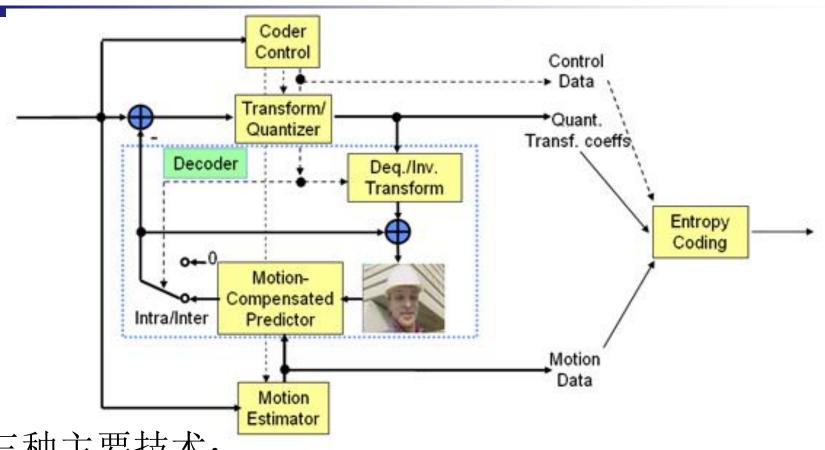
第九章 视频压缩编码及国际标准

主要内容

- 9.1 视频压缩的基本框架
- 9.2 视频压缩标准制定的过程
- 9.3 如何快速掌握视频压缩标准?
 - > 9.3.1 MPEG-2的语法结构
 - > 9.3.2 H.264的语法结构
- 9.4 视频压缩标准在应用中的问题
 - > 9.3.1 码率控制和缓冲区管理
 - > 9.3.2 视频转码
- 9.5 主要的视频压缩标准介绍

9.1 视频压缩的基本框架



三种主要技术:

变换, 去除空间冗余 基于运动补偿的帧间预测,去除时间冗余 熵编码, 去除统计冗余

9.2 标准制定的过程

■ 以ISO为代表,ISO和IEC成立了ISO-IEC/JTC1(联合技术委员会),JTC1分为若干个子委员会(SC)

JTC中的标准委员会SC29中包括了几个著名的工作组

- WG1: JBIG (Joint Bi-level Image experts Group), JPEG (Joint Photographic Experts Group)
- WG11: MPEG (Moving Picture Experts Group) www.mpeg.org
- WG12: MHEG (Multimedia & Hypermedia information coding Experts Group)

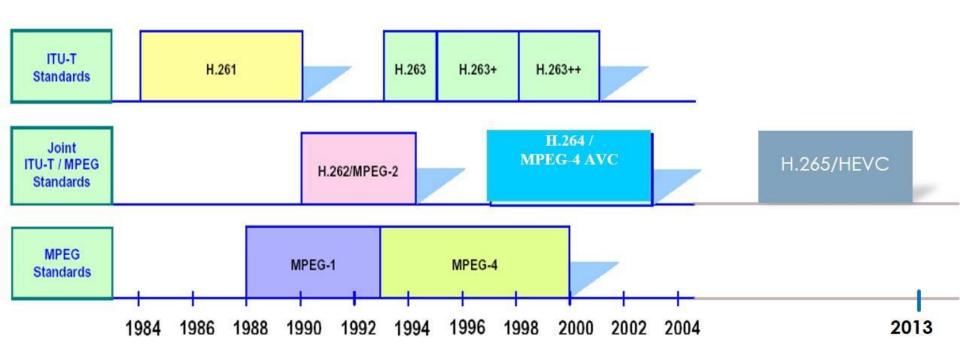
以计算机为中心,强调存储和回放。

■ 以ITU为代表,成立了ICG/AVMMS(视听多媒体业务 联合协调组)

以通信为基础,强调通信网络上多媒体信息的传输。

标准	组织	目标	通过日 期
H.261	ITU-T	运行在p*64kbps(p=1-30) 速率上的编码解码器	1993.3
MPEG-1		运动图像和音频的存储和回 放	1992.1 1
MPEG-2	ISO/IEC	数字电视	1994.1 1
MPEG-3		可伸缩的编码解码器	取消
MPEG-4		多媒体应用	1998.1 0
MPEG-7		基于内容表示的多媒体信息 检索、过滤、管理和处理	2003.1
MPEG-21		多媒体框架	进展
H.264/MPE G-4 AVC	JVT	移动多媒体应用	2003
H.265	JVT	进展中	

9.2 标准制定的过程



9.3 如何快速掌握视频编码标准

■压缩原理

- ▶基本的方法,没有脱离DCT+运动补偿预测+ 熵编码的混合编码框架
- ► 标准中增加了哪些编码工具以进一步提高运 算速度和编码效率
 - DCT变换-〉整数变换
 - ■增加帧内预测方法
 - ■增加帧间预测模式和子像素预测
 - ■增加多参考帧
 - ■运动矢量的预测编码
 - ■提高熵编码的效率

9.3 如何快速掌握视频编码标准

- 调试视频解码器,逐步分析各个功能模块
 - > 理解解码器的输入输出参数
 - > 刚开始先走大的模块,再逐步深入到细节
- ■再调试视频编码器
 - 着重分析编码决策模块,如运动估计、模式选择和码率控制部分
 - ➤ 关键的算法还需进一步学习有关理论知识 http://www.zenith.com/sub_hdtv/mpeg_tutorial/

	视频序列		视频序列				视频序列				
亨列SC	序列头	序列扩展	曼 GOI	PSC GO	OP头	图象数据	序列SC	序列头	序列扩展		序列纠错
						<u> </u>					
图象SC	图象头	图象扩展	夏 I帧数	:据 图:	象SC	图象头	图象扩展	P帧数排	居		B帧数据
			1		_						
条SC	条头	宏块	宏块	宏块	••••	•••••	条SC	条头	宏块	宏块	宏块

宏块类	型 运动矢	量系数	块 系数	块 系	数块	系数块	系数块	系数块	系数块	•••••	•••••
							1				

- ■借助码流分析软件解析比特流文件
- MPEG-2语法结构(基本流)
 - 》视频序列以序列起始码开始,以序列结束码结束。序列头定义了图象的垂直和水平尺寸,宽高比,色度子采样格式,图象速率,逐行还是隔行扫描,类,级和比特率,以及帧内帧间编码所使用的量化表。
 - 》如果没有序列头数据,一个解码器无法理解 比特流,因此序列头是解码器开始操作的入 点

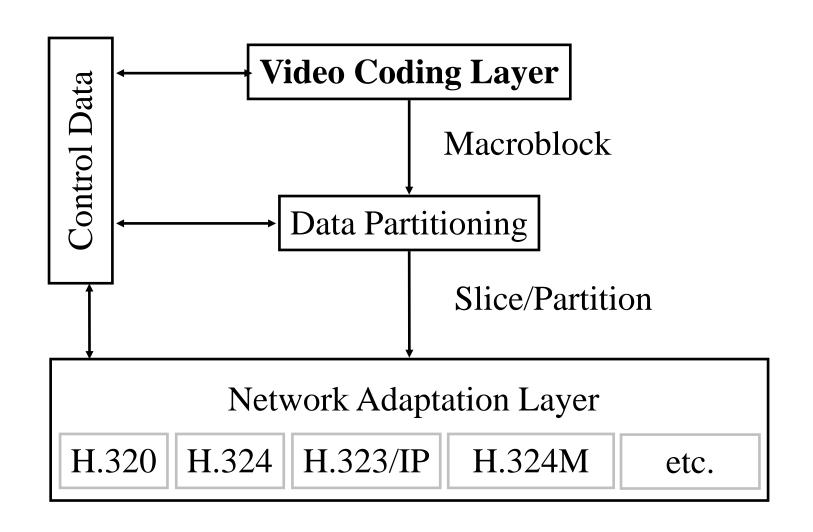
■ MPEG-2语法结构(基本流)

- 》像条组合成图象帧。图象层起始码后的图象 头中给出时间参考信息、图象编码类型和视 频缓存延时信息。图象头后面的图象扩展码 还给出了图象结构(顶场、底场或帧)、量 化因子类型和VLC等信息
- ➤ 若干图象构成GOP。GOP可以是开放或封闭的。在一个封闭的GOP中,最后一个B帧不需要下一个GOP中的I帧作为参考。GOP头中给出时间码、B图象的预测特性和编辑后的B图象能够正确解码的信息

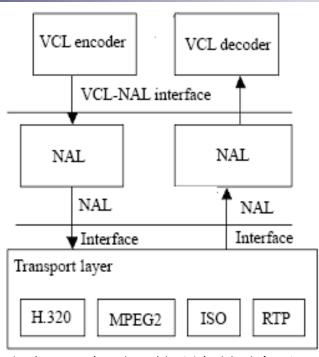
■ MPEG-2语法结构(基本流)

- ▶ DCT块是图象信息的基本单元。DC系数先送出,然后是AC系数,EOB结束DCT块组合成宏块,是运动预测和补偿的基本单元。每个宏块在头标记中有宏块属性和一个二维运动矢量
- > 宏块组合成像条。在MPEG-2中,像条可以从任何地方开始,可以是任意大小,但在ATSC中,必须从图象的左边缘开始。像条是同步的基本单元。因为像条起始码以及上三层的起始码前缀均为16进制的000001,由后面的8bit区分出不同的起始码值,所以可实现相应的数据同步

