow, lav filters...

使用FFMPEG作为内核的转码工具:

ffmpeg,格式工厂...

事实上,FFMPEG的视音频编解码功能确实太强大了,几乎囊括了现存所有的视音频编码标准,因此只要做视音频开发,几乎离不开它。

对于完全没有视音频技术背景的人来说,在学习FFmpeg之前最好先了解一下几种最基本的视音频数据的格式,可以参考下面的文章:

[总结]视音频编解码技术零基础学习方法

视音频数据处理入门: RGB、YUV像素数据处理

视音频数据处理入门: PCM音频采样数据处理

视音频数据处理入门: H.264视频码流解析

视音频数据处理入门: AAC音频码流解析

视音频数据处理入门: FLV封装格式解析

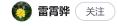
视音频数据处理入门: UDP-RTP协议解析

### 1. ffmpeg程序的使用 (ffmpeg.exe, ffplay.exe, ffprobe.exe)

### 【视频资源】

本文中第1,2章是FFmpeg编程最基础的内容。这部分的内容我在给大二同学代课的时候录制成了视频,有时间的话可以看一下《基于 FFmpeg + SDL 器的制作》课程的视频。

本章主要介绍一下ffmpeg工程包含的三个exe的修









ffmpeg的官方网站是: http://ffmpeg.org/

编译好的windows可用版本的下载地址(官网中可以连接到这个网站,和官方网站保持同步): http://ffmpeg.zeranoe.com/builds/

该网站中的FFMPEG分为3个版本: Static, Shared, Dev。

前两个版本可以直接在命令行中使用,他们的区别在于:Static里面只有3个应用程序:ffmpeg.exe,ffplay.exe,ffprobe.exe,每个exe的体积都很大, 已经被编译到exe里面去了。Shared里面除了3个应用程序:ffmpeg.exe, ffplay.exe, ffprobe.exe之外,还有一些DII,比如说avcodec-54.dII之类的。 的exe体积很小,他们在运行的时候,到相应的DII中调用功能。

Dev版本是用于开发的,里面包含了库文件xxx.lib以及头文件xxx.h,这个版本不包含exe文件。

打开系统命令行接面,切换到ffmpeg所在的目录,就可以使用这3个应用程序了。

## 1.1 ffmpeg.exe

ffmpeg是用于转码的应用程序。

一个简单的转码命令可以这样写:

将input.avi转码成output.ts,并设置视频的码率为640kbps

ffmpeg -i input.avi -b:v 640k output.ts

具体的使用方法可以参考: ffmpeg参数中文详细解释

详细的使用说明(英文): http://ffmpeg.org/ffmpeg.html

## 1.2 ffplay.exe

ffplay是用于播放的应用程序。

一个简单的播放命令可以这样写:

播放test.avi

ffplay test.avi

具体的使用方法可以参考: ffplay的快捷键以及选项

详细的使用说明 (英文): http://ffmpeg.org/ffplay.html

## 1.3 ffprobe.exe

ffprobe是用于查看文件格式的应用程序。

这个就不多介绍了。

详细的使用说明 (英文): http://ffmpeg.org/ffprobe.html

### 2. ffmpeg库的使用:视频播放器

本章开始介绍使用ffmpeg的库进行开发。

### 2.1 ffmpeg库的配置

从http://ffmpeg.zeranoe.com/builds/网站上

- 1.下载Dev版本,里面包含了ffmpeg的xxx.h头文件以及xxx.lib库文件。
- 2.下载Shared版本,里面包含了ffmpeg的dll文件。
- 3.将这两部分文件拷贝到VC工程下面就可以了

注:可能会出现问题,参见:FFMPEG 库移植到 VC 需要的步骤

如果不想自己手动配置,可以下载已经配置好的.











## 2.2 最简单的视频播放器

学习文章《100行代码实现最简单的基于FFMPEG+SDL的视频播放器》中的代码,这是ffmpeg做视频播放器最简单的代码了,是我自己精简出来的, 简化了,每一行都很重要。

原版是基于SDL1.2的视频播放器,后来更新了基于SDL2.0的最简单的视频播放器:最简单的基于FFMPEG+SDL的视频播放器 ver2 (采用SDL2.0)

