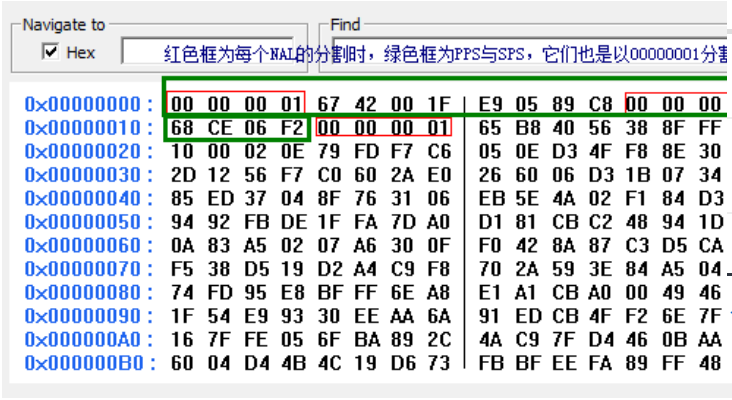
H264(NAL简介与I帧判断)

2016-12-08 15:00 by JG2014, 8864 阅读, 1 评论, 收藏, 编辑

1、NAL全称Network Abstract Layer, 即网络抽象层。

在H.264/AVC视频编码标准中，整个系统框架被分为了两个层面：视频编码层面（VCL）和网络抽象层面（NAL）。其中，前者负责有效表示视频数据的内容，而后者则负责格式化数据并提供头信息，以保证数据适合各种信道和存储介质上的传输。因此我们平时的每帧数据就是一个NAL单元（SPS与PPS除外）。在实际的H264数据帧中，往往帧前面带有00 00 00 01 或 00 00 01分隔符，一般来说编码器编出的首帧数据为PPS与SPS，接着为I帧.....

如下图：



2、如何判断帧类型（是图像参考帧还是I、P帧等）？

NALU类型是我们判断帧类型的利器，从官方文档中得出如下图：

nal_unit_type	NAL类型	C
0	未使用	
1	不分区、非 IDR 图像的片	2, 3, 4
2	片分区 A	2
3	片分区 B	3
4	片分区 C	4
5	IDR 图像中的片	2, 3
6	补充增强信息单元（SEI）	5
7	序列参数集	0
8	图像参数集	1
9	分界符	6
10	序列结束	7
11	码流结束	8
12	填充	9
13..23	保留	
24..31	未使用	

我们还是接着看最上面图的码流对应的数据来层层分析，以00 00 00 01分割之后的下一个字节就是NALU类型，将其转为二进制数据后，解读顺序为从左往右算，如下：

（1）第1位禁止位，值为1表示语法出错

（2）第2~3位为参考级别

（3）第4~8为是nal单元类型

例如上面00000001后有67,68以及65

其中0x67的二进制码为：

0110 0111

4-8为00111，转为十进制7，参考第二幅图：7对应序列参数集SPS

About

昵称: JG2014
园龄: 3年9个月
粉丝: 11
关注: 2
+加关注

最新评论

Re:H264(NAL简介与I帧判断)
图1的工具使什么？ 怎么能看到h.264的数据呢？ -- liyanzhao

Re:iOS-关于微信支付
大神，请指教，总是返回 - 1，错误，retcode = -2, retstr = (null) 我QQ: 841631125 -- wfshjkg

日历

2018年2月 >

日	一	二	三	四	五	六
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	1	2	3
4	5	6	7	8	9	10

随笔分类

推荐排行榜

1. IOS多线程知识总结/队列概念/GCD/主队列/并行队列/全局队列/主队列/串行队列/同步任务/异步任务区别（附代码）(1)
2. iOS开发一字典转模型，KVC设计模式(1)
3. UIKit: UIResponder(转自南峰子博客)(1)
4. iOS-关于微信支付(1)

阅读排行榜

1. H264(NAL简介与I帧判断)(8864)
2. NSCocoaErrorDomain Code=3840 "The operation couldn't be completed. (Cocoa error 3840.)" (7341)
3. NSDictionary或NSArray与JSON字符串相互转换(6829)
4. 直播-拉流和推流概述 转载(6597)
5. IOS多线程知识总结/队列概念/GCD/主队列/并行队列/全局队列/主队列/串行队列/同步任务/异步任务区别（附代码）(6186)

