6篇

2篇

原创 🛂 VIP文章 W2Y 🐧 于 2018-07-10 15:57:31 发布 • 👁 22318 🕏 收藏 66 分类专栏: 音视频 H.264 FFmpeg 文章标签: h264 sps 解析 宽高 帧率fps

在做音视频开发的时候,存在不解码视频帧的前提下需要获取视频宽高、帧率等信息,而H.264中的SPS数据可为我们提供这些相关的信 息。在此之前,我们需要对一些协议和算法有一定的初步了解,后文中有完整的代码展示。

## H.264协议

我们在此不讲解L 264协议,但是我们需要了解ML单元语法(ML unit syntax)、序列参数集数据语法(Sequence parameter set data syntax)、视频可用参数语法(WI parameters syntax)。具体可去1TU-T官阿下载H. 2642[4],我查看的是T-REC-H. 264-201704-I!!PDF-

Sequence parameter set data syntax

### 7.3.2.1.1 Sequence parameter set data syntax

eq_parameter_set_data() {	C	Descrip
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
constraint_set4_flag	0	u(1)
constraint_set5_flag	0	u(1)
reserved_zero_2bits /* equal to 0 */	0	u(2)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v
if( profile_idc == 100    profile_idc == 110    profile_idc == 122    profile_idc == 244    profile_idc == 44    profile_idc == 83    profile_idc == 86    profile_idc == 118    profile_idc == 128    profile_idc == 138    profile_idc == 139    profile_idc == 134    profile_idc == 135 ) {	0	(v
chroma_format_idc	0	ue(v
if( chroma_format_idc == 3 )	-	
separate_colour_plane_flag	0	u(1)
bit_depth_luma_minus8	0	ue(v
bit_depth_chroma_minus8	0	ue(v
qpprime_y_zero_transform_bypass_flag	0	u(1)
seq_scaling_matrix_present_flag	0	u(1)
if( seq_scaling_matrix_present_flag )		
for( i = 0; i < ( ( chroma_format_idc != 3 ) ? 8 : 12 ); i++ ) {		
seq_scaling_list_present_flag[ i ]	0	u(1)
if( seq_scaling_list_present_flag[ i ] )		
if(i < 6)		
scaling_list( ScalingList4x4[ i ], 16, UseDefaultScalingMatrix4x4Flag[ i ] )	0	
else		
$ \begin{array}{l} scaling\_list(\ ScalingList8x8[\ i-6\ ], 64, \\ UseDefaultScalingMatrix8x8Flag[\ i-6\ ]\ ) \end{array} $	0	
}		
https://blog.co	esdn. ne	/lizhiji
log2 max frame num minus4	0	ue(v

log2 max frame num minus4	0	ue(v)
	-	
pic_order_cnt_type	0	ue(v)
if( pic_order_cnt_type == 0 )		
log2_max_pic_order_cnt_lsb_minus4	0	ue(v)
else if( pic_order_cnt_type == 1 ) {		
delta_pic_order_always_zero_flag	0	u(1)
offset_for_non_ref_pic	0	se(v)
offset_for_top_to_bottom_field	0	se(v)
num_ref_frames_in_pic_order_cnt_cycle	0	ue(v)
$for(\ i=0;\ i < num\_ref\_frames\_in\_pic\_order\_cnt\_cycle;\ i++\ )$		
offset_for_ref_frame[ i ]	0	se(v)
}		
max_num_ref_frames	0	ue(v)
gaps_in_frame_num_value_allowed_flag	0	u(1)
pic_width_in_mbs_minus1	0	ue(v)
pic_height_in_map_units_minus1	0	ue(v)
frame_mbs_only_flag	0	u(1)
if( !frame_mbs_only_flag )		
mb_adaptive_frame_field_flag	0	u(1)
direct_8x8_inference_flag	0	u(1)
frame_cropping_flag	0	u(1)
if( frame_cropping_flag ) {		
frame_crop_left_offset	0	ue(v)
frame_crop_right_offset	0	ue(v)
frame_crop_top_offset	0	ue(v)
frame_crop_bottom_offset	0	ue(v)
}		
vui_parameters_present_flag	0	u(1)
if( vui_parameters_present_flag )		
vui parameters()	0	

VUI parameters syntax

# E.1.1 VUI parameters syntax

vui_parameters( ) {	C	Descriptor
aspect_ratio_info_present_flag	0	u(1)
if( aspect_ratio_info_present_flag ) {		
aspect_ratio_idc	0	u(8)