上述播放器使用libavformat和libavcodec两个类库完成了视频的解码工作。实际上解码工作只需要libavcodec就可以了。因此更新了一个"纯净"的解码 器只使用libavcodec完成解码工作: 最简单的基于FFmpeg的解码器-纯净版 (不包含libavformat)

ffmpeg的函数介绍: ffmpeg函数介绍

注1:播放视频或音频数据的时候会用到SDL。有关SDL可以参考:SDL介绍

SDL参考文档: SDL GUIDE 中文译本

注2: 如果想查看解码后的数据,需要用到 YUV播放器: YUV播放器源代码或YUV Player Deluxe都可以

### 2.3 相关结构体的研究

ffmpeg的结构体之间的关系参考文章: FFMPEG中最关键的结构体之间的关系

结构体中每个变量的分析,参考文章:

FFMPEG结构体分析: AVFrame

FFMPEG结构体分析: AVFormatContext FFMPEG结构体分析: AVCodecContext FFMPEG结构体分析: AVIOContext FFMPEG结构体分析: AVCodec FFMPEG结构体分析: AVStream FFMPEG结构体分析: AVPacket

## 3. ffmpeg库的使用: 音频播放器

### 3.1 最简单的音频播放器

学习文章《最简单的基于FFMPEG+SDL的音频播放器》中的代码,和最简单的视频播放器一样,这是最简单的音频播放器,每一行代码都很重要。

原版是基于SDL1.2的音频播放器,后来更新了一个基于SDL2.0的最简单的音频播放器:最简单的基于FFMPEG+SDL的音频播放器 ver2 (采用SDL2

注:如果想要查看解码后的数据 (PCM数据),需要用到Audition。

### 4. ffmpeg库的使用:一个真正的播放器——ffplay

### 4.1 真正的播放器

ffplay流程图如文章《FFplay源代码分析:整体流程图<mark>》 所示。ffplay代码比较复杂,但是其核心代码和《100行代码实现最简单的基于FFMPEG+SDL</mark> 器》是一样的。可以两个工程结合着学习。

ffplay代码简介资料:如何用FFmpeg编写一个简单播放器

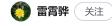
ffplay使用说明: ffplay的快捷键以及选项

ffplay已经移植到VC下的工程:ffplay\_vc2005 (别人做的,质量很不错)

ffplay移植到MFC下的工程,包含了简单的图形界面和一些控制按钮:ffplay播放器移植VC的工程:ffplay for MFC

上述软件的代码简介: ffplay for mfc 代码备忘

ffplay.c函数结构简单分析:ffplay.c函数结构简单<sup>八+c</sup>







(专栏)

## 5. ffmpeg库的使用:编码

### 5.1 编码

ffmpeg编码我自己研究的不是很多,可以参考文章: 使用FFmpeg类库实现YUV视频序列编码为视频

上面那篇文章是用的类库比较旧,新版类库的的使用可以参考下面几篇文章。

图像的编码可以参考: 最简单的基于FFMPEG的图像编码器 (YUV编码为JPEG)

音频的编码可以参考: 最简单的基于FFMPEG的音频编码器 (PCM编码为AAC)

视频的编码可以参考: 最简单的基于FFMPEG的视频编码器 (YUV编码为H.264)

HEVC (H.265) 视频编码可以参考:最简单的基于FFmpeg的视频编码器-更新版 (YUV编码为HEVC(H.265))

上述编码器使用libavformat和libavcodec两个类库完成了视频的编码工作。实际上编码工作只需要libavcodec就可以了。因此更新了一个"纯净"的编码 器只使用libavcodec完成编码工作: 最简单的基于FFmpeg的编码器-纯净版 (不包含libavformat)

### 5.2 转码

转码实际上是先解码然后编码。

不进行转码,只进行封装格式转换的程序可参考:最简单的基于FFMPEG的封装格式转换器(无编解码)

转码程序可参考: 最简单的基于FFMPEG的转码程序

比较复杂的转码程序可以参考ffmpeg.c,它移植到MFC下的工程:ffmpeg转码器移植VC的工程:ffmpeg for MFC

ffmpeg.c函数结构简单分析: ffmpeg.c函数结构简单分析 (画图)

## 6. ffmpeg源代码分析

通晓了ffmpeg库的使用以后,可以看一下ffmpeg的源代码。注意ffmpeg的源代码只有在linux下才能编译,在windows下可以使用MinGW进行编译。推 Eclipse查看ffmpeg的源代码。

有一个很完整的ffmpeg源代码的分析文档: ffdoc

FFmpeg的库函数源代码分析文章列表如下:

#### 【架构图】

FFmpeg源代码结构图 - 解码

FFmpeg源代码结构图 - 编码

【通用】

FFmpeg 源代码简单分析: av\_register\_all()

FFmpeg 源代码简单分析: avcodec\_register\_all()

FFmpeg源代码简单分析:内存的分配和释放 (av\_malloc()、av\_free()等)

FFmpeg 源代码简单分析: 常见结构体的初始化和销毁 (AVFormatContext, AVFrame等)

FFmpeg 源代码简单分析: avio\_open2()

FFmpeg 源代码简单分析: av\_find\_decoder()和av\_find\_encoder()

FFmpeg 源代码简单分析: avcodec\_open2()

FFmpeg 源代码简单分析: avcodec\_close()

【解码】

图解FFMPEG打开媒体的函数avformat\_open\_input

FFmpeg 源代码简单分析: avformat\_open\_inpu











FFmpeg 源代码简单分析: avformat\_find\_stream\_info()

FFmpeg 源代码简单分析: av\_read\_frame()

FFmpeg 源代码简单分析: avcodec decode video2()

FFmpeg 源代码简单分析: avformat\_close\_input()

【编码】

FFmpeg 源代码简单分析: avformat\_alloc\_output\_context2()

FFmpeg 源代码简单分析: avformat\_write\_header()

FFmpeg 源代码简单分析: avcodec\_encode\_video()

FFmpeg 源代码简单分析: av\_write\_frame()

FFmpeg 源代码简单分析: av\_write\_trailer()

【其它】

FFmpeg源代码简单分析:日志输出系统 (av\_log()等)

FFmpeg源代码简单分析: 结构体成员管理系统-AVClass

FFmpeg源代码简单分析: 结构体成员管理系统-AVOption

FFmpeg源代码简单分析: libswscale的sws\_getContext()

FFmpeg源代码简单分析: libswscale的sws\_scale()

FFmpeg源代码简单分析: libavdevice的avdevice\_register\_all()

FFmpeg源代码简单分析: libavdevice的gdigrab

【脚本】

FFmpeg源代码简单分析: makefile

FFmpeg源代码简单分析: configure

偏底层的libavcodec的源代码分析文章列表如下:

【解码-libavcodec H.264 解码器】

FFmpeg的H.264解码器源代码简单分析: 概述

FFmpeg的H.264解码器源代码简单分析:解析器 (Parser)部分

FFmpeg的H.264解码器源代码简单分析:解码器主干部分

FFmpeg的H.264解码器源代码简单分析: 熵解码 (EntropyDecoding) 部分

FFmpeg的H.264解码器源代码简单分析:宏块解码(Decode)部分-帧内宏块(Intra)

FFmpeg的H.264解码器源代码简单分析:宏块解码(Decode)部分-帧间宏块(Inter)

FFmpeg的H.264解码器源代码简单分析:环路滤波 (LoopFilter) 部分

【解码-libavcodec HEVC 解码器】

FFmpeg的HEVC解码器源代码简单分析: 概述

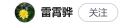
FFmpeg的HEVC解码器源代码简单分析:解析器 (Parser)部分

FFmpeg的HEVC解码器源代码简单分析:解码器主干部分

FFmpeg的HEVC解码器源代码简单分析: CTU解码 (CTUDecode) 部分-PU

FFmpeg的HEVC解码器源代码简单分析: CTU解码 (CTU Decode) 部分-TU

FFmpeg的HEVC解码器源代码简单分析:环路滤波 (LoopFilter)



## 7.FFmpeg其它几个类库的使用

## 7.1.libavfilter (加特效)

AVFilter可以给视音频添加各种滤镜效果。有两个例子,一个是给视频添加水印:

最简单的基于FFmpeg的AVfilter例子(水印叠加)

另一个是给YUV数据加特效:

最简单的基于FFmpeg的AVfilter的例子-纯净版

## 7.2.libavdevice (读设备)

AVDevice可以读取电脑的多媒体设备的数据,或者输出数据到指定的多媒体设备上。

直接使用ffmpeg.exe命令行工具的文章: FFmpeg获取DirectShow设备数据(摄像头,录屏)

编程方面做了2个有关的例子:

读取摄像头: 最简单的基于FFmpeg的AVDevice例子 (读取摄像头)

屏幕录制: 最简单的基于FFmpeg的AVDevice例子(屏幕录制)

