

[illegible]

参考：

最简单的基于FFmpeg的推流器 (以推送RTMP为例) _ffmpeg推流rtmp,ffmpeg图片推流-C代码类资源-CSDN下载

[使用ffmpeg实现rtmp推流_goldfish3的博客-CSDN博客](#)

注意：

1. 需提前安装ffmpeg，并可以成功使用其提供的第三方依赖包，可见链接：[点这里](#)
2. 运行以下代码前，需提前运行服务器（需推流到服务器）

```

1 #include <stdio.h>
2
3 #define __STDC_CONSTANT_MACROS
4
5 extern "C"
6 {
7     #include <libavformat/avformat.h>
8     #include <libavutil/mathematics.h>
9     #include <libavutil/time.h>
10 };
11
12 int main(int argc, char* argv[])
13 {
14     AVOutputFormat *ofmt = NULL;
15     //Input AVFormatContext and Output AVFormatContext
16     //上下文管理对象
17     AVFormatContext *ifmt_ctx = NULL, *ofmt_ctx = NULL;
18     AVPacket pkt;
19     const char *in_filename, *out_filename;
20     int ret, i;
21     int videoindex=-1;
22     int frame_index=0;
23     int64_t start_time=0;
24     //in_filename = "cuc_ieschool.mov";
25     //in_filename = "cuc_ieschool.mhvv";
26     //in_filename = "cuc_ieschool.ts";
27     //in_filename = "cuc_ieschool.mp4";
28     //in_filename = "cuc_ieschool.h264";
29     in_filename = "./video/input.mp4";//输入URL (Input file URL)
30     //in_filename = "shanghai03_p_h264";
31
32     out_filename = "rtmp://localhost/dash/test";//输出 URL (Output URL) [RTMP]
33     //out_filename = "rtp://233.233.233.6666";//输出 URL (Output URL) [UDP]
34
35     av_register_all();
36     //Network
37     avformat_network_init();
38     //Input
39     if ((ret = avformat_open_input(&ifmt_ctx, in_filename, 0, 0)) < 0) {
40         printf(" Could not open input file.");
41         goto end;
42     }
43     if ((ret = avformat_find_stream_info(ifmt_ctx, 0)) < 0) {
44         printf(" Failed to retrieve input stream information");
45         goto end;
46     }
47
48     for(i=0; i<ifmt_ctx->nb_streams; i++)
49         if(ifmt_ctx->streams[i]->codec->codec_type==AVMEDIA_TYPE_VIDEO){
50             videoindex = i;
51             break;
52         }
53
54     av_dump_format(ifmt_ctx, 0, in_filename, 0);
55
56     //Output
57
58     avformat_alloc_output_context2(&ofmt_ctx, NULL, "flv", out_filename); //RTMP
59     //avformat_alloc_output_context2(&ofmt_ctx, NULL, "mpegts", out_filename); //UDP
60
61     if (!ofmt_ctx) {
62         printf(" Could not create output context\n");
63         ret = AVERRROR_UNKNOWN;
64         goto end;
65     }
66     ofmt = ofmt_ctx->oformat;
67     for (i = 0; i < ifmt_ctx->nb_streams; i++) {
68         //Create output AVStream according to input AVStream
69         AVStream *in_stream = ifmt_ctx->streams[i];
70         AVStream *out_stream = avformat_new_stream(ofmt_ctx, in_stream->codec->codec);
71         if (!out_stream) {
72             printf("Failed allocating output stream\n");
73             ret = AVERRROR_UNKNOWN;
74             goto end;
75         }
76         //Copy the settings of AVCodecContext
77         ret = avcodec_copy_context(out_stream->codec, in_stream->codec);
78         if (ret < 0) {
79             printf("Failed to copy context from input to output stream codec context\n");
80             goto end;
81         }
82         out_stream->codec->codec_tag = 0;
83         if (ofmt_ctx->oformat->flags & AVFMT_GLOBALHEADER)
84             out_stream->codec->flags |= CODEC_FLAG_GLOBAL_HEADER;
85     }
86     //Dump Format-----
87     av_dump_format(ofmt_ctx, 0, out_filename, 1);
88     //Open output URL
89     if (!(ofmt->flags & AVFMT_NOFILE)) {
90         ret = avio_open(&ofmt_ctx->pb, out_filename, AVIO_FLAG_WRITE);
91         if (ret < 0) {
92             printf("Could not open output URL '%s'", out_filename);
93             goto end;
94         }
95     }
96     //Write file header