9.3.2 H.264语法结构的设计

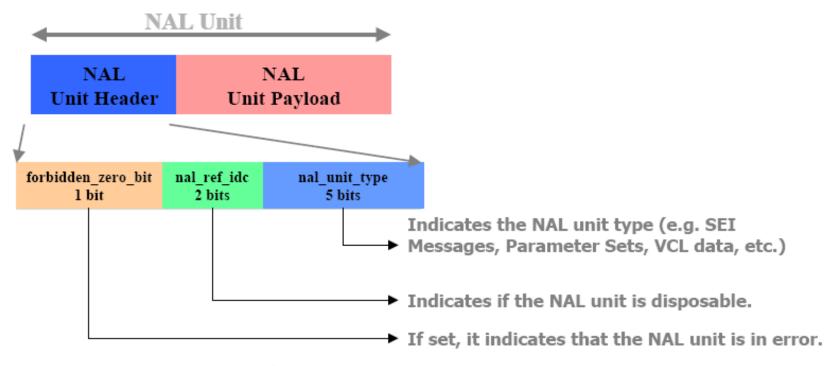


NAL(Network Abstraction Layer)



- NAL用于定义适合于任何信道传输和存储的数据格式,并 提供头信息,从而为视频编码提供与外界的统一接口
- 所有的数据都包含于NAL单元(NAL units)中,每个NAL单元包含整数个字节的数据。
- 如果在有效载荷中包含位错误,则在NALU的头部设置标识

NAL语法结构



- NAL单元由一个字节的头部和有效载荷组成
- 在**包交换系统**中可直接处理NAL单元
- 在电路交换系统中通过定义一种将NAL单元变换成这种流格式的变换机制:使用符合常规视频编码标准的起始码(start code)将NAL单元封装起来,如加入StartCode 0x000001

NAL的处理过程

- 将VCL层输出的SODB(String Of Data Bits) 封装成nal_unit.
 - > SODB字节对齐处理后封装成RBSP(Raw Byte Sequence Payload)
 - > 为防止RBSP的字节流与有序字节流传送方式下的 SCP(start_code_prefix_one_3bytes, 0x000001)出 现字节竞争情形,循环检测RBSP前三个字节,在出 现字节竞争时在第三字节前加入 emulation_prevention_three_byte(0x03)
 - ➤ 防字节竞争处理后的RBSP再加一个字节的 header(forbidden_zero_bit+ nal_ref_idc+ nal_unit_type), 封装成nal_unit.
- 针对不同的传送网络(电路交换|包交换),将nal_unit 封装成针对不同网络的封装格式。

序列参数集和图像参数集

- 几乎所有视频编码标准的编码结构都是基于树状的分层结构,如图像组层,图像层,片层,宏块层和块层。在以往的标准中,每一层都有头部信息描述该层中的重要信息,数据部分包含该层的数据。一旦头部信息丢失,就会导致该层及以下各层的所有数据不可用
- 将片层以上很少变化的系统级信息放在序列参数 集(Sequence parameter set, SPS)和图像 参数集(Picture parameter set, PPS)中传送
- ■由于参数集中的信息大多是系统级信息,与图像 具体内容不相关,因此需要在编解码端建立会话 连接时进行传送或者在会话过程中以可靠的异步 传送方式进行参数集更新

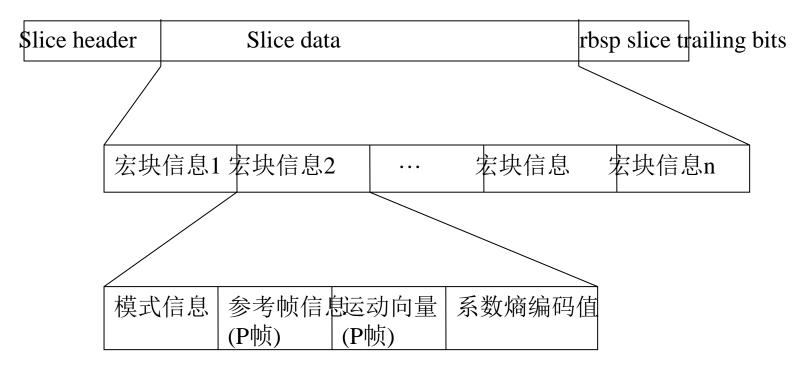
序列参数集

- 序列参数集语法结构说明的是某一段视频序列 的编码特性
 - ▶ 首先需要使用profile idc,level_idc两个语法元素说明该段视频序列遵循的是H.264/AVC标准的哪一种配置,以指导解码端调用合适的解码器进行解码
 - ▶ 其次需要使用pic order_cnt type等语法元素说明该段视频序列中每帧图像的记数方法,也需要使用pic_widthes in-mbs-minusl和height in-mbs-minusl这两个语法元素说明该段视频序列中图像帧的长度和宽度
 - ➤ 各段视频序列的编码特性可能不同,因此序列参数 集的内容也就不会相同,也就需要一个 Sequence_parameter_set_id语法元素来区分不同 的序列参数集,以方便解码端进行参考。

图像参数集

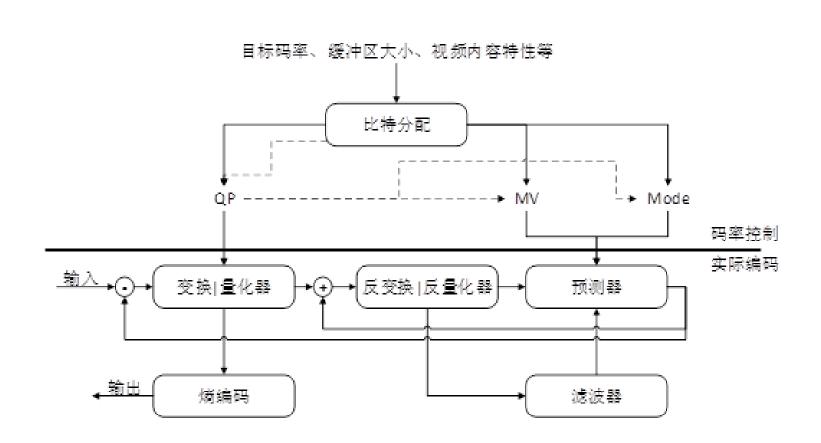
- ■说明的是某个图像帧的编码特性
 - ▶ 包括编码该图像帧所采用的熵编码模式(CAVLC或者CABAC),所使用的参考帧的数目,以及对该图像帧的亮度和色度残差进行量化时所采用的量化参数值等
 - ➤ 图像参数集中最主要的部分是该图像帧的Slice Group划分方式的表示(用于FMO)
 - 其中num_ slice_groupes minusl语法元素说明了该图像帧中Slice Group的数目
 - num_ slice_group- map type语法元素说明了Slice Group 的划分方式
 - ➤ 同序列参数集的情况一样,每个图像帧的编码特性可能各不相同,也就需要一个pic_parameter_set_id来区分不同的图像参数集。

Slice语法结构



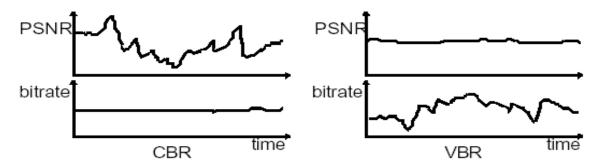
■ Slice头信息中说明了当前Slice的编码特性,包括说明该Slice所采用的图像参数集语法元素pic_parameter set id;表示该Slice的类型(I-Slice P-Slice等)的语法元素slice_type;说明该Slice的第一个宏块在图像帧中的位置的语法元素firstmb in slice;以及对该Slice进行Deblocking操作相关的语法元素和对该Slice的参考帧进行调整的NMCO语法元素等。

- ■每个视频标准的参考软件中均提供了码率控制算法,但都不是最优的
- 编码优化是编码器质量好坏的关键
 - > 快速运动估计算法
 - > 码率控制算法
 - ■比特分配
 - ■编码模式和量化步长的选择

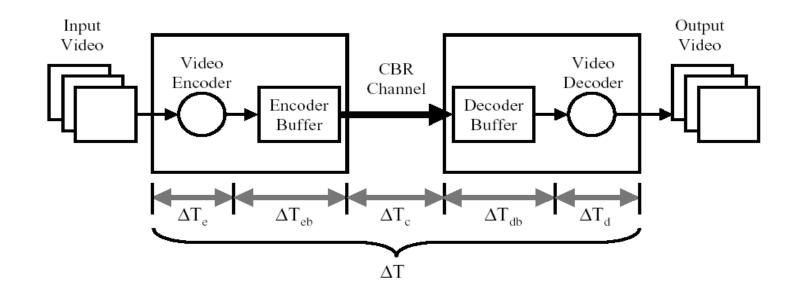


码率控制基本框图

- 视频本身具有变比特率特性
 - > 不同类型的帧
 - > 场景变换
 - > 编码参数的选择
- 信道
 - ▶ 固定比特率 (CBR) (e.g. broadcasting , ISDN,...)
 - > 变比特率 (VBR) (e.g. storage, IP,ATM,...)
 - ➤ 错误特性变化的变比特率信道 (e.g. Internet, Wireless channel,...)