随笔档案

2016年12月(5)
2016年11月(8)
2016年10月(1)
2016年9月(2)
2016年5月(7)
2016年3月(5)
2016年2月(1)
2016年1月(9)
2015年12月(31)
2015年11月(4)
2015年9月(3)
2015年8月(1)
2015年6月(1)
2015年5月(1)
2015年4月(2)
2015年3月(3)
2015年2月(4)
2015年1月(1)
2014年11月(4)
2014年10月(1)
2014年9月(8)
2014年8月(2)
2014年7月(23)
2014年6月(13)
2014年5月(1)
2014年4月(1)

其中0x68的二进制码为：

0110 1000

4-8为01000，转为十进制8，参考第二幅图：8对应图像参数集PPS

其中0x65的二进制码为：

0110 0101

4-8为00101，转为十进制5，参考第二幅图：5对应IDR图像中的片(I帧)

所以判断是否为I帧的算法为： （NALU类型 & 0001 1111） = 5

即 NALU类型 & 31 = 5

比如0x65 & 31 = 5

http://blog.csdn.net/evsqiezi/article/details/8492593

帧格式

H264帧由NALU头和NALU主体组成。

NALU头由一个字节组成,它的语法如下:

```
+---------------+
|0|1|2|3|4|5|6|7|
+-+-+-+-+-+-+-+-+
|F|NRI|  Type   |
+---------------+
```

F: 1个比特.

forbidden_zero_bit. 在 H.264 规范中规定了这一位必须为 0.

NRI: 2个比特.

nal_ref_idc. 取00~11,似乎指示这个NALU的重要性,如00的NALU解码器可以丢弃它而不影响图像的回放,0～3，取值越大，表示当前NAL越重要，需要优先受到保护。如果当前NAL是属于参考帧的片，或是序列参数集，或是图像参数集这些重要的单位时，本句法元素必需大于0。

Type: 5个比特.

nal_unit_type. 这个NALU单元的类型,1～12由H.264使用，24～31由H.264以外的应用使用,简述如下:

0 没有定义
1-23 NAL单元 单个 NAL 单元包
1 不分区，非IDR图像的片
2 片分区A
3 片分区B
4 片分区C
5 IDR图像中的片
6 补充增强信息单元（SEI）
7 SPS
8 PPS
9 序列结束
10 序列结束
11 码流借宿
12 填充
13-23 保留

24 STAP-A 单一时间的组合包
25 STAP-B 单一时间的组合包
26 MTAP16 多个时间的组合包
27 MTAP24 多个时间的组合包
28 FU-A 分片的单元

29 FU-B 分片的单元
30-31 没有定义

AUD

一般文档没有对AUD进行描叙，其实这是一个帧开始的标志，字节顺序为：00 00 00 01 09 f0
从结构上看，有start code, 所以确实是一个NALU，类型09在H264定义里就是AUD（分割器）。大部分播放器可以在没有AUD的情况下正常播放。

紧随AUD，一般是SPS/PPS/SEI/IDR的组合或者简单就是一个SLICE，也就是一个帧的开始。像Flash这样的播放器，每次需要一个完整的帧数据，那么把2个AUD之间的数据按照格式打包给播放器就可以了。

H.264编码时，在每个NAL前添加起始码 0x000001，解码器在码流中检测到起始码，当前NAL结束。为了防止NAL内部出现0x000001的数据，h.264又提出'防止竞争 emulation prevention"机制，在编码完一个NAL时，如果检测出有连续两个0x00字节，就在后面插入一个0x03。当解码器在NAL内部检测到0x000003的数据，就把0x03抛弃，恢复原始数据。

0x000000 >>>>>> 0x00000300
0x000001 >>>>>> 0x00000301
0x000002 >>>>>> 0x00000302
0x000003 >>>>>> 0x00000303

总的来说H264的码流的打包方式有两种,一种为annex-b byte stream format 的格式，这个是绝大部分编码器的默认输出格式，就是每个帧的开头的3~4个字节是H264的start_code,0x00000001或者0x000001。
另一种是原始的NAL打包格式，就是开始的若干字节（1，2，4字节）是NAL的长度，而不是start_code,此时必须借助某个全局的数据来获得编 码器的profile,level,PPS,SPS等信息才可以解码。

