

H.264(八)Slice Header

 blog.csdn.net/qq_40732350/article/details/89605963

1 Slice的概念

H.264的码流结构可以分为两层：

- 网络抽象层NAL：用于传输的二进制码流包
- 视频编码层VCL：保存原始视频的图像数据

H.264的条带：Slice

- 表示视频图像数据的NAL Unit包含的语法元素
- IDR Slice NAL type: 5
- non-IDR Slice NAL type: 1

定义Slice的意义：

- 一个Slice：包含某一帧的全部或部分数据
- 防止误码的扩散
- 不同的slice之间，其解码操作独立；
- 某一个slice的解码过程所参考的数据（例如预测编码）不能越过slice的边界

2 Slice的类型

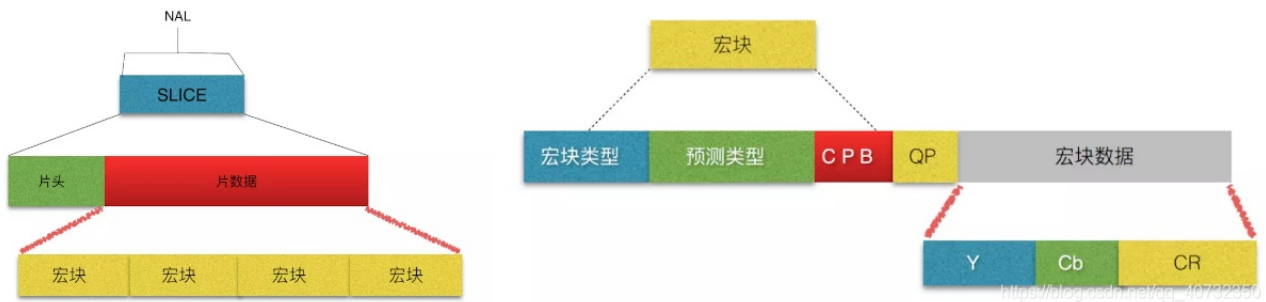
根据码流中不同的数据类型，H.264标准中共定义了5总Slice类型：

1. I slice: 帧内编码的条带；
2. P slice: 单向帧间编码的条带；
3. B slice: 双向帧间编码的条带；
4. SI slice: 切换I条带，用于扩展档次中码流切换使用；
5. SP slice: 切换P条带，用于扩展档次中码流切换使用；

在I slice中只包含I宏块，不能包含P或B宏块；在P和B slice中，除了相应的P和B类型宏块之外，还可以包含I类型宏块。

3 Slice的组成

每一个Slice总体来看都由两部分组成，一部分作为Slice header，用于保存Slice的总体信息（如当前Slice的类型等），另一部分为Slice body，通常是一组连续的宏块结构（或者宏块跳过信息），如下图所示：



4. Slice Header结构

Slice header：中主要保存了当前slice的一些全局的信息，slice body中的宏块在进行解码时需依赖这些信息。其中比较常见的一些语法元素有：

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，宏块都是成对出现，这时本句法元素表示的是第几个宏块对，对应的第一个宏块的真实地址应该是

$$2 * \text{first_mb_in_slice}$$

slice_type：指明片的类型，IDR 图像时, slice_type 等于 2, 4, 7, 9。

pic_parameter_set_id：当前slice所依赖的pps的id；

colour_plane_id：当标识位
separate_colour_plane_flag为true
时，colour_plane_id表示当前的颜色分量，0、1、2分别表示Y、U、V分量。

frame_num：每个参考帧都有一个依次连续的 frame_num 作为它们的标识, 这指明了各图像的解码顺序。但事实上我们在表中可以看到，frame_num 的出现没有 if 语句限定条件，这表明非参考帧的片头也会出现 frame_num。只是当该个图像是参考帧时，它所携带的这个句法元素在解码时才有意义。如表：

slice_type	Name of slice_type
0	P (P slice)
1	B (B slice)
2	I (I slice)
3	SP (SP slice)
4	SI (SI slice)
5	P (P slice)
6	B (B slice)
7	I (I slice)
8	SP (SP slice)
9	SI (SI slice)

