

原 使用FFMPEG类库分离出多媒体文件中的H.264码流

2013年09月18日 11:15:37 阅读数：27968

在使用FFMPEG的类库进行编程的过程中,可以直接输出解复用之后的的视频数据码流。只需要在每次调用 `av_read_frame()` 之后将得到的视频的AVPacket存为本地文件即可。

经试验,在分离MPEG2码流的时候,直接存储AVPacket即可。

在分离H.264码流的时候,直接存储AVPacket后的文件可能是不能播放的。

如果视音频复用格式是TS (MPEG2 Transport Stream),直接存储后的文件是可以播放的。

复用格式是FLV,MP4则不行。

经过长时间资料搜索发现,FLV,MP4这些属于“特殊容器”,需要经过以下处理才能得到可播放的H.264码流:

1.第一次存储AVPacket之前需要在前面加上H.264的SPS和PPS。这些信息存储在AVCodecContext的extradata里面。

并且需要使用FFMPEG中的名为"h264_mp4toannexb"的 `bitstream filter` 进行处理。

然后将处理后的extradata存入文件

具体代码如下:

```
[cpp]
• FILE *fp=fopen( "test.264" , "ab" );
• AVCodecContext *pCodecCtx=...

• unsigned char *dummy=NULL; //输入的指针
• int dummy_len;
• AVBitStreamFilterContext* bsfc = av_bitstream_filter_init( "h264_mp4toannexb" );
• av_bitstream_filter_filter(bsfc, pCodecCtx , NULL, &dummy, &dummy_len, NULL, 0, 0);
• fwrite(pCodecCtx->extradata, pCodecCtx->extradata_size,1,fp);
• av_bitstream_filter_close(bsfc);
• free(dummy);
```

2.通过查看FFMPEG源代码我们发现,AVPacket中的数据起始处没有分隔符(0x00000001),也不是0x65、0x67、0x68、0x41等字节,所以可以AVPacket肯定这不是标准的nalu。其实,AVPacket前4个字表示的是nalu的长度,从第5个字节开始才是nalu的数据。所以直接将AVPacket前4个字节替换为0x00000001即可得到标准的nalu数据。

具体代码如下:

```
[cpp]
• char nal_start[]={0,0,0,1};
• fwrite(nal_start,4,1,fp);
• fwrite(pkt->data+4,pkt->size-4,1,fp);
• fclose(fp);
```

经过以上两步处理之后,我们就得到了可以正常播放的H.264码流

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/leixiaohua1020/article/details/11800877>

文章标签： `ffmpeg` `h.264`

个人分类： `FFMPEG`

所属专栏： `FFmpeg`

此PDF由spygg生成,请尊重原作者版权!!!

我的邮箱:liushidc@163.com