SPS,PPS的解析

SPS

profile_idc和level_idc是指比特流所遵守的配置和级别。

constraint_set0_flag 等于1是指比特流遵从某节中的所有规定。constraint_set0_flag 等于0是指该比特流可以遵从也可以不遵从某节中的所有规定。当profile_idc等于100、110、122或144时，constraint_set0_flag、constraint_set1_flag和constraint_set2_flag都应等于0。

log2_max_frame_num_minus4的值应在0-12范围内（包括0和12），这个句法元素主要是为读取另一个句法元素 frame_num 服务的，frame_num 是最重要的句法元素之一，它标识所属图像的解码顺序 。这个句法元素同时也指明了 frame_num 的所能达到的最大值： MaxFrameNum = 2*exp(log2_max_frame_num_minus4 + 4) 。

pic_order_cnt_type 是指解码图像顺序的计数方法。pic_order_cnt_type 的取值范围是0到2（包括0和2）。

log2_max_pic_order_cnt_lsb_minus4表示用于某节规定的图像顺序数解码过程中的变量MaxPicOrderCntLsb的值，

num_ref_frames规定了可能在视频序列中任何图像帧间预测的解码过程中用到的短期参考帧和长期参考帧、互补参考场对以及不成对的参考场的最大数量。num_ref_frames 的取值范围应该在0到MaxDpbSize。

gaps_in_frame_num_value_allowed_flag 表示某节给出的frame_num 的允许值以及在某节给出的frame_num 值之间存在推测的差异的情况下进行的解码过程。

pic_width_in_mbs_minus1加1是指以宏块为单元的每个解码图像的宽度。pic_height_in_map_units_minus1 的语义依赖于变量 frame_mbs_only_flag，规定如下：-— 如果 frame_mbs_only_flag 等于0，

pic_height_in_map_units_minus1加1就表示以宏块为单位的一场的高度。-- 否则（frame_mbs_only_flag等于1），
pic_height_in_map_units_minus1加1就表示

以宏块为单位的一帧的高度。变量 FrameHeightInMbs 由下列公式得出：
FrameHeightInMbs = (2 – frame_mbs_only_flag) * PicHeightInMapUnits。

mb_adaptive_frame_field_flag 等于0表示在一个图像的帧和场宏块之间没有交换。mb_adaptive_frame_field_flag 等于1表示在帧和帧内的场宏块之间可能会有交换。当mb_adaptive_frame_field_flag没有特别规定时，默认其值为0。

direct_8x8_inference_flag 表示在某节中规定的B_Skip、B_Direct_16x16和B_Direct_8x8亮度运动矢量的计算过程使用的方法。当
frame_mbs_only_flag 等于0时

direct_8x8_inference_flag 应等于1。

frame_cropping_flag 等于1表示帧剪切偏移参数遵从视频序列参数集中的下一个值。frame_cropping_flag 等于0表示不存在帧剪切偏移参数。

vui_parameters_present_flag 等于1 表示存在如附录E 提到的 vui_parameters( ) 语法结构。vui_parameters_present_flag 等于0表示不存在如附录E提到的vui_parameters( ) 语法结构。

PPS
seq_parameter_set_id是指活动的序列参数集。变量seq_parameter_set_id 的值应该在0到31的范围内（包括0和31）。

entropy_coding_mode_flag 用于选取语法元素的熵编码方式，在语法表中由两个标识符代表，具体如下：如果entropy_coding_mode_flag 等于0，那么采用语法表中左边的描述符所指定的方法。
pic_order_present_flag等于1 表示与图像顺序数有关的语法元素将出现于条带头中，pic_order_present_flag 等于0表示条带头中不会出现与图像顺序数有关的语法元素。

num_slice_groups_minus1加1表示一个图像中的条带组数。当
num_slice_groups_minus1 等于0时，图像中所有的条带属于同一个条带组。

