

版权

1 订阅 1 篇文章 (订阅专栏)

HEVC编码的FLV格式视频 -> [https://download.csdn.net/download/qq\\_36236684/13686337](https://download.csdn.net/download/qq_36236684/13686337)

HEVC是High Efficiency Video Coding的缩写(又称为H.265和MPEG-H第2部分),是一种新的视频压缩标准,用来以替代H.264/AVC编码标准,2013年1月26号,HEVC正式成为国际标准。

最近团队需要编解码FLV封装的HEVC编码的视频，但是FFmpeg因为某种原因，没有原生支持上述封装。因此在网上搜寻之后找到了相关的解决方案，在此记录一下。

### 1. 修改部分，可以使用patch命令直接合到自己的项目中

需要注意的是 `FLV_CODECID_HEVC = 14`, 可以根据实际情况更改, 一般情况下国内使用的都是14, 也有用12的

```

1 commit 35b84e7ff5ea2dd314eb9e14fe9698e3c31bb09
2 Author: xxx <xxx@xxx-xx.cn>
3 Date: Mon Dec 14 11:06:49 2020 +0800
4
5     add h265 dec & enc in flv
6
7 diff --git a/libavformat/flv.h b/libavformat/flv.h
8 index df5ce3d..2b025bb 100644
9 --- a/libavformat/flv.h
10 +++ b/libavformat/flv.h
11 @@ -109,6 +109,7 @@ enum {
12     FLV_CODECID_H264 = 7,
13     FLV_CODECID_REALH263 = 8,
14     FLV_CODECID_MPEG4 = 9,
15 +     FLV_CODECID_HEVC = 14,
16 };
17
18 enum {
19     diff --git a/libavformat/flvdec.c b/libavformat/flvdec.c
20     index 2d89bef..bc0ae18 100644
21     --- a/libavformat/flvdec.c
22     +++ b/libavformat/flvdec.c
23 @@ -36,6 +36,7 @@
24     #include "internal.h"
25     #include "avio_internal.h"
26     #include "flv.h"
27     #include "hevc.h"
28
29     #define VALIDATE_INDEX_TS_THRESH 2500
30
31 @@ -291,6 +292,8 @@ static int flv_same_video_codec(AVCodecParameters *vpar, int flags)
32     return vpar->codec_id == AV_CODEC_ID_VP6A;
33     case FLV_CODECID_H264:
34     return vpar->codec_id == AV_CODEC_ID_H264;
35 + case FLV_CODECID_HEVC:
36 + return vpar->codec_id == AV_CODEC_ID_HEVC;
37     default:
38     return vpar->codec_tag == flv_codecid;
39 }
40 @@ -340,6 +343,11 @@ static int flv_set_video_codec(AVFormatContext *s, AVStream *vstream,
41     par->codec_id = AV_CODEC_ID_MPEG4;
42     ret = 3;
43     break;
44 + case FLV_CODECID_HEVC:
45 + par->codec_id = AV_CODEC_ID_HEVC;
46 + vstream->need_parsing = AVSTREAM_PARSE_NONE;
47 + ret = 3; // not 4, reading packet type will consume one byte
48 + break;
49     default:
50     avpriv_request_sample(s, "Video codec (%x)", flv_codecid);
51     par->codec_tag = flv_codecid;
52 @@ -1149,10 +1157,11 @@ retry_duration:
53
54     if (st->codecpar->codec_id == AV_CODEC_ID_AAC ||
55         st->codecpar->codec_id == AV_CODEC_ID_H264 ||
56 -         st->codecpar->codec_id == AV_CODEC_ID_MPEG4) {
57 +         st->codecpar->codec_id == AV_CODEC_ID_MPEG4 ||
58 +         st->codecpar->codec_id == AV_CODEC_ID_HEVC) {
59         int type = avio_r8(s->pb);
60         size--;
61 -         if (st->codecpar->codec_id == AV_CODEC_ID_H264 || st->codecpar->codec_id == AV_CODEC_ID_MPEG4) {
62 +         if (st->codecpar->codec_id == AV_CODEC_ID_H264 || st->codecpar->codec_id == AV_CODEC_ID_MPEG4 || st->codecpar->codec_id == AV_CODEC_ID_HEVC) {
63             // sign extension
64             int32_t cts = (avio_rb24(s->pb) & 0xff800000) ^ 0xff800000;
65             pts = dts + cts;
66 @@ -1168,7 +1177,7 @@ retry_duration:
67         }
68     }
69     if (type == 0 && (1st->codecpar->extradata || st->codecpar->codec_id == AV_CODEC_ID_AAC ||
70 -         st->codecpar->codec_id == AV_CODEC_ID_H264)) {
71 +         st->codecpar->codec_id == AV_CODEC_ID_H264 || st->codecpar->codec_id == AV_CODEC_ID_HEVC)) {
72         AVDictionaryEntry *t;
73
74         if (st->codecpar->extradata) {
75     diff --git a/libavformat/flvenc.c b/libavformat/flvenc.c
76     index e966c36..3d2de1b 100644
77     --- a/libavformat/flvenc.c
78     +++ b/libavformat/flvenc.c
79 @@ -34,7 +34,7 @@
80     #include "libavutil/opt.h"
81     #include "libavcodec/put_bits.h"
82     #include "libavcodec/aacencstab.h"
83     -
84     #include "hevc.h"
85
86     static const AVCodecTag flv_video_codec_ids[] = {

