

最简单的基于FFMPEG的图像编码器（YUV编码为JPEG）

2014年05月09日 00:25:35 阅读数：36754

伴随着毕业论文的完成，这两天终于腾出了空闲，又有时间搞搞FFMPEG的研究了。想着之前一直搞的都是FFMPEG解码方面的工作，很少涉及到FFMPEG编码方面的东西，于是打算研究一下FFMPEG的编码。在网上看了一些例子，发现要不然是难度略微有些大，要不然就是类库比较陈旧，于是就决定自己做一个编码方面的例子，方便以后学习。

简介

本文的编码器实现了YUV420P的数据编码为JPEG图片。本着简单的原则，代码基本上精简到了极限。使用了2014年5月6号编译的最新的FFMPEG类库。

程序很简单，打开工程后直接运行即可将YUV数据编码为JPEG。本程序十分灵活，可以根据需要修改成编码各种图像格式的编码器，比如PNG，GIF等等。平台使用VC2010。

源代码

```
[cpp]
1.  /**
2.   * 最简单的基于FFmpeg的图像编码器
3.   * Simplest Ffmpeg Picture Encoder
4.   *
5.   * 雷霄骅 Lei Xiaohua
6.   * leixiaohua1020@126.com
7.   * 中国传媒大学/数字电视技术
8.   * Communication University of China / Digital TV Technology
9.   * http://blog.csdn.net/leixiaohua1020
10.  *
11.  * 本程序实现了YUV420P像素数据编码为JPEG图片。是最简单的FFmpeg编码方面的教程。
12.  * 通过学习本例子可以了解FFmpeg的编码流程。
13.  */
14.
15.  #include <stdio.h>
16.
17.  #define __STDC_CONSTANT_MACROS
18.
19.  #ifndef _WIN32
20.  //Windows
21.  extern "C"
22.  {
23.  #include "libavcodec/avcodec.h"
24.  #include "libavformat/avformat.h"
25.  };
26.  #else
27.  //Linux...
28.  #ifdef __cplusplus
29.  extern "C"
30.  {
31.  #endif
32.  #include <libavcodec/avcodec.h>
33.  #include <libavformat/avformat.h>
34.  #ifdef __cplusplus
35.  };
36.  #endif
37.  #endif
38.
39.
40.  int main(int argc, char* argv[])
41.  {
42.      AVFormatContext* pFormatCtx;
43.      AVOutputFormat* fmt;
44.      AVStream* video_st;
45.      AVCodecContext* pCodecCtx;
46.      AVCodec* pCodec;
47.
48.      uint8_t* picture_buf;
49.      AVFrame* picture;
50.      AVPacket pkt;
51.      int y_size;
52.      int got_picture=0;
53.      int size;
54.
55.      int ret=0;
56.
57.      FILE *in_file = NULL; //YUV source
58.      int in_w=480,in_h=272; //YUV's width and height
59.      const char* out_file = "cuc_view_encode.jpg"; //Output file
60.
61.      in_file = fopen("cuc_view_480x272.yuv", "rb");
62.
63.      av_register_all();
64.
```

```

65. //Method 1
66. pFormatCtx = avformat_alloc_context();
67. //Guess format
68. fmt = av_guess_format("mjpeg", NULL, NULL);
69. pFormatCtx->oformat = fmt;
70. //Output URL
71. if (avio_open(&pFormatCtx->pb,out_file, AVIO_FLAG_READ_WRITE) < 0){
72.     printf("Couldn't open output file.");
73.     return -1;
74. }
75.
76. //Method 2. More simple
77. //avformat_alloc_output_context2(&pFormatCtx, NULL, NULL, out_file);
78. //fmt = pFormatCtx->oformat;
79.
80. video_st = avformat_new_stream(pFormatCtx, 0);
81. if (video_st==NULL){
82.     return -1;
83. }
84. pCodecCtx = video_st->codec;
85. pCodecCtx->codec_id = fmt->video_codec;
86. pCodecCtx->codec_type = AVMEDIA_TYPE_VIDEO;
87. pCodecCtx->pix_fmt = AV_PIX_FMT_YUVJ420P;
88.
89. pCodecCtx->width = in_w;
90. pCodecCtx->height = in_h;
91.
92. pCodecCtx->time_base.num = 1;
93. pCodecCtx->time_base.den = 25;
94. //Output some information
95. av_dump_format(pFormatCtx, 0, out_file, 1);
96.
97. pCodec = avcodec_find_encoder(pCodecCtx->codec_id);
98. if (!pCodec){
99.     printf("Codec not found.");
100.    return -1;
101. }
102. if (avcodec_open2(pCodecCtx, pCodec,NULL) < 0){
103.     printf("Could not open codec.");
104.     return -1;
105. }
106. picture = av_frame_alloc();
107. size = avpicture_get_size(pCodecCtx->pix_fmt, pCodecCtx->width, pCodecCtx->height);
108. picture_buf = (uint8_t *)av_malloc(size);
109. if (!picture_buf)
110. {
111.     return -1;
112. }
113. avpicture_fill((AVPicture *)picture, picture_buf, pCodecCtx->pix_fmt, pCodecCtx->width, pCodecCtx->height);
114.
115. //Write Header
116. avformat_write_header(pFormatCtx,NULL);
117.
118. y_size = pCodecCtx->width * pCodecCtx->height;
119. av_new_packet(&pkt,y_size*3);
120. //Read YUV
121. if (fread(picture_buf, 1, y_size*3/2, in_file) <=0)
122. {
123.     printf("Could not read input file.");
124.     return -1;
125. }
126. picture->data[0] = picture_buf; // Y
127. picture->data[1] = picture_buf+ y_size; // U
128. picture->data[2] = picture_buf+ y_size*5/4; // V
129.
130. //Encode
131. ret = avcodec_encode_video2(pCodecCtx, &pkt,picture, &got_picture);
132. if(ret < 0){
133.     printf("Encode Error.\n");
134.     return -1;
135. }
136. if (got_picture==1){
137.     pkt.stream_index = video_st->index;
138.     ret = av_write_frame(pFormatCtx, &pkt);
139. }
140.
141. av_free_packet(&pkt);
142. //Write Trailer
143. av_write_trailer(pFormatCtx);
144.
145. printf("Encode Successful.\n");
146.
147. if (video_st){
148.     avcodec_close(video_st->codec);
149.     av_free(picture);
150.     av_free(picture_buf);
151. }
152. avio_close(pFormatCtx->pb);
153. avformat_free_context(pFormatCtx);
154.
155. fclose(in_file);