num_ref_idx_l0_active_minus1表示参考图像列表0 的最大参考索引号，该索引号将用来在一幅图像中num_ref_idx_active_override_flag 等于0 的条带使用列表0 预测时，解码该图像的这些条带。当MbaffFrameFlag等于1时，num_ref_idx_l0_active_minus1 是帧宏块解码的最大索引号值，而2 *num_ref_idx_l0_active_minus1 + 1是场宏块解码的最大索引号值。
num_ref_idx_l0_active_minus1 的值应该在0到31的范围内（包括0和31）。

weighted_pred_flag等于0表示加权的预测不应用于P和SP条带。
weighted_pred_flag等于1表示在P和SP条带中应使用加权的预测。

weighted_bipred_idc等于0表示B条带应该采用默认的加权预测。
weighted_bipred_idc等于1表示B条带应该采用具体指明的加权预测。
weighted_bipred_idc 等于2表示B 条带应该采用隐含的加权预测。
weighted_bipred_idc 的值应该在0到2之间（包括0和2）。

pic_init_qp_minus26表示每个条带的SliceQPY 初始值减26。当解码非0值的slice_qp_delta 时，该初始值在条带层被修正，并且在宏块层解码非0 值的mb_qp_delta 时进一步被修正。pic_init_qp_minus26 的值应该在 - (26 + QpBdOffsetY) 到 +25之间（包括边界值）。

pic_init_qs_minus26表示在SP 或SI 条带中的所有宏块的SliceQSY 初始值减26。当解码非0 值的slice_qs_delta 时，该初始值在条带层被修正。
pic_init_qs_minus26 的值应该在-26 到 +25之间（包括边界值）。

chroma_qp_index_offset表示为在QPC 值的表格中寻找Cb色度分量而应加到参数QPY 和 QSY 上的偏移。chroma_qp_index_offset的值应在-12 到 +12范围内（包括边界值）。

deblocking_filter_control_present_flag等于1 表示控制去块效应滤波器的特征的一组语法元素将出现在条带头中。
deblocking_filter_control_present_flag 等于0 表示控制去块效应滤波器的特征的一组语法元素不会出现在条带头中，并且它们的推定值将会生效。

constrained_intra_pred_flag等于0 表示帧内预测允许使用残余数据，且使用帧内宏块预测模式编码的宏块的预测可以使用帧间宏块预测模式编码的相邻宏块的解码样值。constrained_intra_pred_flag 等于1 表示受限制的帧内预测，在这种情况下，使用帧内宏块预测模式编码的宏块的预测仅使用残余数据和来自I或SI宏块类型的解码样值。

redundant_pic_cnt_present_flag等于0 表示redundant_pic_cnt 语法元素不会在条带头、图像参数集中指明（直接或与相应的数据分割块A关联）的数据分割块B和数据分割块C中出现。redundant_pic_cnt_present_flag等于1表示redundant_pic_cnt 语法元素将出现在条带头、图像参数集中指明（直接或与相应的数据分割块A关联）的数据分割块B和数据分割块C中。

分包

h264包在传输的时候，如果包太大，会被分成多个片。NALU头会被如下的2个自己代替。

The FU indicator octet has the following format:

```
+---------------+
|0|1|2|3|4|5|6|7|
+-+-+-+-+-+-+-+-+
|F|NRI|  Type   |
+---------------+
```

别被名字吓到这个格式就是上面提到的RTP h264负载类型，Type为FU-A

The FU header has the following format:

```
+---------------+
|0|1|2|3|4|5|6|7|
+-+-+-+-+-+-+-+-+
|S|E|R|  Type   |
+---------------+
```

S bit为1表示分片的NAL开始，当它为1时，E不能为1
E bit为1表示结束，当它为1，S不能为1

R bit保留位

Type就是NALU头中的Type,取1-23的那个值

H.264先进的视频编译码标准

http://blog.csdn.net/gl1987807/article/details/11945357

H.264先进视讯编译码标准

郭其昌/工研院电通所

1. 前言

在2001年12月，ITU-T VCEG与ISO MPEG共同组成联合视讯小组(Joint Video Term，JVT)来研订新的视频压缩格式，此新格式在ITU-T组织中称为H.264，在ISO组织中则纳入MPEG-4 Part-10 (ISO/IEC 14496-10)并命名为Advanced VideoCoding (AVC)，通常合并称为H.264/AVC [1]，其国际标准的