## 7.3.libswscale (图像拉伸, 像素格式转换)

Swscale类库可以转换像素数据的格式,同时可以拉伸图像的大小。

libswscale的使用示例: 最简单的基于FFmpeg的libswscale的示例 (YUV转RGB)

此外,这个示例还附带了一个程序,用于生成测试图片: 最简单的基于FFmpeg的libswscale的示例附件:测试图片生成工具

## 8.FFmpeg封装格式的处理

使用FFmpeg进行封装格式的处理,主要是通过AVFormat完成。有关封装格式的处理,做了3个例子:

封装格式转换器:最简单的基于FFMPEG的封装格式转换器(无编解码)

视音频分离器简化版 (demuxer-simple) : 最简单的基于FFmpeg的封装格式处理: 视音频分离器简化版 (demuxer-simple)

视音频分离器 (demuxer): 最简单的基于FFmpeg的封装格式处理: 视音频分离器 (demuxer)

视音频复用器 (muxer) : 最简单的基于FFmpeg的封装格式处理: 视音频复用器 (muxer)

## 9.FFmpeg流媒体方面的应用

使用FFmpeg进行流媒体方面的应用,主要是流媒体的发送和接收。

直接使用ffmpeg.exe命令行工具的文章:

FFmpeg发送流媒体的命令 (UDP, RTP, RTMP)

编程方面做了一个例子:

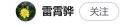
基于FFmpeg的推流器: 最简单的基于FFmpeg的推流器(以推送RTMP为例)

## 10.FFmpeg的其他杂项

使用FFmpeg读写内存 (而非文件) 的例子:

内存播放器: 最简单的基于FFmpeg的内存读写的例子: 内存播放器

内存转码器: 最简单的基于FFmpeg的内存读写的例子: 内存转码器







## 11. ffmpeg在其它平台下的应用

把FFmpeg应用于Android、IOS、Windows Phone的示例程序可以参考:

最简单的基于FFmpeg的移动端例子: Android HelloWorld

最简单的基于FFmpeg的移动端例子: Android 视频解码器

最简单的基于FFmpeg的移动端例子: Android 视频解码器-单个库版

最简单的基于FFmpeg的移动端例子: Android 推流器

最简单的基于FFmpeg的移动端例子: Android 视频转码器

最简单的基于FFmpeg的移动端例子附件: Android 自带播放器

最简单的基于FFmpeg的移动端例子附件: SDL Android HelloWorld

最简单的基于FFmpeg的移动端例子: IOS HelloWorld

最简单的基于FFmpeg的移动端例子: IOS 视频解码器

最简单的基于FFmpeg的移动端例子: IOS 推流器

最简单的基于FFmpeg的移动端例子: IOS 视频转码器

最简单的基于FFmpeg的移动端例子附件: IOS自带播放器

最简单的基于FFmpeg的移动端例子: Windows PhoneHelloWorld

## 12. ffmpeg相关工程的学习

学习完成ffmpeg,还可以了解一下基于ffmpeg的相关的多媒体开源工程,在这里推荐以下几个:

### 12.1 ffdshow

ffdshow是基于ffmpeg的解码器类库libavcodec的DirectShow Filter。广泛安装在PC上。



有关ffdshow的源代码分析文章(更新中):

ffdshow 源代码分析1: 整体结构

ffdshow 源代码分析 2: 位图覆盖滤镜 (对话框部分Dialog)

ffdshow 源代码分析 3: 位图覆盖滤镜 (设置部分Settings)

ffdshow 源代码分析 4: 位图覆盖滤镜(滤镜部分Filter)

ffdshow 源代码分析 5: 位图覆盖滤镜 (总结)

ffdshow 源代码分析 6: 对解码器的dll的封装 (libavcodec)

ffdshow 源代码分析 7: libavcodec视频解码器类(TvideoCodecLibavcodec)

ffdshow 源代码分析 8: 视频解码器类 (TvideoCodecDec)

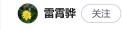
ffdshow 源代码分析 9: 编解码器有关类的总结

## 12.2 LAV filters

LAV Filter是基于ffmpeg的解码器类库libavcodec,以及解封装器类库libavformat的DirectShow Filter。广泛安装在PC上。

有关LAV Filter的源代码分析文章:

LAV Filter 源代码分析 1: 总体结构



**1** 1404 **1** 25073 **2** 916

专栏目

LAV Filter 源代码分析 2: LAV Splitter

LAV Filter 源代码分析 3: LAV Video (1) LAV Filter 源代码分析 4: LAV Video (2)