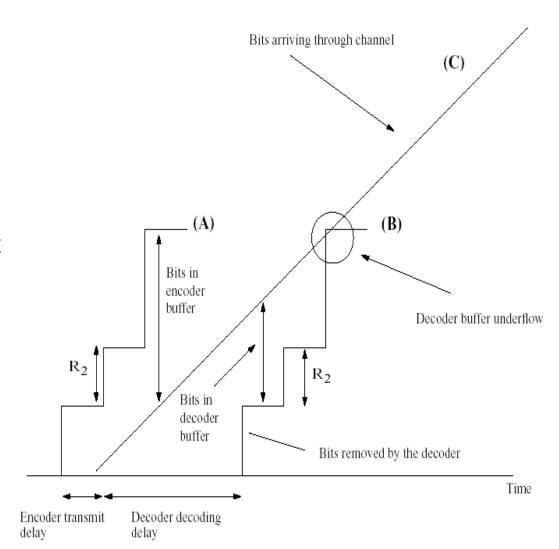
- 对视频序列的编码导致输出变比特率码流
 - » 需要一个编码缓冲区,将变比特率码流 (VBR)与固定码率的信道 (CBR)连接起来
 - > 需要一个解码缓冲区,将 CBR信道与变比特率码流 连接起来
- 视频输入码率 = 视频输出码率->constant △T (端到端延时)
 - \rightarrow 在时刻 t 编码的帧必须在时刻 t + \triangle T显示



编码器码率限制(CBR)

CBR transmission Analysis:

- > A: 发送端缓冲区中积累的比特数
- ▶ B: 已经从解码端缓冲区 移出的比特数
- > C: 从编码器到解码器的 比特传输速率 (CBR)
- A-C>编码器缓冲区的最大容量,编码器缓冲区上溢
- A-C=0:编码器缓冲区下 溢, (filler bits)
- > C-B>解码器缓冲区: decoder buffer overflow
- > C-B<0:解码器缓冲区下 溢(***)



编码器码率限制 (CBR)

■ 假定信道延时可以忽略,数据从发送端直接送到接收端,解码器延时 △N_d 帧后才从解码器缓冲区中取出第一帧数据。

第i个时刻编码缓冲 区中的数据量
$$B^e(i) = \sum_{j=1}^i R_j - \sum_{j=1}^i C$$

$$B^d(i) = \begin{cases} \sum_{j=1}^i C - \sum_{j=1}^{i-\Delta N_d} R_j, & \text{when } i \geq \Delta N_d \\ \sum_{j=1}^i C, & \text{when } i < \Delta N_d. \end{cases}$$
 第i个时刻解码缓冲 区中的数据量

编码器码率限制(CBR)

$$B^{d}(i + \Delta N_{d}) = \sum_{j=1}^{i+\Delta N_{d}} C - \sum_{j=1}^{i} R_{j} = \sum_{j=i+1}^{i+\Delta N_{d}} C - (\sum_{j=1}^{i} R_{j} - \sum_{j=1}^{i} C)$$

$$= \sum_{j=i+1}^{i+\Delta N_{d}} C - B^{e}(i).$$

$$\downarrow \qquad \qquad \downarrow$$

$$B_{eff}(i) = \sum_{j=i+1}^{i+\Delta N_{d}} C \qquad B_{eff}(i) = \Delta N \cdot C$$

编码器码率限制(CBR)

- 如果编码器缓冲区满足 $B_E^{max}=C\cdot \Delta T$ 而且不会上溢,解码器缓冲区就不会下溢
 - ▶ 例如, (MPEG-2 视频传输系统, △T =0.6,C=15000000bps)
 B_E^{max}<=C· △T=1.125MB</p>
 - 在固定比特率信道,监视编码器端缓冲区下溢或上 溢就能确保解码器端缓冲区不会上溢或下溢。

Video Buffer Verifier (VBV)

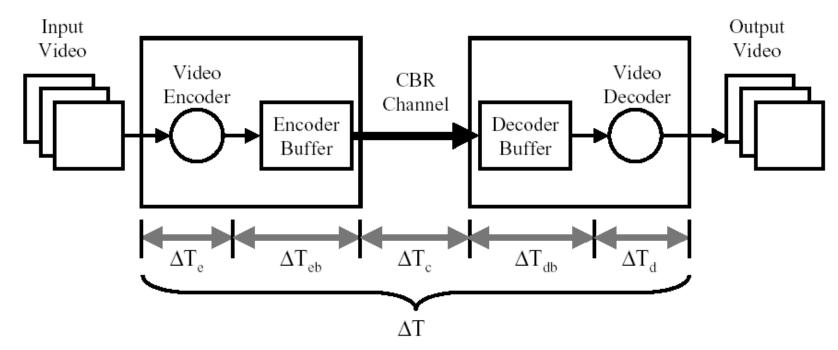
- 当输入码流遵从MPEG-2标准时,既不会上溢出,也不会下 溢出的理论上的解码缓冲器模型
- vbv_buffer_size_value
 - > 是解码缓冲区的最大值,在每个序列头传送
 - ➤ 在ATSC 标准A/53 附件A(ATSC standard A/53 Annex A)中,对缓冲区大小的限制是:
 - maximum video buffer = B = 7995392 bits, vbv_buffer_size_value <= 488 (lower 10 bits of vbv_buffer_size) vbv_buffer_size_extension = 0 (upper 8 bits of vbv_buffer_size) where B = 16*1024*vbv buffer size.
- vbv_delay是从存储一幅图像的开始码到开始解码一幅图像之间的延迟, 在每幅图像头处传送
 - ➤ ATSC对vbv_delay的限制是 vbv_delay <= 45000 vbv_delay是16位的无符号整数,表示从将一幅图像的开始码送入缓存到移除并开始解码一幅图像之间的90kHz时钟周期数量。

注意序列头和vbv_buffer_size在节目正在处理时可能不可用。然而,解码器合适的操作可以从系统时钟和各种时钟标签中得到。

MPEG-2的同步机制

- 是目前为止专为传输所设计的系统层协议
 - > 规定以包方式传输数据的协议
 - > 为收发两端数据流同步创造条件
 - →确定将多个数据流合并和分离(即复用和解复用)
 - > 提供一种进行加密数据传输的可能性

MPEG-2系统层 - 同步机制



在MPEG-2中,所有的时序被定义成一个共同的系统时钟,即系统时间时钟(STC),故编码器、解码器的系统时钟 必须同步。解码器的系统时钟应由编码器的系统时钟经恒 定延迟后恢复出来,以服从于编码器。而实现同步的关键 即为系统层中的各种时间标签。

MPEG-2系统层 - 同步机制

- 采用了PS、TS和PES三种数据包
- 用于视频、音频同步以及系统时钟恢复的时间标签分别在ES, PES和TS/PS这3个层次中
 - ➤ 在MPEG-2码流的ES层,和同步有关的主要是 VBV-Delay域,表示MPEG-2所定义的一个假设的 解码器视频缓存校验器在收到图像起始码的最后一 个字节后,至当前解码帧解码开始所应等待的时间, 用以防止解码器的缓冲器(buffer)出现上溢或下 溢
 - ➤ 在PES层,主要是在PES包头信息中出现的 PTS(显示时间标签)和DTS(解码时间标签)
 - 产在TS流中,TS包头包含了PCR(节目时钟参考),用于解码器的系统时钟恢复。在节目流PS包头中包含SCR,它的定义和作用与PCR域相似。

节目流和传输流(同步机制)

- 指示系统时钟本身的瞬时值的时间标签称为节目参考时钟标签(PCR/SCR)
- PCR/SCR的插入必须在PCR/SCR字段的最后 离开复用器的那一时刻,同时把27 MHz系统 时钟的采样瞬时值作为PCR/SCR字段插入到相应的PCR/SCR域
- PCR/SCR为42 bits的计数值,其中33 bits为PCR Base/SCR Base,是以27 MHz时钟经300分频后的时钟为单位的计数值;9 bits的PCR Ext/SCR Ext,则是以27 MHz时钟为单位的计数值
- SCR的基本部分由MPEG-1与MPEG-2兼容共用。SCR扩展部分是1个9 bit的数,由MPEG-2单独使用

解码端操作(同步机制)

■主要步骤

- ▶首先利用PCR重建和编码器同步的27 MHz 系统时钟,恢复27 MHz系统时钟后
- ▶ 再利用PES流中的DTS,PTS进行音、视频的同步
- → 同时利用VBV Delay在解码器中的Buffer充 盈到相应程度后启动初始解码

解码端操作(同步机制)

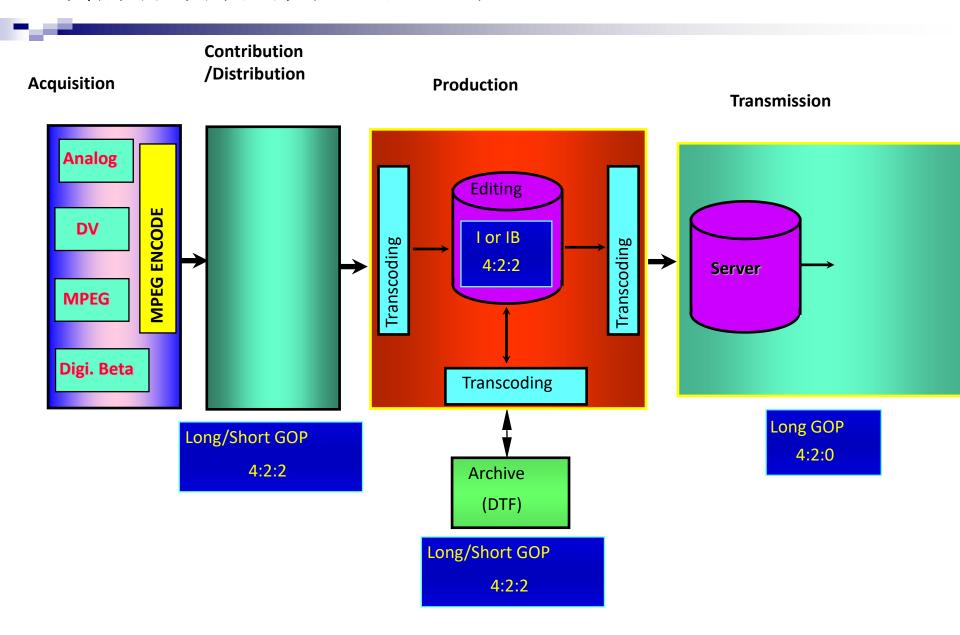
■详细步骤

- ➤ MPEG-2复用器从输入码流的包头中解出时间信息 PCR/SCR送入到系统时间时钟恢复电路
- ➤ 系统时间时钟恢复电路在接收到每一个新的 SCR/PCR时,进行本地系统时间时钟恢复和锁相
- 》解复用器从输入码流的PES包头中解出显示时间标签PTS和解码时间标签DTS,并送入到基本流解码器中
- ▶ 基本流解码器在接收到新的PTS/DTS存入对应的 FIFO中进行管理
- ▶ 对于没有DTS/PTS的显示单元,需要对其时间标签 进行插值,并送入到FIFO中管理
- ▶ 每一显示单元开始解码前,用其对应的DTS与STC 进行比较,当STC与DTS相等时开始解码
- ▶ 每一显示单元开始显示前,用其对应的PTS与STC 进行比较,当STC与PTS相等时开始显示

