

数字图像与视频处理

卢官明 唐贵进 崔子冠 编著

机械工业出版社  
CHINA MACHINE PRESS



“十三五”江苏省高等学校重点教材 (编号: 2017-2-029)  
高等院校通信与信息专业规划教材

# 数字图像 与视频处理

DIGITAL IMAGE AND VIDEO PROCESSING



卢官明 唐贵进 崔子冠 编著

机械工业出版社  
CHINA MACHINE PRESS

教学资源下载网站  
<http://www.cmpedu.com>

# 第6章 数字图像与视频压缩 编码标准

熊健

南邮通信学院



## 第6章 数字图像与视频压缩编码标准

---

- 6.1 静止图像编码标准
- 6.2 数字视频编码标准概述
- 6.3 H.264/AVC视频编码标准
- 6.4 H.265/HEVC视频编码标准
- 6.5 AVS/AVS+视频编码标准



## 6.1 静止图像编码标准

---

- 静止图像包括两类：黑白（二值）静止图像和连续色调（彩色或灰度）静止图像。
- 对于静止图像压缩编码，已有多个国际标准，如国际标准化组织（ISO）制订的JBIG标准（ISO 11544）、JPEG标准（ISO 10918）、JPEG2000标准（编号为ISO 15444，等同的ITU-T编号为T.800）等。本节将主要介绍JPEG和JPEG2000标准。



## 6.1.1 JPEG 标准概述

- **JPEG**是Joint Photographic Experts Group (**联合图片专家组**)的缩写。
  - 由ISO和IEC两个组织机构**联合**组成的专家组，负责制定**静态的**数字图像数据压缩编码标准
- **JPEG标准**
  - **标准号**: ISO/IEC 10918或ITU-T T.81
  - **标准名**: Information technology — Digital compression and coding of continuous-tone still images  
(信息技术 — 连续色调静态图像的数字压缩和编码)
- **JPEG 标准适用范围**
  - **灰度**图像, **彩色**图像
  - **静止**图像的压缩, 视频序列**帧内**图像压缩
  - JPEG可以大范围地调节图像**码率**和**质量**



## ■ JPEG算法与颜色空间无关

- RGB和YUV颜色空间之间的转换不包含在JPEG算法中。
- 可压缩来自不同颜色空间的数据，如RGB,  $YC_bC_r$ , CMYK。
- JPEG算法对每个彩色分量进行**单独**编码，对不同的分量可以采用**不同的**量化参数和熵编码表。



## 6.1.2 JPEG 基本编码系统

### ■ 基于 DCT 的顺序编码算法

- 利用视觉系统特性，使用**正交变换**、**量化**和**熵编码**相结合的方法，以去掉或减少视觉的冗余和数据本身的冗余
- JPEG基本编码系统的算法大致分成三个步骤：
  - **正交变换**：使用正向离散余弦变换(FDCT)把空间域上的图像数据变换成频率域上的变换系数
  - **量化**：使用加权函数对DCT系数进行量化，加权函数对人的视觉系统是最佳的
  - **熵编码**：使用霍夫曼编码器对量化系数进行编码（专利因素没有使用算术编码）



# 基于DCT的顺序编码算法基本步骤

- 将原图像用亮度、色差表示（分量图像采样4：1：1）
- 分成 $8 \times 8$ 数据块，数据[0~255]转换为[-128~127]
- 进行正向离散余弦变换(FDCT)
- 量化(quantization)
- Z字形(zig-zag)扫描
- 使用DPCM对直流系数(DC)进行编码
- 使用行程编码对交流系数(AC)进行编码
- 熵编码(entropy coding)：哈夫曼或算术编码





# JPEG 压缩效果



压缩 21:1  
(12KB)



原始图像  
(512×512×8)  
(256KB)



压缩 51:1  
(5KB)



## 6.1.3 基于DCT的渐进编码

### ■ 基本思想

- 此模式与顺序模式编码步骤基本一致，不同之处在于渐进编码模式每个图像分量的编码要经过多次扫描才完成。第一次扫描只进行一次粗糙的压缩，然后根据此数据先重建一幅质量低的图像，以后的扫描再作较细的扫描，使重建图像质量不断提高，直到满意为止



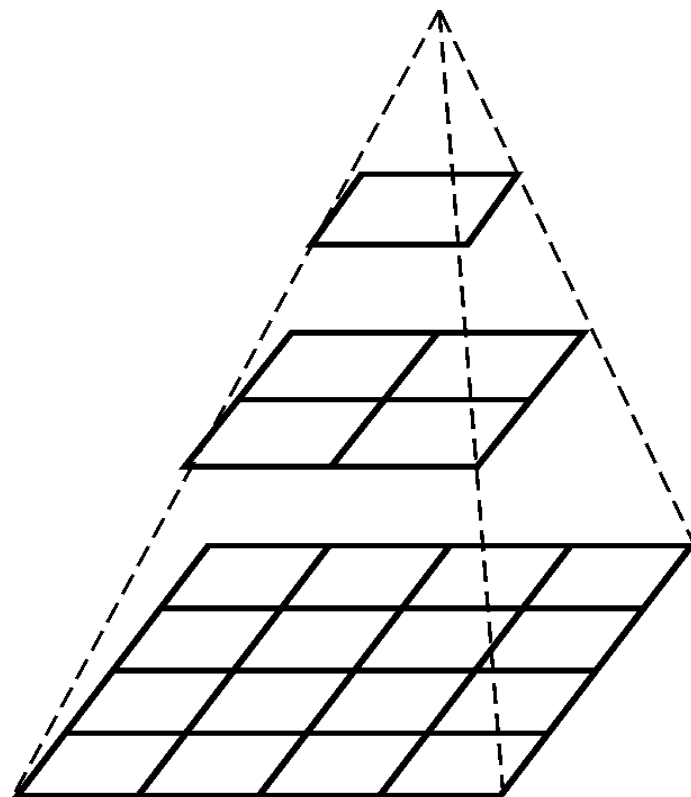
## 6.1.3 基于DCT的渐进编码

### ■ 基本步骤

- (1) **降低**原始图像的空间分辨率。
- (2) 对已经降低分辨率的图像按照**顺序编码**模式进行压缩并存储或传输。
- (3) 对低分辨率图像进行**解码**，然后用**插值**法提高图像的分辨率。
- (4) 将分辨率已经升高的图像作为原图像的**预测值**，并把它与原图像的**差值**进行基于DCT的编码。
- (5) **重复**步骤3、4直到图像达到完整的分辨率。

## 6.1.4 分级编码

- 分级编码的操作模式是把一幅原始图像的空间分辨率分成多个低分辨图像进行“锥形”编码的方法。例如，水平方向和垂直方向分辨率均以  $2^n$  的倍数改变，如图所示。

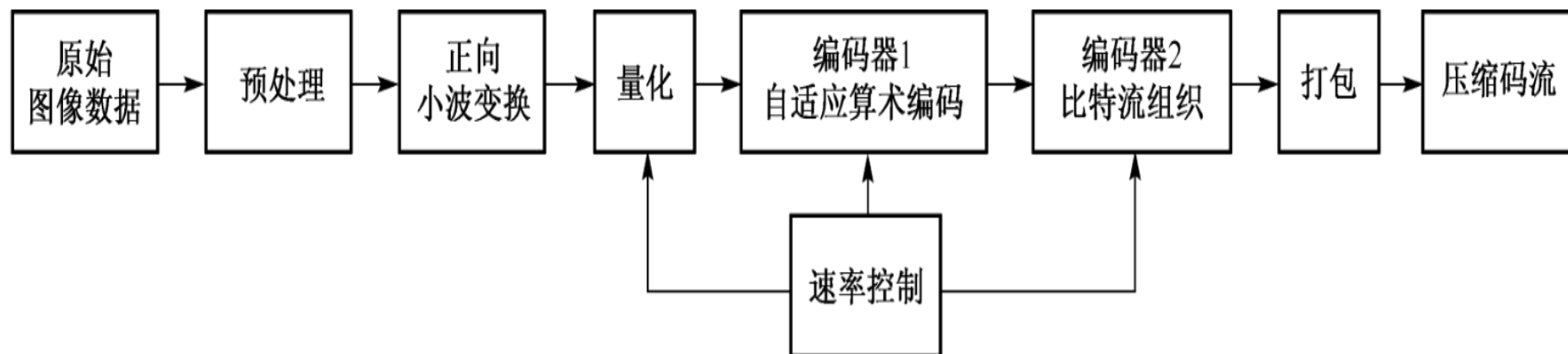




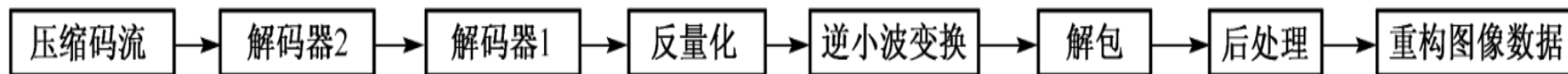
## 6.1.5 JPEG2000 标准概述

- ITU-T SG8和ISO/IEC JTC1 SC29 WG1联合开发的静态图像压缩编码标准(ISO/IEC 15444)，标准名为“**JPEG 2000: Image Coding System**”。
- 开发工作始于1996年1月，其目标是增强对连续色调图像的压缩效率、管理和传输，而又不使图像质量有明显的损失。
- 使用**小波变换**技术提高压缩比，用户可控制图像的分辨率，用在网络上传输时可按照用户要求下载各种分辨率的图像。
- 可提供**无损压缩**的图像，在文档中可提供更多的颜色信息。

## 6.1.6 JPEG2000 标准的基本框架



(a) 编码器



(b) 解码器

**JPEG 2000的编、解码原理框图**

# JPEG2000 压缩效果



Original Carol Image (512 x 512 Pixels, 24-Bit RGB, Size



75:1, 10.6 Kbyte

# JPEG2000 压缩效果



150:1, 5.3 Kbyte



300:1, 2.6 Kbyte





## 6.1.5 JPEG2000 标准概述

---

### JPEG 2000标准的主要内容

- (1) 良好的低比特压缩性能（比JPEG提高10%-30%，适应窄带网络、移动通信）
- (2) 连续色调图像压缩和二值图像压缩（动态范围：1-16bit）
- (3) 同时支持无损压缩和有损压缩
- (4) 渐进传输；
- (5) 支持感兴趣区编码
- (6) 良好的抗误码性
- (7) 向下兼容JPEG



# 第6章 数字图像与视频压缩编码标准

---

- 6.1 静止图像编码标准
- **6.2 数字视频编码标准概述**
- 6.3 H.264/AVC视频编码标准
- 6.4 H.265/HEVC视频编码标准
- 6.5 AVS/AVS+视频编码标准



## 6.2 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



## 6.2 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

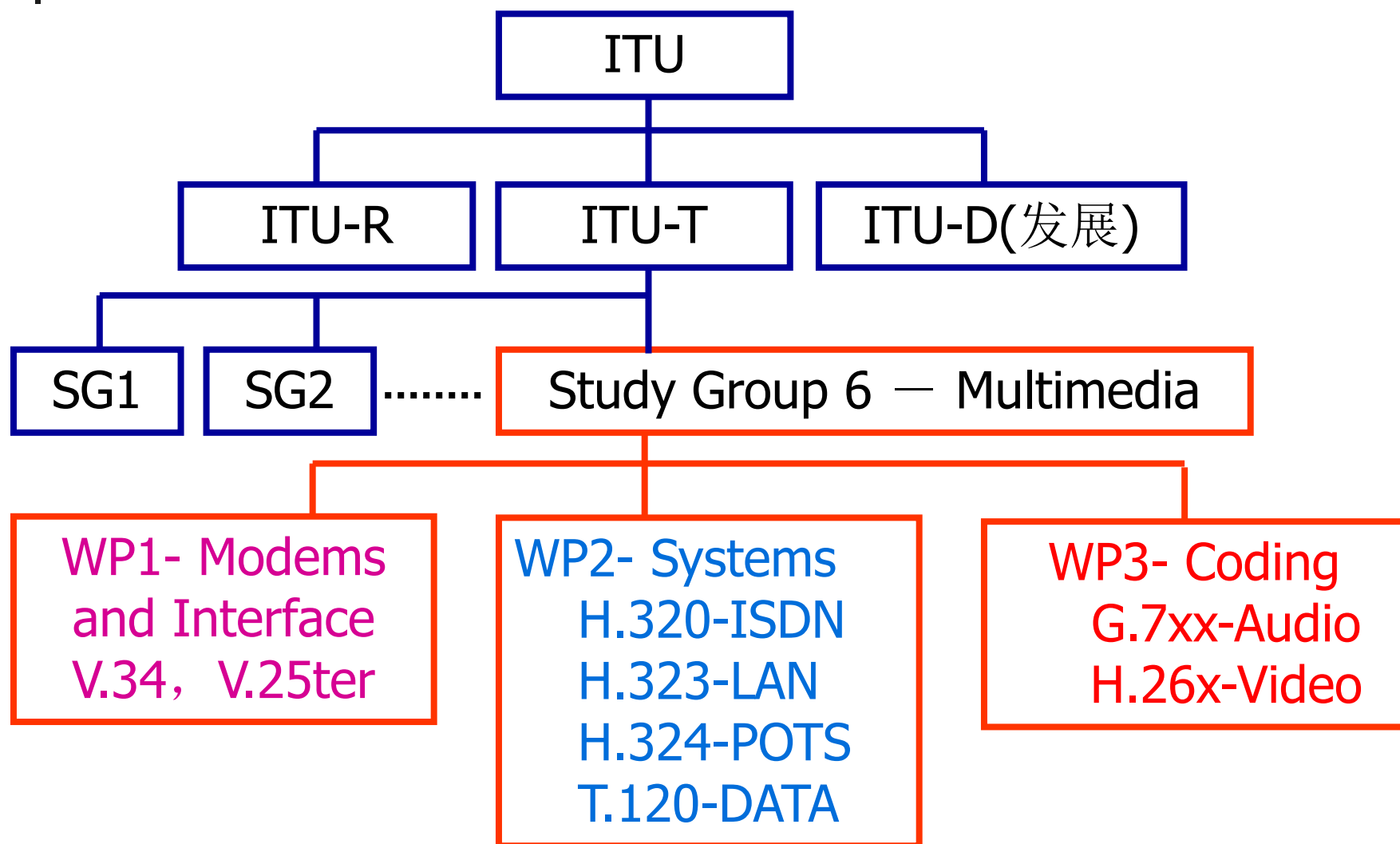
- CCIR (Consultative Committee on International Radio), 1927~1992

