Lok'tar

事情若有可能出错，就一定会出错

使用libx264将I420编码为H.264

libx264

libx264是一个自由的H.264编码库，是x264项目的一部分，使用广泛，ffmpeg的H.264实现就是用的libx264。

代码

要把一个I420视频文件编码为H264格式。I420是YUV中planar格式的一种，一张I420图片中有三个plane，分别存放整张图片的Y、U、V分量;采样比例为4：2：0，12bpp，Y：U：V的分量长度是4：1：1。

头文件

```
#include <stdint.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <x264.h>
```

变量声明和参数

```
int width = 480;
int height = 360;
int fps = 25;
size_t yuv_size = width * height * 3 / 2;
x264_t *encoder;
x264_picture_t pic_in, pic_out;
int inf, outf;
uint8_t *yuv_buffer;

if (argc != 3) {
    printf("usage: %s input output\n", argv[0]);
}
```

- 视频尺寸是480×360,YUV I420格式，每个像素1.5个字节，所以一

< 2012年9月 >

日 一 二 三 四 五 六
26 27 28 29 30 31 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 1 2 3 4 5 6

公告

昵称：Lamo
园龄：2年9个月
粉丝：7
关注：10
+加关注

搜索

找找看

谷歌搜索

常用链接

- 我的随笔
- 我的评论
- 我的参与
- 最新评论
- 我的标签

我的标签

- h264(1)
- iOS(1)
- libx264(1)
- sqlite(1)
- unicode(1)

随笔分类

张YUV图片大小是width * height * 1.5

- encoder就是编码器，x264_t格式在x264.h文件中只有

  typedef struct x264_t x264_t

  编码器类型只需要也只能声明为x264_t的指针类型

- 每次编码时，YUV图片的信息都保存在pic_in中
- 输入输出的文件描述符
- 从文件读入的YUV的缓冲区

初始化encoder

```
 1 x264_param_t param;
 2 x264_param_default_preset(&param, "veryfast", "zerolatency
");
 3 param.i_threads = 1;
 4 param.i_width = width;
 5 param.i_height = height;
 6 param.i_fps_num = fps;
 7 param.i_fps_den = 1;
 8
 9 param.i_keyint_max = 25;
10 param.b_intra_refresh = 1;
11
12 param.b_annexb = 1;
13
14 x264_param_apply_profile(&param, "baseline");
15 encoder = x264_encoder_open(&param);
```

初始化pic_in

```
1 x264_picture_alloc(&pic_in, X264_CSP_I420, width, height);
2
3 yuv_buffer = malloc(yuv_size);
4
5 pic_in.img.plane[0] = yuv_buffer;
6 pic_in.img.plane[1] = pic_in.img.plane[0] + width * height;
7 pic_in.img.plane[2] = pic_in.img.plane[1] + width * height
/ 4;
```

- pic_in.img中保存YUV图片各分量的信息

```
typedef struct {
    int i_csp;
    int i_plane;
    int i_stride[4];
```


- C(2)
- H.264(1)
- iOS(3)
- Lua
- Objective-C(3)
- 读书笔记(1)
- 计算机网络

随笔档案

- 2012年10月 (1)
- 2012年9月 (1)
- 2012年6月 (1)
- 2012年3月 (1)
- 2011年12月 (1)

最新评论

阅读排行榜

- 1. 使用libx264将I420编码为H.264(1751)
- 2. 一个简单的segmentedControl实现(1412)
- 3. Core Data - Constraint failed(886)
- 4. UTF32字符串转换成NSString(314)
- 5. 程序员的思维修炼读后总结-第二章:从新手到专家的历程(191)

评论排行榜

- 1. 程序员的思维修炼读后总结-第二章:从新手到专家的历程(1)
- 2. 使用libx264将I420编码为H.264(1)
- 3. UTF32字符串转换成NSString(0)
- 4. 一个简单的se

```
    uint8_t *plane[4];
} x264_image_t;
```

其中*i*csp, *i*plane, i*stride的值在pic*in初始化的时候已经被赋值，代码中只需要将plane数组指向正确的位置

- 程序中每一帧的图片都是读取到yuv_buffer中，所以在这里设置一次就行了

初始化文件描述符

```
1 inf = open(argv[1], O_RDONLY);
2 if (inf < 0) {
3     return -1;
4 }
5 outf = open(argv[2], O_CREAT | O_WRONLY, 444);
6 if (outf < 0) {
7     return -1;
8 }
```

编码

```
 1 int64_t i_pts = 0;
 2
 3 x264_nal_t *nals;
 4 int nnal;
 5 while (read(inf, yuv_buffer, yuv_size) > 0) {
 6     pic_in.i_pts = i_pts++;
 7     x264_encoder_encode(encoder, &nals, &nnal, &pic_in, &p
ic_out);
 8     x264_nal_t *nal;
 9     for (nal = nals; nal < nals + nnal; nal++) {
10         write(outf, nal->p_payload, nal->i_payload);
11     }
12 }
```

- 关于ffmpeg的pts，网上有好多种公式，其实只要步长为1递增就行了
- H.264的NAL层是为了适应网络传输的需要，将大的编码后的帧分成多个块
- p_payload就是编码后的H.264的帧数据，写入输出文件

扫尾

```
1 x264_encoder_close(encoder);
2 close(inf);
3 close(outf);
```

gmentedControl实现(0)
- 5. Core Data - Constraint failed(0)

推荐排行榜

- 1. 程序员的思维修炼读后总结-第二章:从新手到专家的历程(1)
- 2. UTF32字符串转换成NSString(1)
- 3. 使用libx264将I420编码为H.264(1)

```
4 free(yuv_buffer);
5 return 0;
```

编译

gcc sourcefile -lx264 -Wall -o execfile

这里有一段I420视频可供测试。

参考

1. How does one encode a series of images into H264 using the x264 C API? - Stack Overflow
2. YUV RGB 常见视频格式解析 - 一指流砂 - 博客园

分类: C, H.264

标签: h264, libx264

Lamo
关注 - 10
粉丝 - 7
+加关注

1 0

(请您对文章做出评价)

« 上一篇：一个简单的segmentedControl实现
» 下一篇：UTF32字符串转换成NSString

发表于 2012-09-01 14:19 LAMO 阅读(1751) 评论(1) 编辑 收藏

评论

#1楼

楼主，辛苦了！
补充一点：
编译的时候要加上 '-pthread -lm '，即
gcc sourcefile -lx264 -Wall -o execfile -pthread -lm
否则会出现很多的undefined reference to。。。。。

支持(0) 反对(0)

scalerzj 评论于 2012-12-05 22:09

最新IT新闻:

· 新东方邓弘：单凭技术不可能搞教育颠覆
· 云计算技术背后的那些天才程序员：Qemu的作者法布里斯贝拉
· 云计算技术背后的那些天才程序员：KVM之父Avi Kivity
· 魅族高管回顾与老罗交锋史：曾欲收购锤子
· 滴滴打车回应交通部意见稿：手机更适合叫车
» 更多新闻...

最新知识库文章:

· 「我只是认真」——聊聊工匠情怀
· 什么是互联网思维？给你最全面的解释
· 豆瓣的基础架构
· 大公司？小公司？我的经历和建议
· 阿里负责人揭秘面试潜规则
» 更多知识库文章...