原 最简单的视频编码器：基于libx264（编码YUV为H.264）

2014年12月23日 00:18:18　阅读数：26052

==========

最简单的视频编码器系列文章列表：

最简单的视频编码器：编译

最简单的视频编码器：基于libx264（编码YUV为H.264）

最简单的视频编码器：基于libx265（编码YUV为H.265）

最简单的视频编码器：libvpx（编码YUV为VP8）

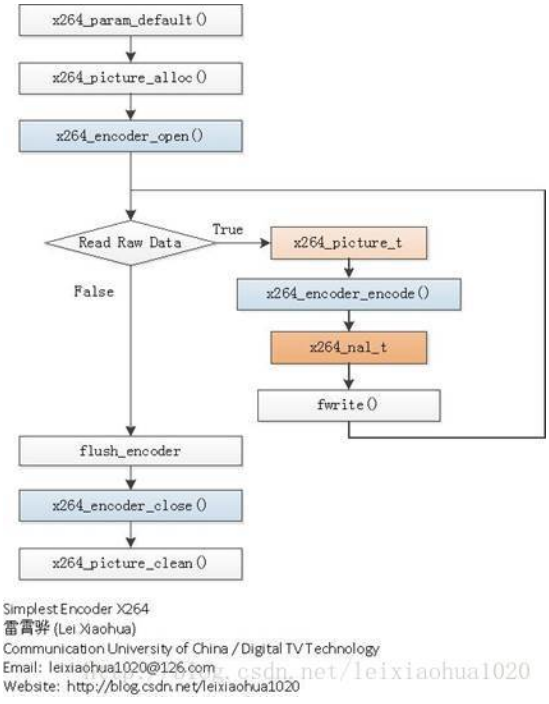
==========

本文记录一个最简单的基于libx264的H.264视频编码器。此前记录的H.264编码器都是基于FFmpeg调用libx264完成编码的，例如：

《最简单的基于FFMPEG的视频编码器（YUV编码为H.264）》
相比与上文中的编码器，本文记录的编码器属于“轻量级”的编码器。因为它不再包含FFmpeg的代码，直接调用libx264完成编码。因此项目的体积非常小巧。该编码器可以将输入的YUV数据编码为H.264码流文件。

流程图

调用libx264进行视频编码的流程图如下所示。



流程图中主要的函数如下所示。
x264_param_default()：设置参数集结构体x264_param_t的缺省值。
x264_picture_alloc()：为图像结构体x264_picture_t分配内存。
x264_encoder_open()：打开编码器。
x264_encoder_encode()：编码一帧图像。
x264_encoder_close()：关闭编码器。
x264_picture_clean()：释放x264_picture_alloc()申请的资源。

存储数据的结构体如下所示。
x264_picture_t：存储压缩编码前的像素数据。
x264_nal_t：存储压缩编码后的码流数据。

此外流程图中还包括一个“flush_encoder”模块，该模块使用的函数和编码模块是一样的。唯一的不同在于不再输入视频像素数据。它的作用在于输出编码器中剩余的码流数据。

