





第4章 数字视频编码标准



本章学习目标

- 理解**H.264/AVC**标准中“类”和“级”的含义。
- 熟悉**H.264/AVC**标准的主要特点及性能。
- 了解**H.265/HEVC**标准的主要特点及性能。
- 了解我国具备自主知识产权的音视频编码标准（**AVS**）的性能及应用。



第4章 数字视频编码标准

- 4.1 数字视频编码标准概述
- 4.2 H.264/AVC视频编码标准
- 4.3 H.265/HEVC视频编码标准
- 4.4 AVS与AVS+视频编码标准



4.1 数字视频编码标准概述

国际上数字视频编码标准主要有两大系列。一个系列由国际标准化组织（**ISO**）和国际电工委员会（**IEC**）制定，另一个系列由国际电信联盟电信标准部（**ITU-T**）制定。

- **ISO/IEC**: JPEG, JPEG2000, MPEG-1/2/4/7
- **ITU-T**: H.261/2/3/4

制定这些标准的背景有所不同，面向的主要应用也有所区别，它们采用的技术有很多共同点，应用领域有所重叠。

■ **两者合作制定标准**: H.262/MPEG-2/、H.264/MPEG-4 AVC 和 H.265/HEVC



4.1 数字视频编码标准概述

- **制定视频编码标准的组织**

- **国际电信联盟 (ITU)**

- **ISO/IEC**

- **International Telecommunication Union (ITU)**

- **其委员会包括:**

- **CCITT (Consultative Committee on International Telegraph and Telephone), 1956~1992**

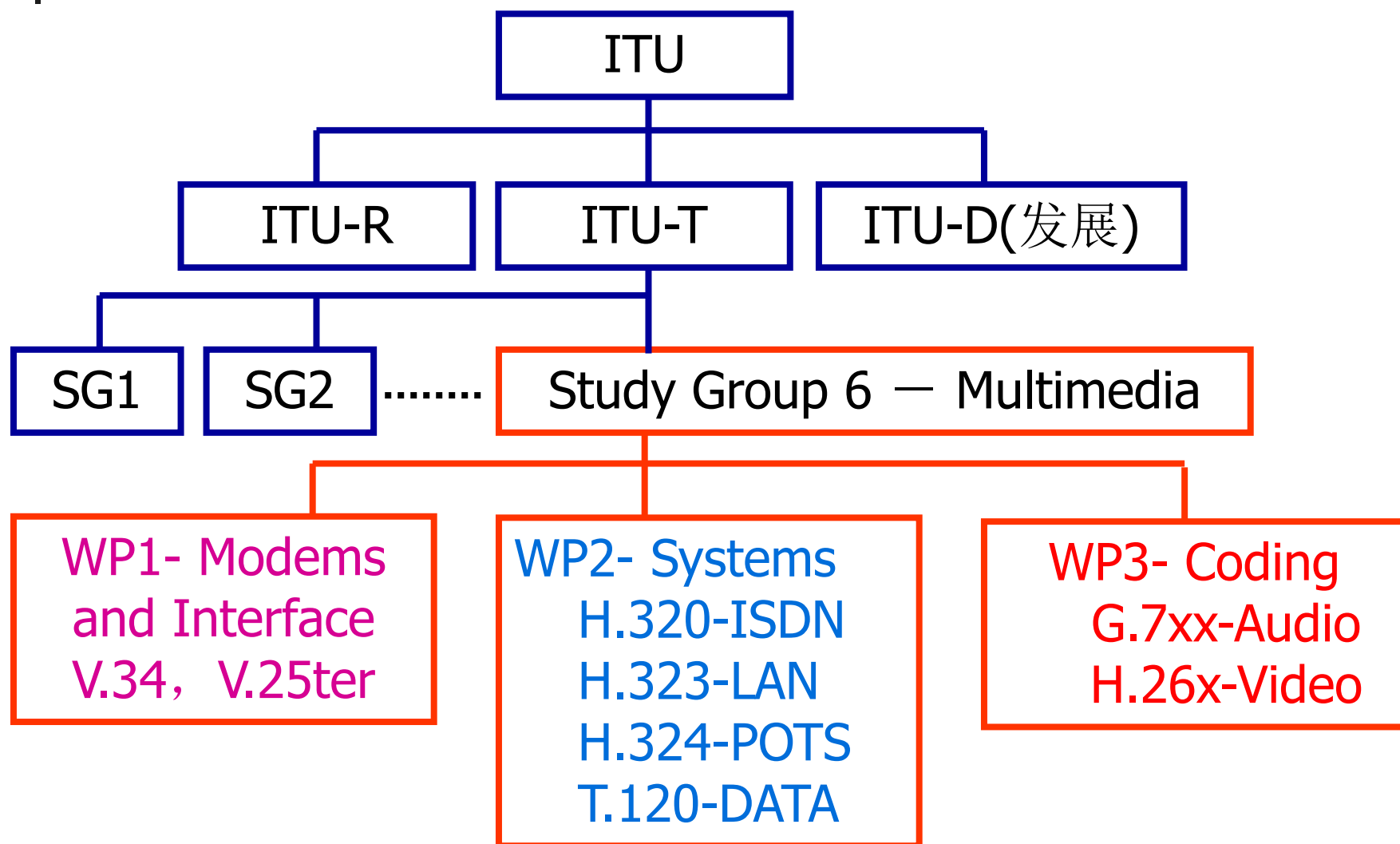
- **CCIR (Consultative Committee on International Radio), 1927~1992**

- **1992 重组**

- **CCITT → ITU-T**

- **CCIR → ITU-R**

ITU 视频编码相关的研究小组



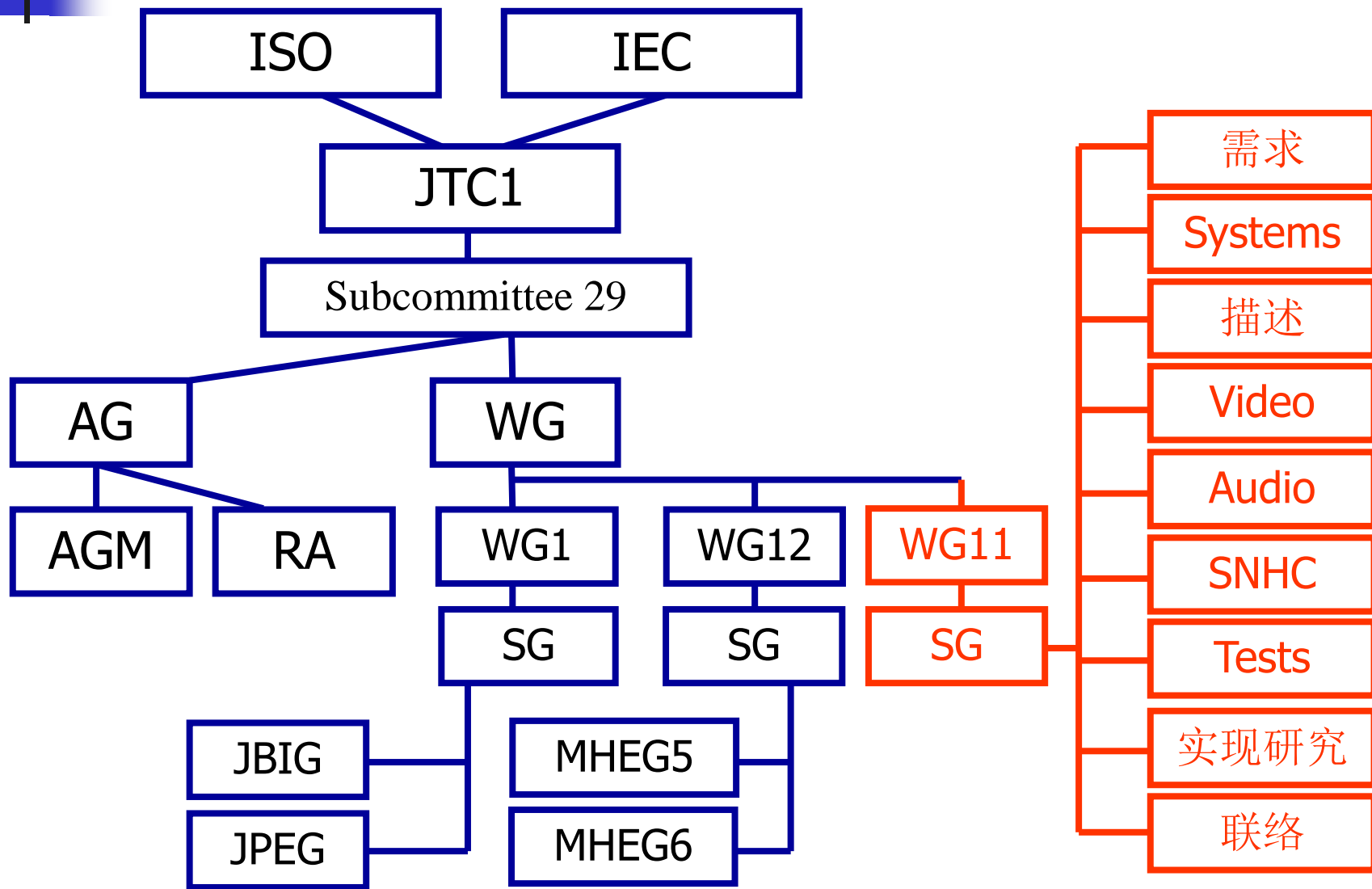


视频编码标准组织

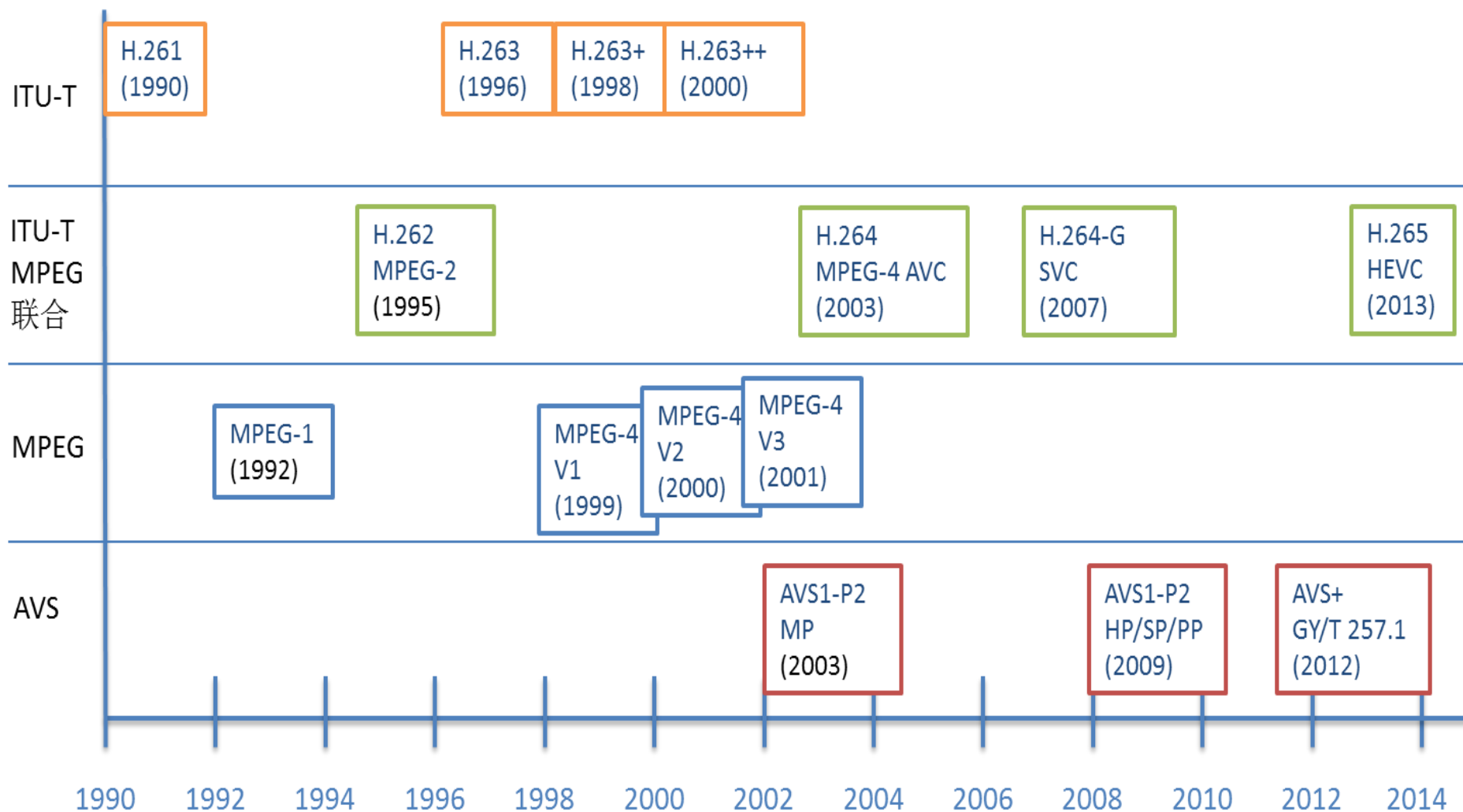
■ ISO/IEC

- **IEC** - International Electrotechnical Commission
 - 成立于 1906 年，致力于建立所有电工技术方面的国际标准
 - 瑞士法律下的非盈利、私立机构
- **ISO** - International Standardization Organization
 - 成立于 1947 年，“推动工业标准的国际协调和统一”
 - 瑞士法律下的非盈利、私立机构
- **ISO/IEC 联合技术委员会** (ISO/IEC Joint Technical Committee 1, JTC1)
 - 联合致力于计算机相关的标准活动
 - 占 ISO 和 IEC 总标准的 30%