```

分类专栏



```
98     ret = avformat_write_header(&ofmt_ctx, NULL);
99     if (ret < 0) {
100         printf( "Error occurred when opening output URL\n");
101         goto end;
102     }
103     start_time=av_gettime();
104     while (1) {
105         AVStream *in_stream, *out_stream;
106         //Get an AVPacket
107         ret = av_read_frame(ifmt_ctx, &pkt);
108         if (ret < 0)
109             break;
110         //FIX: No PTS (Example: Raw H.264)
111         //Simple Write PTS
112         if(pkt.pts==AV_NOPTS_VALUE){
113             //Write PTS
114             AVRational time_base=ifmt_ctx->streams[videoindex]->time_base;
115             //Duration between 2 frames (us)
116             int64_t calc_duration=(double)AV_TIME_BASE/av_q2d(ifmt_ctx->streams[videoindex]->r_frame_rate);
117             //Parameters
118             pkt.pts = (double)(frame_index*calc_duration)/(double)(av_q2d(time_base1)*AV_TIME_BASE);
119             pkt.dts=pkt.pts;
120             pkt.duration = (double)calc_duration/(double)(av_q2d(time_base1)*AV_TIME_BASE);
121         }
122         //Important:Delay
123         if(pkt.stream_index==videoindex){
124             AVRational time_base=ifmt_ctx->streams[videoindex]->time_base;
125             AVRational time_base_q={1,AV_TIME_BASE};
126             int64_t pts_time = av_rescale_q(pkt.dts, time_base, time_base_q);
127             int64_t now_time = av_gettime() - start_time;
128             if (pts_time > now_time)
129                 av_usleep(pts_time - now_time);
130         }
131     }
132     in_stream = ifmt_ctx->streams[pkt.stream_index];
133     out_stream = ofmt_ctx->streams[pkt.stream_index];
134     /* copy packet */
135     //Convert PTS/DTS
136     pkt.pts = av_rescale_q(pkt.pts, in_stream->time_base, out_stream->time_base, (AVRounding) (AV_ROUND_NEARLY|
137     (1<AV_ROUND_NEARLY|
138     pkt.dts = av_rescale_q(pkt.dts, in_stream->time_base, out_stream->time_base, (AVRounding) (AV_ROUND_NEARLY|
139     pkt.duration = av_rescale_q(pkt.duration, in_stream->time_base, out_stream->time_base);
140     pkt.pos = -1;
141     //Print to Screen
142     if(pkt.stream_index==videoindex){
143         //printf("Send %d video frames to output URL\n",frame_index);
144         frame_index++;
145     }
146     //ret = av_write_frame(ofmt_ctx, &pkt);
147     ret = av_interleaved_write_frame(ofmt_ctx, &pkt);
148
149     if (ret < 0) {
150         printf( "Error muxing packet\n");
151         break;
152     }
153
154     av_free_packet(&pkt);
155 }
156 //Write file trailer
157 av_write_trailer(&ofmt_ctx);
158 end:
159 avformat_close_input(&ifmt_ctx);
160 /* close output */
161 if (ofmt_ctx && !(ofmt->flags & AVFMT_NOFILE))
162     avio_close(ofmt->pb);
163 avformat_free_context(&ofmt_ctx);
164 if (ret < 0 && ret != AVERRROR_EOF) {
165     printf( "Error occurred.\n");
166     return -1;
167 }
168 return 0;
169 }
170
171
172
```

C++编程FFMpeg (QT5+OpenCV) 实战-实时美颜直播推流 08-22
C++编程FFMpeg实时美颜直播推流实战视频培训课程，本课程基于ffmpeg，qt5，opencv进行实战教学。基于c++编程，掌握录制视频（rtsp和系统相机...
ffmpeg c++代码推流RTSP/RTMP（命令行推流） m0_37400316的博客 5411
FFMPEG c++代码实现推流

