转 使用FFmpeg类库实现YUV视频序列编码为视频

2013年09月22日 01:36:13 阅读数：20385

搞视频处理的朋友肯定比较熟悉YUV视频序列，很多测试库提供的视频数据都是YUV视频序列，我们这里就用用YUV视频序列来做视频。关于YUV视频序列，我就不多讲了，可以看书学习，通常的视频序列都是YUV420格式的。

步骤也就那几步，添加视频流，打开编码器，开辟相应的内存空间，然后就可以打开YUV序列逐帧写入数据了，so easy！记得最后要做好文件的关闭和内存的释放，因为FFmpeg是c风格的（不知道新版本是否是c++风格的），这些工作都需要自己做好啊。过多的说明是没用的，直接上代码：

这里我补充一下， 大多数的视频格式好像只支持YUV格式的视频帧AVFrame，我试图直接把RGB的视频序列直接编码到视频这条路好像走不通，都需要把RGB的视频帧再转成YUV视频帧才行， 不知道高手有没有其他高见。

[cpp]

```cpp
#include <stdio.h>
#include <string.h>

extern "C"
{
#include <libavcodec\avcodec.h>
#include <libavformat\avformat.h>
#include <libswscale\swscale.h>
};

void main( int argc, char ** argv)
{
AVFormatContext* oc;
AVOutputFormat* fmt;
AVStream* video_st;
double video_pts;
uint8_t* video_outbuf;
uint8_t* picture_buf;
AVFrame* picture;
//  AVFrame* pictureRGB;
int size;
int ret;
int video_outbuf_size;

FILE *fin = fopen( "akiyo_qcif.yuv" , "rb" ); //视频源文件

const char * filename = "test.mpg" ;
//  const char* filename;
//  filename = argv[1];

av_register_all();

//  avcodec_init(); // 初始化codec库
//  avcodec_register_all(); // 注册编码器

fmt = guess_format(NULL, filename, NULL);
oc = av_alloc_format_context();
oc->oformat = fmt;
snprintf(oc->filename, sizeof (oc->filename), "%s" , filename);

video_st = NULL;
if (fmt->video_codec != CODEC_ID_NONE)
{
AVCodecContext* c;
video_st = av_new_stream(oc, 0);
c = video_st->codec;
c->codec_id = fmt->video_codec;
c->codec_type = CODEC_TYPE_VIDEO;
c->bit_rate = 400000;
c->width = 176;
c->height = 144;
c->time_base.num = 1;
c->time_base.den = 25;
c->gop_size = 12;
c->pix_fmt = PIX_FMT_YUV420P;
if (c->codec_id == CODEC_ID_MPEG2VIDEO)
{
c->max_b_frames = 2;
}
if (c->codec_id == CODEC_ID_MPEG1VIDEO)
{
c->mb_decision = 2;
}
if (!strcmp(oc->oformat->name, "mp4" ) || !strcmp(oc->oformat->name, "mov" ) || !strcmp(oc->oformat->name, "3gp" ))
{
c->flags |= CODEC_FLAG_GLOBAL_HEADER;
}
}

if (av_set_parameters(oc, NULL)<0)
{
```

```
return ;
}

dump_format(oc, 0, filename, 1);
if (video_st)
{
AVCodecContext* c;
AVCodec* codec;
c = video_st->codec;
codec = avcodec_find_encoder(c->codec_id);
if (!codec)
{
return ;
}
if (avcodec_open(c, codec) < 0)
{
return ;
}
if (!(oc->oformat->flags & AVFMT_RAWPICTURE))
{
video_outbuf_size = 200000;
video_outbuf = (uint8_t*)av_malloc(video_outbuf_size);
}
picture = avcodec_alloc_frame();
size = avpicture_get_size(c->pix_fmt, c->width, c->height);
picture_buf = (uint8_t*)av_malloc(size);
if (!picture_buf)
{
av_free(picture);
}
avpicture_fill((AVPicture*)picture, picture_buf, c->pix_fmt, c->width, c->height);
}

if (!(fmt->flags & AVFMT_NOFILE))
{
if (url_fopen(&oc->pb, filename, URL_WRONLY) < 0)
{
return ;
}
}
av_write_header(oc);

for ( int i=0; i<300; i++)
{
if (video_st)
{
video_pts = ( double )(video_st->pts.val * video_st->time_base.num / video_st->time_base.den);
}
else
{
video_pts = 0.0;
}
if (!video_st /* || video_pts >= 5.0*/ )
{
break ;
}
AVCodecContext* c;
c = video_st->codec;
size = c->width * c->height;

if (fread(picture_buf, 1, size*3/2, fin) < 0)
{
break ;
}

picture->data[0] = picture_buf; // 亮度
picture->data[1] = picture_buf+ size; // 色度
picture->data[2] = picture_buf+ size*5/4; // 色度

// 如果是rgb序列，可能需要如下代码
//      SwsContext* img_convert_ctx;
//      img_convert_ctx = sws_getContext(c->width, c->height, PIX_FMT_RGB24, c->width, c->height, c-
>pix_fmt, SWS_BICUBIC, NULL, NULL, NULL);
//      sws_scale(img_convert_ctx, pictureRGB->data, pictureRGB->linesize, 0, c->height, picture->data, picture->linesize);

if (oc->oformat->flags & AVFMT_RAWPICTURE)
{
AVPacket pkt;
av_init_packet(&pkt);
pkt.flags |= PKT_FLAG_KEY;
pkt.stream_index = video_st->index;
pkt.data = (uint8_t*)picture;
pkt.size = sizeof (AVPicture);
ret = av_write_frame(oc, &pkt);
}
else
{
int out_size = avcodec_encode_video(c, video_outbuf, video_outbuf_size, picture);
if (out_size > 0)
{
AVPacket pkt;
av_init_packet(&pkt);
pkt.pts = av_rescale_q(c->coded_frame->pts, c->time_base, video_st->time_base);
if (c->coded_frame->key_frame)
{
pkt.flags |= PKT_FLAG_KEY;
}
pkt.stream_index = video_st->index;
pkt.data = video_outbuf;
pkt.size = out_size;
ret = av_write_frame(oc, &pkt);
}
}
}

if (video_st)
{
avcodec_close(video_st->codec);
//      av_free(picture->data[0]);
av_free(picture);
av_free(video_outbuf);
//      av_free(picture_buf);
}
av_write_trailer(oc);
for ( int i=0; i<oc->nb_streams; i++)
{
av_freep(&oc->streams[i]->codec);
av_freep(&oc->streams[i]);
}
if (!(fmt->flags & AVFMT_NOFILE))
{
url_fclose(oc->pb);
}
av_free(oc);
}
```

原文地址： http://blog.csdn.net/yang_xian521/article/details/7698742

文章标签： ffmpeg 编码 视频 yuv 类库

个人分类： FFMPEG

所属专栏： FFmpeg