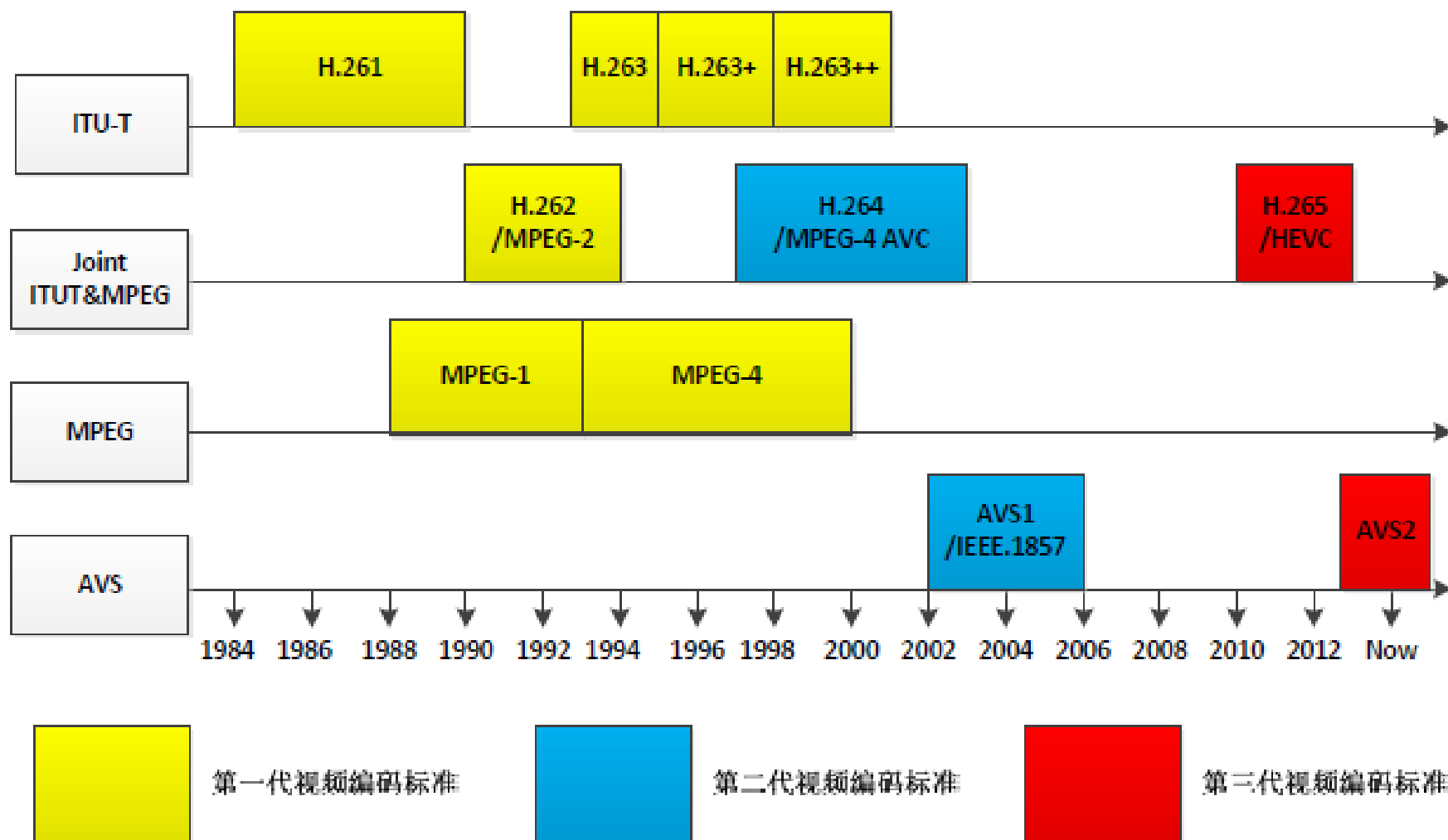
ISO/IEC 视频编码相关的研究小组



视频编码标准发展历程

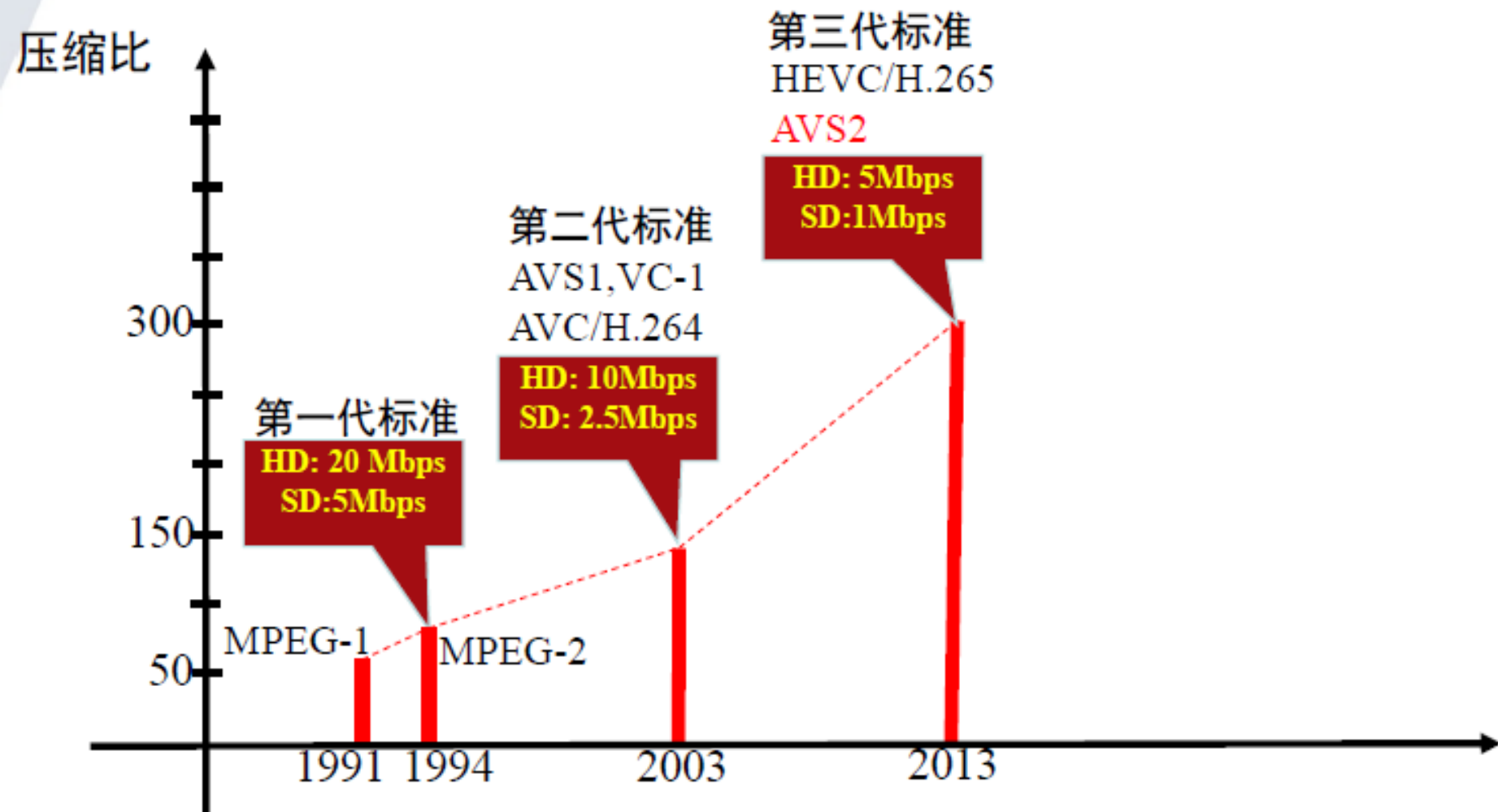


视频编码标准发展历程



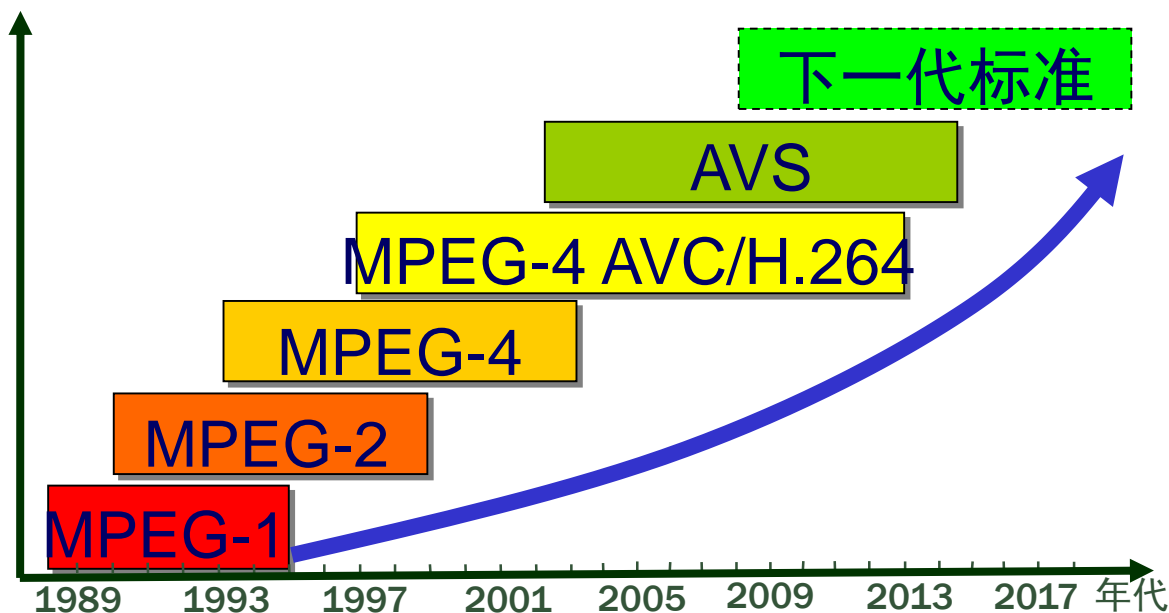
视频编码标准发展历程

➤ 压缩效率变化情况



视频编码标准发展历程

- ❖ **ISO/IEC**--MPEG-1, MPEG-2, MPEG-4, MPEG-4 AVC/H.264, HEVC
- ❖ **ITU-T** --H.261, H.262(MPEG-2), H.263, H.264, H.265
- ❖ **中国 AVS**--AVS, AVS+
- ❖ **下一代标准?** ---方向、技术?