9.4.2 码流转换问题(Transcodig)

- 把已经压缩编码的视频码流转换成另一种不同 格式、不同码率以及不同分辨率的视频码流
- 适应不同的网络带宽、终端处理能力和不同的 用户需求
- 不同标准码流之间的转换
 - DV->MPEG MPEG2->MPEG4
 - H.264->AVS
- 同一标准,不同码率之间的转换
 - 编辑码流->播出码流 (不同GOP)
 - 高码率->低码率
- ■不同分辨率的转换

传煤体领域的典型码流流程

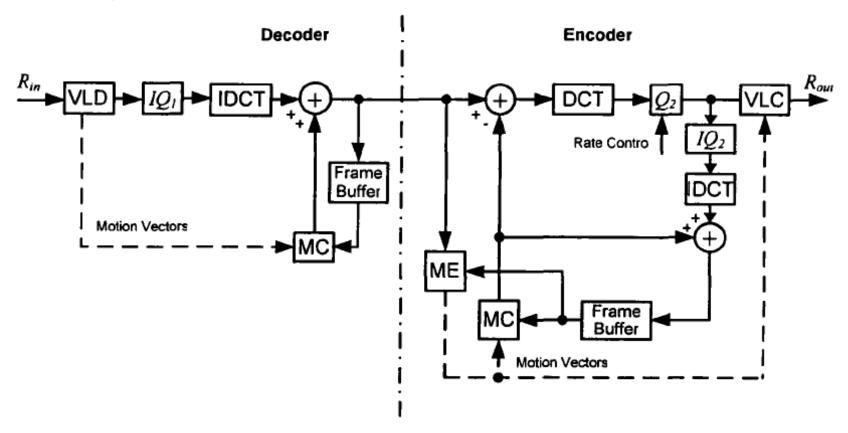


9.4.2 码流转换问题(Transcodig)

- ■全解全编的方法
 - > 实现简单
 - > 完全没有利用原始视频码流中的相关编码信息进行视频数据的转换
 - >复杂度很高,很难做到实时
- ■转码技术研究的重点
 - > 结合具体的应用环境和压缩编码方法
 - ▶ 充分利用原始视频码流中的编码信息进行重新压缩编码
 - ▶ 简化视频转码器的结构,低复杂度&尽可能 少的图像失真

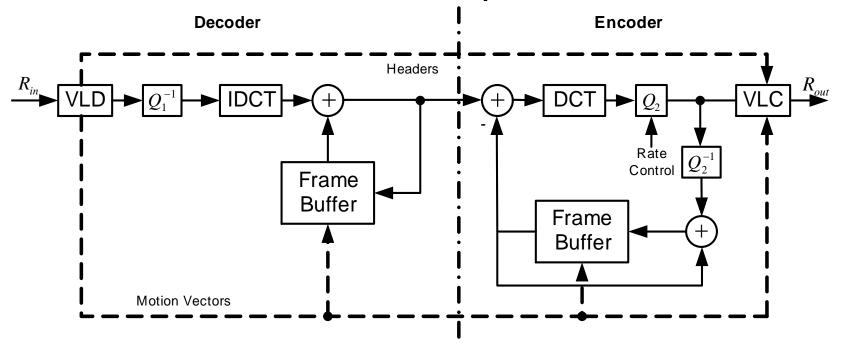
■分析

采用全解全编技术中,运动估计和模式选择是开销最大的部分



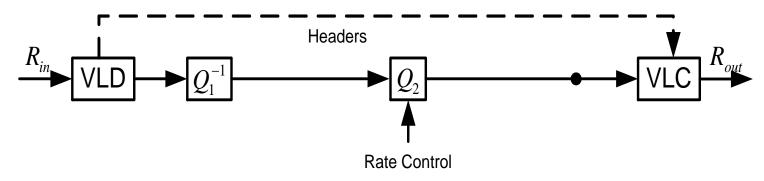
■ 分析- 级联方案

- > 尽量地重用原码流中的运动矢量和模式信息
- 为改善转码后的图像质量,可在小范围内(如2个像 素范围内进行运动估计的修正)



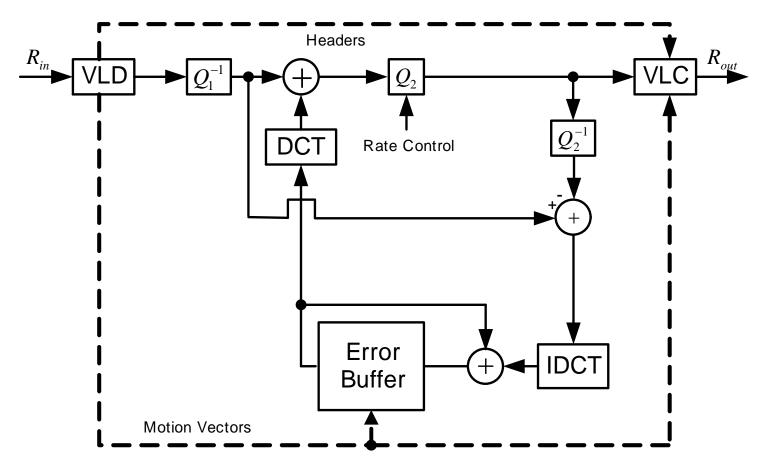
■ 分析

▶ 转码器中的解码器和编码器可进一步耦合,得到更 为简单的结构(开环结构)

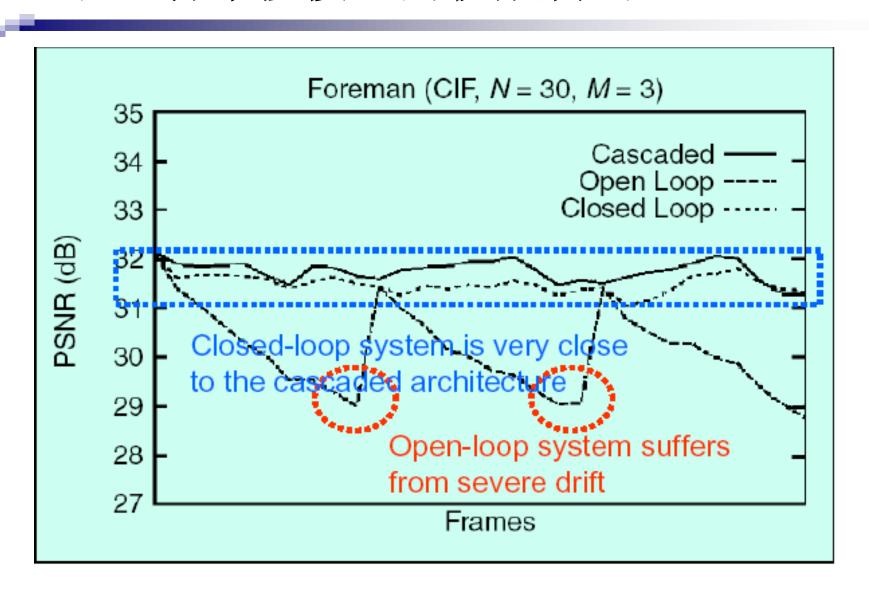


- > 直接在压缩域上进行转码
- 产生严重的误差积累
 - 进行运动补偿的时候,当前帧的预测部分从转码后劣化的 参考帧中获得,而残差部分从原始码流中解码获得,导致 预测部分和残差部分的不匹配

■ 闭环实现尽管可以避免这种"漂移"问题,但 实现复杂度较高



基于运动漂移模型的视频转码



9.5 主要的视频压缩标准介绍

- H.261
- MPEG-1
- MPEG-2
- MPEG-4
- H.263/H.263+/H.263++
- H.264/MPEG-4 AVC
- AVS
- H.265

■ MPEG-1的原则

- > 运动图象压缩与计算机应用密切结合构成多媒体终端
- ▶ 标准给系统设计提供充分自由,图象质量足以适应广泛 应用
- > 努力延长标准寿命、技术先进可靠
- > 集中科研界和工业界的智慧
- 产严格测试
- > 协调好质量与价格
- ▶ 广泛鼓励提出方案

■ MPEG-1的目标

- ▶ 音像质量应高于可视电话,达到VHS录象机的质量
- ▶可以存储在光盘、数字录音带 DAT (Digital Audio T) 和可读写光盘上
- ➤ 码率应符合当前计算机网络的传输率: 1 ~ 1.5 Mbps, 以 1.2 Mbps 为宜
- > 可以适应多种通信网络,如LAN、ISDN等
- > 满足不对称和对称应用
 - 不对称应用指一次编码、多次解码,如电视娱乐片、电子游戏、 电子出版物等,只需解码器用于播放;
 - 对称应用指同时进行编码和解码,如可视电话、双向图象邮递、同时录放的电子图象编辑。