- 1992 重组

- CCITT → ITU-T

- CCIR → ITU-R

# ITU 视频编码相关的研究小组





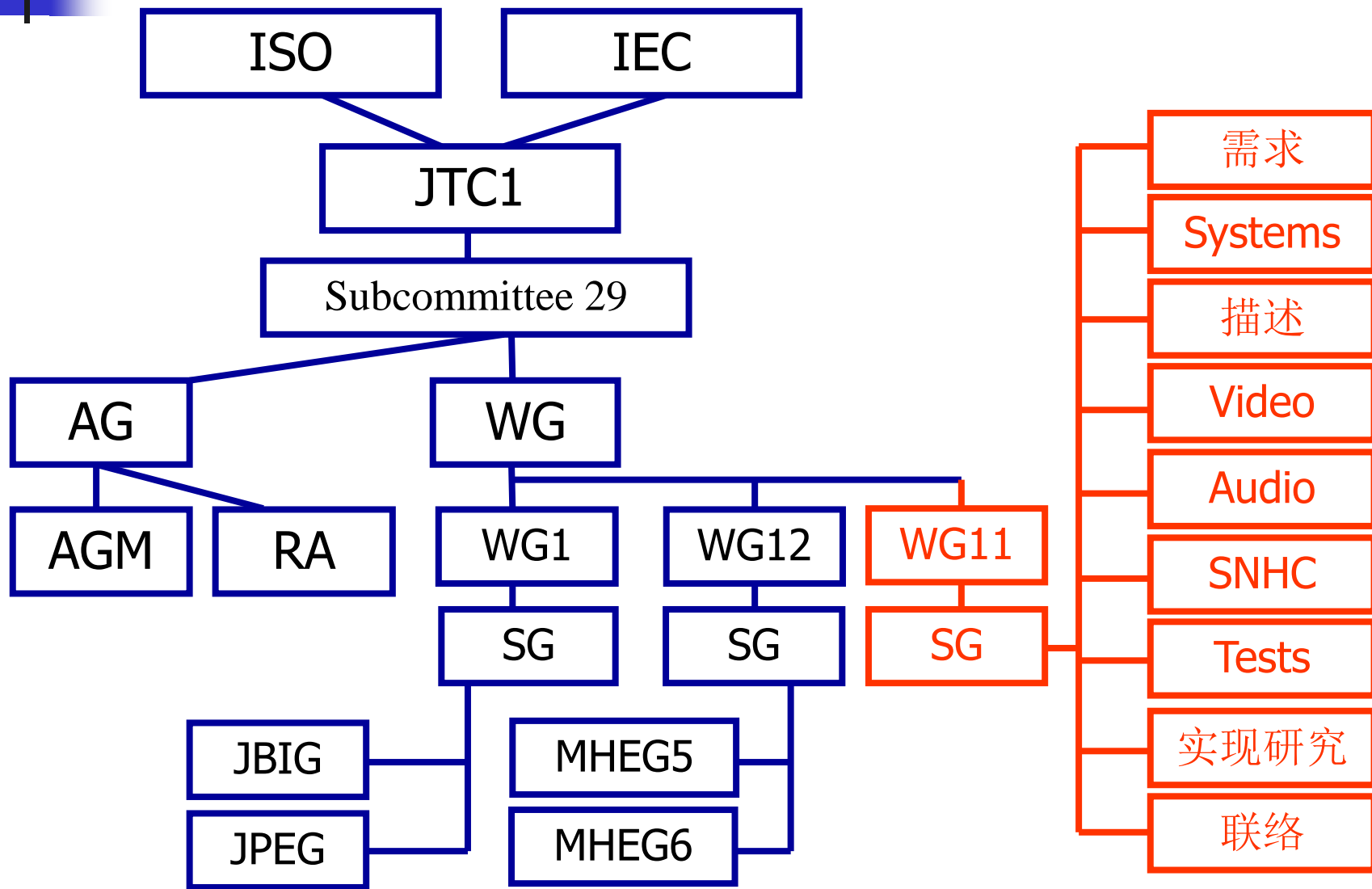
# 视频编码标准组织

---

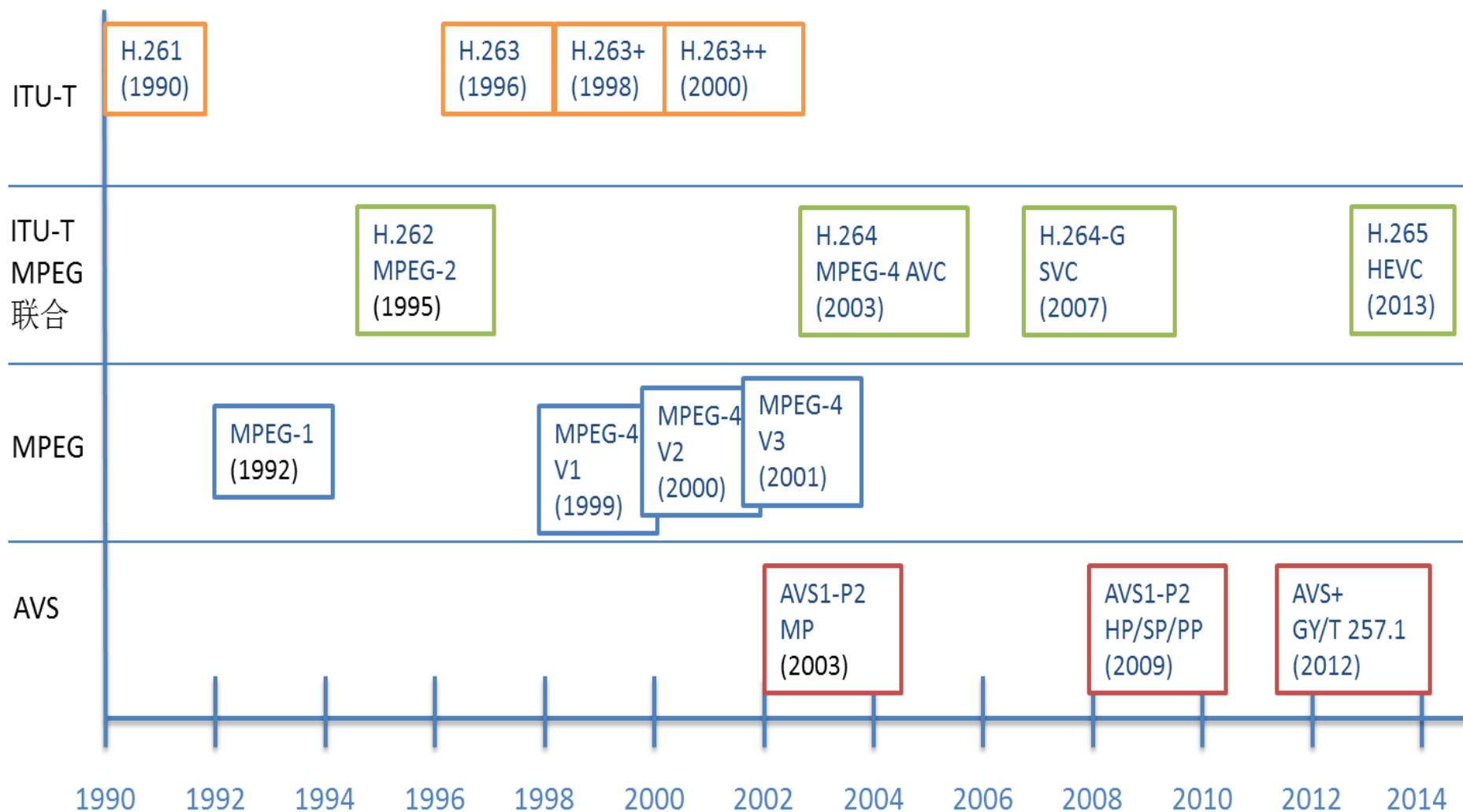
## ■ ISO/IEC

- **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

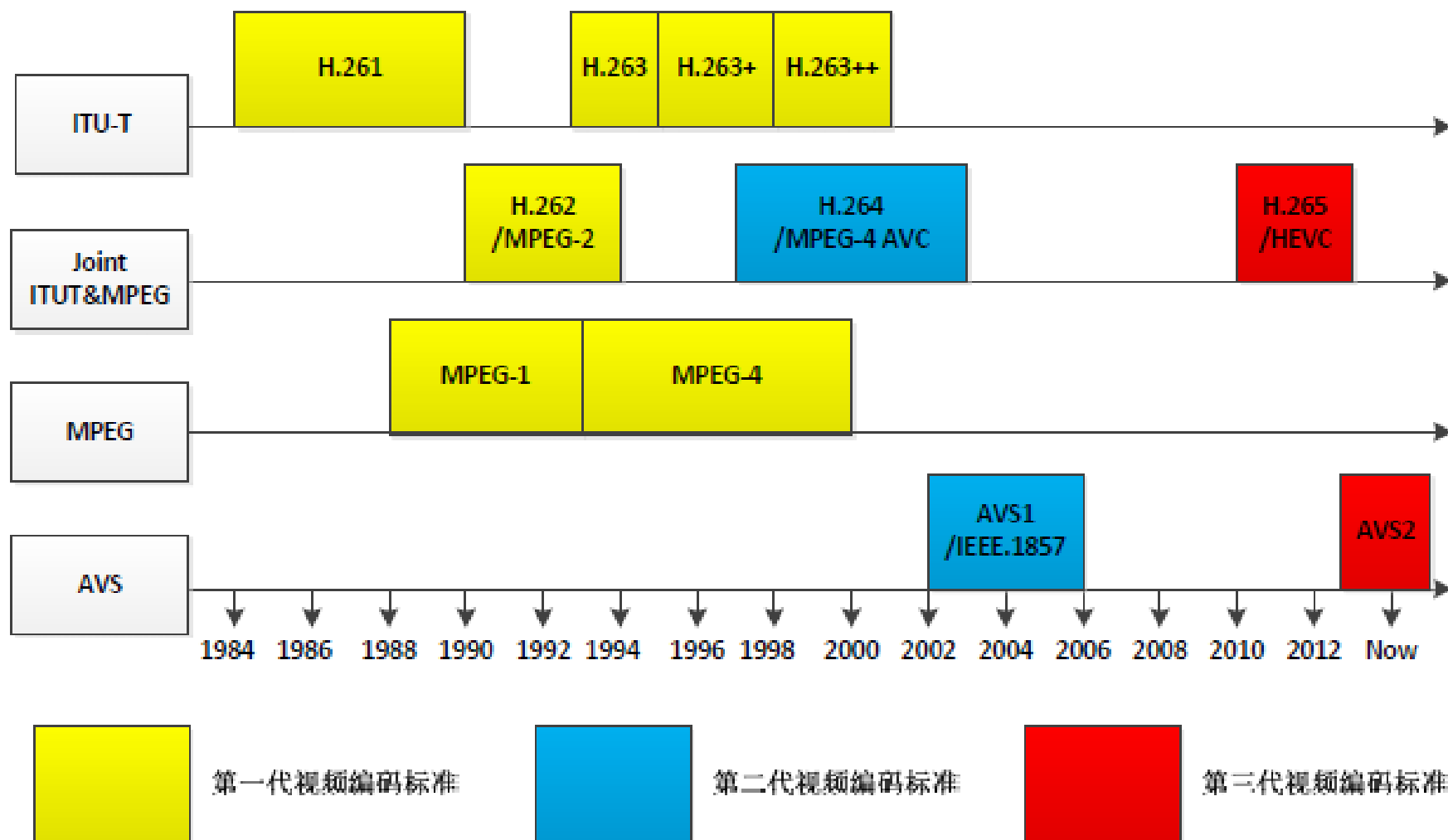


# 视频编码标准发展历程



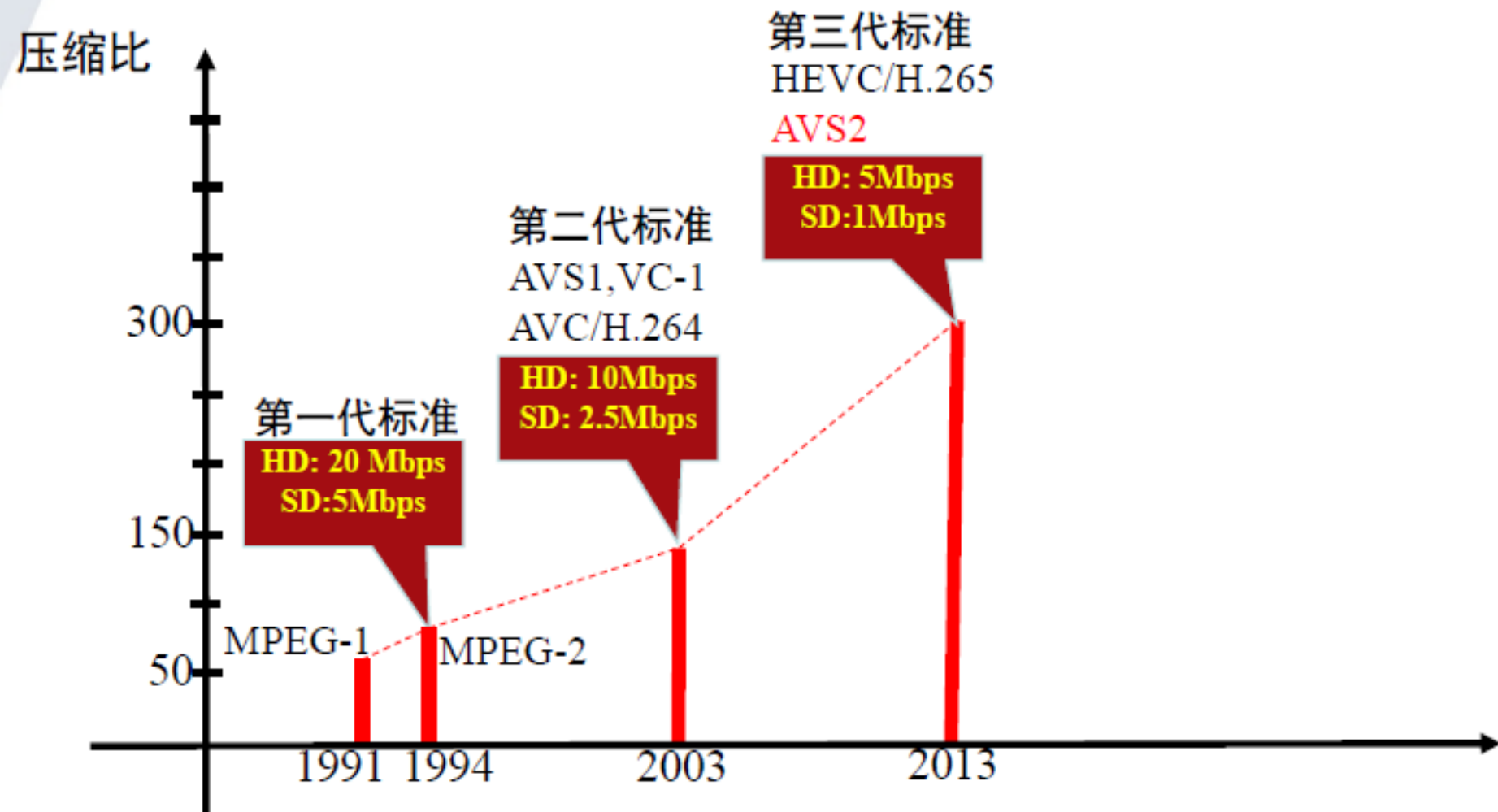


# 视频编码标准发展历程



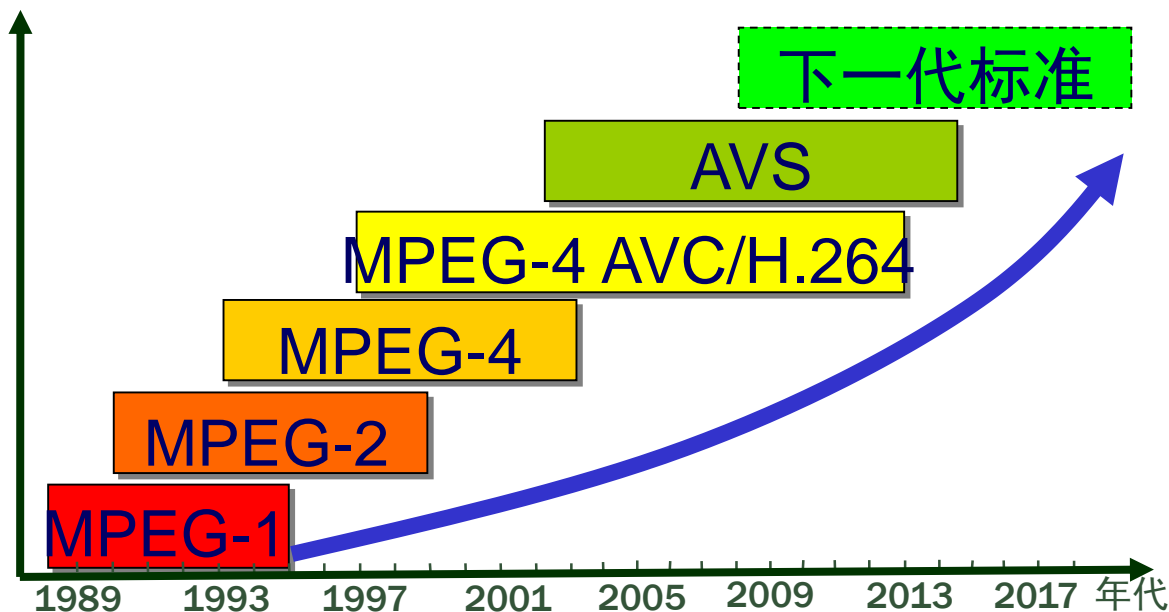
# 视频编码标准发展历程

## ➤ 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? Versatile Video Coding (VVC), deep learning





## 6.2.1 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  $16 \times 16$ 宏块运动补偿预测、 $8 \times 8$  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