```

```

87 { AV_CODEC_ID_FLV1, FLV_CODECID_H263 },
88 @@ -46,6 +46,7 @@ static const AVCodecTag flv_video_codec_ids[] = {
89 { AV_CODEC_ID_VP6, FLV_CODECID_VP6 },
90 { AV_CODEC_ID_VP6A, FLV_CODECID_VP6A },
91 { AV_CODEC_ID_H264, FLV_CODECID_H264 },
92 + { AV_CODEC_ID_HEVC, FLV_CODECID_HEVC },
93 { AV_CODEC_ID_NONE, 0 }
94 };
95
96 @@ -486,7 +487,7 @@ static void flv_write_codec_header(AVFormatContext* s, AVCodecParameters* par) {
97     FLVContext *flv = s->priv_data;
98
99     if (par->codec_id == AV_CODEC_ID_AAC || par->codec_id == AV_CODEC_ID_H264
100 -        || par->codec_id == AV_CODEC_ID_MPEG4) {
101 +        || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC) {
102         int64_t pos;
103         avio_w8(pb,
104             par->codec_type == AVMEDIA_TYPE_VIDEO ?
105 @@ -533,7 +534,11 @@ static void flv_write_codec_header(AVFormatContext* s, AVCodecParameters* par) {
106         avio_w8(pb, par->codec_tag | FLV_FRAME_KEY); // flags
107         avio_w8(pb, 0); // AVC sequence header
108         avio_wb24(pb, 0); // composition time
109 -        ff_isom_write_avcc(pb, par->extradata, par->extradata_size);
110 +        if (par->codec_id == AV_CODEC_ID_HEVC) {
111 +            ff_isom_write_hvcc(pb, par->extradata, par->extradata_size, 0);
112 +        } else {
113 +            ff_isom_write_avcc(pb, par->extradata, par->extradata_size);
114 +        }
115     }
116     data_size = avio_tell(pb) - pos;
117     avio_seek(pb, -data_size - 10, SEEK_CUR);
118 @@ -836,7 +841,7 @@ end:
119     AVCodecParameters *par = s->streams[i]->codecpar;
120     FLVStreamContext *sc = s->streams[i]->priv_data;
121     if (par->codec_type == AVMEDIA_TYPE_VIDEO &&
122 -        (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4))
123 +        (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id ==
124         put_avc_eos_tag(pb, sc->last_ts);
125     }
126 }
127 @@ -887,13 +892,13 @@ static int flv_write_packet(AVFormatContext *s, AVPacket *pkt)
128     if (par->codec_id == AV_CODEC_ID_VP6F || par->codec_id == AV_CODEC_ID_VP6A ||
129         par->codec_id == AV_CODEC_ID_VP6 || par->codec_id == AV_CODEC_ID_AAC)
130         flags_size = 2;
131 -    else if (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4)
132 +    else if (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC)
133         flags_size = 5;
134     else
135         flags_size = 1;
136
137     if (par->codec_id == AV_CODEC_ID_AAC || par->codec_id == AV_CODEC_ID_H264
138 -        || par->codec_id == AV_CODEC_ID_MPEG4) {
139 +        || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC) {
140         int side_size = 0;
141         uint8_t *side = av_packet_get_side_data(pkt, AV_PKT_DATA_NEW_EXTRADATA, &side_size);
142         if (side && side_size > 0 && (side_size != par->extradata_size || memcmp(side, par->extradata, side_size) != 0))
143 @@ -956,6 +961,10 @@ static int flv_write_packet(AVFormatContext *s, AVPacket *pkt)
144         if (par->extradata_size > 0 && *(uint8_t*)par->extradata != 1)
145             if ((ret = ff_avc_parse_nal_units_buf(pkt->data, &data, &size)) < 0)
146                 return ret;
147 +        } else if (par->codec_id == AV_CODEC_ID_HEVC) {
148 +            if (par->extradata_size > 0 && *(uint8_t*)par->extradata != 1)
149 +                if ((ret = ff_hevc_annexb2mp4_buf(pkt->data, &data, &size, 0, NULL)) < 0)
150 +                    return ret;
151         } else if (par->codec_id == AV_CODEC_ID_AAC && pkt->size > 2 &&
152             (AV_RB16(pkt->data) & 0xfff0) == 0xfff0) {
153             if (!s->streams[pkt->stream_index]->nb_frames) {
154 @@ -1028,7 +1037,7 @@ static int flv_write_packet(AVFormatContext *s, AVPacket *pkt)
155                 (FFALIGN(par->height, 16) - par->height));
156             } else if (par->codec_id == AV_CODEC_ID_AAC)
157                 avio_w8(pb, 1); // AAC raw
158 -            else if (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4) {
159 +            else if (par->codec_id == AV_CODEC_ID_H264 || par->codec_id == AV_CODEC_ID_MPEG4 || par->codec_id == AV_CODEC_ID_HEVC)
160                 avio_w8(pb, 1); // AVC NALU
161                 avio_wb24(pb, pkt->pts - pkt->dts);
162             }