## 12.3 Mplayer

Mplayer是Linux下使用最广泛的播放器,也有Windows版本的。其中使用了ffmpeg。



有关Mplayer的源代码分析文章:

MPlayer源代码分析

## 12.4 Media Player Classic - HC

现在广为使用很多播放器都是构建于Media Player Classic - HC的基础之上的。



有关Media Player Classic - HC的源代码分析文章:

Media Player Classic - HC 源代码分析 1:整体结构

Media Player Classic - HC 源代码分析 2:核心类(CMainFrame)(1) Media Player Classic - HC 源代码分析 3:核心类(CMainFrame)(2) Media Player Classic - HC 源代码分析 4:核心类(CMainFrame)(3) Media Player Classic - HC 源代码分析 5:关于对话框(CAboutDlg)

Media Player Classic - HC 源代码分析 6: MediaInfo选项卡 (CPPageFileMediaInfo) Media Player Classic - HC 源代码分析 7: 详细信息选项卡 (CPPageFileInfoDetails)

## 12.5 XBMC

XBMC是一个优秀的自由和开源的 (GPL) 媒体中心软件。



### 有关XBMC源代码分析文章:

XBMC源代码分析 1:整体结构以及编译方法 XBMC源代码分析 2:Addons(皮肤Skin)

XBMC源代码分析 3:核心部分 (core) -综述

XBMC源代码分析 4:视频播放器 (dvdplayer) -解码器 (以ffmpeg为例)

XBMC源代码简析 5: 视频播放器 (dvdplayer) -解复用器 (以ffmpeg为例)

XBMC源代码分析 6: 视频播放器 (dvdplayer) -文件头 (以ffmpeg为例)

XBMC源代码分析 7: 视频播放器 (dvdplayer)













### 音视频入门基础知识笔记

音视频<mark>编解码</mark>相关的基础知识点。RGB与YUV表示。预测编码:帧内预测与帧间预测。I帧、P帧、B帧概念。PTS与DTS。GOP序列。H.264编码标准。H.264码流分析。数:

#### 雷神霄骅博士1020篇博客集录

一代流媒体璀璨巨星雷神雷<mark>霄骅</mark>博士终身的1020篇技术博客chm集录。雷神为流媒体技术潜心专研并无私共享他的成果,引导了一代流媒体开发者,开拓了流媒体事业。

916 条评论 您还未登录, 请先 登录 后发表或查看评论

#### 使用FFmpeg做视频解码\_ffmpeg视频解码

[总结]视音频编解码技术零基础学习方法\_雷霄骅的博客-CSDN博客 最简单的基于FFMPEG+SDL的视频播放器 ver2 (采用SDL2.0) 视频编解码四:FFmpeg视频解码流程 一、补

#### FFmpeg简易播放器的实现2-视频播放\_non-interleaved avi

[5]. FFmpeg简易播放器的实现5-音视频同步 1. 视频播放器基本原理 下图引用自 "<mark>雷霄骅视音频编解码</mark>技术<mark>零基础学习方法"</mark>,因原图太小,看不太清楚,故重新制作了一张图片:

### 基于javaweb+mysql的ssm+maven高校后勤报修系统(java+ssm+jsp+mysql)

欢迎私信,一直

基于javaweb+mysql的ssm+maven高校后勤报修系统(java+ssm+jsp+mysql)基于javaweb的SSM+Maven高校后勤报修系统(java+ssm+jsp+mysql)图片存放路径: src\mai

#### FFmpeg教程 (超级详细版)

一、参考资料 通过ffmpeg把图片转换成视频 FFmpeg命令(一)、使用filter\_complex命令拼接视频 FFmpeg 视频处理入门教程给新手的 20 多个 FFmpeg 命令示例 FFmpeg命

### FFmpeg的H.264解码器源代码简单分析:解析器(Parser)部分\_雷霄骅-CSDN博...

FFmpeg的H.264解码器源代码简单分析:解码器主干部分 FFmpeg的H.264解码器源代码简单分析:摘解码(EntropyDecoding)部分 FFmpeg的H.264解码器源代码简单分析:宏

#### 最简单的基于FFMPEG的转码程序\_最简单的ffmpeg電雪骅 leixiaohua...

\*最简单的基于FFmpeg的转码器 \*Simplest FFmpeg Transcoder \* \*雷霄骅 Lei Xiaohua \*leixiaohua1020@126.com \*中国传媒大学/数字电视技术 \*Communication Universi