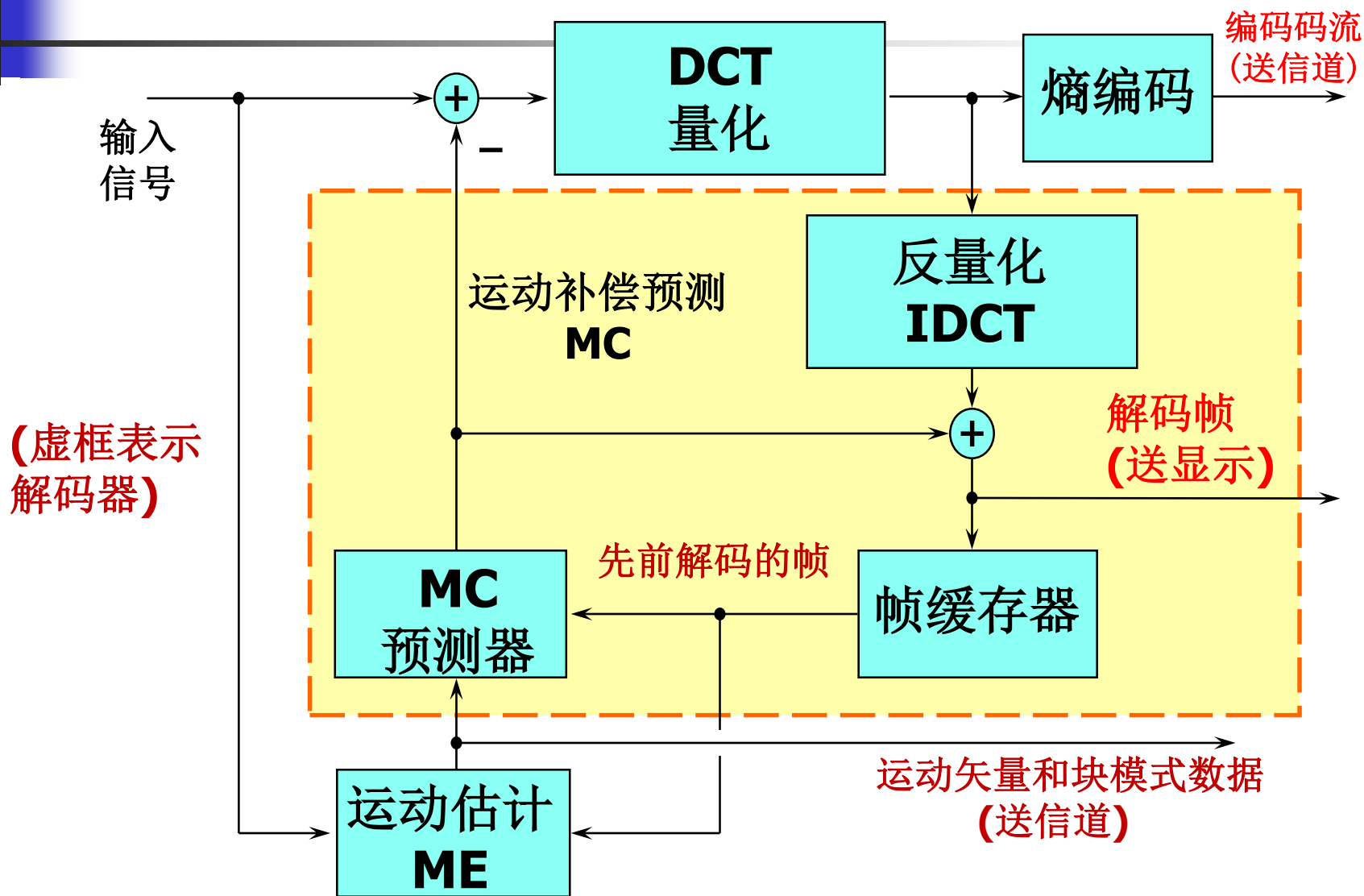
352 × 288

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



## 6.2.2 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**。



## 6.2.2 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。



## 6.2.2 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。



## 6.2.2 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

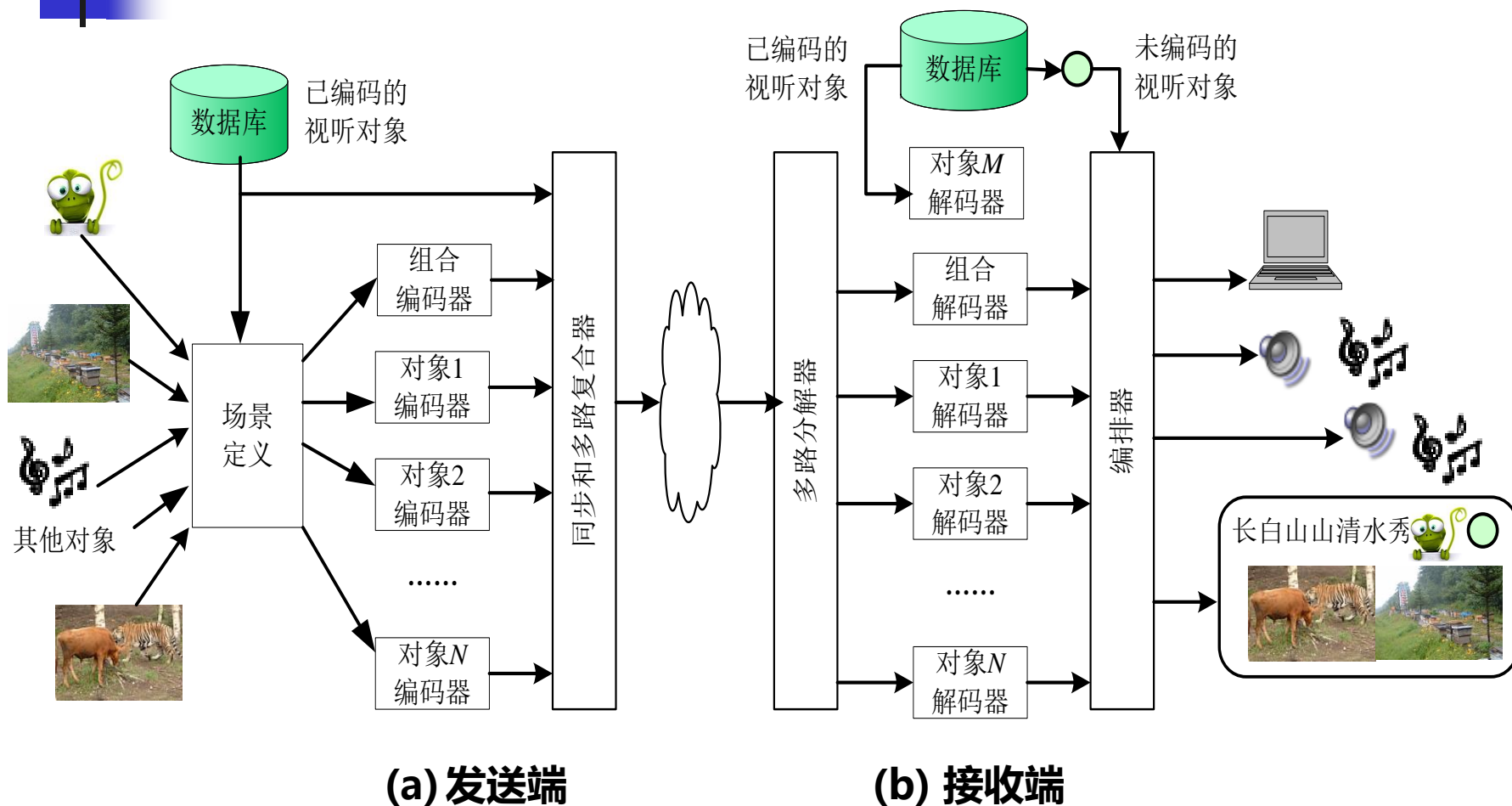


## 6.2.2 MPEG-x系列标准

---

- 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

## 6.2.2 MPEG-x系列标准



MPEG-4基于对象的系统模型



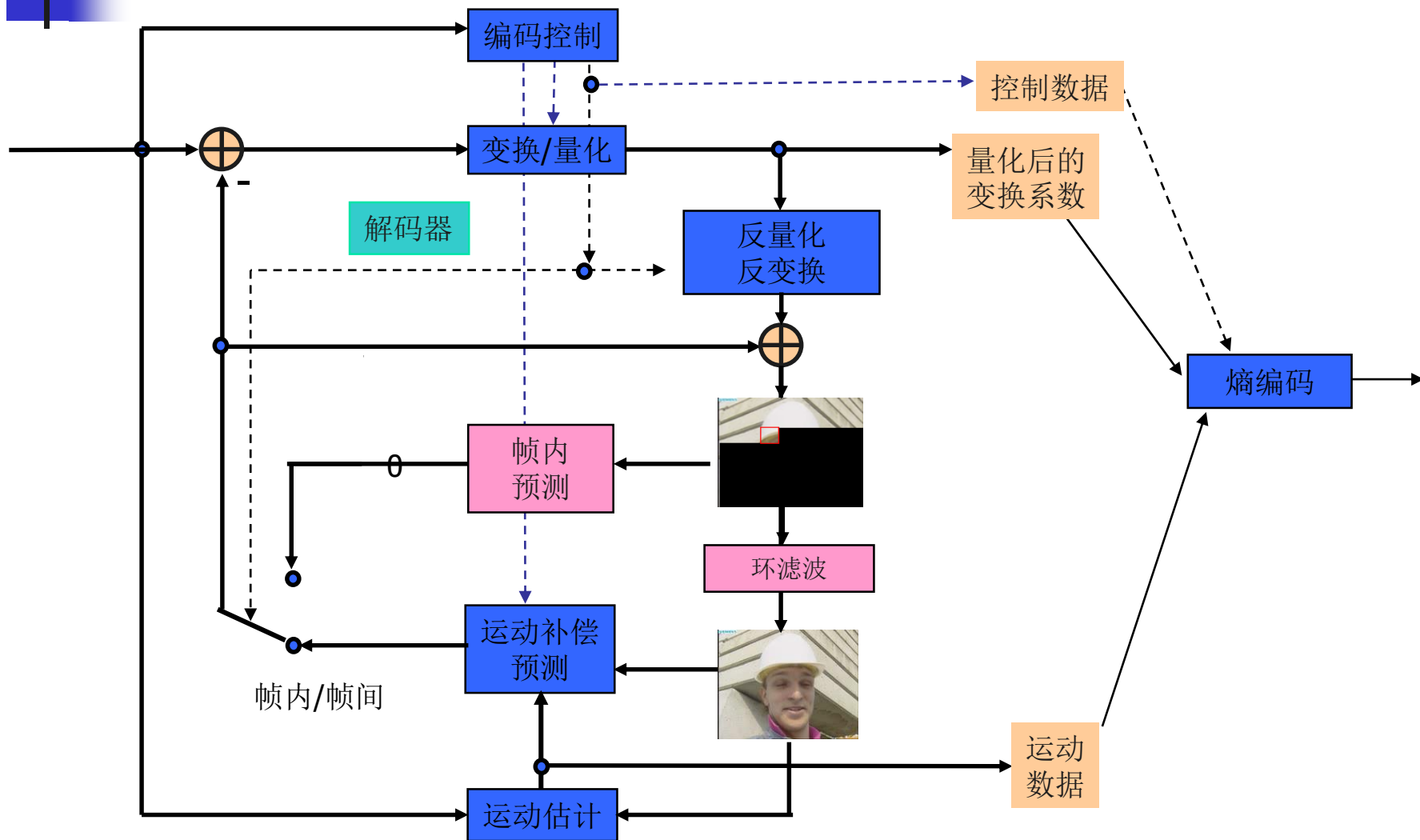
## 6.2.3 AVS和AVS+标准

---

- 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。**



## 6.2.3 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

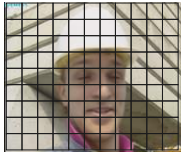
- AVS标准针对视频监控的伸展档**AVS-S2**又于2013年被国际电子电气工程师协会(**IEEE**)标准化委员会接受并颁布为**IEEE 1857**标准。2013年12月31日国家质检总局、国家标准委批准发布了《信息技术 先进音视频编码 第2部：视频》（修订）（国标代号GB/T 20090.2-2013）、《信息技术 先进音视频编码 第10部：移动语音和音频》（国标代号GB/T 20090.10-2013）为国家标准，标准将于2014年7月15日正式实施。
- AVS2是AVS+的下一代，即《信息技术 --先进音视频编码 第2部：视频》（修订）。
- 国标代号GB/T 20090.2-2013，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6章 数字图像与视频压缩编码标准

---

- 6.1 静止图像编码标准
- 6.2 数字视频编码标准概述
- **6.3 H.264/AVC视频编码标准**
- 6.4 H.265/HEVC视频编码标准
- 6.5 AVS/AVS+视频编码标准





# 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

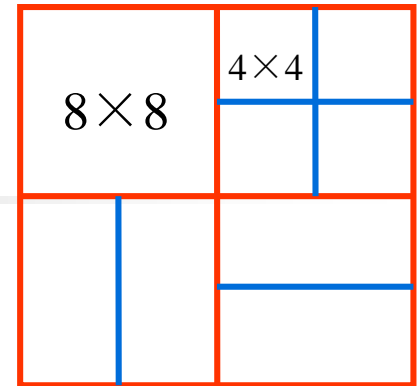
---

- **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)**



16×16

## H.264多模式、多参考帧运动补偿



- **动机：运动/固定目标的大小是可变的**
  - 许多小块也可能需要花费许多比特来编码
  - 不同块大小和形状的运动补偿
- **1/4-pixel 精度 (源自 MPEG-4 ASP, 但更复杂)**
  - 以往的视频编码标准都采用了整像素或半像素的运动估计,
  - 1/4-pixel 比 1/2-pixel 获得 1.5~2dB 增益, 约节省20%的比特率
  - 1/2-pixel 精度采用 6 抽头滤波器得到
- **7 种可变块大小：H.264把每个 16×16 宏块再细分, 每个子块分别进行预测, 提高了预测精度：**
  - 保持完整的宏块,
  - 水平 (垂直) 划分为两个子块, 大小为 16×8 (8x16)
  - 划分成 4 个子块, 大小为 8×8
  - 上述的 4 个 8×8 子块可以进一步划分成更小的 2 或 4 个块 (8×4、4×8、4×4) 。
  - 节省了15%以上的比特率
- **多参考帧预测：假设为5个参考帧预测, 相对于一个参考帧来说, 可节省 5% 到 10% 比特率。**