4.1.1 H.26x 系列标准

- ITU-T的视频编码专家组 (**VCEG**) 制定了H.26x 标准系列, 主要针对实时视频通信领域的应用, 如可视电话、会议电视等。
- **H.261**: “速率为 $p \times 64\text{ kbit/s}$ ($p=1, 2, \dots, 30$) 视听业务的视频编解码”, 简称为 $p \times 64\text{ kbit/s}$ 标准。
- **H.262**: 同MPEG-2的视频部分 (ISO/IEC 13818-2) 。
- **H.263**: 低码率视频编码标准。
- **H.264**: 等同于MPEG-4 AVC (ISO/IEC 14496-10) 。
- **H.265**: 等同于MPEG HEVC (ISO/IEC 23008-2) 。



ITU-T H.261

- 1988年制定，奠定了现代视频编码的基础：
 - 编码结构为：**运动补偿预测+DCT 编码+熵编码**
 - **关键技术**包括 16×16 宏块运动补偿预测、 8×8 DCT、量化、环路滤波和变字长编码（VLC）等。
 - 这些关键技术后来被其它标准采用。
- H.261v2 (1993年早期) 增加了后向兼容的高清晰度图片模式(例如远程教学中的白板)
- 图像格式：CIF or QCIF，帧率 29.97fps
- 数码率为 $n \times 64 \text{ kbit/s}$ (64~1920 kbit/s)，典型为 384kbit/s
- 特性：
 - 低复杂、低时延
 - 后向兼容的一个标准
 - **被 H.263 和 H.264 超越**

H.261 图像格式



720 × 576

ITU-R BT.601

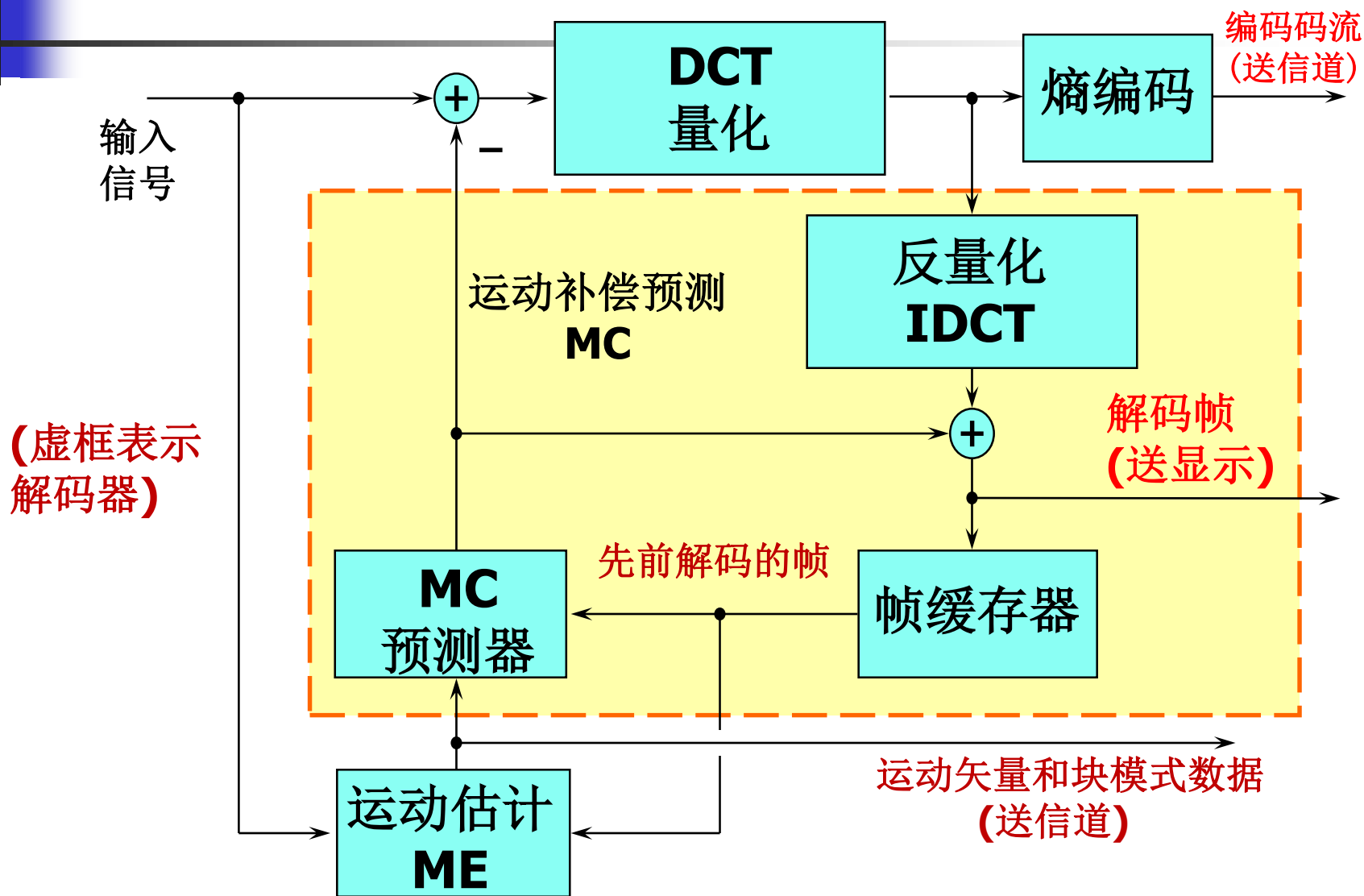
352 × 288

CIF

176 × 144

QCIF

基于 MC+DCT 混合视频编码器





H.263 : 低码率视频编码标准

- ITU-T H.263 (V1: 1995): 取代 H.261, 成为视频会议、可视电话和 Internet 视频流的压缩标准。
 - H.263 + (1997/1998年) & H.263 + + (2000年)
 - 图像格式通常为 CIF、QCIF or Sub-QCIF (128×96) , 帧率一般低于 10fps。
 - 数码率: 任意, 典型 20kbps (对于 PSTN) 。
 - 在所有的数码率上对优于 H.261, 在一半数码率上与 H.261 具有相同的质量
 - 是 MPEG4 标准的压缩 Core。



H.263 vs. H.261

■ 改进的**运动补偿**

- H.261: 整数像素精度, 环路滤波器, 每个 MB 一个运动矢量
- H.263: **半像素精度, 无环路滤波器**, 每个 MB 一个运动矢量

■ 减少了 Overhead

■ 支持更多的图像格式

■ **附录**中定义了选项

- 无限制的运动矢量 (附录 D)
- 基于语法的算术编码 (SAC) (附录 E)
- **高级预测模式 (AP)** (附录 F)
 - 重叠块运动补偿 (Overlapped Block Motion Compensation, OBMC)
 - 每个 MB 中 1 或 4 个运动矢量切换
- PB 帧图像 (附录 G)

■ 在 H.263++ 增加了更多选项



4.1.2 MPEG-x系列标准

MPEG (Moving Picture Experts Group)是**运动图像专家组**的英文缩写。即ISO/IEC JTC1/SC29/WG11。这个专家组开发的标准通常称为MPEG标准。

- **MPEG-1**: 针对1.5 Mbit/s以下数码率的数字存储媒体应用的运动图像及其伴音编码, 标准号**ISO/IEC 11172**。
- **MPEG-2**: 运动图像及其伴音信息的通用编码, 标准号**ISO/IEC 13818**。
- **MPEG-4**: 视听对象编码, 标准号**ISO/IEC 14496**。
- **MPEG-7**: 多媒体内容描述接口, 标准号**ISO/IEC 15938**。
- **MPEG-H Part 2**: 同**H.265**, 标准号**ISO/IEC 23008-2**。



4.1.2 MPEG-x系列标准

- MPEG-1是**针对1.5Mbit/s以下数据传输率的数字存储媒介应用的运动图像及其伴音编码的国际标准。**
- MPEG-1的目标是将压缩后的视/音频码流存入光盘（如CD-ROM、VCD等），数据传输率为**1.416 Mbit/s**，其中**1.1 Mbit/s用于视频**，**128 kbit/s用于音频**，其余的用于系统开销。
- MPEG-1是一个开放的、统一的标准，在商业上获得了巨大的成功。尽管其图像质量仅相当于**VHS**视频的质量，还不能满足广播级的要求，但已广泛应用于VCD等家庭视听产品中。



4.1.2 MPEG-x系列标准

- 在制订MPEG-1的过程中，广播电视的设备制造商立即意识到MPEG技术对提高卫星转发器和有线电视信道效率的潜力。但数字电视广播不能满足于VHS的图像质量，数码率也不必像MPEG-1限制得那样低。于是MPEG为数字电视广播的应用制订了MPEG-2标准。
- MPEG-1是MPEG-2的一个子集，任何MPEG-2的解码器要能够对MPEG-1的码流进行解码。
- MPEG-2不是MPEG-1的简单升级，它在系统和传送方面作了更加详细的规定和进一步的完善。
- MPEG-2的应用领域非常广泛，包括存储媒介中的DVD、广播电视中的数字电视和HDTV、以及交互式的视频点播（VOD）等。



4.1.2 MPEG-x系列标准

■ MPEG-4

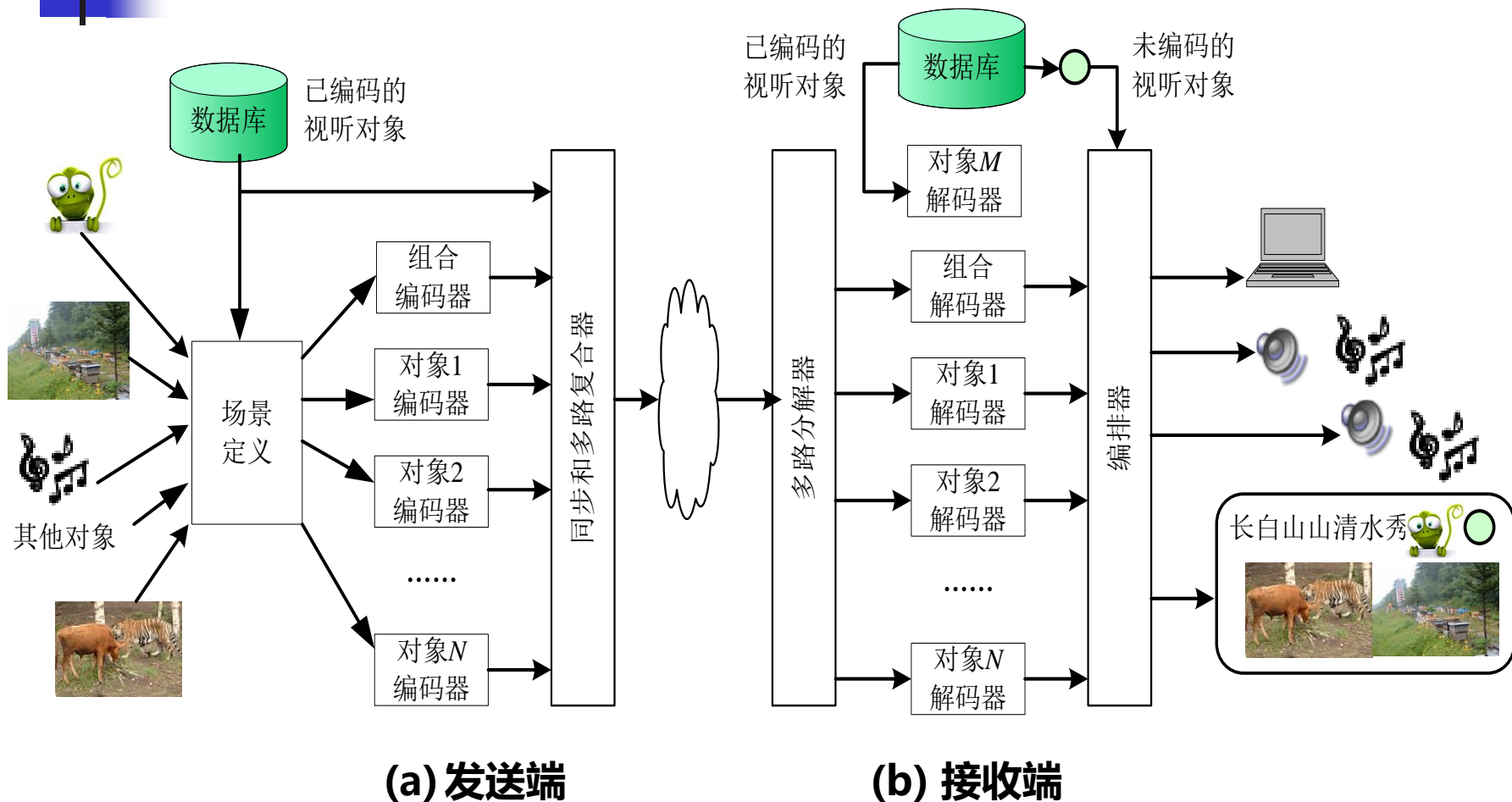
- 视听对象编码(Coding of audio-visual objects)标准(ISO/IEC 14496), 始于1993年的多媒体应用标准
 - 1998年10月发布了版本1
 - 1999年底完成了版本2
 - 2000年初正式成为国际标准
 - 不断更新和完善.....
- 目标是为通信、广播、存储和其他应用提供数据速率低而视听质量高的数据编码方法和交互播放工具
 - 吸收了MPEG-1、MPEG-2和其他相关标准的许多特性
 - 引入了视听对象(audio-visual objects, **AVO**)编码的概念