第一版于2003年公布，而增修的第二版也于2005年3月定案。相关研究显示H.264/AVC与MPEG-2及MPEG-4相较之下，无论是压缩率或视讯质量皆有大幅的提升[2]，而且H.264/AVC也首次将视讯编码层(Video Coding Layer，VCL)与网络提取层(Network Abstraction Layer，NAL)的概念涵盖进来，以往视讯标准着重的是压缩效能部分，而H.264/AVC包含一个内建的NAL网络协议适应层，藉由NAL来提供网络的状态，可以让VCL有更好的编译码弹性与纠错能力，使得H.264/AVC非常适用于多媒体串流(multimedia streaming)及行动电视(mobile TV)的相关应用。在第一版的标准规范中，H.264/AVC根据使用的编码工具种类来提供三种编码规模(Profile)，如表1所示分别为Baseline Profile、Main Profile、Extension Profile，而相对应的影片尺寸与比特率等级由Level 1至Level 5.1，涵盖小画面与高分辨率画面的应用范围。Baseline Profile主要是着眼于低比特率的应用(例如：影像通讯)，而且其运算复杂度低，所以也适合应用于个人随身的多媒体拨放机；Main Profile因为有支持交错式影片(interlaced content)的编码，所以适合应用于HDTV数字电视广播，而且非常容易整合在传统的MPEG-2 Transport/Program Stream上来传送H.264/AVC比特流；对于IP-TV或是MOD(Multimedia On Demand)等应用，使用包含高抗错性编码工具(error resilient tools)的Extension Profile即可以满足这些需求。然而，微软□公司在2003年将其视频压缩技术向美国的电影电视工程师协会(Society of Motion Picture and Television Engineers，SMPTE)提出公开标准化的申请，并以VC-1(Video Codec 1)为此新标准的命名[3]，由于VC-1在高分辨率影片上的表现出色，导致H.264/AVC在DVD Forum与Blu-ray Disc Association的高分辨率DVD影片测试中败阵下来，其主要原因是H.264/AVC使用较小尺寸的转换公式与无法调整的量化矩阵，造成不能完整保留影像的高频细节信息，因此H.264/AVC于2004年展开标准增修的讨论，来纳入称之为Fidelity Range Extensions (FREXt) [4]的新编码工具，并以先前MainProfile为基础来扩充增加4个新的等级(Table 1)，期望能够在高分辨率影片的应用上扳回劣势，目前增修的H.264/AVC第二版标准已于2005年3月发表。本文后段将探讨网络提取层的相关特性，接着来说明视讯编码层的原理，最后并讨论H.264/AVC的应用现况。

2. 网络提取层 (Network Abstraction Layer，NAL)

H.264/AVC标准的特色是将网络提取层的概念涵盖进来，亦即以NAL封包为单位的方式来做为VCL编译码的运算单位，这样传输层拿到NAL封包之后不需要再进行切割，只需附加该传输协议的文件头信息(adding header only)就可以交由底层传送出去，如图1所示，可以将NAL当成是一个专作封装(packaging)的模块，用来将VCL压缩过的bitstream封装成适当大小的封包单位(NAL-unit)，并在NAL-unit Header中的NAL-unit Type字段记载此封包的型式，每种型式分别对应到VCL中不同的编解碼工具。NAL另外一个重要的功能为当网络发生壅塞而导致封包错误或接收次序错乱(out-of-order)的状况时，传输层协议会在Reference Flag作设定的动作，接收端的VCL在收到这种NAL封包时，就知道要进行所谓的纠错运算(error concealment)，在解压缩的同时也会尝试将错误修正回来。如图2所示，一个完整的H.264/AVC bitstream是由多个NAL-units所组成的，所以此bitstream也称之为NAL unit stream，一个NAL unit stream内可以包含多个压缩视讯序列(coded video sequence)，一个单独的压缩视讯序列代表一部视讯影片，而压缩视讯序列又是由多个access units所组成，当接收端收到一个access unit后，可以完整地译码成单张的画面，而每个压缩视讯序列的第一个access unit必须为Instantaneous Decoding Refresh (IDR) access unit，IDRaccess unit的内容全是采用intra-prediction编码，所以自己本身即可完全译码，不用参考其他access unit的数据。access unit亦是由多个NAL-units所组成，标准中总共规范12种的NAL-unit型式，这些可以进一步分类成VCL NAL-unit及non-VCL NAL-unit，所谓的VCL NAL-unit纯粹是压缩影像的内容，而所谓的non-VCL NAL-unit则有两种：Parameter Sets与Supplemental Enhancement Information (SEI)，SEI可以存放影片简介、版权宣告、用户自行定义的数据…等；Parameter Sets主要是描述整个压缩视讯序列的参数，例如：长宽比例、影像显现的时间点(timestamp)、相关译码所需的参数…等，这些信息非常重要，万一在传送的过程中发生错误，会导致整段影片无法译码，以往像