# 1/4 像素精度运动补偿

- 1/2 像素内插采用 6-TAP 滤波器

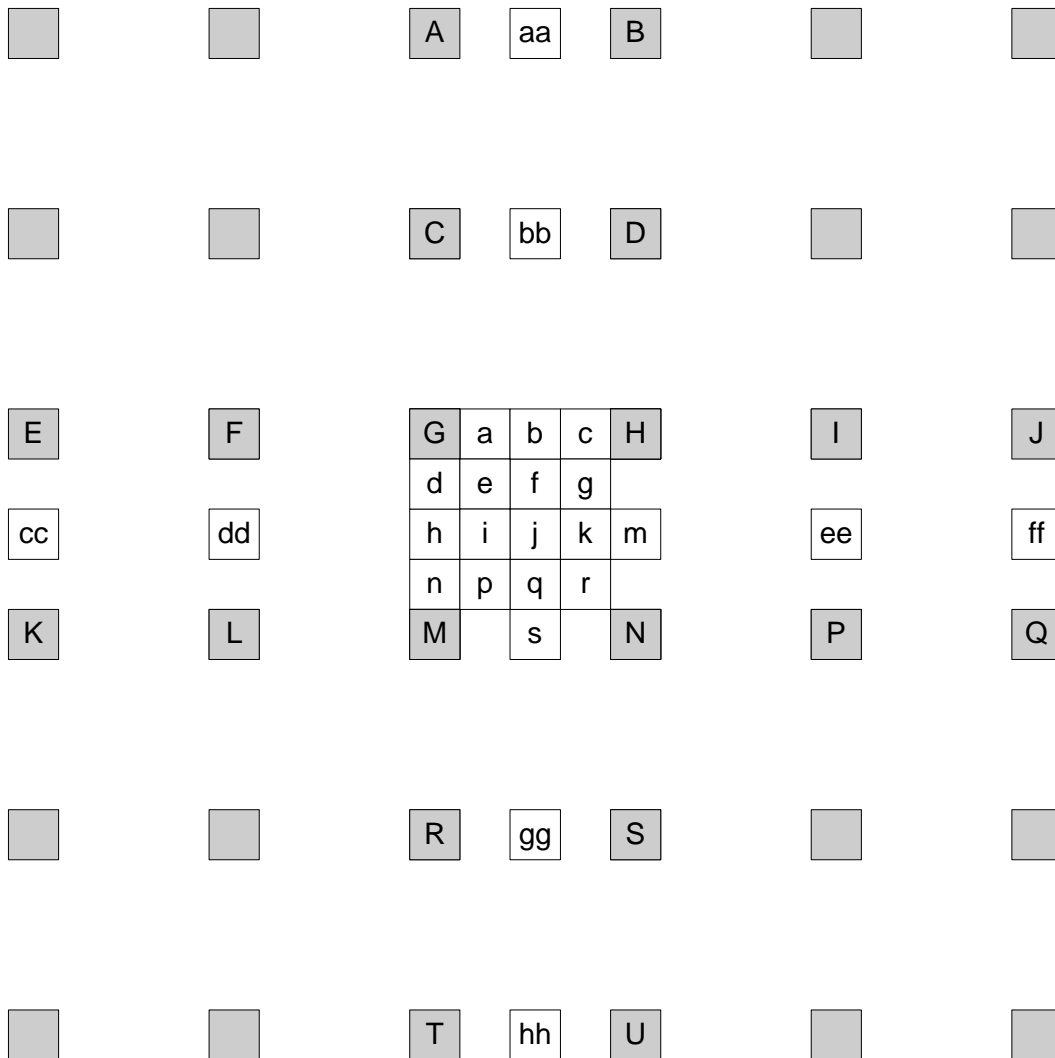
$$(1, -5, 20, 20, -5, 1)$$

例如 b 点是由 E、F、G、H、I、J 计算得出：

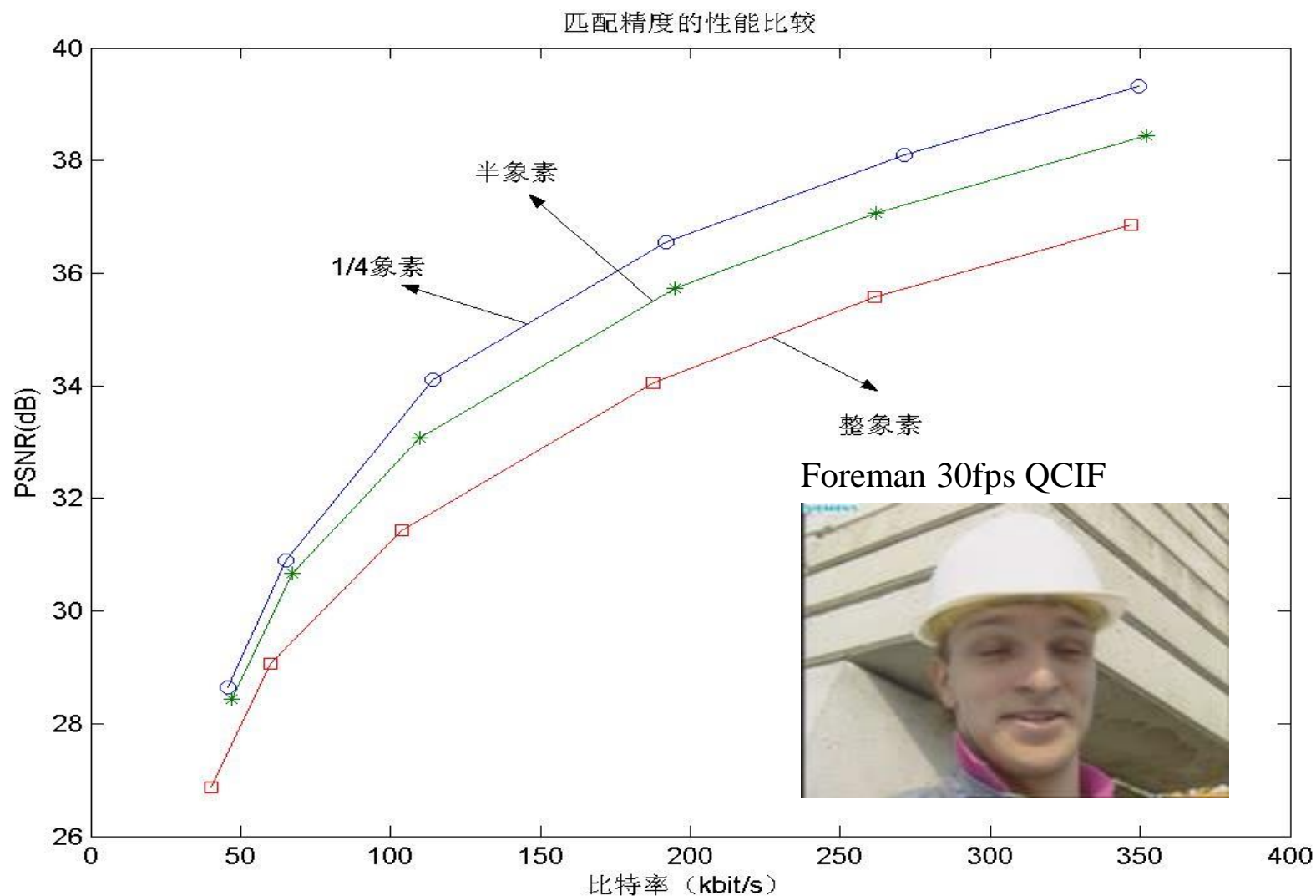
$$b = (E - 5F + 20G + 20H - 5I + J) / 32$$

- 1/4 像素则由 1/2 像素线性内插得到，例如：

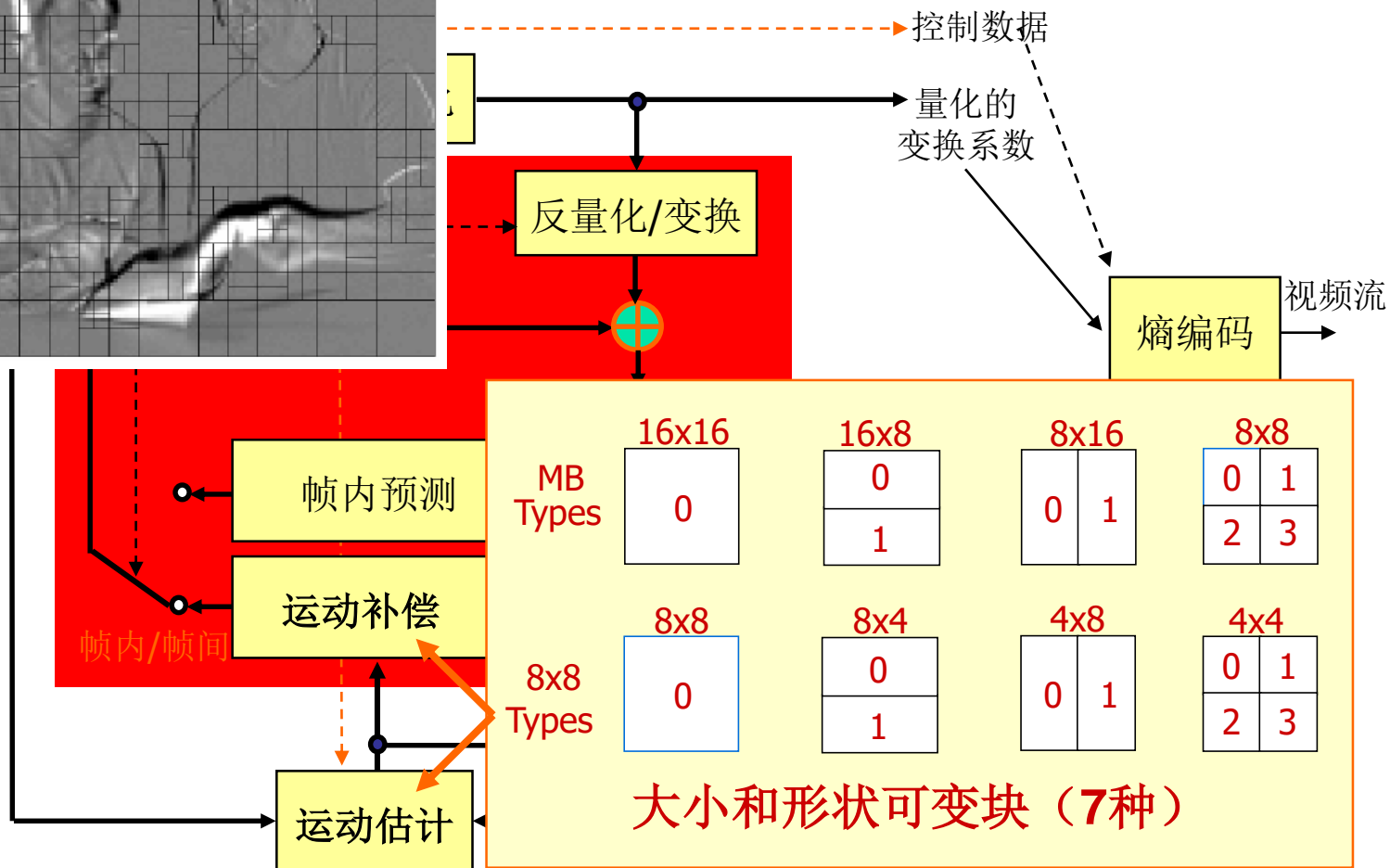
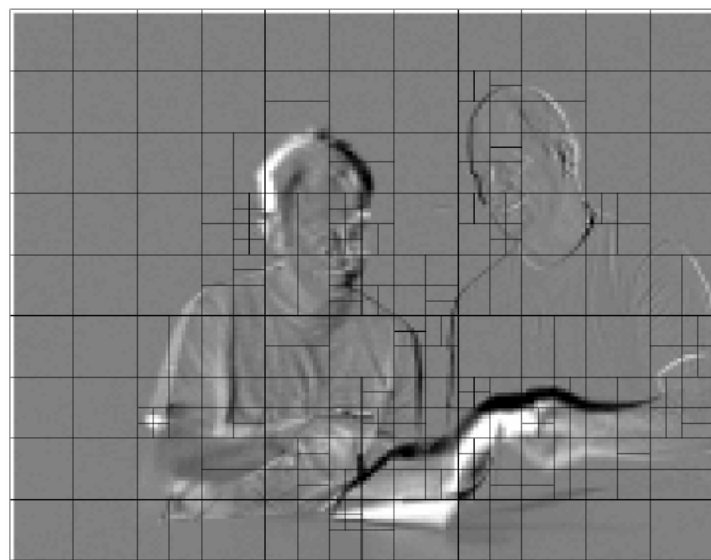
$$a = (G + b) / 2$$