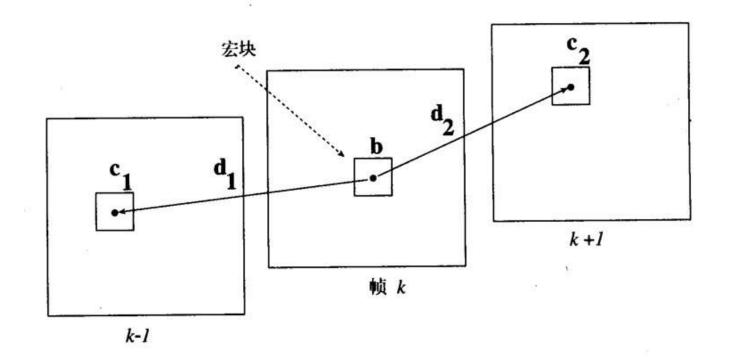
- MPEG-1标准由下列三部分组成
 - > 系统: 伴音压缩数据和图象数据的复用及同步
 - > 视频: 视频压缩
 - 》伴音: 伴音压缩,有64、128和192kbps三种码率
- MPEG-1的特点
 - > 定义了一套标准语法用于编码流的表示及解码
 - > 并不限定具体的编码算法,只要求产生有效编码流
 - > 支持多种图象尺寸、宽高比、信道传输率
 - ▶ 随机访问: 通过I帧定位可以迅速解码任一帧
 - > 快进快退: 实现快速交互

■ MPEG-1的数据结构

- > 采用层次化的数据结构,分为6个层次
 - 序列sequence: 由若干图象组构成
 - 图象组(Group Of Picture) GOP: 由各种类型图象组成
 - 图象picture,分为I、P、B、D四种类型,代表了各种压缩模式
 - 切片slice: 宏块的集合,主要用于误差恢复
 - 宏块macro block: 和H.261中的宏块定义相同
 - 块block: 8×8的象素矩阵,最小的DCT编码单位

- MPEG-1中的图象类型
 - ▶相当于帧
 - ▶ I型图象:使用类似JPEG算法进行帧内编码,是随机访问点
 - ▶ P型图象:只参考前面的I图象和P图象,进行前向预测
 - ▶ B型图象:可以参考其它的I图象和P图象,进行前向、 后向和双向预测
 - D型图象: 仅包含每一块的直流分量,用于在甚低速率下提供浏览功能

- MPEG-1中的双向预测
 - ▶B图象的双向预测是MPEG-1的重要特点

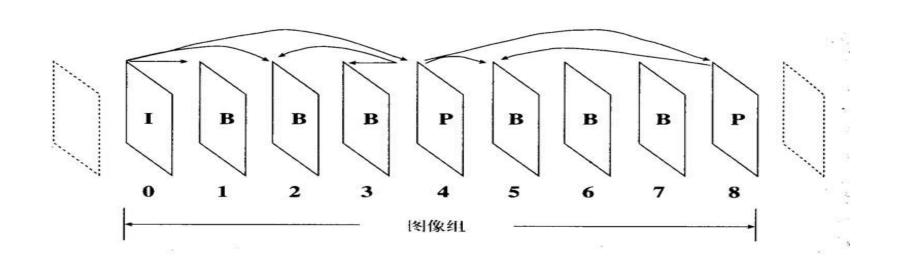


■ MPEG-1中的双向预测

- ▶ 用两个运动矢量进行运动补偿可以获得更高的信噪比 SNR
- ▶ 不把B图象作为预测的参考图象,因此不会引起误差扩 散
- ➤ 需要在编码器和解码器中存储至少两帧图象,因为双向 预测需要两个参考图象(I图象或者P图象)
- > 如果使用太多B图象,也会产生影响
 - 参考图象距离增加,相关性下降
 - 产生较大的编码延迟

■ MPEG-1中的图象组GOP

- > GOP由各类图象构成,但是第一帧总是是I型图象
- ▶ 由于引入双向预测,图象处理的顺序不必和时序相同, 下面例子中的顺序可以是



■ MPEG-1中宏块的压缩 模式选择

▶ Intra:帧内编码

> Inter:帧间预测

▶ A:自适应量化

D:传递预测误差的DCT系数

> F:有运动补偿的前向预测

> B:有运动补偿的后向预测

▶ I: 有运动补偿内插(双向) 预测

> Skipped:两MB相同,不 需传输

I图象	P图象	B图象	
Intra	Intra	Intra	
Intra-A	Intra-A	Intra-A	
	Inter-D	Inter-F	
	Inter-DA	Inter-FD	
	Inter-F	Inter-FDA	
	Inter-FD	Inter-B	
	Inter-FDA	Inter-BD	
	skipped	Inter-BDA	
		Inter-I	
		Inter-ID	
		Inter-IDA	
		skipped	

■ MPEG-2

- 一种高质量的运动图象及其伴音压缩标准
- ▶最初要求
 - 图象分辨率720×480,与CCIR 601演播室数字电视分辨率相同
 - 传输码率4~10Mbps
 - 与MPEG-1兼容,发挥设备兼容能力、降低价格
- ➤ 后来许多提案要求囊括视频通信领域的一些应用如多点电视会议、工作站窗口显示、ATM视频通信及HDTV等,支持可分级(Scalable)视频编解码,因此最终覆盖了2~40Mbps的许多应用,导致MPEG-3的制定中途取消。

■ MPEG-2标准的组成

- > 系统: 1994/11 (完成时间)
- > 视频: 1994/11
- > 音频: 1994/11
- > 一致性测试: 1995/07
- > 软件: 1995/07
- > 数字存储媒体的指令和控制: 1995/09
- > 非向后兼容音频: 1997/03
- > 10比特视频: 1996/07
- > 实时接口: 1995/11

■ MPEG-2的类和级

- 》如果将各种应用需求协调成单一语法,标准 将非常复杂
- ➤ MPEG-2将语法分为5个子集:类(Profile),每个类最多支持4级(Level)
- >具体应用可根据需要选取相应的类@级
- ▶同时,复杂的类**②**级解码器可以解简单的类 **②**级比特流

■ MPEG-2的类和级

- ▶ MPEG-2划分了5个类
 - 简单类(Simple Profile):除了没有B帧外与主类相同
 - 主类 (Main Profile): 不可分级,质量尽可能地好
 - SNR可分级类(SNR Scalable Profile):信噪比可分级
 - 空间可分级类(Spatially Scalable Profile): 空间分辨率可分级
 - 高类 (High Profile): 支持4:2:2、4:4:4及全部的可分级性

> 4个级别

- 低级(Low Level):对应H.261 CIF或MPEG-1 SIF。
- 主级(Main Level):对应常规电视。
- 高级-1440(High-1440 Level): 对应HDTV-1440。
- 高级(High Level):对应HDTV-1920。

■ MPEG-2的类和级

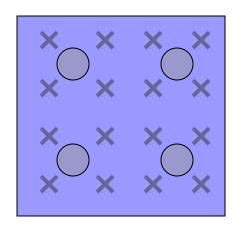
> 类与级的有效组合共有11种

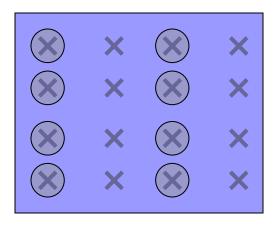
等级/类	简单类	主类	SNR可分	空间可分级类	高类
			级类		
低级		MP@LL	SNR@LL		
352×288					
主级	SP@ML	MP@ML	SNR@ML		HP@ML
720×576					
高级1440		MP@H1440		SSP@H1440	HP@H1440
1440×1152					
高级		MP@HL			HP@HL
1920×1152					

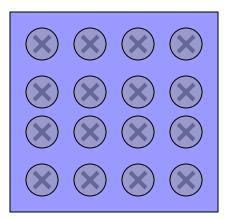
■ MPEG-2的类和级

- > 有四个重要的类级组合
 - 主类@主级(MP@ML): DVB、DVD、数字有 线电视、ITV等
 - ■简单类@主级(SP@ML): 数字有线电视、数字录象机
 - 主类@高级(MP@HL): 全数字HDTV
 - 空间可分级类@高级-1440(SSP@H1440): 欧洲的HDTV

- MPEG-2的宏块
 - ▶ MPEG-2中的宏块可以有三种格式
 - 4:2:0模式, 4:2:2模式, 4:4:4模式
 - X代表亮度分量, O代表色度分量







■ MPEG-2的图象类型

- ▶ 逐行视频: 支持帧图象(I/P/B图象)
- > 隔行视频: 支持帧图象和场图象
 - Field Picture: 每场独立编码,适合于描述剧烈的运动
 - Frame Picture: 奇偶两场合在一起基于帧编码,适合于静态或运动缓慢视频图象
- > 隔行视频补偿预测模式
 - 简单场预测(Simple Field Prediction): 基于场作预测和补偿, Filed Picture中只能采用简单场预测
 - 简单帧预测(Simple Frame Prediction): 奇偶两场合在 一起基于帧作预测和补偿, Frame Picture中既可采用简单 场预测,又可采用简单帧预测

■ MPEG-2的可分级性(Scalability)