```

```
156.
157.     return 0;
158. }
```

结果

编码前的YUV420P数据：

□

编码后的JPEG：

□

下载

simplest ffmpeg picture encoder

项目主页

SourceForge：<https://sourceforge.net/projects/simplestffmpegpictureencoder/>

Github：https://github.com/leixiaohua1020/simplest_ffmpeg_picture_encoder

开源中国：http://git.oschina.net/leixiaohua1020/simplest_ffmpeg_picture_encoder

CSDN工程下载地址：

<http://download.csdn.net/detail/leixiaohua1020/7319265>

PUDN工程下载地址：

<http://www.pudn.com/downloads644/sourcecode/multimedia/detail2605252.html>

本程序实现了YUV420P像素数据编码为JPEG图片。是最简单的FFmpeg编码方面的教程。通过学习本例子可以了解FFmpeg的编码流程。

更新-1.1(2015.2.13)=====

这次考虑到了跨平台的要求，调整了源代码。经过这次调整之后，源代码可以在以下平台编译通过：

VC++：打开sln文件即可编译，无需配置。

cl.exe：打开compile_cl.bat即可命令行下使用cl.exe进行编译，注意可能需要按照VC的安装路径调整脚本里面的参数。编译命令如下。

```
[plain]
1.  ::VS2010 Environment
2.  call "D:\Program Files\Microsoft Visual Studio 10.0\VC\vcvarsall.bat"
3.  ::include
4.  @set INCLUDE=include;%INCLUDE%
5.  ::lib
6.  @set LIB=lib;%LIB%
7.  ::compile and link
8.  cl simplest_ffmpeg_picture_encoder.cpp /link avcodec.lib avformat.lib avutil.lib ^
9.  avdevice.lib avfilter.lib postproc.lib swresample.lib swscale.lib /OPT:NOREF
```

MinGW：MinGW命令行下运行compile_mingw.sh即可使用MinGW的g++进行编译。编译命令如下。

```
[plain]
1.  g++ simplest_ffmpeg_picture_encoder.cpp -g -o simplest_ffmpeg_picture_encoder.exe \
2.  -I /usr/local/include -L /usr/local/lib \
3.  -lavformat -lavcodec -lavutil
```

GCC：Linux或者MacOS命令行下运行compile_gcc.sh即可使用GCC进行编译。编译命令如下。

[plain]  

```
1. gcc simplest_ffmpeg_picture_encoder.cpp -g -o simplest_ffmpeg_picture_encoder.out \  
2. -I /usr/local/include -L /usr/local/lib -lavformat -lavcodec -lavutil
```

PS：相关的编译命令已经保存到了工程文件夹中

CSDN下载地址：<http://download.csdn.net/detail/leixiaohua1020/8444893>

SourceForge上已经更新。

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/25346147>

文章标签：[ffmpeg](#) [图像](#) [编码](#) [yuv](#) [jpeg](#)

个人分类：[我的开源项目](#) [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