4.1.2 MPEG-x系列标准

- 扩充了编码类型，由自然对象扩展到合成对象
- 采用了合成对象与自然对象混合编码(Synthetic/Natural Hybrid Coding, SNHC)算法
- 引入了组合、合成和编排等重要概念，以实现交互功能和对象重用

4.1.2 MPEG-x系列标准



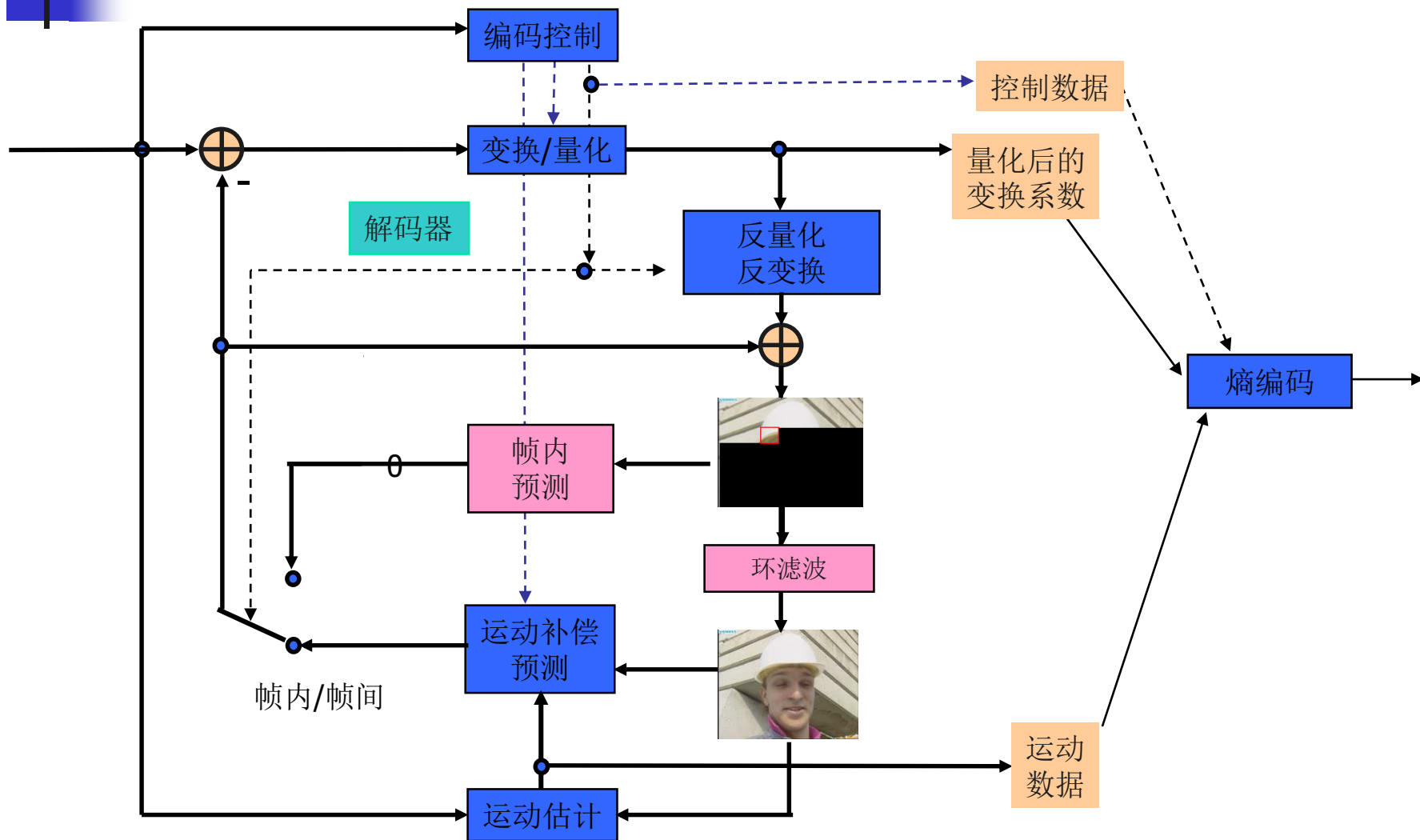
MPEG-4基于对象的系统模型



4.1.3 AVS和AVS+标准

- AVS (Audio Video coding Standard) 是我国具备自主知识产权的第二代信源编码标准，是《信息技术——先进音视频编码》系列标准的简称。
- AVS视频标准 (GB / T 20090.2-2006) 主要面向高清晰度和高质量数字电视广播、网络电视、高密度激光数字存储媒体和其他相关应用，具有以下特点：
 - (1) **性能高**，编码效率是MPEG-2的2倍以上，与H.264的编码效率处于同一水平；
 - (2) **复杂度低**，算法复杂度比H.264明显低，软硬件实现成本都低于H.264；
 - (3) **我国掌握主要知识产权，专利授权模式简单，费用低。**

4.1.3 AVS和AVS+标准





AVS视频编码标准工具集

- 双向预测
- 隔行编码
- 运动矢量预测
- 子像素插值
- 多参考帧预测
- 可变块大小预测
- 帧内预测
- 变换和量化
- 熵编码
- 环路滤波
- 缓冲区管理
- 其它
 - 图像组头
 - 防伪起始码
 - 码流顺序
 - 时间参考索引



AVS+视频标准

- AVS工作组制定了AVS1-P2等视频编码标准，并于2006年颁布为国标GB/T 20090.2-2006，之后于2012年在国标的基础上升级为AVS+，形成了广电行业标准GY/T 257.1-2012。



AVS2视频标准

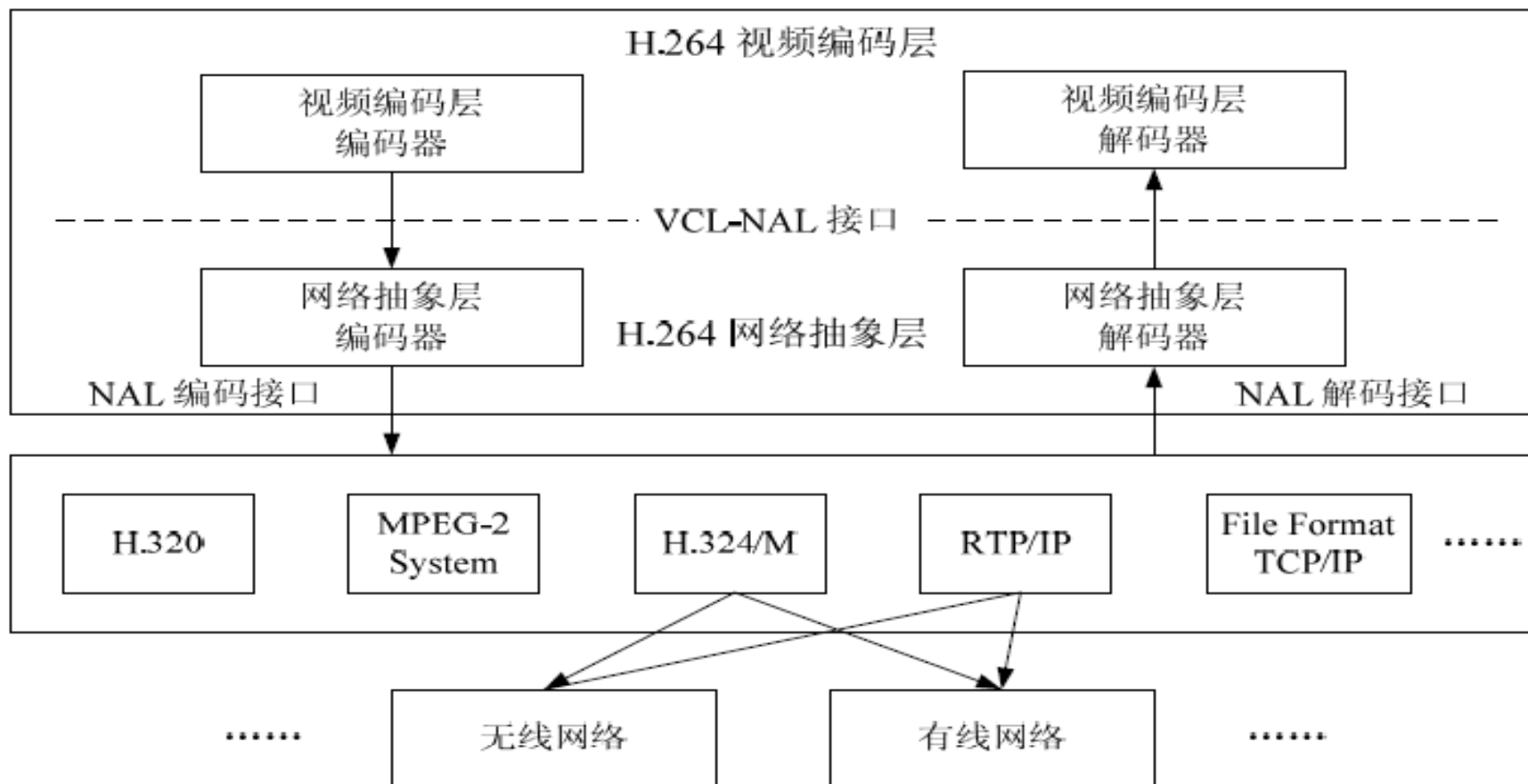
- AVS标准针对视频监控的伸展档**AVS-S2**又于2013年被国际电子电气工程师协会(**IEEE**)标准化委员会接受并颁布为**IEEE 1857**标准。2013年12月31日国家质检总局、国家标准委批准发布了《信息技术 先进音视频编码 第2部：视频》（修订）（国标代号GB/T 20090.2-2013）、《信息技术 先进音视频编码 第10部：移动语音和音频》（国标代号GB/T 20090.10-2013）为国家标准，标准将于2014年7月15日正式实施。
- AVS2是AVS+的下一代，即《信息技术 --先进音视频编码 第2部：视频》（修订）。
- 国标代号GB/T 20090.2-2013，简称AVS2视频标准
- 首要应用目标是超高清晰度（**4K或8K**）视频。
- 测试表明，AVS2视频标准的压缩效率已经比上一代AVS国家标准和AVC/H.264国际标准提高了一倍，在场景类视频编码方面大幅度领先于最新国际标准HEVC，实现复杂度不高于同等级的编码标准。