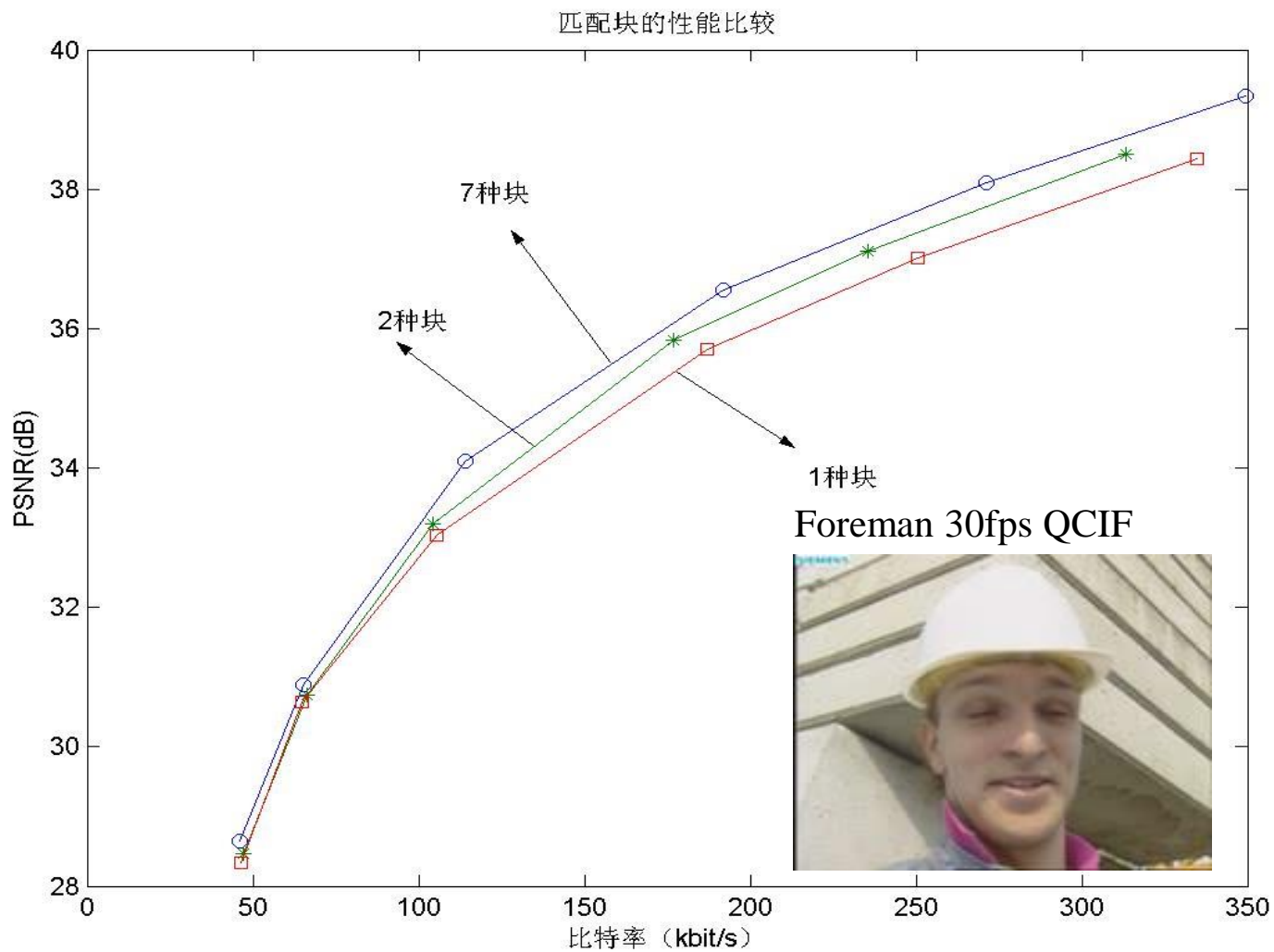
# 1/4像素精度运动补偿的性能比较



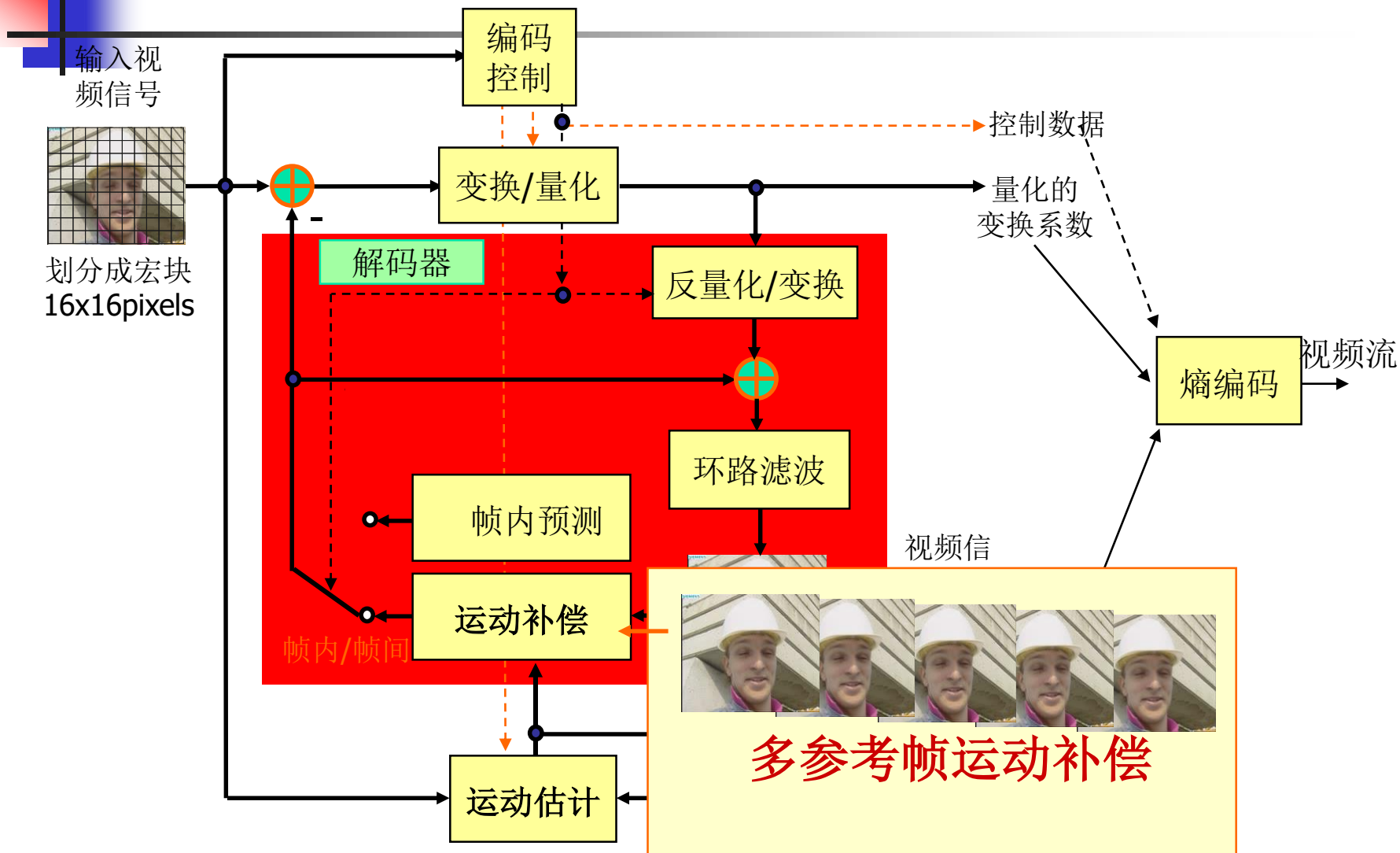
# H.264 可变大小块运动补偿



# 不同匹配块数量的性能比较



# H.264 多参考帧运动补偿



# H.264 多参考帧运动补偿

## ■ 多参考帧的性能比较



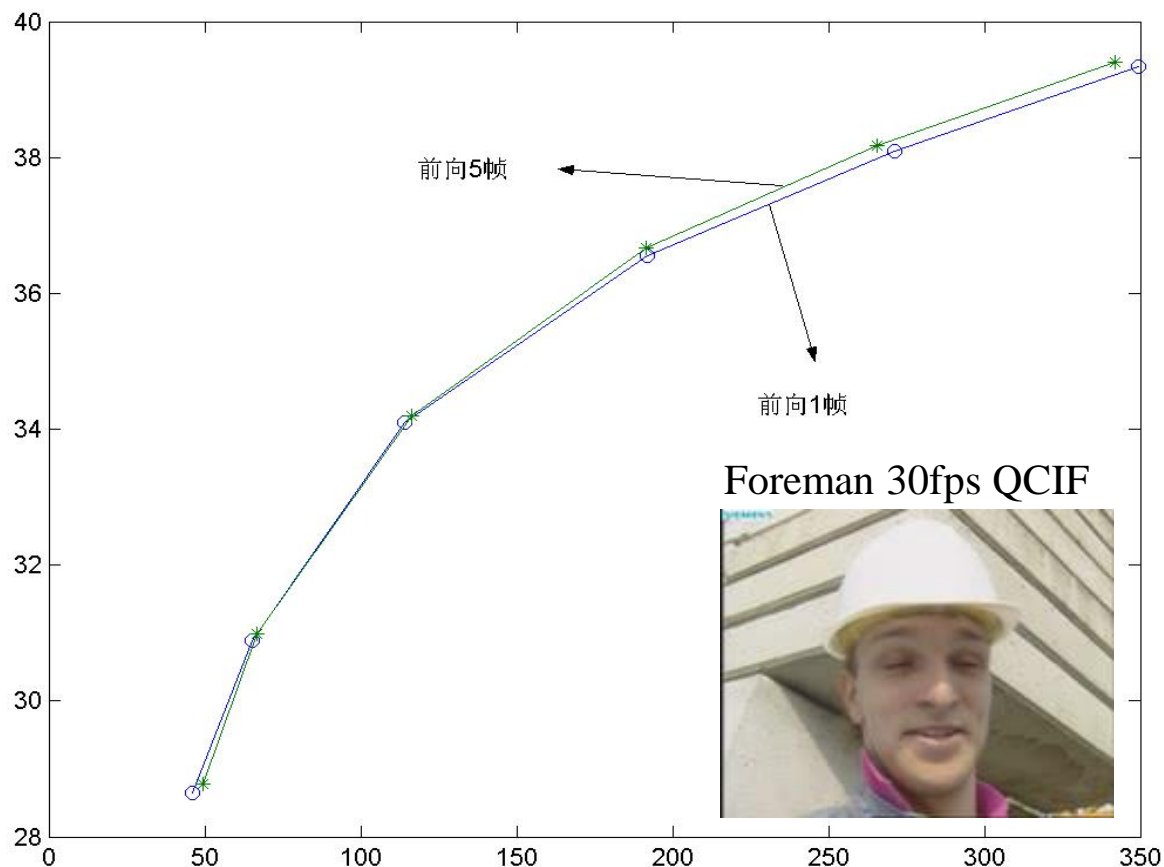
1个参考帧(31.65dB)



5个参考帧(33.19dB)

# H.264 多参考帧运动补偿

- 多参考帧的性能提高不明显，这与图像序列有很大关系。



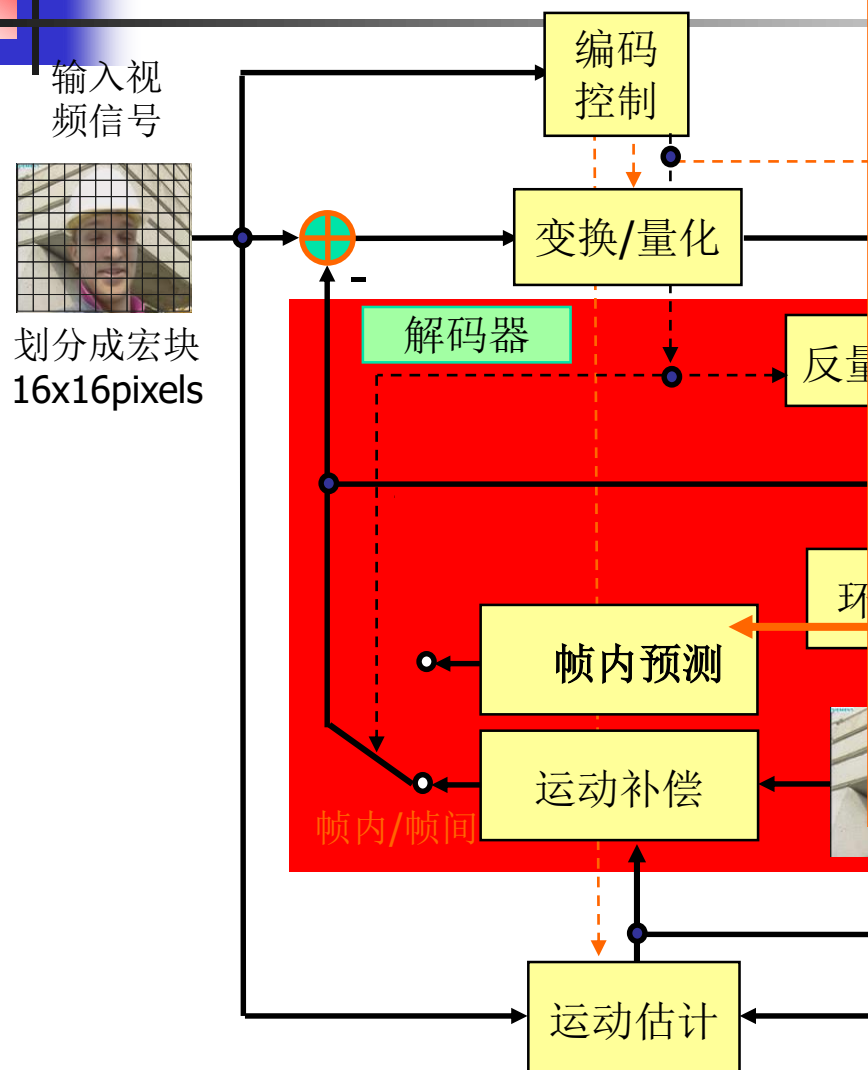




# 多模式帧内预测 (I 帧空域预测)

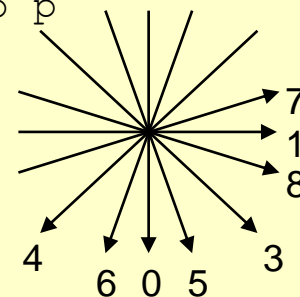
- 宏块或块基于先前编码的块进行预测：
  - 先前编码的块位于当前块的上面和/或左面的块
  - **亮度块**两类帧内预测方式：
    - 9 种  **$4 \times 4$  帧内预测模式**，用于纹理区域。
    - 4 种  **$16 \times 16$  帧内预测模式**，用于平滑区域。
      - Mode 0(垂直)
      - Mode 1(水平)
      - Mode 2(DC)
      - Mode 3(面)
  - **色度样值**只有 4 种  **$8 \times 8$  帧内预测模式**，类似于亮度样值的  $16 \times 16$  帧内预测模式