1 条评论 MONKEY_D_LUFFY 热评 推h264—卡—卡的，延时设置需要修改 写评论

Linux下C语言实现ffmpeg视频+音频推流 没用的阿鸡的博客 7-26
1.环境需求 ①ffmpeg源码编译 https://trac.ffmpeg.org/wiki/CompilationGuide/Ubuntu官网编译教程 一定要支持h264 ②alsa支持 2.思路 ①ffmpeg调用摄像...
ffmpeg源码优化之推流发送篇_ffmpeg 代码推流_音视频开发老马的博客-CSD... 7-31
环境PC: 推流摄像头: ffmpeg -i dshow -i video="Integrated Camera" -vcodec libx264 -acodec copy -preset:v ultrafast -tune:v zerolatency -f flv rtmp://19...
FFmpeg 代码实现流媒体推流 (RTSP) 热门推荐 m0_60259116的博客 1万+
FFmpeg 代码实现流媒体推流 (RTSP)

c++ ffmpeg结合opencv推流 qq_45620342的博客 95
c++ 利用ffmpeg推流的一个简单的demo实现代码。

FFmpeg/opencv + C++ 实现直播拉流和直播推流(对视频帧进行处理) 8-2
本文主要使用C++ ffmpeg库实现对除去webrtc的视频流进行拉流,而后经过自身的处理,而后通过将处理后的视频帧进行编码,最后进行推流处理。_opencv...

FFmpeg常用推流命令_ffmpeg 推流_音视频开发老马的博客 8-2
二、FFmpeg推流1.推流MP4文件 视频文件地址:Users\xu\Desktop\bangbangbang.mp4 推流地址:rtmp://localhost:1935/rtmplive/home acc:RTMP的...

基于ffmpeg的windows c++程序 05-31
基于ffmpeg的windows c++程序,使用c++语言调用ffmpeg函数库,实现音视频媒体文件信息输出。可以参考博客: http://blog.csdn.net/grafx/article/detail...

FFmpeg/opencv + C++ 实现直播拉流和直播推流 (对视频帧进行处理) weixin_43482623的博客 755
本文主要使用C++ ffmpeg库实现对除去webrtc的视频流进行拉流,而后经过自身的处理,而后通过将处理后的视频帧进行编码,最后进行推流处理。

直播推流 ffmpeg 拉流二次转发 记录_ffmpeg c# 拉流后转发_tnan2522的... 7-24
这种的话我直接就完成了推流到live552 服务器,再用ffmpeg 进行拉流转推倒 live553服务器上 了 哦,对了,ffmpeg 拉流转发的命令是: ffmpeg -i 拉流地址 -vc...

FFmpeg流媒体处理-收流与推流_ffmpeg 推流_音视频开发老马的博客-CSD... 8-2
ffmpeg 推流测试: ffmpeg -re -i tnhaorc.flv -c copy -f flv rtmp://192.168.0.104/live "-re":按视频帧率的速度读取输入 "-c copy" 输出流使用和输入流相同的...

C++封装FFMPEG拉取RTSP的音视频流接口类 09-27
C++封装的FFMPEG拉取音视频流接口类,调用简单,采用回调方式读取数据,工程采用VS2015,希望对需要的人有帮助!

ubuntu 基于ffmpeg和Visual GDB 实现本地推流功能 turetony的专栏 167
ubuntu 下载ffmpeg的相关组件 并且安装安装如下 1 ./configure --prefix=/usr/local/ffmpeg2--enable-shared--enable-libmp3lame --enable-libvorbis --en...

基于ffmpeg的推流示例_ffmpeg实现推流_IT_阿水的博客 8-3



我要出家当道士

码龄7年

暂无认证

312

5617

2939

57万+

原创

周排名

总排名

访问

等级

6805

8250

462

213

1569

积分

粉丝

获赞

评论

收藏



私信

已关注

搜博文文章

热门文章



二叉树遍历（已知前序和后序遍历，求中序遍历的可能序列数）

27527

Android-计算器的实现（界面布局，计算逻辑处理）

25020

C++中string类型insert方法用法集锦

19615

如何更新word文档中的文献引用编号？

18330

VMWare虚拟机启动img文件

14618

最新评论

BP神经网络Python代码实现

liuchujie 1225: 请问博主读取的文件在哪找呀

二叉树遍历 (已知前序和后序遍历, 求中...
看着夕阳敲代码: 最近准备初赛无意间刷到老...
老师这篇文章, 比较好懂, 果断收藏哈哈嘻嘻...
进程信息查看脚本