```

## 2. 编译:

```

1 ./configure --prefix=/usr/local --enable-libx265 --enable-gpl --enable-ffmpeg
2 make
3 make install

```

## 3. 转码为h265:

-i 为输入, -acodec 为音频编码, -vcodec为视频编码

```
ffmpeg -y -i xxx.mp4 -acodec copy -vcodec libx265 xxx.flv
```

## 4. 播放

```
ffplay xxx.flv
```

拉戈蓝人 已关注

👍 5

💬

🌟 16

📁

💬 1

🔖

专栏目录

1 条评论 朽木自雕 热评 持续更新, 加油👍

写评论

FFmpeg hevc codec\_tag兼容问题\_音视频开发老马的博客

8-29

那是因为FFmpeg3.3与最新版本表示codec\_tag的方式改变了(同样是hev1 tag,FFmpeg3.3是0x23,最新版本是MKTAG('h','e','v','1')),虽然都是hev1,但是...

视频文件太大?使用FFmpeg来无损压缩它\_视频极限压缩 github\_音视频开发...

8-24

首先,我们打开FFmpeg的官网<https://ffmpeg.org>然后点击'Download'按钮,进入下载页面,下载对应的版本即可。然后在电脑上安装。Windows还需要将安...

FLV HEVC(H265)

12-14

FLV封装的HEVC编码的测试视频,用于FFmpeg的测试,使用libx265进行编码。是一种网络视频格式,用作串流媒体格式,它的出现有效地解决了视频文...