- 》可以在所期望的分辨率下,仅对位流中的某一部分 进行解码而获得视频图象
- ➤ 可解码的最小比特流子集称为基层(Base Layer), 其余部分称为增强层(Enhancement Layer),在 基层基础上提高分辨率
- > 可分级性有几种不同的形式
 - ■空间(象素分辨率)可分级:基层包含了低分辨率视频图象,增强层在含有连续的较高分辨率的信息
 - SNR可分级: 指的是对DCT系数使用不同的量化阶距后的解码能力, 基层视频图象是对DCT系数进行粗略的量化, 和初始视频图象空间分辨率是相同的
 - 时间可分级: 可以以不同的帧率进行解码

■ MPEG-4背景

- ➤ MPEG-4以前的音频视频编码标准已经覆盖了 64kbps以上的码率
- ➤ MPEG-4在92 年末的初衷是制定 64kbps 以下甚低比特率 (VLBR: Very Low Bit Rate) 音频视频编码标准
- ➤ 但94年7月ITU-T针对PSTN 和GSM开始制定H.263,MPEG-4作为纯粹的压缩标准意义骤减
- ➤ 而且随着Internet的飞速发展,对音视(AV)信息的灵活有效交互、存储与传输的需求日益高涨,因此MPEG-4及时调整方向,除压缩外还侧重于功能、尚未支持的大量应用。

■ MPEG-4目标

- ▶制定一个基于内容/对象(Content /Object Based)的、高效的(Efficient)、灵活的(Flexible)、通用的(Universal)、可分级/伸缩的(Scalable)、可扩展的(Extensible)视频编码标准
- ▶ 通过定制(Customization)便于特定应用充分利用 现有的乃至未来的技术。
- ➤ 一个重要特色是支持Internet上对音视信息有效的传输和灵活的操作。

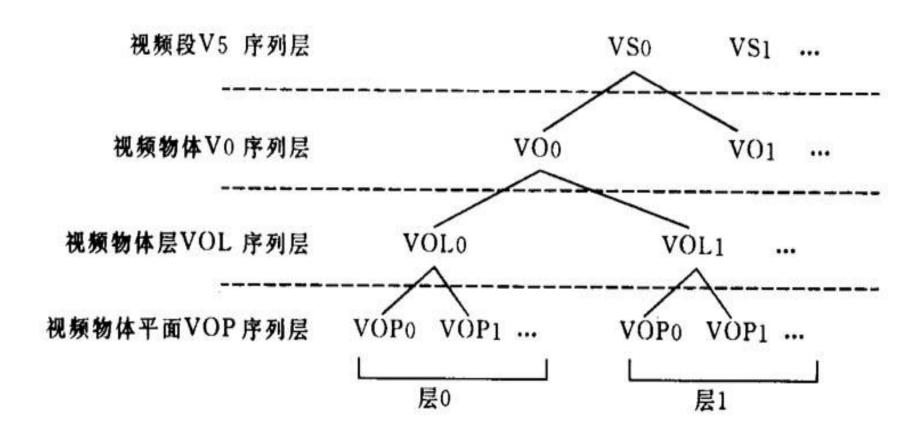
■ MPEG-4的功能

- > 内容交互的功能
 - 基于内容的多媒体访问(Access)
 - 基于内容的操作、比特流编辑(Bit Stream Editing)
 - 自然/合成混合编码 (SNHC: Synthetic & Natural Hybrid Coding)
 - 完善的时域访问(Temporal Access)
- > 压缩的功能
 - 高效编码: 形状、纹理与运动
 - 多同步流编码(Multiple Concurrent Stream Coding)
- > 通用的访问功能
 - 鲁棒性 (Robust)
 - 基于内容的可分级/伸缩(Scalable)

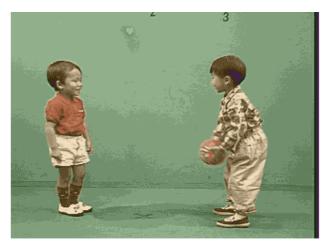
■ MPEG-4的应用

- ▶ 基于Internet的视频流(Video Streaming)传输
- > 多媒体制作(authoring)与展示(presentation)
- > 可分级/伸缩:不同分辨率/帧率/质量的视频播放
- ► 低带宽、易错(error-prone)环境下(例如GSM、PDA)的多媒体传输、多媒体数据库检索
- ➤ 多点远程会议(Multi-Point Tele-Conferencing),可选性(Selective)传输、解码、显示
- > 交互式家庭视频选购
- ➤ 视频邮件(Video Mail)
- ➤ 远程教育(Tele-Education)

- MPEG-4的数据结构
 - ➤视频段(VS: Video Segment),完整的视频 序列由VS构成
 - ➤ 视频对象(VO: Video Object),一个视频段 由若干隔视频对象构成
 - ➤ 视频对象层(VOL: Video Object Level),每个VOL代表VO的一个层次(基本层、增强层),每个层表示一级分辨率(空间或者时间的分辨率)
 - ➤ 视频对象面(VOP: Video Object Plane),每个VOL中都有一系列时间上连续的VOP



- MPEG-4中视频编码的过程
 - >VO的形成,从原始视频流中分割出VO;
 - >编码,对各个VO分别独立编码,即对不同的 VO的三类信息(运动、形状、纹理信息)分别 编码
 - ▶ 复合,将各个VO的码流复合成一个符合MPEG-4标准的码流



Version 1:

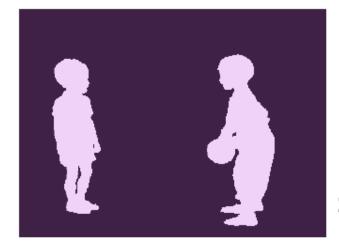
- Shape coding
- Texture coding
- Motion coding

Scene

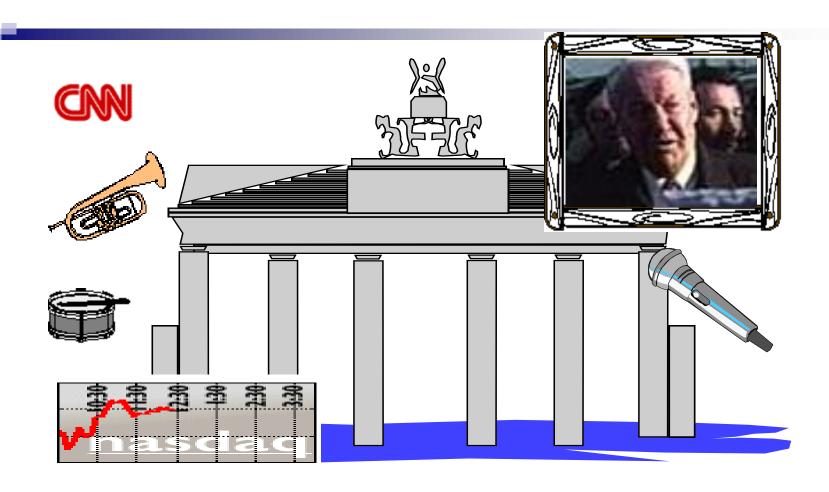
- Error resilience coding
- Scalable coding
- Sprite coding
- Still texture coding
- Face object coding
- Mesh object coding

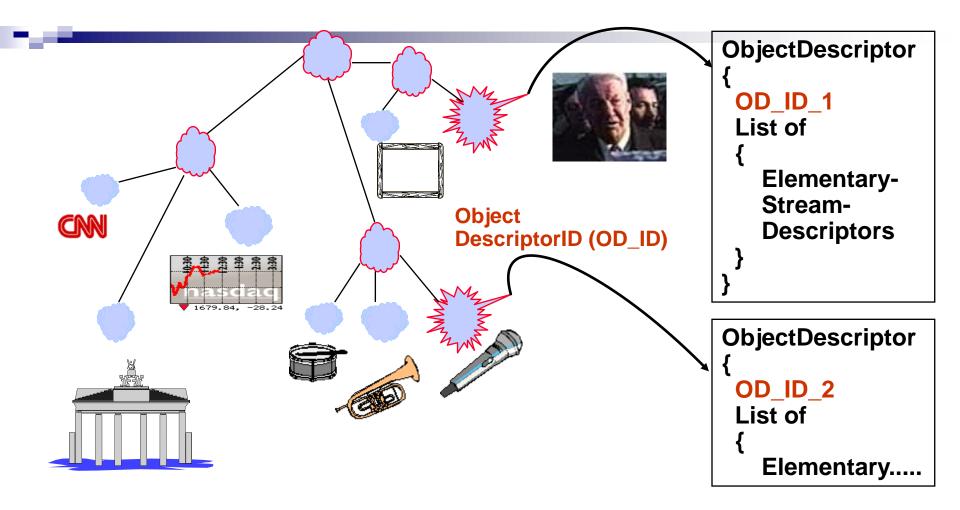


Object

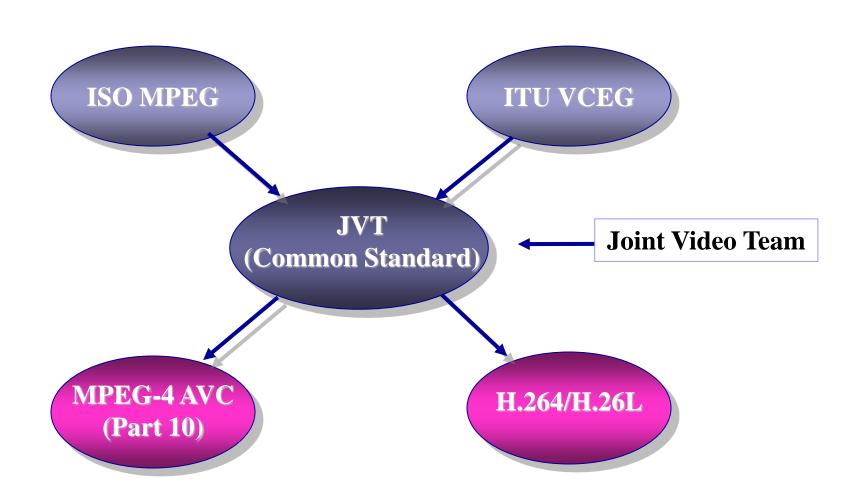


Shape





视频压缩标准



视频压缩标准

■ H.264/MPEG-4 AVC的特点

- > 简单的语法结构
- > 编码效率大为提高
- > 友好的网络接口
- > 应用的主要目标是移动网络和因特网

视频压缩标准

H.264/AVC

Conversational H.32X Services

- H.320 Conversational
- 3GPP Conversational H.324/M
- 3GPP Conversational IP/RTP/SIP
- H.323 Conversational Internet/unmanaged/best effort IP/RTP