MPEG-2/-4都把这些信息放在一般的packet header，所以很容易随着packet loss而消失，现在H.264/AVC将这些信息独立出来成为特殊的parameter set，可以采用所谓的out-of-band的方式来传送，以便将out-of-band channel用最高层级的信道编码(channel coding)保护机制，来保证传输的正确性。

3. 视讯编码层 (Video Coding Layer，VCL)

视频压缩的原理是利用影像在时间与空间上存有相似性，这些相似的数据经过压缩算法处理之后，可以将人眼无法感知的部分抽离出来，这些称为视觉冗余(visual redundancy)的部分在去除之后，就可以达到视频压缩的目的。如图1所示，H.264/AVC的视讯编码机制是以图块(block-based)为基础单元，也就是说先将整张影像分割成许多矩形的小区域，称之为巨图块(macroblock，MB)，再将这些巨图块进行编码，先使用画面内预测(intra-prediction)与画面间预测(inter-prediction)技术，以去除影像之间的相似性来得到所谓的差余影像(residual)，再将差余影像施以空间转换(transform)与量化(quantize)来去除视觉冗余，最后视讯编码层会输出编码过的比特流(bitstream)，之后再包装成网络提取层的单元封包(NAL-unit)，经由网络传送到远程或储存在储存媒体中。H.264/AVC允许视讯影片以frame或是以filed的方式来进行编码，两者可以共存，而frame可以是progress或是interlace形式，对同一段影片来说也可使用两者来混合编码，这个特性与MPEG-2相同。而在影像色彩格式的支持上，H.264/AVC第一版的标准只支持YCrCb 4:2:0取样的方式，而在增修的第二版标准中增加4:2:2与4:4:4取样格式，通常这些格式会被数字电影或HDTV影片所采用。

3.1 H.264/AVC影像格式阶层架构

H.264/AVC的阶层架构由小到大依序是sub-block、block、macroblock、slice、slicegroup、frame/field-picture、sequence。对一个采用4:2:0取样的MB而言，它是由16x16点的Luma与相对应的2个8x8点Chroma来组成，而在H.264/AVC的规范中，MB可再分割成多个16x8、8x16、8x8、8x4、4x8、4x4格式的sub-blocks。所谓的slice是许多MB的集合，而一张影像是由许多slice所组成(图3)，slice为H.264/AVC格式中的最小可译码单位(self-decodable unit)，也就是说一个slice单靠本身的压缩数据就能译码，而不必依靠其他slice，这样的好处是当传送到远程时，每接收完一笔slice的压缩数据就能马上译码，不用等待整张的数据接收完后才能开始，而且万一传送的过程中发生数据遗失或错误，也只是影响该笔slice，不会对其他slice有所影响，但跟MPEG-2的slice不同处在于它允许slice的范围可以超过一行MB，也就是说H.264/AVC允许整张影像只由单一个slice组成。H.264/AVC的slice架构还有一项特性称为Flexible Macroblock Ordering (FMO)，也就是说组成slice的MB可以不必局限于循序扫描(rasterscan)的排列方式，例如：图3最右侧的排法就非常适用于多个前景(foreground) slice groups与一个独自的背景(background) slice group，好处是对不同的slice group可以用不同质量的压缩参数，例如：对于前景物件通常是人眼较感兴趣的区域，可以用较小的压缩率来维持较好的质量。