拉戈蓝人

10  
原创

2万+  
周排名

119万+  
总排名

4万+  
访问

等级

432  
积分

2932  
粉丝

28  
获赞

17  
评论

164  
收藏

私信

已关注

使用Ffmpeg 编解码 FLV的HEVC(H265)格式的视频 解决了你的问题么?  
可以写篇文章记录加深印象哦~

写文章

搜博文文章

🔍

热门文章

Asp .Net 连接SQL Server 的两种方式 ④ 12486

使用Ffmpeg 编解码 FLV的HEVC(H265)格式的视频 ④ 9419

使用STC89C52单片机完成一个温度预警系统 ④ 3643

Ubuntu 启动关闭服务 ④ 3610

deepin 深度系统 更换阿里源 ④ 3430

最新评论

使用STC89C52单片机完成一个温度预警...  
逆风要翻盘: 可以私信吗

使用STC89C52单片机完成一个温度预警...  
拉戈蓝人: 这个肯定是需要先编译, 然后再把板上运行才可以的

使用STC89C52单片机完成一个温度预警...  
逆风要翻盘: 下载到板子上不显示温度呀 [安卓] 实现悬浮通知效果

新手-99: 谢谢, 我试试  
[安卓] 实现悬浮通知效果  
拉戈蓝人: 我这个就可以在其他应用上显示了

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

[安卓] 实现悬浮通知效果

Sourceinsight 豆沙绿背景颜色

deepin 深度系统 更换阿里源

2022年 1篇

2020年 3篇

2019年 4篇

2018年 2篇

<b>FFmpeg修改源码支持H265</b> <b>最新发布</b>	ynshin007的博客	👍 97		
如果出现错误: nasm/yasm not found or too old. Use --disable-x86asm for a crippled build. 请先执行: brew install yasm. 此时我们用 fffmpeg 一个h265...				
<b>FFmpeg的HEVC解码器源代码简单分析-解码器主千部分_ffdecoder.lib_雷霄...</b>		8-28		
FFmpeg HEVC解码器主千部分在整个HEVC解码器中的位置如下图所示。 单击查看更清晰的大图 HEVC解码器主千部分的源代码的调用关系如下图所示...		...		
<b>Python自动生成ffmpeg转码HEVC (X265,H265) 命令_ffmpeg hevc 转码...</b>		8-31		
7. 本程序生成ffmpeg命令并保存在D:\BatchConvert.bat,不会自动转码。 需要手动执行,这点很重要。 接下来是代码 from pathlib import Path from pymedi...		...		
<b>【FFmpeg】支持flv+hevc编解码转码</b>	m0_52257114的博客	👍 2383		
背景 当前FFmpeg支持flv+h264的编解码和转码, 但是并不支持flv+h265 如果需要在FFmpeg中支持flv+h265, 只需要简单的修改代码 源码修改 flv.h diff ...		...		
<b>用于解决ffmpeg不支持flv+h265编解码的文件.zip</b>		09-26		
Video codec hevc not compatible with flv, 用于解决ffmpeg不支持flv+h265的情况。				
<b>FFmpeg的HEVC解码器源代码简单分析-解析器(Parser)部分_雷霄骅的博客...</b>		8-25		
上篇文章概述了FFmpeg的libavcodec中HEVC(H.265)解码器的结构,从这篇文章开始,具体研究HEVC解码器的源代码。本文分析HEVC解码器中解析器(Par...		...		
<b>ffmpeg实战教程(三)音频PCM采样为AAC,视频YUV编码为H264/HEVC</b>		8-28		
av_register_all()注册FFmpeg所有编解码器。 avformat_alloc_output_context2()初始化输出码流的AVFormatContext。 avio_open();打开输出文件。 av...		...		
<b>最简单的基于FFMPEG的视频编码器 (YUV编码为HEVC H 265 )</b>		10-04		
FFMPEG工程浩大, 可以参考的书籍又不是很多, 因此很多刚学习FFMPEG的人常常感觉到无从下手。 因此特地分离出了一个简单的视频编码器供学习之...		...		
<b>FFmpeg的HEVC解码器源代码简单分析: 概述</b>	雷霄骅@xiaohua1020的专栏	👍 17+		