Streaming Services

- 3GPP Streaming IP/RTP/RTSP
- Streaming IP/RTP/RTSP

Other Services

- Entertainment Satellite/Cable/DVD, 0.5 8 Mbit/s
- Digital Cinema Application
- 3GPP Multimedia Messaging Services

与其它标准之间的共同处

- 16x16 宏块
- ■色度子采样
- ■基于块的运动补偿
- ■可变块大小的运动补偿
- ■基于块的正交变换
- 变换系数的游程及变长编码
- ■标量量化
- I和P帧

AVC Coding Features

- I, P, B, SP, SI
- 4x4 Integer DCT-like transform
- Improved Intra Prediction
- Variable block size -7 Block Patterns
 - > 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4
- Multiple reference frames
- Up to 16MVs per MB
- Advanced MV coding
- De-blocking filter
- Up to 1/4th Pel Interpolation
- Universal VLC, Context Adaptive VLC (CAVLC) and Arithmetic Coding (CABAC)
- Adaptive Coding is the key feature of JVT
- Motion Compensation brings most of the gain

MC related

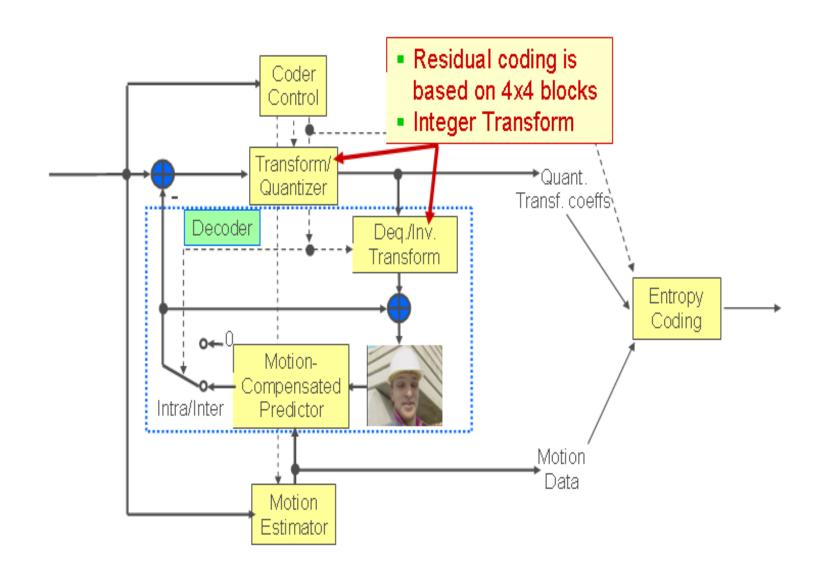
Motion-Compensated Prediction

- Various block sizes and shapes for motion compensation (7 segmentations of the macroblock: 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4)
- 1/4 sample accuracy motion
 - 6x6 tap filtering to 1/2 sample accuracy, bilinear filtering to 1/4 sample accuracy, special position with heavier filtering
- Multiple reference pictures (per H.263++ Annex U)
- Temporally-reversed motion and generalized B-frames
 - > B frame can be also references of other frames
- B-frame prediction weighting

Advanced Motion Vector Coding

- P-picture
 - Median Prediction (except for 16x8 and 8x16 modes)
 - Directional Segmentation Prediction
- B-picture
 - > Direct, forward, backward, bi-directional and intra
 - > 16x16, 8x16, 16x8, 8x8, 4x8, 8x4, 4x4
 - Direct mode:
 - No MV data is transmitted
 - Same block structure as co-located MB in temporally subsequent picture
 - MVs are computed as scaled version of corresponding MV of the co-located MB

Residual Coding



Residual and Intra Coding

Transform

Based primarily on 4x4 transform (all prior standards: 8x8)

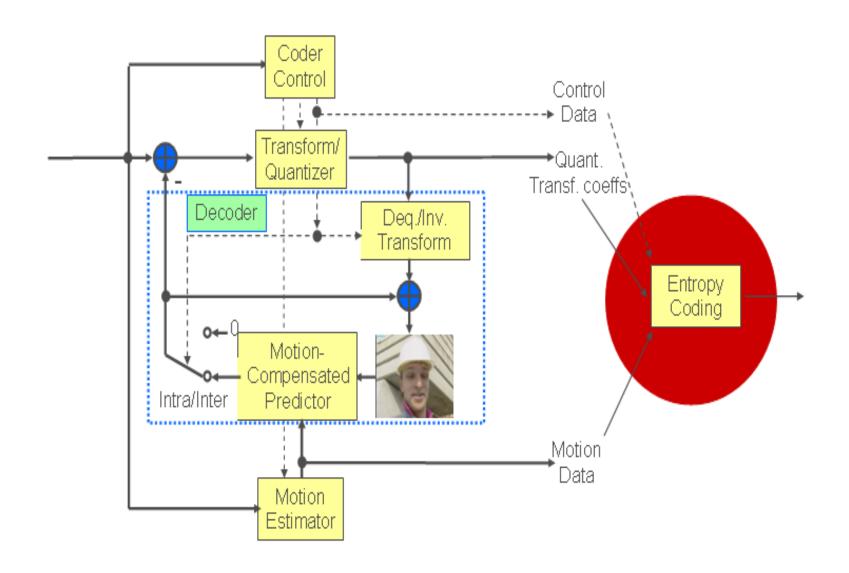
$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

- Expanded to 8x8 for chroma by 2x2 transform of the DC values
- Intra Coding Structure
 - Directional spatial prediction (9 types luma, 1 chroma)
 - Expanded to 16x16 for luma intra by 4x4 transform of the DC values

Quantization and De-blocking

- Quantization
 - Logarithmic step size control
 - Smaller step size for chroma (per H.263 Annex T)
- De-blocking Filter (in loop)
 - Filtering done on an MB basis
 - Content dependent filtering

Entropy Coding



Context-Adaptive VLC (CAVLC)

- Coding and decoding of transform coefficients
- String of coefficients at the high frequencies at +-1
- Parameter Num-Trail –number of coefficients and Trailing 1s (T1s); T1 = 0,1,2,3
- Coefficients other than T1s Level information is coded
- For decoding of coefficient data (both Level and Run) scanning is done in reverse order
 - In Level information, signs of T1s are decoded first (in reverse order) then the Level Information
 - Similarly Run information

Context-based Adaptive Binary Arithmetic Codes (CABAC)

- Usage of adaptive probability models
- Exploiting symbol correlations by using contexts
- Non-integer number of bits per symbol by using arithmetic codes
- Restriction to binary arithmetic coding
 - Simple and fast adaptation mechanism
 - > Fast binary arithmetic coders are available
 - Binarization is done using the UVLC

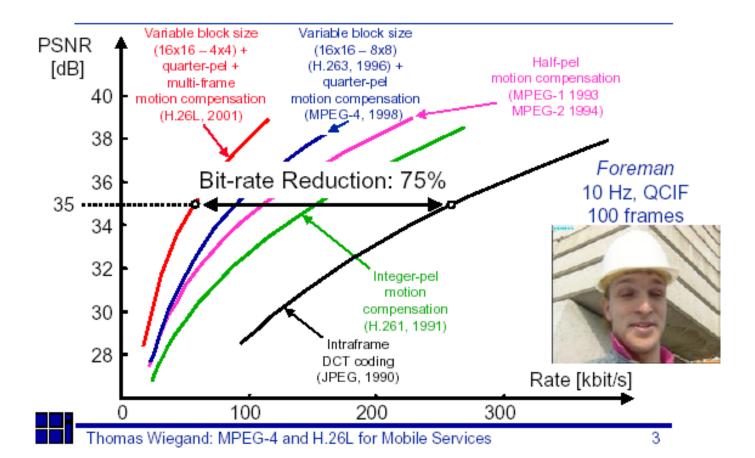
AVC Profile

	Profile		
Visual Tools	Baseline	Main	"X"
CABAC Entropy Coding		X	
Adaptive Block Cize Transform		X	
B-Pictures		X	X
S-Pictures			X
Data Partitioning			X
Flexible Macroblock Ordering	X	X	X
VLC Entropy Coding	X	X	X
I-Picture	X	X	X
P-Picture	X	X	X
In-loop de-blocking filter	X	X	X
Interlaced	X	X	X
1/4-sample motion compensation	X	X	X
4x4 Motion Compensation (only 8x8 for B-pictures).	Х	X	Х
Chrominance Format	4:2:0	4:2:0	4:2:0