第4章 数字视频编码标准

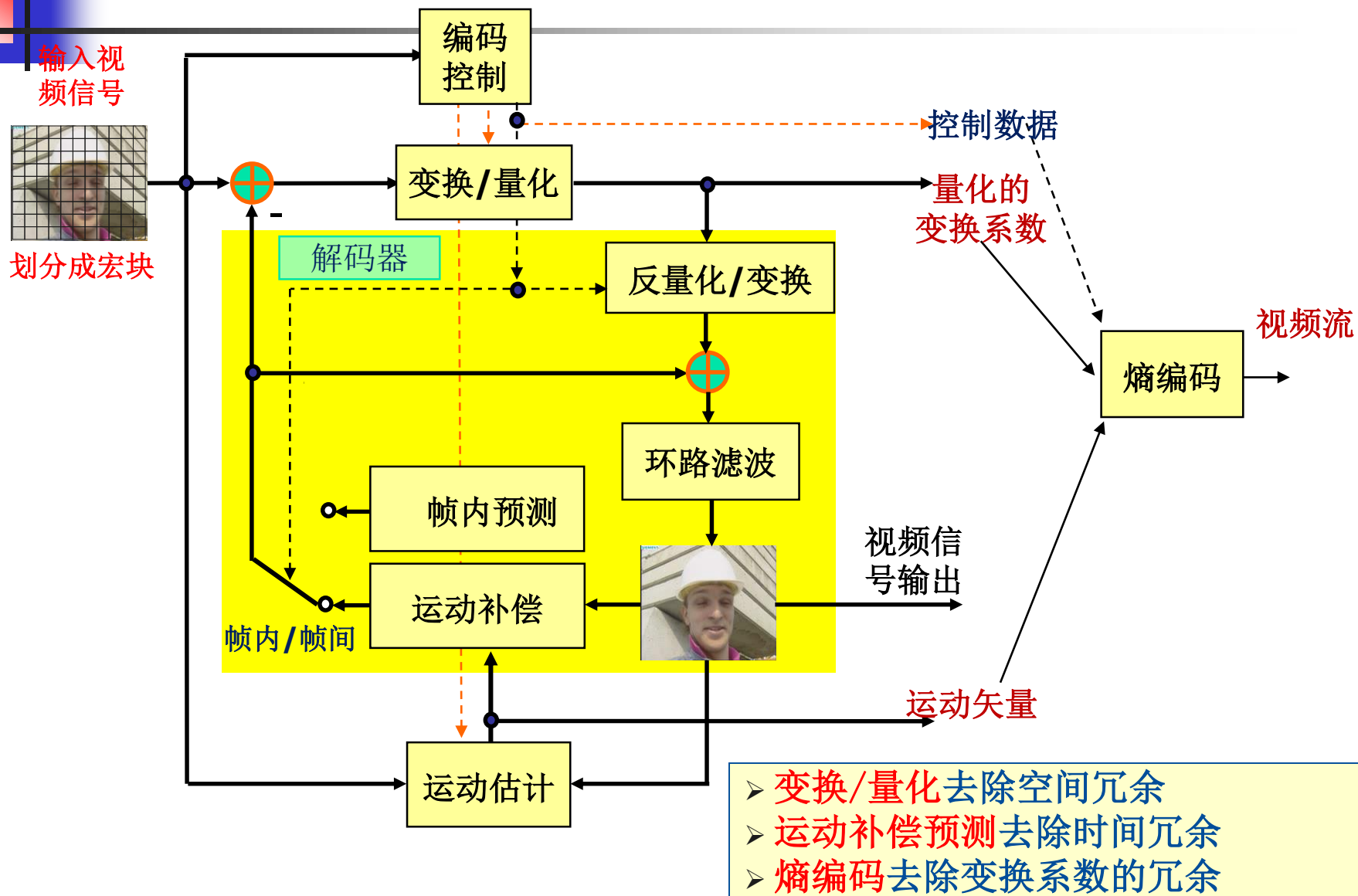
- 4.1 数字视频编码标准概述
- **4.2 H.264/AVC视频编码标准**
- 4.3 H.265/HEVC视频编码标准
- 4.4 AVS与AVS+视频编码标准

H.264/AVC视频编码器的分层结构



- VCL实现视频数据的编解码;
- NAL定义数据封装格式, 为VCL提供与网络无关的统一接口

H.264/AVC视频编码框图





H.264/AVC 标准采用的已有技术

■ 仍然采用先前标准中的 MC + DCT 混合编码算法， **相同的部分**包括：

- 16×16 宏块
- 传统的亮度和色度采样格式 4:2:0
- 块运动位移或矢量
- 运动矢量越过图像边界的算法
- 块大小可变的运动补偿
- 块变换（没有采用小波或分形 wavelets or fractals）
- 标量量化
- I、P 和 B 帧图像类型
- 变字长编码 Variable-length coding



H.264/AVC 标准采用的新技术

- **多模式、多参考帧运动补偿**
- **1/4pixel 运动矢量精度**
- **多模式帧内预测 (I 帧空域预测)**
- **去方块效应环路滤波器**
- **4×4 整数变换**
- **熵编码采用CAVLC 或 CABAC**
- **SP-slices**
- **网络适应层 NAL (Network Abstraction Layer)**



亮度分量 4×4 块预测模式

模式	描述
模式0(垂直)	由A、B、C、D垂直推出相应像素值
模式1(水平)	由I、J、K、L水平推出相应像素值
模式2(DC)	由A~D及I~L平均值推出所有像素值
模式3(下左对角线)	由 225° 方向像素值内插值得出相应像素值
模式4(下右对角线)	由 315° 方向像素值内插得出相应像素值
模式5(右垂直)	由 296.6° 方向像素值内插得出相应像素值
模式6(下水平)	由 333.4° 方向像素值内插得出相应像素值
模式7(左垂直)	由 243.4° 方向像素值内插得出相应像素值
模式8(上水平)	由 26.6° 方向像素值内插得出相应像素值



色度分量/亮度分量 16×16 块预测模式

模式	描述
模式0（垂直）	由上边像素推出相应像素值
模式1（水平）	由左边像素推出相应像素值
模式2（DC）	由上边和左边像素平均值推出相应像素值
模式3（平面）	利用线形“plane”函数及左、上像素推出相应像素值



H.264中的 4×4 整数变换

■ 动机:

- DCT 需要实数运算，反变换中引起了精度的损失。
- 更小的图像，更好的运动补偿，意味着更小的空间相关性，不需要 8×8 变换
- 4×4 的块可以有效的减弱重建图像的块效应

■ H.264 使用了非常简单的整数 4×4 变换:

- 对 4×4 DCT 非常精巧的逼近
- 变换矩阵只包含 ± 1 和 ± 2
 - 计算只需要加法、减法和移位

■ 结果显示仅有可以忽略不计的质量损失 ($\sim 0.02\text{dB}$)



H.264中的 4×4 整数变换

4 x 4 integer DCT

- X : input pixels, Y : output coefficients

$$Y = (C_f \times C_f^T) \otimes E_f$$

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix}$$

$$a = \frac{1}{2}, b = \sqrt{\frac{2}{5}}, d = \frac{1}{2}$$

\otimes Implies element by element multiplication

H.264中的 4×4 整数变换

- 4×4 整数变换的表达式为：

$$Y' = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix}$$

- 基核矩阵 **H** 为：

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

- 由于矩阵 **H** 只有 ± 1 与 ± 2 ，只需加减法和左移即可完成变换，所以 **H.264** 的整数变换是无乘法的。

H.264中的 4×4 整数变换

- 相应的反变换表达式如下：

$$X = C(Y \otimes E_{inv})C^T =$$
$$\begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} y_{00} & y_{01} & y_{02} & y_{03} \\ y_{10} & y_{11} & y_{12} & y_{13} \\ y_{20} & y_{21} & y_{22} & y_{23} \\ y_{30} & y_{31} & y_{32} & y_{33} \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$

- 最终得到反变换表达式如下：

$$X = \begin{bmatrix} 1 & 1 & 1 & \frac{1}{2} \\ 1 & \frac{1}{2} & -1 & -1 \\ 1 & -\frac{1}{2} & -1 & 1 \\ 1 & -1 & 1 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} y'_{00} & y'_{01} & y'_{02} & y'_{03} \\ y'_{10} & y'_{11} & y'_{12} & y'_{13} \\ y'_{20} & y'_{21} & y'_{22} & y'_{23} \\ y'_{30} & y'_{31} & y'_{32} & y'_{33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ 1 & -1 & -1 & 1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} \end{bmatrix}$$

- 只需加法和右移的操作即可完成反变换，而反变换的输入数据具有大的动态范围，所以造成的误差很小。

H.264中4×4 亮度的 DC 系数变换

- 此变换仅用于 16×16 的帧内编码块的亮度信号，每个 4×4 块整数变换后，取出 DC 系数进行再次进行 4×4 Hadamard 变换。
 - 正变换表达式如下：

$$Y_D = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} X_{D00} & X_{D01} & X_{D02} & X_{D03} \\ X_{D10} & X_{D11} & X_{D12} & X_{D13} \\ X_{D20} & X_{D21} & X_{D22} & X_{D23} \\ X_{D30} & X_{D31} & X_{D32} & X_{D33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \right) // 2$$

- 反变换表达式为：

$$X_{QD} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} Y_{QD00} & Y_{QD01} & Y_{QD02} & Y_{QD03} \\ Y_{QD10} & Y_{QD11} & Y_{QD12} & Y_{QD13} \\ Y_{QD20} & Y_{QD21} & Y_{QD22} & Y_{QD23} \\ Y_{QD30} & Y_{QD31} & Y_{QD32} & Y_{QD33} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

H.264 中 2×2 色度的 DC 系数变换

- 每个宏块有两个 8×8 色度块，每个色度块分为 4 个 4×4 块。
- 色度块经过变换之后，可分离出两个 2×2 的 DC 系数块，同亮度一样再进行一次 2×2 Hadamard 变换：
 - 正变换表达式为：

$$Y_D = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} X_{D00} & X_{D01} \\ X_{D10} & X_{D11} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- 反变换表达式为：

$$X_{QD} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} Y_{QD00} & Y_{QD01} \\ Y_{QD10} & Y_{QD11} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$



H.264 量化技术

$$Z_{i,j} = \text{round} \left(\frac{Y_{i,j}}{Q_{step}} \right)$$

$Y_{i,j}$ —— 变换后的系数

Q_{step} —— 量化步长

$Z_{i,j}$ —— 量化后的系数

$$Z_{i,j} = \text{round} \left(W_{i,j} \frac{PF}{Q_{step}} \right)$$

$W_{i,j}$ —— 经 $C_f X C_f^T$
变换后未缩放的矩阵系数

PF —— 缩放系数

H.264 量化技术

■ H.264 量化需要注意:

- 避免除法运算和浮点运算;
- 将整数变换中的放大矩阵包含进去

$$Z_{ij} = \text{round} \left(W_{i,j} \cdot \frac{PF}{Q_{\text{step}}} \right)$$

- 其中 W 为整数变换后系数
- PF 为放大因子, 其值取决于其在 4×4 块中的位置

$$\begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}$$



量化参数 Q 与量化步长 Q_{Step} 的对应关系

- H.264 标准定义了52个量化步长，对应52个量化参数。量化参数每增加6，量化步长增加一倍。增加量化步长的范围使得编码器能够在精确自如的控制比特率和图像质量间的均衡。

Q	0	1	2	3	4	5	6	7	8	9	10	11	
Q_{Step}	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	
Q	12		...	18	...	24	...	30	...	36	...	51	
Q_{Step}	2.5		...	5	...	10	...	20	...	40	...	224	



H.264 熵编码

- 动机：传统的编码器使用了固定的变字长码
 - 本质上都是 Huffman 类型码
 - 没有自适应，不能对概率 > 0.5 的符号进行有效编码，因为至少需要一个bit。
 - H.263 Annex E 定义了一个算术编码器
 - 仍然是非自适应的，使用了多个非二进制的字母表，计算高度复杂
- H.264 标准规定了两种熵编码：
 - 基于上下文的自适应变长编码（Context-based Adaptive Variable Length Coding, **CAVLC**）
 - 基于上下文的自适应二进制算术编码（Context-based Adaptive Binary Arithmetic Coding, **CABAC**）



H.264 CAVLC 熵编码

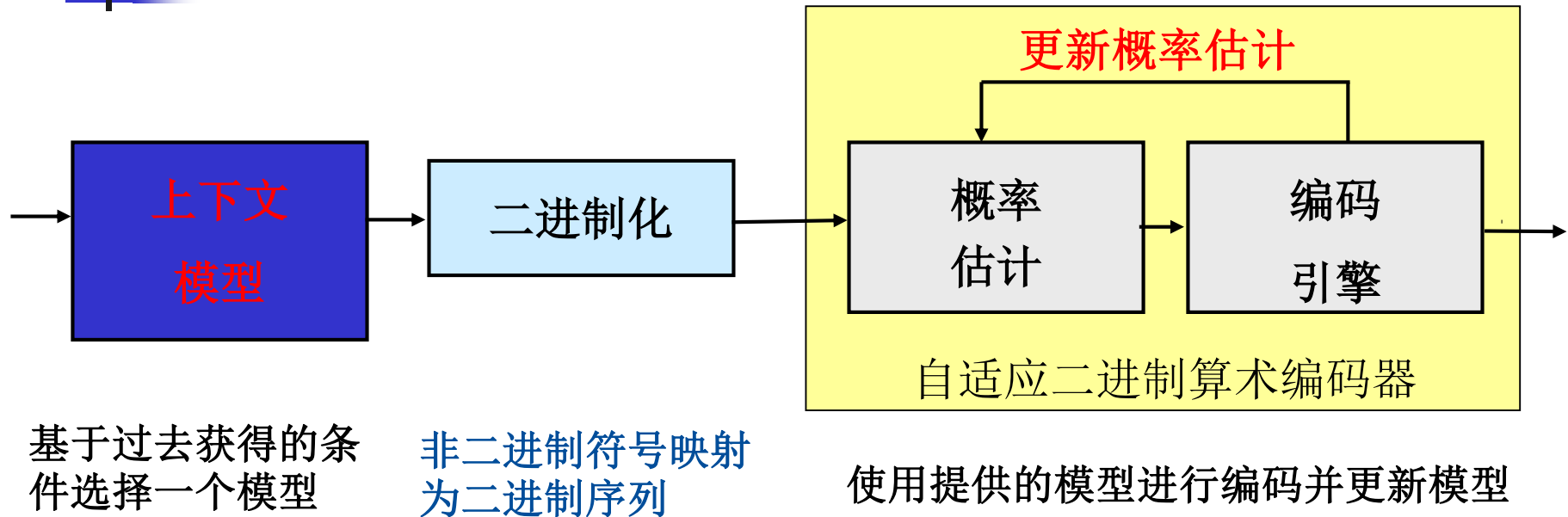
- 变换量化后的 4×4 (或 2×2) 块进行 zig-zag 扫描后, 进行 CAVLC 编码。
量化后的 4×4 块有一些特点:
 - ✧ 经预测、变换和量化后, 块包含许多 0, 采用游程 (Run) 与幅度 (level) 的编码方式可以有效的表示变换系数, 这也是以往传统的编码方式用到的。
 - ✧ 经过量化后, 许多 Level 是 ± 1 , 通过 zig-zag 扫描可以把尾 1 (trailing 1s) 集中起来, CAVLC 把这些 Level 统计出来。
 - ✧ 没有块结束标志, 非零系数的数目是需要编码的。
 - ✧ 非零系数的 Level 越接近 DC 系数越大, 而在高频处比较小。CAVLC 利用这个特点根据当前编码的变换系数选择码表。
- CAVLC 编码对象共有 5 个 (或者编码步骤有 5 步) :
 - ① 非零系数的个数和尾 1 (trailing 1s) 的个数
 - ② 尾 1 的正负号
 - ③ 其余非零系数的幅度 (level)
 - ④ 最后一个非零系数前零的总数
 - ⑤ 游程 (Run)