从这篇文章开始, 简单分析记录FFmpeg中libavcodec的HEVC (H.265) 解码器 (HEVC Decoder) 的源代码。 本文综述整个解码器的框架, 后续几篇文章...		...		
<b>怎么编译 ffmpeg 让其支持 hevc(h265)_ffmpeg hevc_freeabc的博客-CSDN...</b>		8-28		
#push stream \$./ffmpeg -re -i 1.mp4 -c:a copy -c:v libx265 -f flv rtmp://192.168.1.1/live/hevc #pull stream \$./ffmpeg rtmp://192.168.1.1/live/hevc good luck!				
<b>基于 FFMPEG 的视频编码器 (libavcodec, 致敬雷霄骅)</b>	lvan 的专栏	👍 2006		
基于 FFMPEG 的视频编码 (libavcodec, 致敬雷霄骅) 本文参考了雷博士的博客: 最简单的基于FFmpeg的视频编码器-更新版 (YUV编码为HEVC(H.26...		...		
<b>ffmpeg h265(1v封装) 转码成 h264 (1v封装)</b>	glw@glw的博客	👍 17+		
ffmpeg h265(1v封装) 转码成 h264 (1v封装) ffmpeg -v -c:dec hevc -i glw265.flv -acodec copy -v:codec libx264 glw264.flv				
<b>用ffmpeg解析H265视频编码</b>	weixin_35755562的博客	👍 2139		
使用ffmpeg 解析 H265 视频编码的方法如下: 下载并安装 ffmpeg。 如果您的计算机上已经安装了 ffmpeg, 则可以跳过此步骤。 打开命令行窗口(Windo...		...		
<b>多种测试视频文件(TS、h264_mp4、h265_mp4、mov、mpg、flv等共计17种格式)</b>		02-22		
多种视频测试文件(TS、h264_mp4、h265_mp4、mov、mpg、flv等共计17种格式)				
<b>用Ffmpeg进行h265编解码的实例</b>		02-28		
这是一个用Ffmpeg进行h265编解码的实例, 集成了demux, 解码, 编码相关的代码。				
<b>ffmpeg5.0 msvc x64 支持flv-h265</b>		01-29		
ffmpeg5.0 msvc x64 支持flv-h265,带开发库和Fplay				
<b>【FFmpeg实战】H265/HEVC&amp;码流结构</b>	C/C++Linux、音视频、DPDK	👍 87		
H265 又被叫做HEVC (全称叫做 HVR Efficiency Video Coding, 高效率视频编码), 它同H264一样也是ITU-T和ISO两个组织共同制定的视频压缩标准, ...		...		
<b>Nginx+FFmpeg 海康、大华NVR实现实时转flv实时预览+录像回放</b>	akai9808的专栏	👍 3186		
工具准备FFmpeg, 流媒体Nginx, 要想实现FLV还需要安装模块nginx-http-flv-module, 这个模块需要编译, 如果是linux环境很方便, 如果是windows环...		...		
<b>解决FFmpeg播放RTSP推送的H265码流报错问题</b>	lrainsa的专栏	👍 906		
RTSP(Real Time Streaming Protocol), 实时流传输协议, 是TCP/IP协议体系中的一个应用层协议。 RTP (Real-time Transport Protocol) 实时传输协议...		...		
<b>ffmpeg之H265解码 热门推荐</b>		👍 2万+		
本文详细介绍如何使用Fmpeg把H265解码成YUV420P格式格式的数据。 首先由于采用VS2017开发, 我禁了一个4996的错误: #pragma error(disable...		...		
<b>ffmpeg解析实时H265/264视频流, 出现Could not find ref with POCXX问题解决方法</b>	u013241583的博客	👍 27+		
这是由于处理的机器性能低, 导致播放流的速度大于解码速度, 怎么解决这种问题呢, 就是提高解码速度, 采取的就是增加解码的线程, 参考文章 FFmp...		...		
<b>linux ffmpeg如何支持flv+h265</b>		07-13		
要在Linux上使用FFmpeg来支持FLV + H.265, 你需要确保你的FFmpeg版本已经编译了x265编码器。 以下是一些步骤供参考: 1. 检查FFmpeg版本: 运...		...		
<b>“相关推荐”对你有帮助么?</b>				
	没帮助			