### FFmpeg源代码结构图 - 解码 热门推荐

雷霄骅(leixiaohua1020)的专

近期研究了一下FFmpeg的内部的源代码。之前对于FFmpeg的研究主要在它的应用层面上,因此制作的很多示例程序都是调用的FFmpeg的API。但是一直感觉这样对FFm

### 《FFmpeg+SDL的视频播放器的制作》学习记录(1):视音频基础知识

雷神博客:<mark>雷霄骅</mark>(leixiaohua1020)的专栏<mark>.雷霄骅\_</mark>CSDN博客-FFMPEG,FFmpeg,视频质量评价领域博主 视频:基于FFmpeg+SDL的视频播放器的制作——<mark>雷霄骅</mark> 一、视频扩

### (ffmpeg)ffmpeg+SDL的简单播放器(雷霄骅)更新版

(ffmpeg)ubuntu18.04安装ffmpeg4.4 ffmpeg版本:4.41.音频播放器 /\*\* \* 最简单的基于FFmpeg的音频播放器 2 \* Simplest FFmpeg Audio Player 2 \* \* <mark>雷霄骅</mark> Lei Xiaohua \*

### FFmpeg源代码简单分析:configure\_雷霄晔

FFmpeg源代码简单分析:av\_register\_all() FFmpeg源代码简单分析:avcodec\_register\_all() FFmpeg源代码简单分析:内存的分配和释放(av\_malloc()、av\_free()等) FFmpeg源

### 深入浅出理解视频编码H264结构

fei的引

https://abson.github.io/2016/11/15/%E6%B7%B1%E5%85%A5%E6%B5%85%E5%87%BA%E7%90%86%E8%A7%A3%E8%A7%86%E9%A2%91%E7%BC%96%E7%A0%81H2(Continuous) and the sum of the continuous and the cont

### FFmpeq入门教程(非常详细)从零基础入门到精通,看完这一篇就够了。

Javachichi的t

FFmpeg全称为Fast Forward Moving Picture Experts Group,于2000年诞生,是一款免费,开源的音视频<mark>编解码</mark>工具及开发套件。它的功能强大,用途广泛,大量用于视划

## 基于FFMPEG 的视频编码(libavcodec, 致敬雷霄骅)\_ffmpeg 雷

Qt与FFmpeg联合开发指南(三)——编码(1):代码流程演示在为了代码简洁、代码中还用到了Qt。先不讲解具体的实现代码。大家先看看我封装后的类的使用方法。下面是一

### 什么是HEVC?解释了高效视频编码,H.265和4K压缩

Mr.Wing的博

什么是HEVC?解释了高效视频编码,H.265和4K压缩 高效视频编码(HEVC),也称为H.265,可以通过蓝光最佳视频压缩<mark>方法</mark>实现两倍的压缩。但它是如何工作的,是否

## 雷霄骅系列博客汇总

mytzs123的t

这两天快速浏览了一下<mark>雷霄骅</mark>的博客,相信看过他博客的人内心都是无比钦佩,说句天妒英才确不为过,可能雷神说过最多的话大概就是,...没有我想要的,打开源代码...

## FFmpeg学习之一 (音视频理论知识)

kvl282889543的

FFmpeg学习之一 (音视频理论知识) 1. 音视频理论知识 1. 音视频理论知识

### ffmpeg视音频解码【一】的SourceCode

多媒体的时代,得多了解点<mark>编解码</mark>的技术才行,而<mark>ffmpeg</mark>为我们提供了一系列多媒体<mark>编解码</mark>的接口,如何用好这些接口达到自己所需要的目的,这也是一门重要的学问。

## ffmpeg Captura

再找开源录屏软件的时候发现Captura,需要依赖ffmpeg,官网连接下载太慢,特地搬运过来,方便大家下载

### FFMPEG视音频编解码零基础学习方法

FFMPEG视音频编解码零基础学习方法 FFMPEG视音频编解码零基础学习方法

### FFMPEG3.4视音频编解码代码库.rar

FFMPEG3.4视音频编解码代码库.rar FFMPEG3.4视音频编解码代码库.rar FFMPEG3.4视音频编解码代码库.rar

### 《FFmpeg视、音频解码》源码+视频文件

Android与FFmpeg: 音、视频文件解码Demo。 音频生



**1**404 **1** 25073 **1** 916