H.264 CABAC 熵编码

- 对多数符号使用了自适应概率模型
- 通过使用上下文关系，利用了符号相关性。
- 限制为二进制算术编码 (binary arithmetic coding)
 - 所有的语法符号都转换为 bit 串
 - 简单和快速自适应机制
 - 基于只用查表和移位方式的快速二进制算术编解码器
- 399 种预定义的上下文模型，分为组：
 - 例如模型 14-20 用于帧间宏块类型的编码
 - 模型的选择基于前面编码的信息 (上下文关系)
 - 每个上下文模型适应实验分布
- CAVLC 平均节省了比特率 10~15%

H.264 CABAC 熵编码



■ 获得好的性能，源于：

- 通过上下文选择模型
- 基于本地统计的自适应估计
- 算术编码减少计算复杂度



H.264 CABAC 熵编码

■ CABAC 性能 (ITU H.26L (TML4) 测试模型)

- QCIF: 节省 4.5%~15% 比特率
- CIF: 节省 5%~32% 比特率

■ CABAC

- 通过使用**上下文**，利用符号间相关性
- 使用**自适应概率模型**
- 使用**算术编码**使得每个符号有非整数比特

■ CAVLC

- 概率分布是**静态的**
- 码字只能是**整数比特**



H.264 帧和片

- 帧 (Frame) 、片 (Slice)
- 在H.263 和 MPEG 中, 每个帧是P 帧 或I 帧:
 - 例外: P 帧中的宏块可能是帧内编码的, 称之为 I 块 (I-block).
- H.264 推广这点: 每个帧包含一个或多个片(slices):
 - Slice 是临近的宏块序列, 当不采用 FMO 时, 按其内部的光栅扫描顺序进行处理。
 - 每个 slice 独立地编码和解码
 - I-slices, P-slices, B-slices (两个参考帧)
 - FMO (Flexible Macroblock Ordering) 通过 Slice Group 改变了图像分割成 Slice 的方式。每个 Slice Group 以光栅扫描方式分割成一个或多个 Slice。



H.264 帧和片

- 帧 Frames、片 Slices
- 在H.263 和 MPEG 中，每个帧是P 帧 或I 帧：
 - 例外：P 帧中的宏块可能是帧内编码的，称之为 I 块 (I-block).
- H.264 推广这点：每个帧包含一个或多个片(*slices*):
 - Slice 是临近的宏块序列，当不采用 FMO 时，按其内部的光栅扫描顺序进行处理。
 - 每个 slice 独立地编码和解码
 - *I-slices, P-slices, B-slices* (两个参考帧)
 - FMO (Flexible Macroblock Ordering) 通过 Slice Group 改变了图像分割成 Slice 的方式。每个 Slice Group 以光栅扫描方式分割成一个或多个 Slice。



FMO—灵活的宏块排序



The diagram shows a large orange rectangle representing a video frame. Inside this frame, there are three distinct regions: a blue rectangle at the top left labeled 'Slice Group 0', the text 'Slice Group 1' in the center, and a purple rectangle at the bottom left labeled 'Slice Group 2'.

Slice Group 0

Slice Group 1

Slice Group 2



The diagram shows a 16x16 grid of squares in a checkerboard pattern of blue and orange. Two overlapping rectangles are shown: a blue one on the left labeled 'Slice Group 0' and an orange one on the right labeled 'Slice Group 1'.

Slice Group 0

Slice Group 1

H.264 与 MPEG-2 的性能比较

Foreman, QCIF, 30 fps



MPEG-2
203 kbps



H.264
39 kbps

H.264 与 MPEG-2 的性能比较

Comparison of MPEG-2 and H.264/AVC

H.264/AVC @ 340 kbit/s

MPEG-2 @ 1024 kbit/s



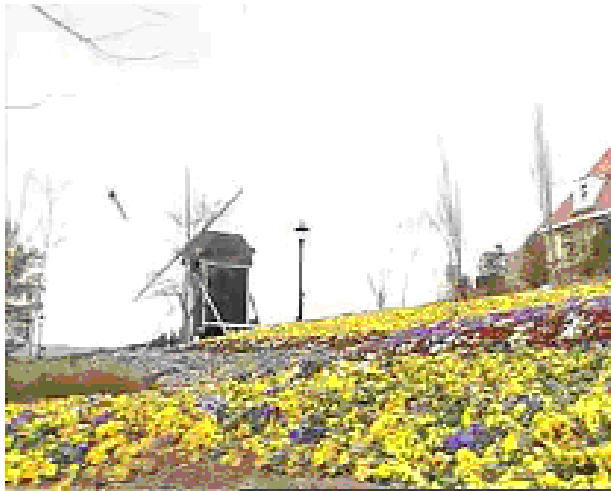
CIF, 30Hz : 340 & 1024 kbit/s



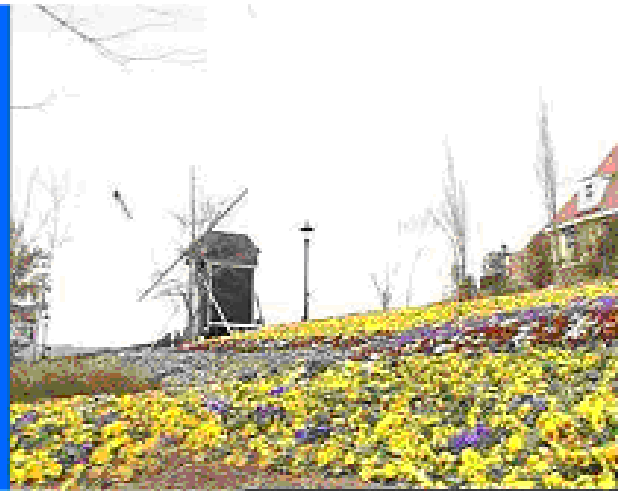
H.264 与 MPEG-2 的性能比较

Comparison of MPEG-2 and H.264/AVC

H.264/AVC @ 340 kbit/s



MPEG-2 @ 1024 kbit/s



CIF, 30Hz : 340 & 1024 kbit/s





H.264/AVC 小结

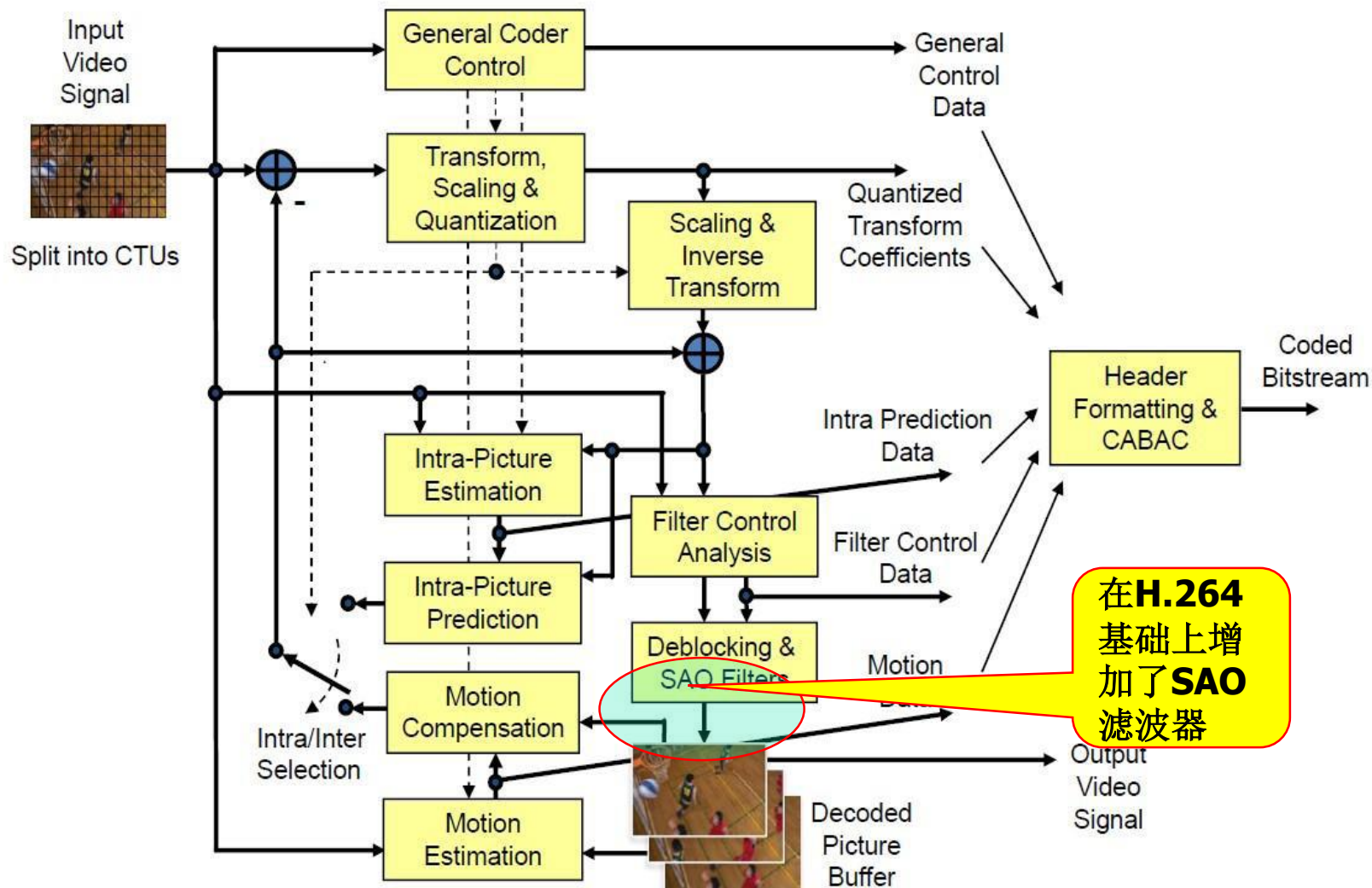
- 视频编码基于混合视频编码方案，原理上与其它标准类似，但有重要差别。
- **新的关键技术特征：**
 - 增强的运动补偿
 - 小尺寸的整数变换
 - 增强的熵编码 CAVLC 和 CABAC
 - 其它：增强的去块效应滤波器等
- 和其它标准相比，在相同质量下，**节省比特率 50% 以上**，特别是对于允许使用 B 帧的较高延时应用。
- 主观质量感觉比客观 PSNR 指标更好。
- **复杂度增加，编码器 3~4倍，解码器 2~3倍。**
- ITU-T VCEG 和 ISO/IEC MPEG 联合制定的标准 H.264/MPEG4-10 AVC



第4章 数字视频编码标准

- 4.1 数字视频编码标准概述
- 4.2 H.264/AVC视频编码标准
- **4.3 H.265/HEVC视频编码标准**
- 4.4 AVS与AVS+视频编码标准