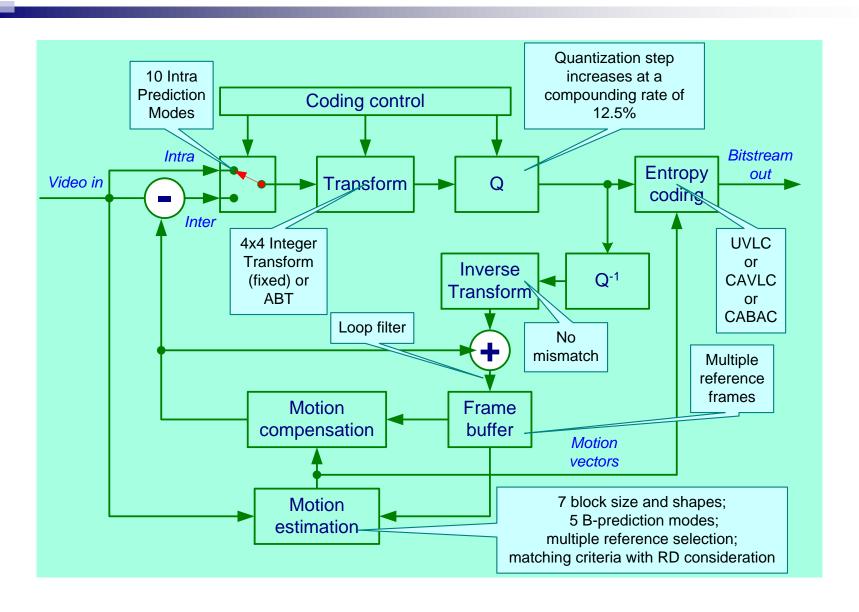
Comparison of AVC/H.264 and MPEG-4

Both:

- Sequence structure IBBPBBP....
- Search range: 32x32 around 16x16 predictor
- \triangleright Encoders use similar D+ λ R optimization techniques
- MPEG-4: Advanced Simple Profile (ASP)
 - Global Motion Compensation
 - > 8x8 DCT
 - \rightarrow QP_B=1.2 x QP_P
- AVC/H.264:
 - > 4x4 Integer transform
 - Using CABAC entropy coding
 - > 5 reference frames
 - \triangleright QP_B=QP_P+2



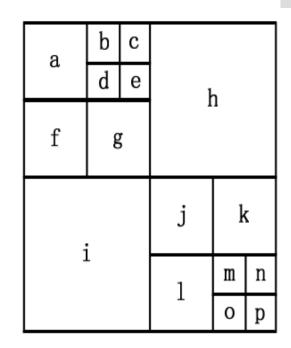
Summary of AVC Coding Tools

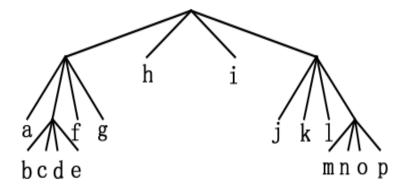


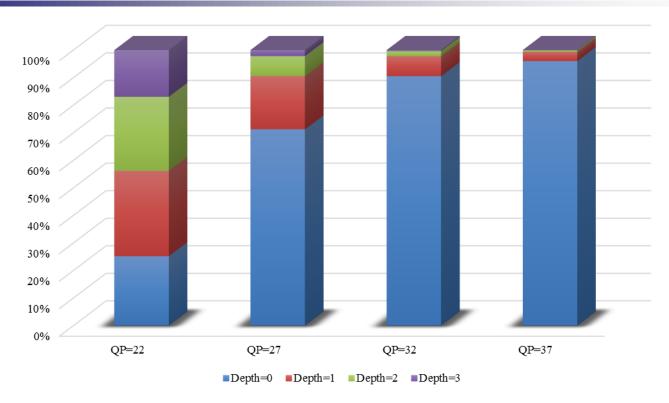
Upgrade H.264 to HEVC Control Motion Input Data Video **Transfor** Control Signal m-skip **▶**Data ∖ Transform/ Ouant. Scal./Quant. Transf. coeffs Scaling & Split into Inv. Largest Coding Unit Transform Entrop 64x64 pixels Coding Deblocking Filter Intra-frame Prediction Motion-Sample Compensation Adaptive Intra/Inter Offset Motion Data Motion Estimation

- CTU(Coding TreeUnits,编码树单元)
 - 》类似于H.264 /AVC中的宏块,HEVC将编码帧分为若干编码树块(Coding Tree Blocks,CTB),它们是进行预测、变换、量化和熵编码等处理的基本单元,其尺寸或者所包含的像素数可以是16×16、32×32或64×64。
 - ▶ 同一位置的亮度CTB和两块色度CTB,再加上相应的语法元素形成一个CTU。
 - ➤ 一般来说,Unit是从语法结构角度进行描述,而 Block是从采样值角度进行描述,并不特意区分这两 个概念。
 - > CTU是解码器处理的基本单元。

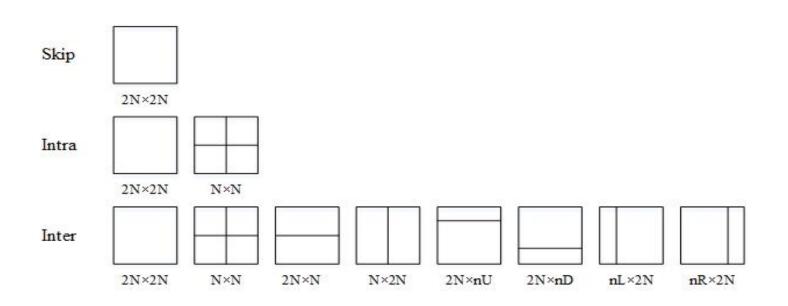
- **4**⁰**+4**¹**+4**²**+4**³
- CU(Coding Unit, 编码单元)
 - ▶ 同一层次的CU必须是同一尺寸的 4个方块,最多可有4层分解。
 - ➤ 每个CU包含一块亮度编码块(Coding Blocks, CB)、两个色度 CB以及相应的语法元素。
 - ➤ CU是决定进行帧内预测还是帧间 预测的单元,也就是说整个CU只 能是一种预测模式,不是帧内就 是帧间。CU还可以按照四叉树层 次分解(或不分解) 为更小的预测 单元(Prediction Units,PU) 和变 换单元(TB,Transform Units)





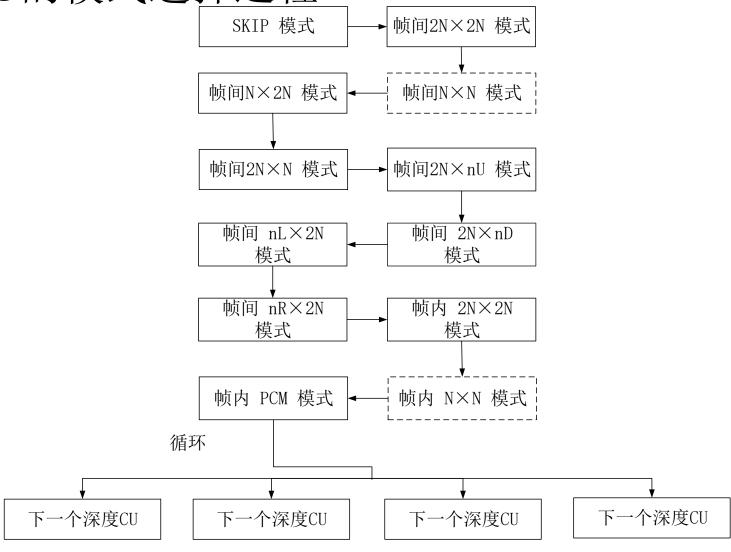


- PU(Prediction Unit, 预测单元) (共11种)
 - ▶ 对于图像的帧内预测,预测单元PU的尺寸与所属的编码单元CU相同,只有当编码单元CU在最低层即0层时,才能被划分为N*N的预测单元PU。
 - 对于图像的帧间预测,预测单元的分割模式有8种,4种对称模式和4种非对称模式。



■PU的模式选择过程

95



- TU(Transform Unit, 变换和量化的基本单元)
 - ➤ 在CU的基础上划分的,但受到所在的预测单元PU的限制。如PU是正方形,则TU也必须是正方形,尺寸为4×4、8×8、16×16、32×32。
 - 》帧内编码模式中变换单元的尺寸需小于或者等于预测单元, 而帧间编码模式中变换单元可以大于预测单元,但是不能超 过编码单元 CU。一个编码单元中多个变换单元也是按照四 叉树的结构排列,即每向下一层划分为4个小的正方形。
 - ➤ 如 PU为非正方形时,TU 也需非正方形,其大小为32×8、8×32、16×4、4×16,可用于亮度分量,其中只有32×8、8×32 可用于色度分量。HEVC中允许使用的变换包括4x4、8x8、16x16和32x32整数DCT变换以及对于Intra预测中4x4亮度块使用的整数DST变换。

- TU(Transform Unit, 变换和量化的基本单元)
 - ➤ 虽然Intra预测的模式信息是以PU为单位传递的,但Intra 预测是以TU为单位进行的。
 - ▶ 所有PU内部的各个TU的预测方向是一致的,以TU为基本单元进行Intra预测的目的是使已重建像素和待预测像素之间的距离尽量短,进而提高预测效率,即进行帧内预测的基本单元是TB。
 - ▶ 这也是在Intra编码的CU中,TU永远不会比PU大的原因。

- HEVC中共有3种片(slice)类型。
 - ▶I slice仅能使用帧内预测对当前slice进行预测
 - ➤ P slice可以使用帧内预测或者不超过一组运动矢量和参考索引(reference index)的帧间预测对当前 slice进行预测
 - ➤ Bslice可以使用帧内预测或者不超过两组运动矢量和参考索引的帧间预测对当前slice进行预测。