    - Mode 0(Vertical)
    - Mode 1(Horizontal)
    - Mode 2(DC)
    - Mode 3(Plane)
- 编码参数规定了哪些相邻块用于预测，以及如何预测？

# H.264中的 4×4 帧内预测



## ■ 方向性空间预测模式 (亮度 9 种 4×4 预测模式)

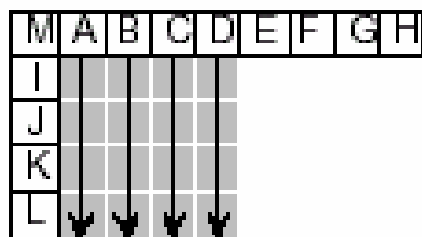
Q	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				
M								
N								
O								
P								



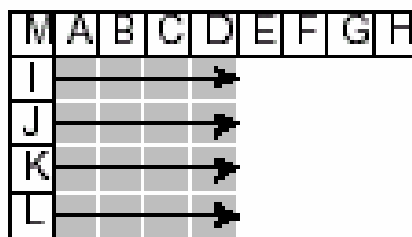
- 像素 (a-p) 由周围解码的像素 (A-Q) 进行预测。例如模式3：右下对角线预测  
通过下式预测：  
$$(A + 2Q + I + 2) \gg 2$$

# H.264中的 4×4 帧内预测

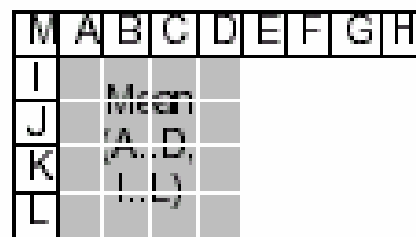
0 (vertical)



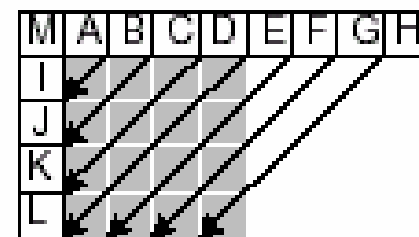
1 (horizontal)



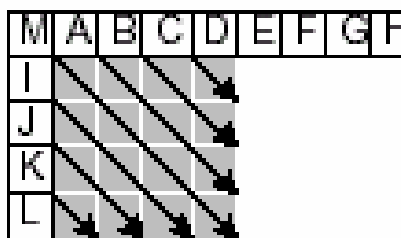
2 (DC)



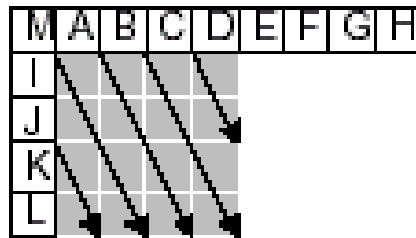
3 (diagonal down-left)



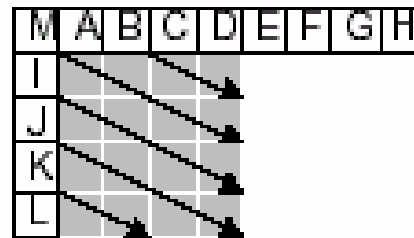
4 (diagonal down-right)