会员中心 🔐	消息	历史	创作中	Þ心 🧶	<b></b>	AI发
目录		发布首篇 点 <b>亮新</b> 秀		得大额流	<b>允量券</b>	
h.264协议						
sequen	ce paran	neter se	t data s	yntax		
vui para	meters	syntax				
指数哥伦布	编码					
无符号	旨数哥伦	布编码(L	ie)			
有符号	旨数哥伦	布编码(s	e)			
sps参数分析	Ť					
视频宽	高(width、	height)	获取			
视频帧	堅(fps)获	取				
sps解析代码	3					
分类专栏						
C She	ell			1篇		
- C And	Iroid			1篇		
Coo	oaPods			1篇		
- 優 嵌/	土			1篇		
C 音初	顺			10篇		
FFr	npeg			6篇		
C H.2	64			3篇		
ios				7篇		

C++

React-Native

if( aspect_ratio_idc == Extended_SAR) {		
sar_width	0	u(16
sar_height	0	u(16
}		
}		
overscan_info_present_flag	0	u(1)
if( overscan_info_present_flag )		
overscan_appropriate_flag	0	u(1)
video_signal_type_present_flag	0	u(1)
if( video_signal_type_present_flag ) {		
video_format	0	u(3)
video_full_range_flag	0	u(1)
colour_description_present_flag	0	u(1)
if( colour_description_present_flag ) {		
colour_primaries	0	u(8)
transfer_characteristics	0	u(8)
matrix_coefficients	0	u(8)
}		
}		
chroma_loc_info_present_flag	0	u(1)
if( chroma_loc_info_present_flag ) {		
chroma_sample_loc_type_top_field	0	ue(v)
chroma sample loc type bottom field	0	ue(v

timing_info_present_flag	0	u(1)
<pre>if( timing_info_present_flag ) {</pre>		
num_units_in_tick	0	u(32)
time_scale	0	u(32)
fixed_frame_rate_flag	0	u(1)
}		
nal_hrd_parameters_present_flag	0	u(1)
if( nal_hrd_parameters_present_flag )		
hrd_parameters( )	0	
vcl_hrd_parameters_present_flag	0	u(1)
if( vcl_hrd_parameters_present_flag )		
hrd_parameters( )	0	
if( nal_hrd_parameters_present_flag    vcl_hrd_parameters_present_flag )		
low_delay_hrd_flag	0	u(1)
pic_struct_present_flag	0	u(1)
bitstream_restriction_flag	0	u(1)
if( bitstream_restriction_flag ) {		
motion_vectors_over_pic_boundaries_flag	0	u(1)
max_bytes_per_pic_denom	0	ue(v)
max_bits_per_mb_denom	0	ue(v)
log2_max_mv_length_horizontal	0	ue(v)
log2_max_mv_length_vertical	0	ue(v)
max_num_reorder_frames	0	ue(v)
max_dec_frame_buffering	0	ue(v)
}		
https://blog.	csdn. ne	/lizhijia

第一列: 相关参数信息名称;

第二列: 规定条带数据可以至多划分为三种条带数据类别(这个可以不用理解);

第三列中的u(n)表示:使用n位无符号整数表示,由n位bit換算得到,即从左到右读取n个bit位所表示的值;ue(v)表示:无符号指数哥伦布编码值;ue(v)表示:有符号指数哥伦布编码值。

u(n)代码实现:

## 指数哥伦布编码

Golomb编码是一种无规的数据压缩方法,由数学家Solomon W.Golomb在1960年代发明。Golomb编码只能对非负整数进行编码,符号表中的符号出现的概率符合几何分布(Geometric Distribution)时,使用Golomb编码可以取得最优效果,也就是说Golomb编码比较适合小的数字比大的数字出现概率比较高的编码。它使用较短的码长编码较小的数字,较长的码长编码较大的数字。在此,我们只需要了解:无符号指数哥伦布编码ue(v)。有符号指数哥伦布编码se(v)。

# 无符号指数哥伦布编码(UE)

```
哥伦布編码的码字code_word由三部分组成: code_word = [以个0] + [1] + [Info]
其中, Info是一个携帶信息的如位数据。時个哥伦布码的长度为(2M+1)位,每个码字都可由code_num产生。
根据码字code_word等码出code_num值的过程如下:
1. 育先能入M位以「"为结尾的0;
2. 根据得到的机。该入接下来的地位Info数据。
3. 根据这个公式得到计算结果code_num = Info - 1 + 2M
```