3.2 Slice的编码模式

H.264/AVC的slice依照编码的类型可以分成下列种类：(1)I-slice: slice的全部MB都采用intra-prediction的方式来编码；(2) P-slice: slice中的MB使用intra-prediction和inter-prediction的方式来编码，但每一个inter-prediction block最多只能使用一个移动向量；(3) B-slice:与P-slice类似，但每一个inter-prediction block可以使用二个移动向量。比较特别的是B-slice的'B'是指Bi-predictive，与MPEG-2/-4 B-frame的Bi-directional概念有很大的不同，MPEG-2/-4 B-frame被限定只能由前一张和后一张的I(或P)-frame来做inter-prediction，但是H.264/AVC B-slice除了可由前一张和后一张影像的I(或P、B)-slice外，也能从前二张不同影像的I(或P、B)-slice来做inter- prediction，而H.264/AVC另外增加两种特殊slice类型：(1) SP-slice:即所谓的Switching P slice，为P-slice的一种特殊类型，用来串接两个不同bitrate的bitstream；(2) SI-slice: 即所谓的Switching I slice，为I-slice的一种特殊类型，除了用来串接

两个不同content的bitstream外，也可用来执行随机存取(random access)来达到网络VCR的功能。这两种特殊的slice主要是考虑当进行Video-On-Demand streaming的应用时，对同一个视讯内容的影片来说，server会预先存放不同bitrate的压缩影片，而当带宽改变时，server就会送出适合当时带宽比特率的影片，传统的做法是需要等到适当的时间点来传送新的I-slice (容量较P-slice大上许多)，但因为带宽变小导致需要较多的时间来传送I-slice，如此会让client端的影像有所延迟，为了让相同content但不同bitrate的bitstream可以较平顺地串接，使用SP-slice会很容易来达成(图4)，不仅可以直接送出新的bitstream，也因为传送的P-slice的容量较小，所以不会有时间延迟的情形出现。当client端的使用者要切换到新的接收频道(channel)时，因为与目前传送的bitstream不但内容不同连比特率也不同，传统的做法需让client重新缓冲(buffering)一段新频道的内容(图5)，此时是为了要接收新频道bitstream的I-slice，然后再开始传送新频道bitstream后续的P-slice，如此client也会发生延迟接收的现象，而且当client要进行所谓的快转、倒转、随机存取(random access)的动作时，传统的做法无法达到实时的反应，H.264/AVC利用SI-slice就可以轻易地达到目的。

3.3画面内预测技术(Intra-frame Prediction)

以往的压缩标准在进行intra-prediction时，多半只是将转换系数做差值编码，而H.264/AVC在空间领域(spatial domain)来进行像点之间的预测，而不是用转换过的系数，它提供两种intra-prediction的型式：intra_4x4及intra_16x16，所谓的intra_4x4是以Luma 4x4 sub-block为单位，找出它的参考对象(predictor)后，再将其与参考对象相减后所产生的差余影像(residual)送入转换算法，而寻找参考对象的模式共有9种预测的方向(图6)，以mode 0 (vertical)为例，{a,e,i,m}、{b,f,j,n}、{c,g,k,o}、{d,h,l,p}的参考对象分别为A、B、C、D；Luma intra_16x16与Chroma的模式跟Luma intra_4x4类似，详细的运算公式可以参考[1]。

3.4画面间预测技术(Inter-frame Prediction)

至于横跨每张画面之间的预测技术，H.264/AVC提供了更丰富的编码模式，计有下述几种区块分割(partition)的方法：16x16、16x8、8x16、8x8、8x4、4x8、4x4，多样的分割方式可以让移动向量的预测更准确，如图7所示，画面中有些移动的区域并不是正方形，使用长方形或较小的4x4分割来做预测的区域，可以大幅降低差余影像的数值来增加了压缩比，但也因此P-slice中的MB最多可有16个移动向量(motion vector)，而B-slice中的MB最多可拥有32个移动向量，虽然这些会增加移动向量档头(header)的容量，但整体来说对压缩比仍有正面的益处。再者，以往的压缩标准所使用的动态估测(motion estimation)，只有使用前一张图像来作为预测的对象，H.264/AVC提供了多重参考图像(multiple reference frames)的概念，使得移动向量不再只限于前后相邻的影像，而是可以跨过多张影像，如图8所示，在时间点t的图块，可以使用t-1到t-2图像中的图块来作为预测的对象，当影片有周期重复性的内容时，例如：背景影像周期性的出现或被遮盖、对象有来回跳动的行为、形状忽大忽小，或是摄影机在拍摄时，因为有多处的取景点，并且摄影画面在取景点之间来回移动，这种情形在球类比赛转播时常出现，这些状况都能得到较好的动态预测结果，因而提高了压缩的效能。