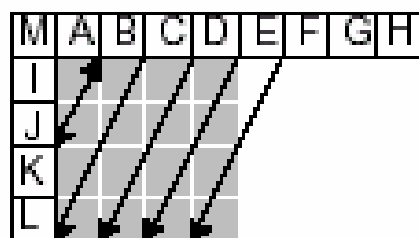
5 (vertical-right)



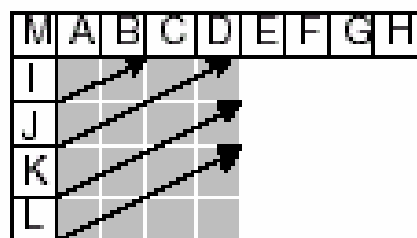
6 (horizontal-down)



7 (vertical-left)



8 (horizontal-up)



- Mode 0: Extrapolate from upper samples A-D
- Mode 1: Extrapolate from left samples I-L
- Mode 2: Predict by the mean of A-D and I-L
- Mode 3: Diagonal down-left 45 degrees
- Mode 4: Diagonal down-right 45 degrees
- Mode 5: Vertical-left 22.5 degrees
- Mode 6: Horizontal-down 22.5 degrees
- Mode 7: Vertical-right 22.5 degrees
- Mode 8: Horizontal-up 22.5 degrees



# 亮度分量 $4\times 4$ 块预测模式

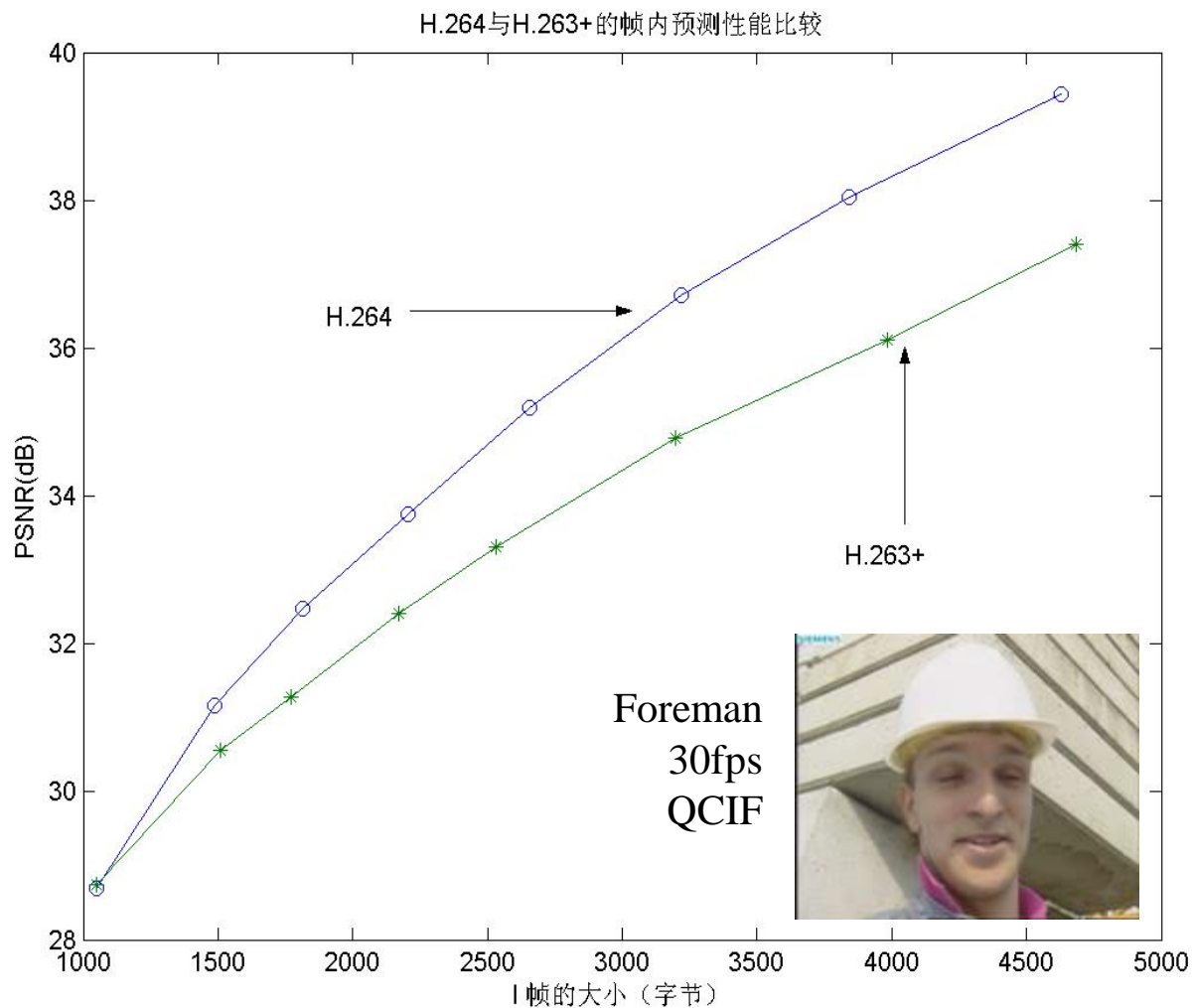
模式	描述
模式0(垂直)	由A、B、C、D垂直推出相应像素值
模式1(水平)	由I、J、K、L水平推出相应像素值
模式2(DC)	由A~D及I~L平均值推出所有像素值
模式3(下左对角线)	由 $225^{\circ}$ 方向像素值内插值得出相应像素值
模式4(下右对角线)	由 $315^{\circ}$ 方向像素值内插得出相应像素值
模式5(右垂直)	由 $296.6^{\circ}$ 方向像素值内插得出相应像素值
模式6(下水平)	由 $333.4^{\circ}$ 方向像素值内插得出相应像素值
模式7(左垂直)	由 $243.4^{\circ}$ 方向像素值内插得出相应像素值
模式8(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



## 色度分量/亮度分量 $16 \times 16$ 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

# 帧内预测性能分析





# H.264中的 $4 \times 4$ 整数变换

## ■ 动机:

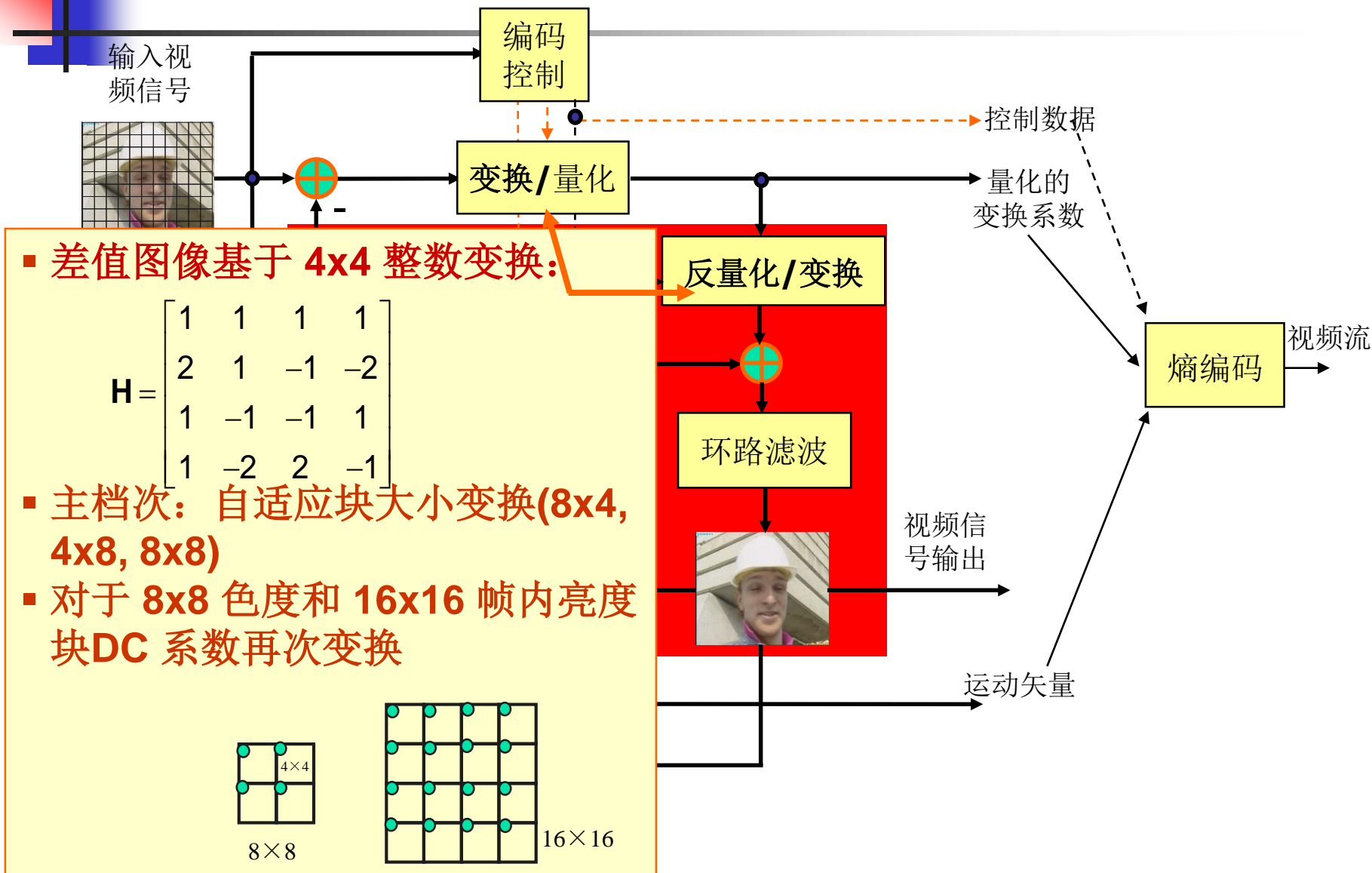
- DCT 需要实数运算，反变换中引起了精度的损失。
- 更小的图像，更好的运动补偿，意味着更小的空间相关性，不需要  $8 \times 8$  变换
- $4 \times 4$  的块可以有效的减弱重建图像的块效应

## ■ H.264 使用了非常简单的整数 $4 \times 4$ 变换:

- 对  $4 \times 4$  DCT 非常精巧的逼近
- 变换矩阵只包含  $\pm 1$  和  $\pm 2$ 
  - 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





# 2D-DCT 回顾

■ 2D-DCT的基为:

$$a_{k,l}(m,n) = \sqrt{\frac{2}{N}} \cos \left[ \frac{\pi(2m+1)k}{2N} \right] \sqrt{\frac{2}{N}} \cos \left[ \frac{\pi(2n+1)l}{2N} \right]$$
$$0 \leq k, l \leq N-1 \quad 0 \leq m, n \leq N-1$$

正变换可表示为:

$$y(k,l) = d(k,l) \frac{2}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) \cos \frac{\pi(2m+1)k}{2N} \cos \frac{\pi(2n+1)l}{2N}$$
$$d(0,0) = \frac{1}{2}, \quad d(k,l) = 1 \text{ 对其它}$$

2D-DCT 矩阵表示:

正变换:  $Y = AXA^T$

反变换:  $X = A^T Y A$

A 为离散余弦变换矩阵

$A^T$  为 A 的转置矩阵

# 2D-DCT 回顾

## 2D-DCT 变换矩阵 A

$$A = \sqrt{\frac{2}{N}} \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \cdots & \sqrt{\frac{1}{2}} \\ \cos \frac{\pi}{2N} & \cos \frac{3\pi}{2N} & \cdots & \cos \frac{(2N-1)\pi}{2N} \\ \vdots & \vdots & \vdots & \vdots \\ \cos \frac{(N-1)\pi}{2N} & \cos \frac{3(N-1)\pi}{2N} & \cdots & \cos \frac{(N-1)(2N-1)\pi}{2N} \end{bmatrix}_{N \times N}$$

当N=2时，变换矩阵A为：

$$A = \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ \cos \frac{\pi}{4} & \cos \frac{3\pi}{4} \end{bmatrix}$$

当N=4时，变换矩阵A为：

$$A = \sqrt{\frac{1}{2}} \begin{bmatrix} \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} & \sqrt{\frac{1}{2}} \\ \cos \frac{\pi}{8} & \cos \frac{3\pi}{8} & \cos \frac{5\pi}{8} & \cos \frac{7\pi}{8} \\ \cos \frac{2\pi}{8} & \cos \frac{6\pi}{8} & \cos \frac{10\pi}{8} & \cos \frac{14\pi}{8} \\ \cos \frac{3\pi}{8} & \cos \frac{9\pi}{8} & \cos \frac{15\pi}{8} & \cos \frac{21\pi}{8} \end{bmatrix}$$



# H.264中的 $4 \times 4$ 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 $4 \times 4$ 整数变换

- $4 \times 4$  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  $\pm 1$  与  $\pm 2$ ，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 $4 \times 4$ 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整数变换与量化的性能