### 代码实现

```
1 static UINT ue(sps_bit_stream *bs)
2 {
3     UINT zeroNum = 0;
4     while (u(bs, 1) == 0 && !eof(bs) && zeroNum < 32) {
5     zeroNum ++;
```

```
return (UINT)((1 << zeroNum) - 1 + u(bs, zeroNum));
                             有符号指数哥伦布编码(SE)
                             有符号的指数哥伦布编码值是通过无符号的指数哥伦布编码的值通过换算得到的。其换算关系为: n = (-1)^(k+1) * ceil(k/2).
                               1 | INT se(sps_bit_stream *bs)
                                     INT ueVal = (INT)ue(bs);
                                                          ; //ceil:返回大于或者等于指定表达式的最小整数
//偶数取反,即(-1)^(k+1)
                                                                  200万优质内容无限畅学
                                                      ○ 最低0.62元/天解锁文章
                                                                               ○ 向"C知道"追问
                             T-REC-H.264-200503-S2PDF-C (中文版) .pdf
                             该资源为h264编码的标准文档,介绍了h264编码流程以及SPS、PPS等相关概念
                             视频编解码(SPS重要参数)
                             SPS重要参数 pic_width_in_minus1-> 图像宽度包含的宏块个数-1 第一行中mbs部分指的是宏块的数量,pic_width_in_minus1即图像宽度包含的宏块个数.
                             15 冬评论 高欢 喜欢 热评 大佬我是UDP接收H264流然后解析。可以指导一下嘛?
                                                                                                                  写评论
原创 周排名 总排名 访问 等级
                              【编解码】从零开始写H264解码器(5) SPS解析-参数含义.伪代码 taotao...
                     385
                                                                          和 是是从main profile开始就加入了interlace的交错场功能,也算支持比较早.
    粉丝 获赞 评论 收藏
                             ffmpeg h264解码器分析sps(二)_sps->mb_aff_fantasy_arch的博客-CSDN博 ...
(A) (B) (B) (T) (B) (D)
                                 frame_mbs_only_flag = get_bits1(gb); /*frame_mbs_only_flag 等于 0 表示编码视频序列的编码图像可能是编码场或编码帧。 frame_mbs_only_flag...
                                                                                                       u011110953的专栏 @ 3521
                             音视频 1 H264中SPS帧分辨率的解析
                             自己解析SPS中分辨率的宽和高,具体的语法在方文档ITU-T H.264(03/2005)中有解释,具体在7.4.2.1 序列参数集RBSP语义中。重点字段有: 1、chrom...
                             H264 NAL单元简介
解析H264的SPS信息解决了你的问题么?可以写篇文章记录加深印象哦~
                                          ·
为若干slice,编码以slice为单位进行,即slice之间编码相互独立。每个slice分为若干宏块MB,宏块MB是<mark>H264</mark>编码的基本单位。编码..
                             通过SPS计算视频的宽高及帧率等参数_视频时序参数计算公式_又下雨的...
                             crop_unit_y = 2 - sps->frame_mbs_only_flag; } else if (1 == sps->chroma_format_idc) // 4:2:0 { crop_unit_x = 2; crop_unit_y = 2 * (2 - sps->frame_mbs...
                             H264视频解码器C++工程说明
                                                                                                           jfu22的专栏 ⊙ 5197
                             h264编码概述七(SPS解析)
                                              ·/
格式位于文档的7.3.2.1.1,如下图所示:根据《T-REC-H.264-201402-I!!PDF-E》的附件A.2定义,profiles有如下类型:profil
                             1、片头的field_pic_flag ,指定当前图像是帧编码(0)还是场编码(1)。这个元素在同一图像的所有片中应具有相同值。 2、sps中的frame_mbs_only_f...
FFmpeg解码H264视频裸流(直接可用) ①
                             H264/AVC-编码方式: 帧、场以及宏块帧场自适应 (MBAFF)
                                                                                                       qq 42139383的博客 ① 1478
                             本文主要介绍H264编码方式: 帧、场、帧场自适应