图像序号	图像类型	是否用作参考	frame_num
1	I	是	0
2	P	是	1
3	B	否	2
4	P	是	2
5	B	否	3
6	P	是	3
7	B	否	4
8	P	是	4
...

field_pic_flag：场编码标识位。当该标识位为1时表示当前slice按照场进行编码；该标识位为0时表示当前slice按照帧进行编码。

bottom_field_flag：底场标识位。该标志位为1表示当前slice是某一帧的底场；为0表示当前slice为某一帧的顶场。

idr_pic_id：表示IDR帧的序号。某一个IDR帧所属的所有slice，其idr_pic_id应保持一致。IDR 图像的标识。不同的 IDR 图像有不同的 idr_pic_id 值。值得注意的是，IDR 图像有不等价于 I 图像，只有在作为 IDR 图像的 I 帧才有这个句法元素，在场模式下，IDR 帧的两个场有相同的 idr_pic_id 值。idr_pic_id 的取值范围是 [0，65535]，和 frame_num 类似，当它的值超出这个范围时，它会以循环的方式重新开始计数。

pic_order_cnt_lsb：表示当前帧序号的另一种计量方式。

delta_pic_order_cnt_bottom：表示顶场与底场POC差值的计算方法，不存在则默认为0；

slice_qp_delta：指出在用于当前片的所有宏块的量化参数的初始值。

$\text{SliceQPY} = 26 + \text{pic_init_qp_minus26} + \text{slice_qp_delta}$

QPY 的范围是 0 to 51。

我们前文已经提到，H.264 中量化参数是分图像参数集、片头、宏块头三层给出的，前两层各自给出一个偏移值，这个句法元素就是片层的偏移。

整个slice header的结构如下表所示：

slice_header() {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
if(separate_colour_plane_flag == 1)		
colour_plane_id	2	u(2)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(IdrPicFlag)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type == 0) {		
pic_order_cnt_lsb	2	u(v)
if(bottom_field_pic_order_in_frame_present_flag && !field_pic_flag)		
delta_pic_order_cnt_bottom	2	se(v)
}		
if(pic_order_cnt_type == 1 && !delta_pic_order_always_zero_flag) {		
delta_pic_order_cnt[0]	2	se(v)
if(bottom_field_pic_order_in_frame_present_flag && !field_pic_flag)		
delta_pic_order_cnt[1]	2	se(v)

}		
if(redundant_pic_cnt_present_flag)		
redundant_pic_cnt	2	ue(v)
if(slice_type == B)		
direct_spatial_mv_pred_flag	2	u(1)
if(slice_type == P slice_type == SP slice_type == B) {		
num_ref_idx_active_override_flag	2	u(1)
if(num_ref_idx_active_override_flag) {		
num_ref_idx_l0_active_minus1	2	ue(v)
if(slice_type == B)		
num_ref_idx_l1_active_minus1	2	ue(v)
}		
}		
if(nal_unit_type == 20)		
ref_pic_list_mvc_modification() /* specified in Annex H */	2	
else		
ref_pic_list_modification()	2	
if((weighted_pred_flag && (slice_type == P slice_type == SP)) (weighted_bipred_idc == 1 && slice_type == B))		
pred_weight_table()	2	
if(nal_ref_idc != 0)		
dec_ref_pic_marking()	2	
if(entropy_coding_mode_flag && slice_type != I && slice_type != SI)		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)

}		
if(deblocking_filter_control_present_flag) {		
disable_deblocking_filter_idc	2	ue(v)
if(disable_deblocking_filter_idc != 1) {		
slice_alpha_c0_offset_div2	2	se(v)
slice_beta_offset_div2	2	se(v)
}		
}		
if(num_slice_groups_minus1 > 0 && slice_group_map_type >= 3 && slice_group_map_type <= 5)		
slice_group_change_cycle	2	u(v)
}		