4.3.1 H.265/HEVC视频编码原理





H.265与H.264各模块技术差异

	H.264	H.265
MB/CU大小	$4 \times 4 \sim 16 \times 16$	$4 \times 4 \sim 64 \times 64$
Inter插值	Luma-为6抽头系数插值 Chroma双线性插值	Luma-1/2像素采用8抽头插值滤波 Luma-1/4像素采用7抽头插值滤波器 Chroma所有分数像素点采用4抽头系数插值
Inter MVP预测方法	空域中值MVP预测	空域+时域MVP预测候选列表（AMVP） 空域+时域的Merge/Skip的候选列表
Intra预测	亮度 4×4 块：9种模式 亮度 8×8 块：9种预测模式 亮度 16×16 块：4种预测模式 色度：4种预测模式	亮度所有尺寸的CU块：总共35种预测模式 色度所有尺寸的CU块：5种预测模式

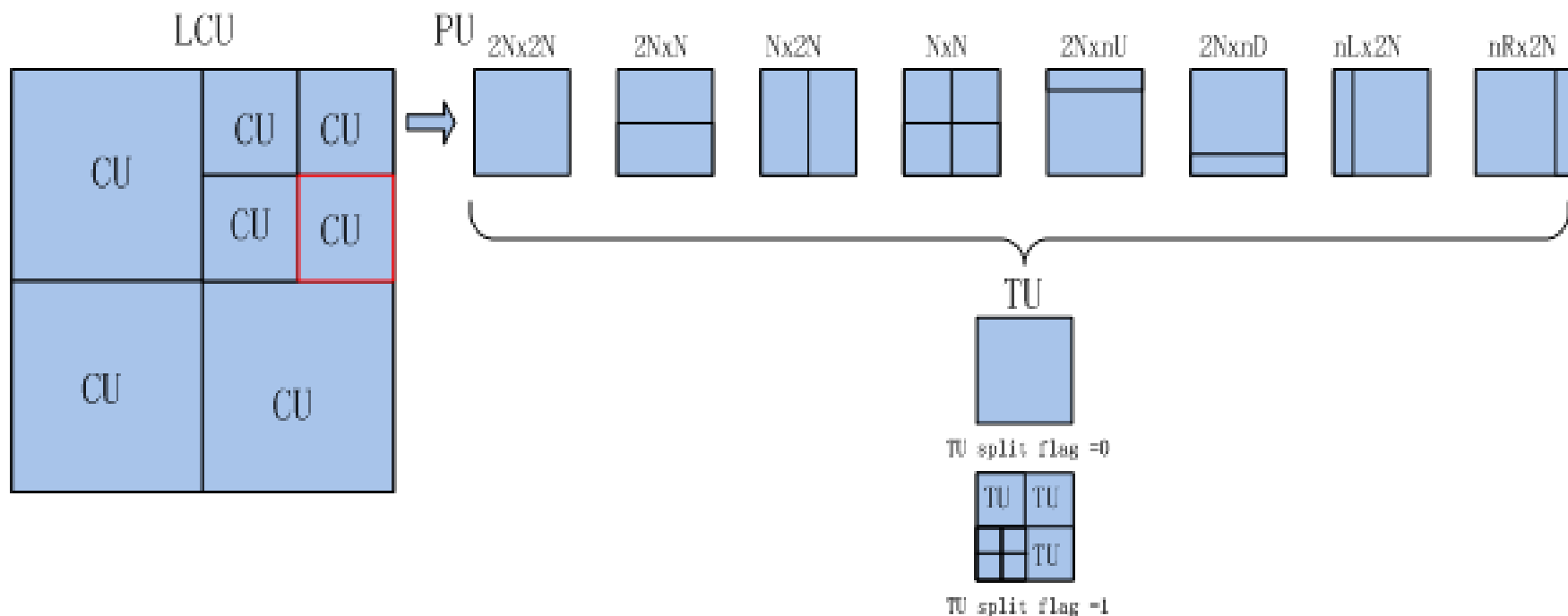


H.265与H.264各模块技术差异(续)

变换	DCT $4 \times 4/8 \times 8$	DCT $4 \times 4/8 \times 8/16 \times 16/32 \times 32$ DST 4×4
滤波器	4×4 和 8×8 边界去块滤波	● 8×8 及以上的CU、PU、TU边界去块滤波 ●SAO滤波器
熵编解码技术	CAVLC及CABAC	CABAC
其他技术	FMO映射关系等	Tile、WPP以及dependent Slice

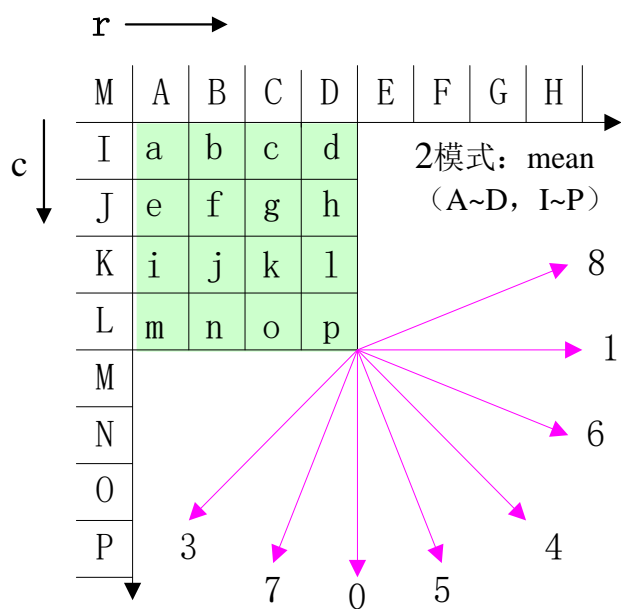
4.3.2 基于四叉树结构的编码单元划分

- 采用**CU (Coding Unit)**、**PU(Prediction Unit)**和**TU(Transform Unit)**的块划分结构
- 这三者之间的关系主要是以**LCU**为基本编码单元，在**LCU**递归划分为**CU**块，每个**CU**块可以划分成不同的**PU**块，同时也在**CU**的基础上可以进行**TU**块的递归划分

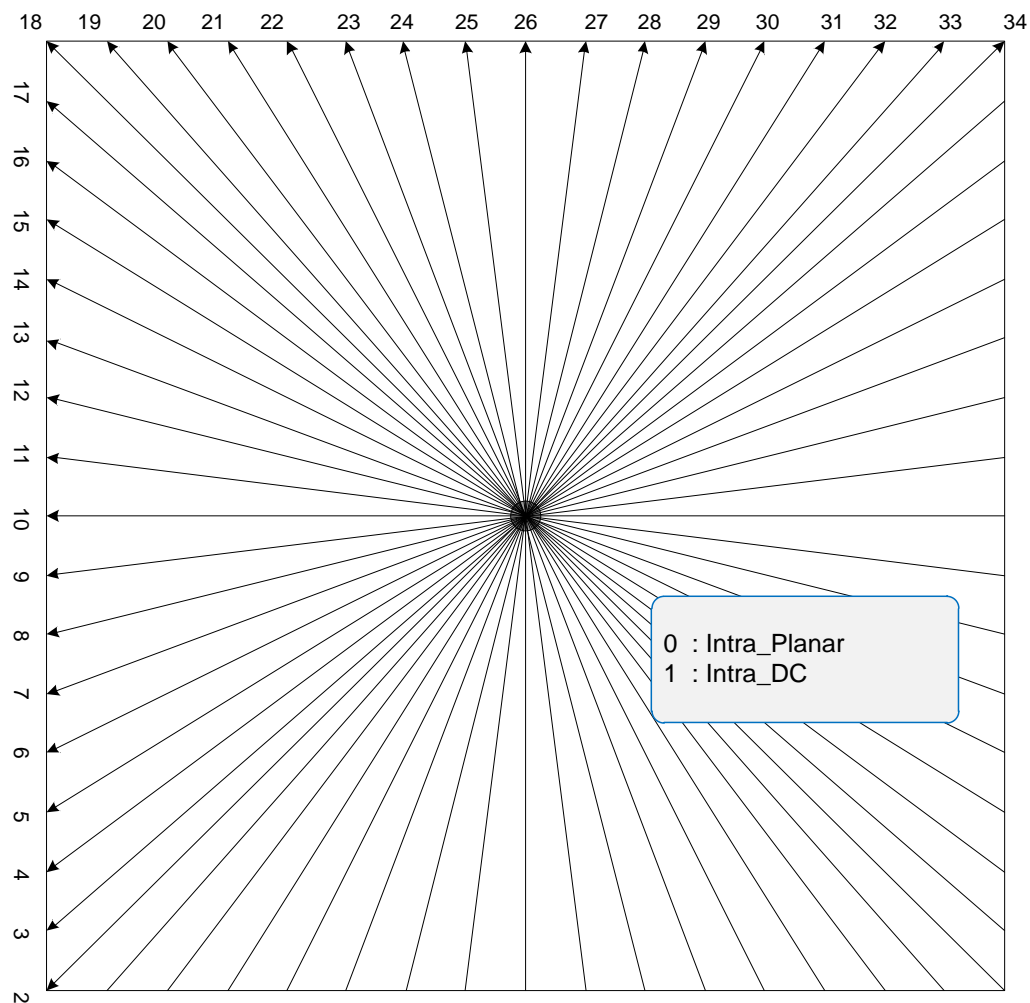


4.3.3 帧内预测

H.264帧内预测方向



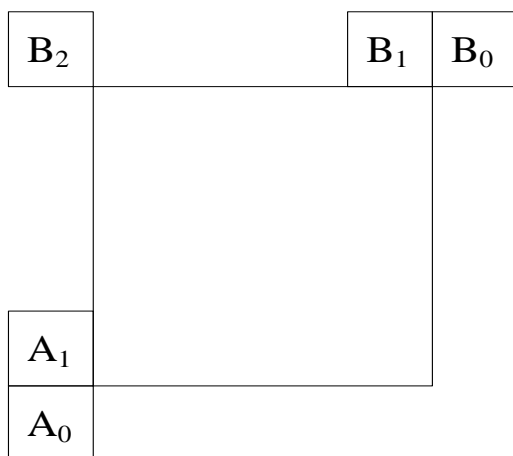
H.265帧内预测方向



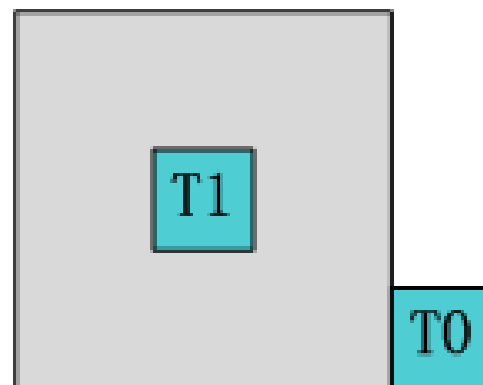
4.3.4 帧间预测

- 本质上H.265是在H.264基础上增加插值的抽头系数个数，改变抽头系数值以及增加运动矢量预测值的候选个数，以达到减少预测残差的目的。
- H.265与H.264一样插值精度都是亮度到1/4，色度到1/8精度，但插值滤波器抽头长度和系数不同
- H.265的增加了运动矢量预测值候选的个数，而H.264预测值只有一个