源代码

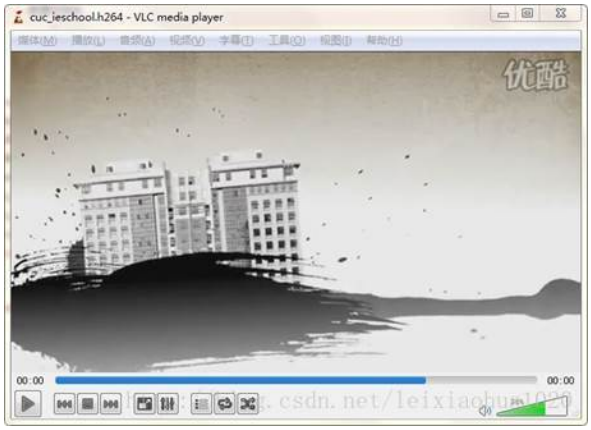
```cpp
/**
 * 最简单的基于X264的视频编码器
 * Simplest X264 Encoder
 *
 * 雷霄骅 Lei Xiaohua
 * leixiaohua1020@126.com
 * 中国传媒大学/数字电视技术
 * Communication University of China / Digital TV Technology
 * http://blog.csdn.net/leixiaohua1020
 *
 * 本程序可以YUV格式的像素数据编码为H.264码流，是最简单的
 * 基于libx264的视频编码器
 *
 * This software encode YUV data to H.264 bitstream.
 * It's the simplest encoder example based on libx264.
 */
#include <stdio.h>
#include <stdlib.h>

#include "stdint.h"

#if defined ( __cplusplus)
extern "C"
{
#include "x264.h"
};
#else
#include "x264.h"
#endif


int main(int argc, char** argv)
{

	int ret;
	int y_size;
	int i,j;

	//FILE* fp_src  = fopen("../cuc_ieschool_640x360_yuv444p.yuv", "rb");
	FILE* fp_src  = fopen("../cuc_ieschool_640x360_yuv420p.yuv", "rb");

	FILE* fp_dst = fopen("cuc_ieschool.h264", "wb");

	//Encode 50 frame
	//if set 0, encode all frame
	int frame_num=50;
	int csp=X264_CSP_I420;
	int width=640,height=360;

	int iNal   = 0;
	x264_nal_t* pNals = NULL;
	x264_t* pHandle   = NULL;
	x264_picture_t* pPic_in = (x264_picture_t*)malloc(sizeof(x264_picture_t));
	x264_picture_t* pPic_out = (x264_picture_t*)malloc(sizeof(x264_picture_t));
	x264_param_t* pParam = (x264_param_t*)malloc(sizeof(x264_param_t));

	//Check
	if(fp_src==NULL||fp_dst==NULL){
		printf("Error open files.\n");
		return -1;
	}

	x264_param_default(pParam);
	pParam->i_width   = width;
	pParam->i_height  = height;
	/*
	//Param
	pParam->i_log_level  = X264_LOG_DEBUG;
	pParam->i_threads  = X264_SYNC_LOOKAHEAD_AUTO;
	pParam->i_frame_total = 0;
	pParam->i_keyint_max = 10;
	pParam->i_bframe  = 5;
	pParam->b_open_gop  = 0;
	pParam->i_bframe_pyramid = 0;
	pParam->rc.i_qp_constant=0;
	pParam->rc.i_qp_max=0;
	pParam->rc.i_qp_min=0;
	pParam->i_bframe_adaptive = X264_B_ADAPT_TRELLIS;
	pParam->i_fps_den  = 1;
	pParam->i_fps_num  = 25;
```

```
        pParam->i_timebase_den = pParam->i_fps_num;
        pParam->i_timebase_num = pParam->i_fps_den;
        */
        pParam->i_csp=csp;
        x264_param_apply_profile(pParam, x264_profile_names[5]);

        pHandle = x264_encoder_open(pParam);

        x264_picture_init(pPic_out);
        x264_picture_alloc(pPic_in, csp, pParam->i_width, pParam->i_height);

        //ret = x264_encoder_headers(pHandle, &pNals, &iNal);

        y_size = pParam->i_width * pParam->i_height;
        //detect frame number
        if(frame_num==0){
                fseek(fp_src,0,SEEK_END);
                switch(csp){
                case X264_CSP_I444:frame_num=ftell(fp_src)/(y_size*3);break;
                case X264_CSP_I420:frame_num=ftell(fp_src)/(y_size*3/2);break;
                default:printf("Colorspace Not Support.\n");return -1;
                }
                fseek(fp_src,0,SEEK_SET);
        }

        //Loop to Encode
        for( i=0;i<frame_num;i++){
                switch(csp){
                case X264_CSP_I444:{
                        fread(pPic_in->img.plane[0],y_size,1,fp_src);         //Y
                        fread(pPic_in->img.plane[1],y_size,1,fp_src);         //U
                        fread(pPic_in->img.plane[2],y_size,1,fp_src);         //V
                        break;}
                case X264_CSP_I420:{
                        fread(pPic_in->img.plane[0],y_size,1,fp_src);         //Y
                        fread(pPic_in->img.plane[1],y_size/4,1,fp_src);     //U
                        fread(pPic_in->img.plane[2],y_size/4,1,fp_src);     //V
                        break;}
                default:{
                        printf("Colorspace Not Support.\n");
                        return -1;}
                }
                pPic_in->i_pts = i;

                ret = x264_encoder_encode(pHandle, &pNals, &iNal, pPic_in, pPic_out);
                if (ret< 0){
                        printf("Error.\n");
                        return -1;
                }

                printf("Succeed encode frame: %5d\n",i);

                for ( j = 0; j < iNal; ++j){
                         fwrite(pNals[j].p_payload, 1, pNals[j].i_payload, fp_dst);
                }
        }
        i=0;
        //flush encoder
        while(1){
                ret = x264_encoder_encode(pHandle, &pNals, &iNal, NULL, pPic_out);
                if(ret==0){
                        break;
                }
                printf("Flush 1 frame.\n");
                for (j = 0; j < iNal; ++j){
                        fwrite(pNals[j].p_payload, 1, pNals[j].i_payload, fp_dst);
                }
                i++;
        }
        x264_picture_clean(pPic_in);
        x264_encoder_close(pHandle);
        pHandle = NULL;

        free(pPic_in);
        free(pPic_out);
        free(pParam);

        fclose(fp_src);
        fclose(fp_dst);

        return 0;
}
```

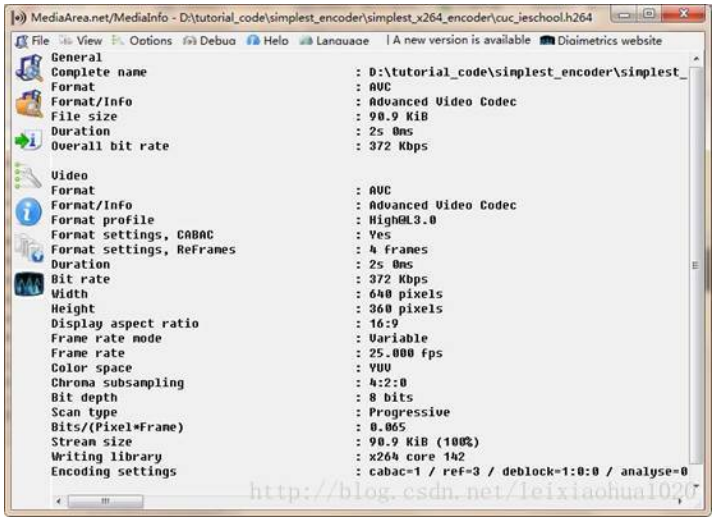
运行结果

程序的输入为一个YUV文件（已经测试过YUV444P和YUV420P两种格式）。

输出为H.264码流文件。



H.264码流文件的信息如下所示。



下载

Simplest Encoder

项目主页

SourceForge： https://sourceforge.net/projects/simplestencoder/

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

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

CDSN下载地址： http://download.csdn.net/detail/leixiaohua1020/8284105

该解决方案包含了几个常见的编码器的使用示例：
simplest_vpx_encoder：最简单的基于libvpx的视频编码器
simplest_x264_encoder：最简单的基于libx264的视频编码器
simplest_x265_encoder：最简单的基于libx265的视频编码器

文章标签： x264 h264 视频 编码 YUV

个人分类： x264 我的开源项目