看着夕阳敲代码: 博主写的非常好, 文章干货满满, 三联+关注! 欢迎作者闲暇之余...
集线器、交换机和网桥三者有什么区别
weixin_46467363: 所以他们的区别和联系是啥?

python实现ftp文件传输
superbeeze: 您好可以再发一下吗, 分享过期了

您愿意向朋友推荐“博客详情页”吗?

😞

😐

😌

😊

😄

强烈不推荐 不推荐 一般般 推荐 强烈推荐

最新文章

进程信息查看脚本

Oracle中Archived redolog的生成

内存映射详细介绍

2023年 11篇

2022年 50篇

2021年 56篇

2020年 93篇

2019年 59篇

2018年 71篇

2017年 5篇

基于ffmpeg的推流示例 基于ffmpeg的推流示例 流媒体(streaming media)是指将一连串的媒体数据压缩后,经过网上分段发送数据,在网上即时传输影音...

ffmpeg C++推流

ffmpeg 推流器代码如下: #include <iostream> using namespace std; //引入头文件 extern "C" { #include "libavformat/avformat.h" //引入时间 #inclu...

FFmpeg入门 - rtmp推流

介绍了怎样用ffmpeg去播放视频,里面用于打开视频流的avformat_open_input函数除了打开本地视频之外,实际上也能打开rtmp协议的远程视频,实现拉流。/...

基于ffmpeg实现的rtmp推流模块

需要创建推流的流媒体地址,然后再创建一个对象启动推流,创建过程如下。RtmpPublishManager* rpm = new RtmpPublishManager("rtmp://192.168...

基于ffmpeg-4.2.1-win-x86推拉流接口库.zip

基于ffmpeg 4.2.1 32位实现的rtmp推拉流接口库,依赖pthread2.9.1、opencv4.1.1、SDL2-2.0.10 包含vs2019的C#测试程序,可以在WINDOWS10 WIN7...

利用ffmpeg实现RTSP, RTMP推流以及保存到本地录像文件

demo目录是简单的调用例子,简单实现了抓取屏幕然后推流到RTSP,RTMP服务器,以及保存到本地。bin目录是已经编译好的二进制文件。有兴趣可查看...

读取H264文件到内存推流到RTMP流媒体服务器

纯C++代码,读取H264文件到内存推流到RTMP流媒体服务器,关键使用libRTMP库封装成RTMP协议格式!方便初学者

基于ffmpeg实现了解析rtsp视频为cv::Mat格式,将Mat数据推送到rtmp服务

基于ffmpeg实现了解析rtsp视频为cv::Mat格式,此时可以对cv::Mat进行处理,然后再将cv::Mat数据推送到rtmp流媒体服务。处理、转码一条龙。另外,工...

C++调用ffmpeg命令推流【详细代码】

我们可以使用ffmpeg命令进行很多工作,也可以在代码中调用ffmpeg指令进行相关操作,例如推流,录屏等,线面介绍C++调用ffmpeg命令进行rtmp推流...

FFMPEG对USB摄像头视频进行UDP推流

FFMPEG对USB摄像头视频进行UDP推流 利用FFMPEG对USB摄像头进行推流直播,采用命令行的方式无论是rtmp还是udp推流,延迟均较为严重,延迟...

C++调用ffmpeg进行rtmp推流

C++调用ffmpeg进行rtmp推流

ffmpeg rtsp推流 c++ 最新发布

回答1: ffmpeg可以用来进行rtsp推流。在C语言中,我们可以使用ffmpeg提供的C API来实现这个功能。首先,我们需要初始化ffmpeg。这可以通过...

“相关推荐”对你有帮助?

😞 非常没帮助

😐 没帮助

😌 一般

😊 有帮助

😄 非常有帮助

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

我要出家当道士 已关注

👍 2 🗨 27 🌟 1 📄 1 🗑 1 专栏目录

Beta

🏠

📝

🔍

📁

🔖

🔖