H.265空域候选项



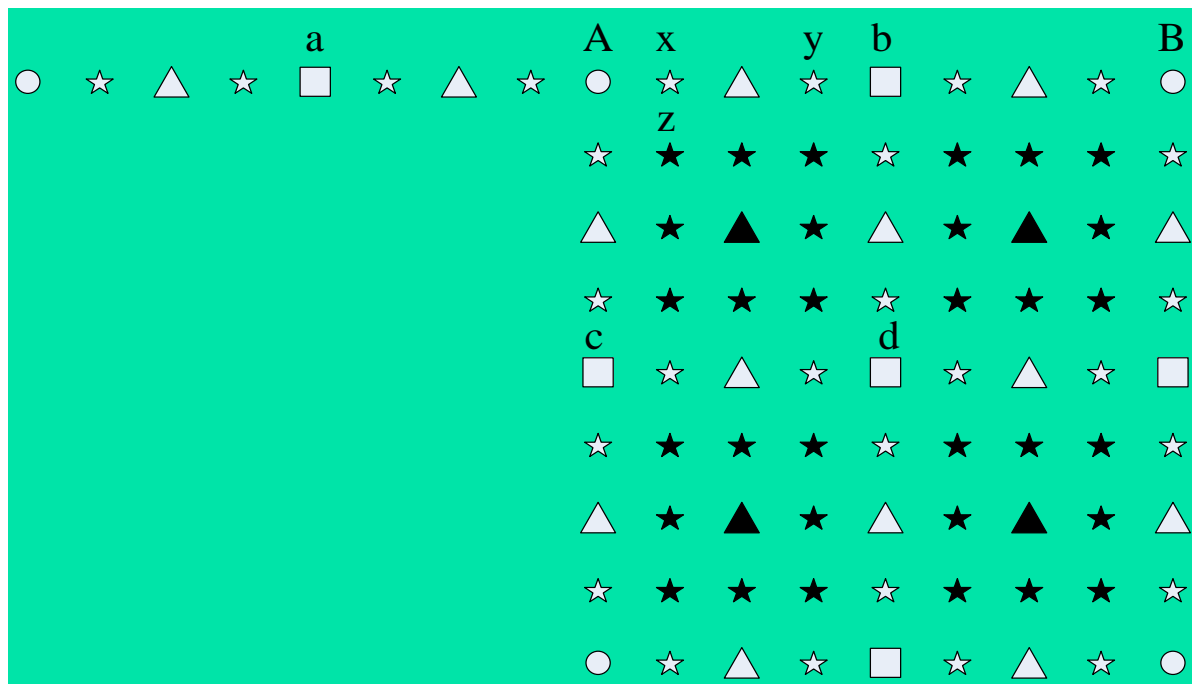
H.265时域共同位置候选项



4.3.4 帧间预测

■ 1/8像素插值

- 插值滤波器设计对数据存取的吞吐量、计算复杂度起着重要的影响作用，例如，AVS 1/2像素插值滤波器为 $(-1, 5, 5, -1)$ 与H.264的 $(1, -5, 20, 20, -5, 1)$ 相比，插值一个 8×8 块所需取点数分别为 11×11 , 13×13





4.3.5 变换与量化

- H.265/HEVC采用的变换运算和H.264/AVC类似，也是一种对预测残差进行近似DCT的整数变换，但为适应较大的编码单元而进行了改进。
- H.265/HEVC的量化机理和H.264/AVC基本相同，是在进行近似DCT的整数变换时一并完成的。



4.3.5 变换与量化

- H.265/HEVC采用的变换运算和H.264/AVC类似，也是一种对预测残差进行近似DCT的整数变换，但为适应较大的编码单元而进行了改进。
- H.265/HEVC的量化机理和H.264/AVC基本相同，是在进行近似DCT的整数变换时一并完成的。



4.3.6 环路滤波

■ 去方块效应滤波器

H.265的去块滤波与H.264的去块滤波及流程本质上是一致的，但做了如下**改变**：

- **滤波边界**：H.264最小到4x4边界滤波；而H.265适应最新的CU、PU和TU划分结构的滤波边缘，最小滤波边界为8x8
- **滤波顺序**：H264先宏块内采用垂直边界，再当前宏块内水平边界；而H.265先整帧的垂直边界，再整帧的水平边界

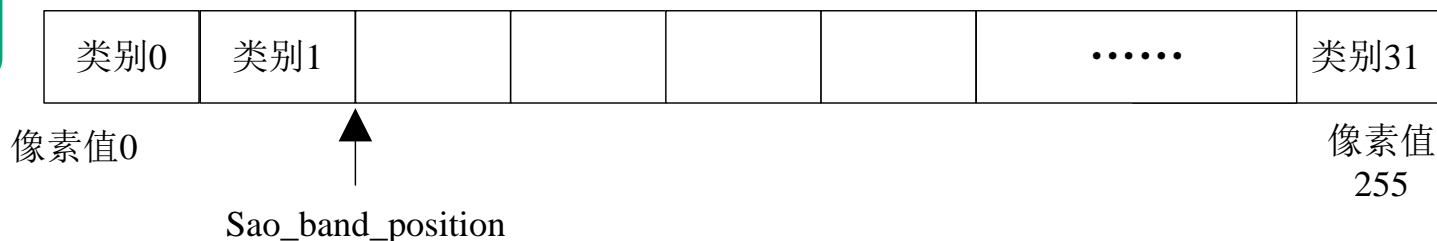
4.3.6 环路滤波

■ 样值自适应偏移 (SAO)

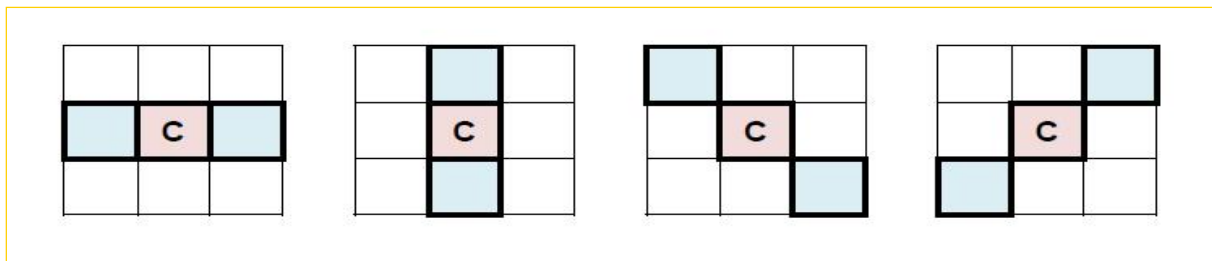
SAO(sample adaptive offset)滤波其实就是对去块滤波后的重建像素按照不同的模式进行分类，并对每一种分类像素进行补偿，分类模式分为BO(Band offset)和EO(Edge offset)。

BO总共有32中类型，只取从sao_band_position开始的连续四个类别作为当前SAO类别

BO模式



EO模式





4.3.7 上下文自适应的熵编码

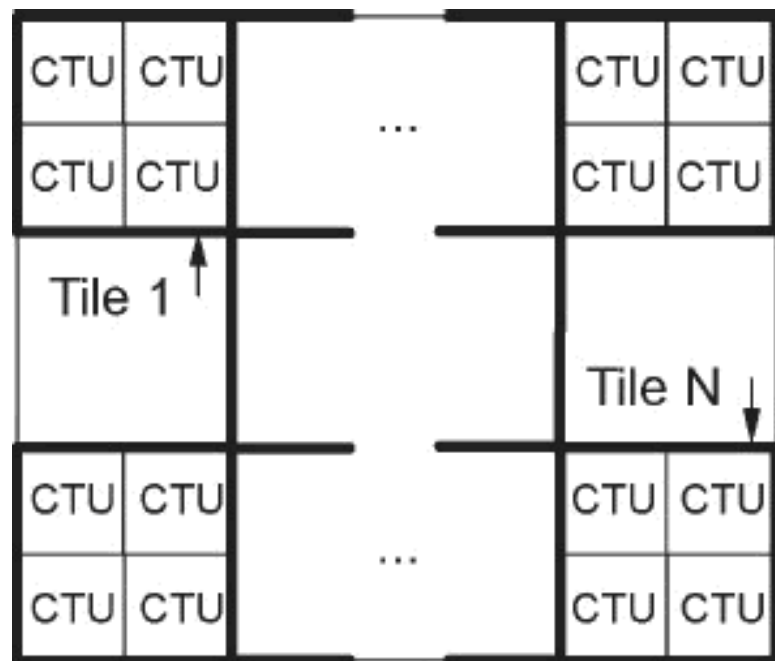
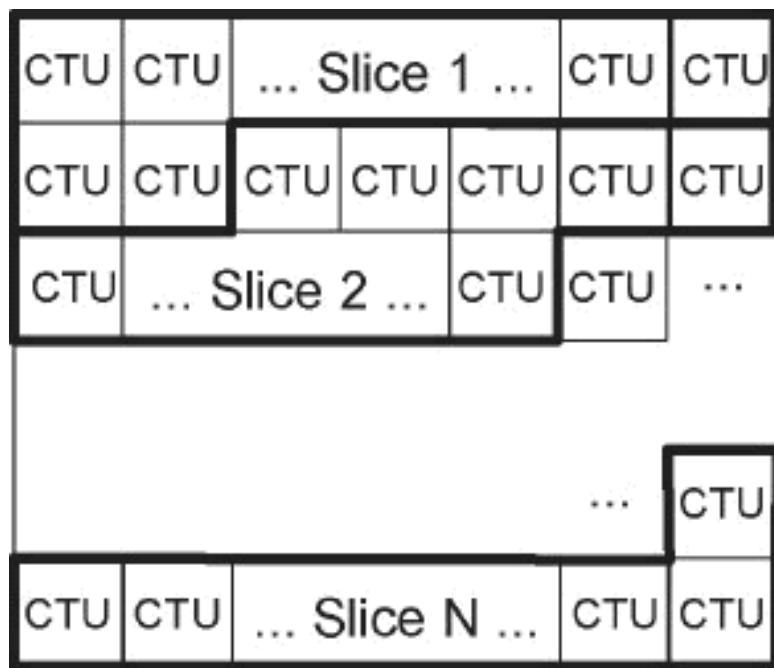
H.265/HEVC标准中使用的上下文自适应的二进制算术编码（CABAC）与H.264/AVC中使用的CABAC基本类似，除了上下文建模过程中概率码表需要重新布置以外，在算法上并没有什么变化。

CABAC 主要包括以下三个模块。

- 语法元素的二值化
- 上下文建模
- 算术编码

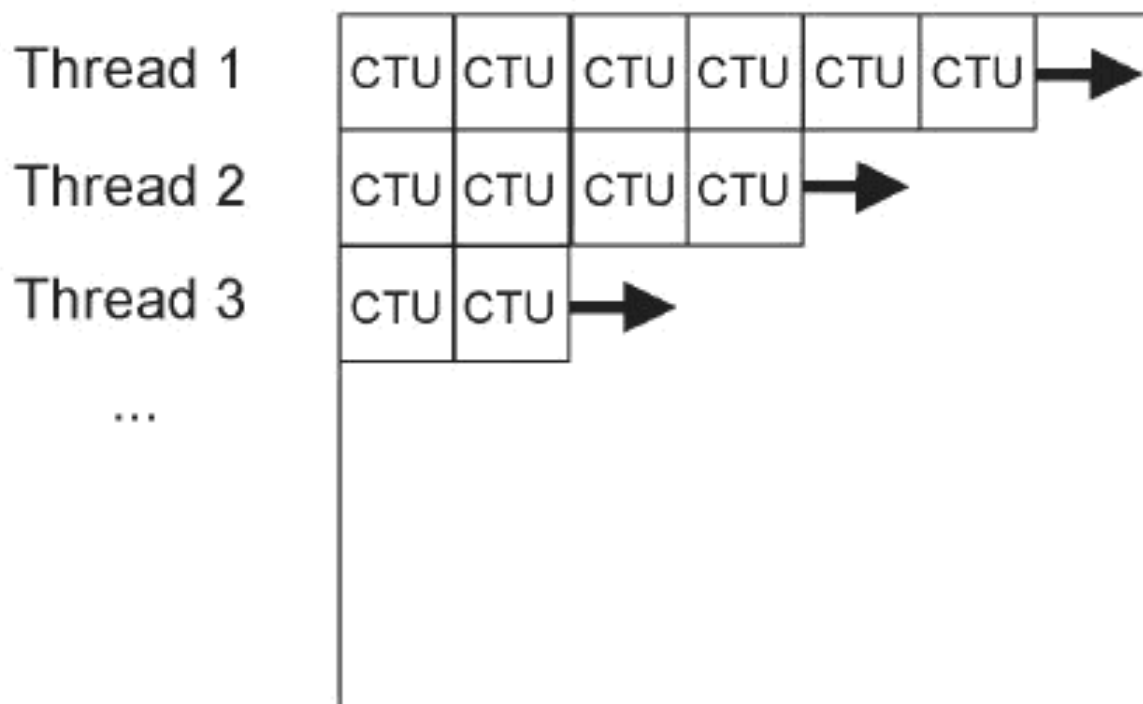
4.3.8 并行化处理

- **Tile:** 将图像分割为**矩形区域**。其主要**目的**增强**并行处理**性能。每个**Tile**区域相当于一幅子图像，可以独立的以LCU块为单位进行编解码。一个**Tile**块为基本的并行单元，每个**Tile**为一个子码流



4.3.8 并行化处理

- **波前并行处理(WPP):** 全称为 wavefront parallel process, 以LCU行为基本的编码单位。以一行LCU块为基本的并行单元, 每一行LCU为一个子码流





Question?