H264关于一帧完整帧被分切成多个Slice时
                             H.264句法和语法总结 (五) 片头句法
                                                                                                       破晓而出一飞冲天 ② 9086
                             slice_header(){ // first_mb_in_slice 片中的第一个宏块的地址,片通过这个句法元素来标定它自己的地址。 要注意的是在帧场自适应模式下,宏块都…
                                                                                                        qq 51210935的博客 ② 635
                             数据压缩实验七: H.264码流分析
                                          res,图像组) 是将一个图像序列中连续的几个图像组成的一个小组,是对编码后的视频码流进行编辑、存取和压缩编码的基本单元,
                             h264中文官方文档__T-REC-H.264_1.pdf
                             h264中文官方文档,h264标准文档,T-REC-H.264-200503-S!!PDF-C.pdf,h264文档(中文版),h264原版最全文档
喜欢口: 大佬我是UDP接收H264流然后解析
                             T-REC-H.264建议书 PDF 中文
                             ITU-T H.264标准 建议书 中文版
iszlittlecat 720: 因为FFmpeg编译脚本没有
                             解析h264 sps c源码
配置 --disable-static \ --enable-shared \ ...
                             H264的sps帧完整解析代码,包含测试代码,实际环境测试通过
zizbruce: CTimer *pTimer = new CTimer ∜∏
                             解析h264 sps获取分辨率
果这个函数调用多次,new的pTimer,能...
                             此资源转载于pudn,经本学渣亲测可用。嘿嘿
                             h264sps解析获取分辨率
                                                                                                                     05-17
cloud_yq: 博主,这种解裸视频流,怎样获
                             h264sps解析获取分辨
取到AVCodecParameters结构体? 我这证...
                             H.264 序列参数集 (SPS)
                                                                                                          键盘敲的劈啪响 ② 241
猿诗人: 比如上图B帧中数据为0x00 0x00 0x
                             H.264 中的 SPS(Sequence Parameter Set)是一种参数集,用于描述视频序列的特征和配置信息。SPS 是在 H.264 视频流中的一个元数据单元,它包...
00 0x01 0x41 0xE2,读取第6位转成二进
                             FFmpeg的H.264解码器源代码简单分析:解析器 (Parser)部分 热门推荐
                                        npeg中libavcodec的H.264解码器(H.264 Decoder)。上篇文章概述了FFmpeg中H.264解码器的结构,从这篇文章开始,具体研究H.26.
                             H.264句法和语法总结 (三) 序列参数集层 (SPS)句法
                             seq_parameter_set_rbsp() { // profile_idc level_idc 指明所用 profile、level profile_idc // constraint_set0_<mark>flag</mark> 等于 1 时表示必须遵从附录 A.2.1 所指明..
 强烈不推荐 不推荐 一般般 推荐 强烈推荐
                                                                                                      qq_42139383的博客 ① 1507
                             H264/AVC-SPS (序列参数集) 解析
                             1.什么是SPS 一个H264码流序列中,每个slice都会有各自的编码参数,其中有些参数在整个码流序列中不会改变,为节省码流,把这类参数提取出来放入...
                             H264码流sps/pps 最新发布
                                          meter Set)和PPS(Picture Parameter Set)是H.264/AVC视频编码标准中的两个重要概念。它们是一些编码参数的集合,用来描述
                                                                 "相关推荐"对你有帮助么?
                                                     关于我们 招贤纳士 商务合作 寻求报道 ☎400-660-0108 ☑ kefu@csdn.net ⑤ 在线客服 工作时间 8:30-22:00
```

▲ 8 📭 🏡 66 😭 🖃 15 🔍 🕏 長栏目录

码龄12年 🛡 暂无认证

关注

23 9万+ 121万+ 13万+

积分

私信

搜博主文章

的合并思路 ① 11360

解析H264的SPS信息

可以指导一下嘛?

C++定时器CTimer的实现

最新评论

C++定时器CTimer的实现 ① 39527

解析H264的SPS信息 @ 22315

iOS硬解码H264视频流 ⊙ 7045

Mac下编译FFmpeg(Android平台)

FFmpeg解码H264视频裸流(直接可用)

H264关于一帧完整帧被分切成多个Slice.

您愿意向朋友推荐"博客详情页"吗?

批量修改/替换文件名称和内容

2020年 3篇

2018年 11篇

── W2Y 关注

最新文章

C++线程池

C++线程

2021年 2篇

2019年 9篇

热门文章



举报