3.5 转换、量化与熵编码算法 (Transform, Quantization, and EntropyCoding)

H.264/AVC的转换算法采用所谓的4x4与8x8整数转换，跟MPEG-2/-4的8x8 DCT(Discrete Cosine Transform)有很大的不同，因为是整数运算的缘故，不像小数运算的DCT有系数还原后无法匹配的问题，而且以4x4的区块大小来进

行转换也可减低区块效应的程度。在量化技术方面，H.264/AVC只使用加法与乘法而没有除法运算，有利于集成电路的实现。跟以往MPEG-2/-4的熵编码技术(entropy coding)不同的是，H.264/AVC针对量化过的转换系数与非转换系数数据(文件头数据、移动向量...等)，分别使用二个不同的编码法则。非转换系数数据使用单一个的编码表，好处是可以节省编码表所占用的内存空间；针对量化过的转换系数数据来说，不像MPEG-2/-4对每种影像都使用固定的编码表，H.264/AVC使用所谓的内容适应性编码技术(context-adaptive)，也就是会根据编码的内容来统计某些代码(code-word)的出现机率，而产生一个最适合于目前影像的编码表，好处是能够提高压缩比，但要使用额外的带宽来传送这些编码表。H.264/AVC内容适应性编码技术有两种：Context Adaptive Variable Length Coding (CAVLC)以及ContextAdaptive Arithmetic Binary Coding (CABAC)，CAVLC的基本原理跟MPEG-2/-4的VLC相同，而CABAC的复杂度比CAVLC高，但却可以提供较高的压缩比，尤其是用在压缩交错式的数字电视影片。

3.6 内嵌式去区块效应滤波器(In-Loop De-blocking Filter)

先前有提到H.264/AVC也是一种block-based的压缩方法，所以会有区块效应(blocking-effect)的现象，虽然它采用4x4转换可以稍减区块效应的程度，但是在影像较平滑的区域，仍需依靠去区块效应滤波器来做影像质量的修补。通常去区块效应滤波器分成两种：post filter及in-loop filter，所谓post filter就是在译码的流程之后再进行的，而不在解压缩标准的规范中，好处是厂商可以依应用的复杂度，有弹性地决定滤波器的实现方式，而所谓的in-loop filter就是直接规范在编译码的流程中，虽然会增加复杂度，但由于经过滤波器处理后的影像质量较好，若以此作为画面间预测的参考图像，其预测精确度会大幅的提升，因而增加了压缩比。

4. 结论

由于H.264/AVC在视讯编码算法上的改进，其压缩比及视讯质量与MPEG-2/-4相较下有大幅度的提升，而其NAL概念有助于在有限带宽的传输通道上来传送高质量的视讯内容，此外，对于高画质数字电视或高画质DVD，以H.264/AVC的编码技术都可以很轻易地满足应用需求，但就市场面来看，VC-1标准凭借着微软□在PC平台的优势与低价授权的策略，今后将成为H.264/AVC最强大的挑战者。

« 上一篇：iOS面向编码|iOSVideoToolbox：读写解码回调函数CVImageBufferRef的YUV图像

最新IT新闻:
· 2018年最值得关注的15大技术趋势
· 微软精心打造的Vista系统，为什么死得这么快?
· 除了微信淘宝这是中国人最爱用的App 安装量逆天
· 看完猎鹰重型火箭的发射视频 你能算出这道题吗?
· 管理者在数据分析上常犯的9个错误
» 更多新闻...

最新知识库文章:
· 作为一个程序员，数学对你到底有多重要
· 领域驱动设计在互联网业务开发中的实践
· 步入云计算
· 以操作系统的角度述说线程与进程
· 软件测试转型之路
» 更多知识库文章...

历史上的今天:
2015-12-08 Auto Layout 使用心得