

## 转 avcodec\_decode\_video2()解码视频后丢帧的问题解决

2014年02月09日 20:50:06 阅读数：17265

使用libav转码视频时发现一个问题：使用下面这段代码解码视频时，视频尾巴上会丢掉几帧。

```
[cpp]
1. while(av_read_frame(ifmt_ctx,&packet) >= 0){
2.     ret = avcodec_decode_video2(video_dec_ctx, vframe, &got_frame, &packet);
3.     if (got_frame) {
4.         packet.pts = av_rescale_q(packet.pts,video_dec_st->time_base,video_enc_st->time_base);
5.         write_video_frame(ofmt_ctx,video_enc_st,vframe);
6.     }
7. }
```

这是因为源视频中PTS与DTS的不同造成的。

av\_read\_frame()按照PTS顺序读帧的时候，如果此帧需要参考后面的帧，那么此时avcodec\_decode\_video2()是没有能力解码此帧的，表现为got\_frame返回0。

比如说遇上如下EFGH四帧：

ID :  
E F G H  
KIND:  
I B P P  
PTS :  
1 2 3 4  
DTS :  
1 4 2 3

那么顺序读到F时，由于F需要参考G帧，而此时我们还没读到G帧，我们是没有解码F的能力的，got\_frame就返回0了。如果我们对此事不做处理，那么我们会丢掉一个帧（但丢掉的未必是F，因为av\_read\_frame()和avcodec\_decode\_video2()是1:1调用的）。

所以我们需要在while(av\_read\_frame())读完整个视频后，继续调用avcodec\_decode\_video2()把之前那些没有成功解码的帧都解出来。调用的次数就是之前got\_frame返回0的次数。

按照上述思路变更代码为以下，成功找回丢失的帧。

```
[cpp]
1. int skipped_frame = 0;
2.
3. while(av_read_frame(ifmt_ctx,&packet) >= 0){
4.     ret = avcodec_decode_video2(video_dec_ctx, vframe, &got_frame, &packet);
5.     if (got_frame) {
6.         packet.pts = av_rescale_q(packet.pts,video_dec_st->time_base,video_enc_st->time_base);
7.         write_video_frame(ofmt_ctx,video_enc_st,vframe);
8.     }
9.     else
10.    {
11.        skipped_frame++;
12.    }
13. }
14.
15. for(int i=skipped_frame; i>0; i--)
16. {
17.     ret = avcodec_decode_video2(video_dec_ctx, vframe, &got_frame, &packet);
18.     if (got_frame) {
19.         packet.pts = av_rescale_q(packet.pts,video_dec_st->time_base,video_enc_st->time_base);
20.         write_video_frame(ofmt_ctx,video_enc_st,vframe);
21.     }
22. }
```

原文地址：<http://blog.yikuyiku.com/?p=3486>

文章标签：[ffmpeg](#) [解码](#) [avcodec\\_decode\\_video](#) [帧重排](#) [丢帧](#)

个人分类：[FFMPEG](#)

所属专栏：[FFmpeg](#)