- 我们用变换编码增益来衡量整数变换的性能，而**编码增益通常定义为，信号经过变换以后再量化得到的信噪比，与直接对原始信号量化得到的信噪比，前者比后者的提高。**
- 如果输入信号为一个平稳的高斯-马尔科夫随机过程，相关系数是0.9，DCT、H.26L 以及 H.264 的变换编码增益分别是5.39dB、5.39dB、5.38dB。因此，H.264的整数变换约有0.01dB 的损失。





# 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

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

$$Z_{i,j} = |W_{i,j}MF + f| \gg q$$

$$\text{sgn}(Z_{i,j}) = \text{sgn}(W_{i,j})$$

$$MF = PF \times 2^q / Q_{\text{step}}; \quad q = 15 + \text{floor}(QP/6)$$

$$f = 2^q / 3 \quad f = 2^q / 6$$

- 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 $Q_{\text{Step}}$ 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。

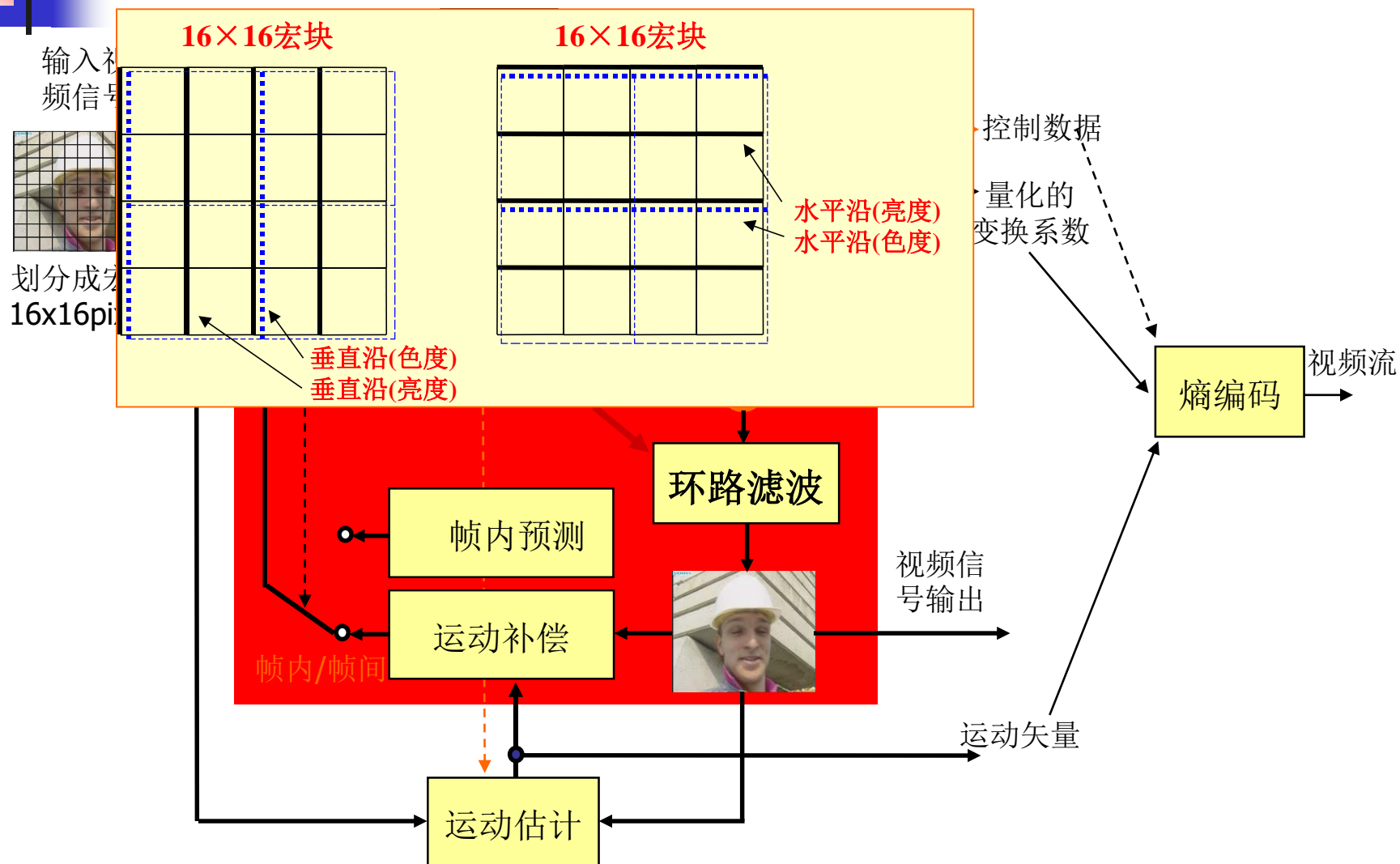
$Q$	0	1	2	3	4	5	6	7	8	9	10	11	
$Q_{\text{Step}}$	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	
$Q$	12		...	18	...	24	...	30	...	36	...	51	
$Q_{\text{Step}}$	2.5		...	5	...	10	...	20	...	40	...	224	



## H. 264去方块效应环路滤波器

- **动机：基于块匹配的 MC 和块变换将产生块效应**
  - **在低码率时，人眼易于观察到。**
- **以前的标准采用简单的滤波器去 “模糊” 块之间的边界。**
- **H.264 对每个边界自适应地选择应用 5 个去方块效应滤波器之一。**
  - **例如，若两个块具有相同的运动矢量，就需要较小的滤波。**
- **提高了客观质量，以及在相同的 PSNR 下，减少了约 7~9% 比特率。**

# H. 264去方块效应环路滤波器



## H. 264去方块效应环路滤波器



没有去块效应滤波



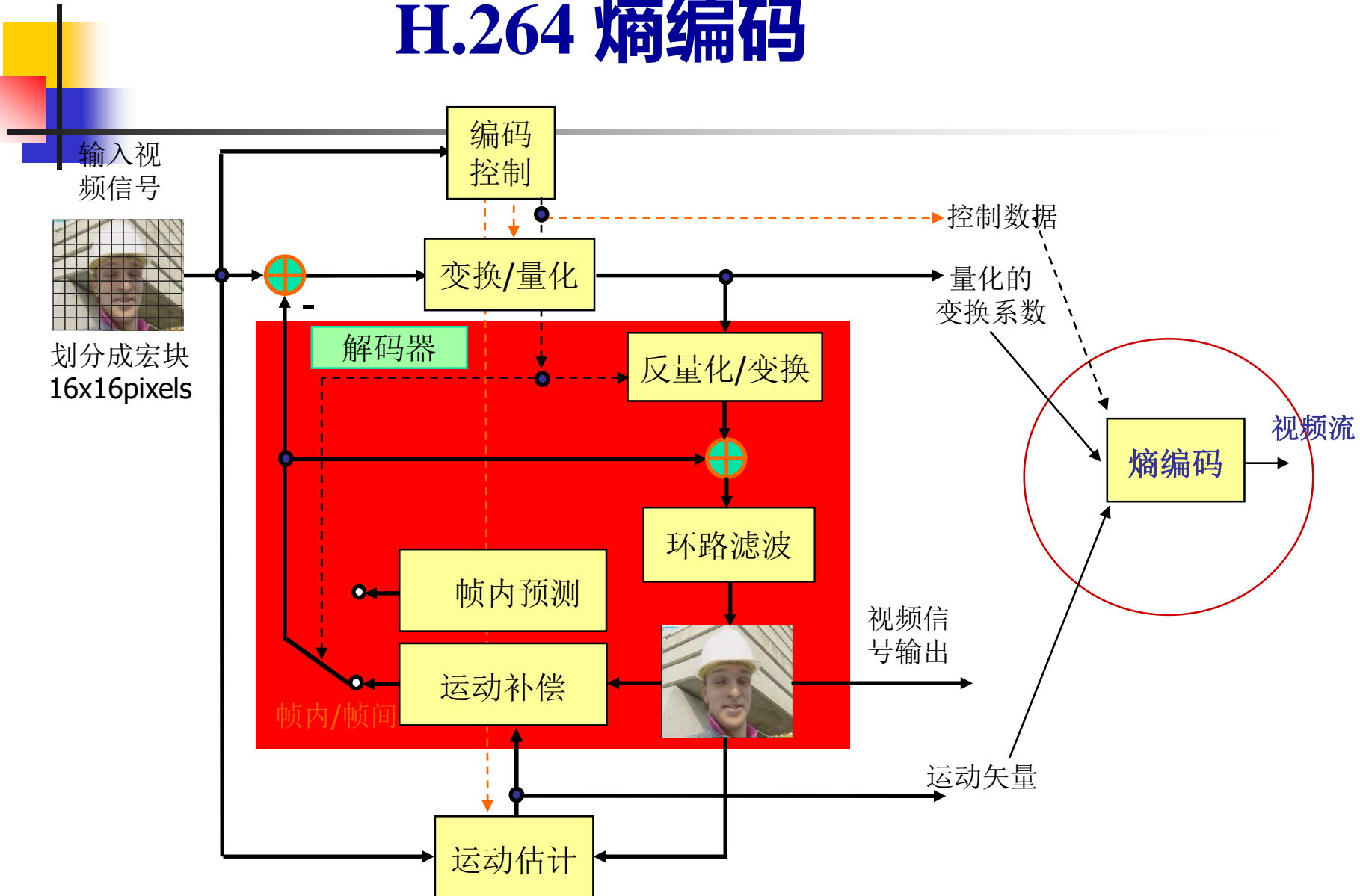
**H.264/AVC** 去块效应滤波



# 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 熵编码







# H.264 CAVLC 熵编码

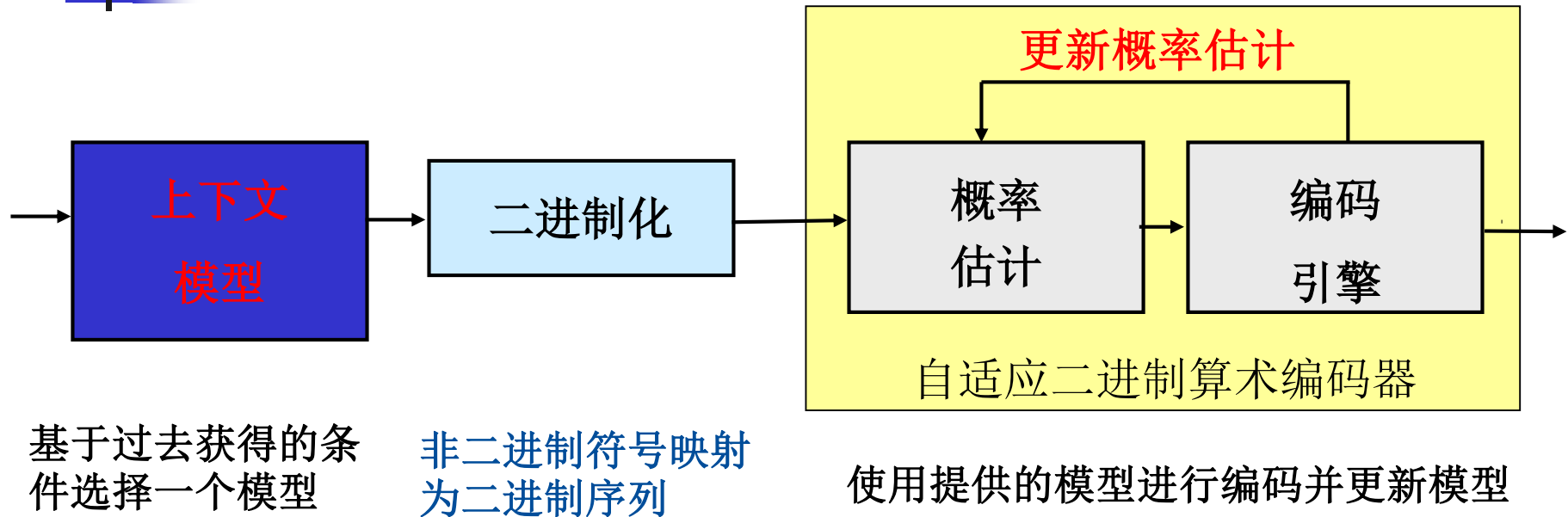
- 变换量化后的  $4 \times 4$  (或  $2 \times 2$ ) 块进行 zig-zag 扫描后, 进行 CAVLC 编码。量化后的  $4 \times 4$  块有一些特点:
  - ✧ 经预测、变换和量化后, 块包含许多 0, 采用游程 (Run) 与幅度 (level) 的编码方式可以有效的表示变换系数, 这也是以往传统的编码方式用到的。
  - ✧ 经过量化后, 许多 Level 是  $\pm 1$ , 通过 zig-zag 扫描可以把尾 1 (trailing 1s) 集中起来, CAVLC 把这些 **Level 统计**出来。
  - ✧ 没有块结束标志, 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 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



Slice Group 0

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', a white text label 'Slice Group 1' in the center, and a purple rectangle at the bottom left labeled 'Slice Group 2'.

Slice Group 1

Slice Group 2



Slice Group 0

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'.

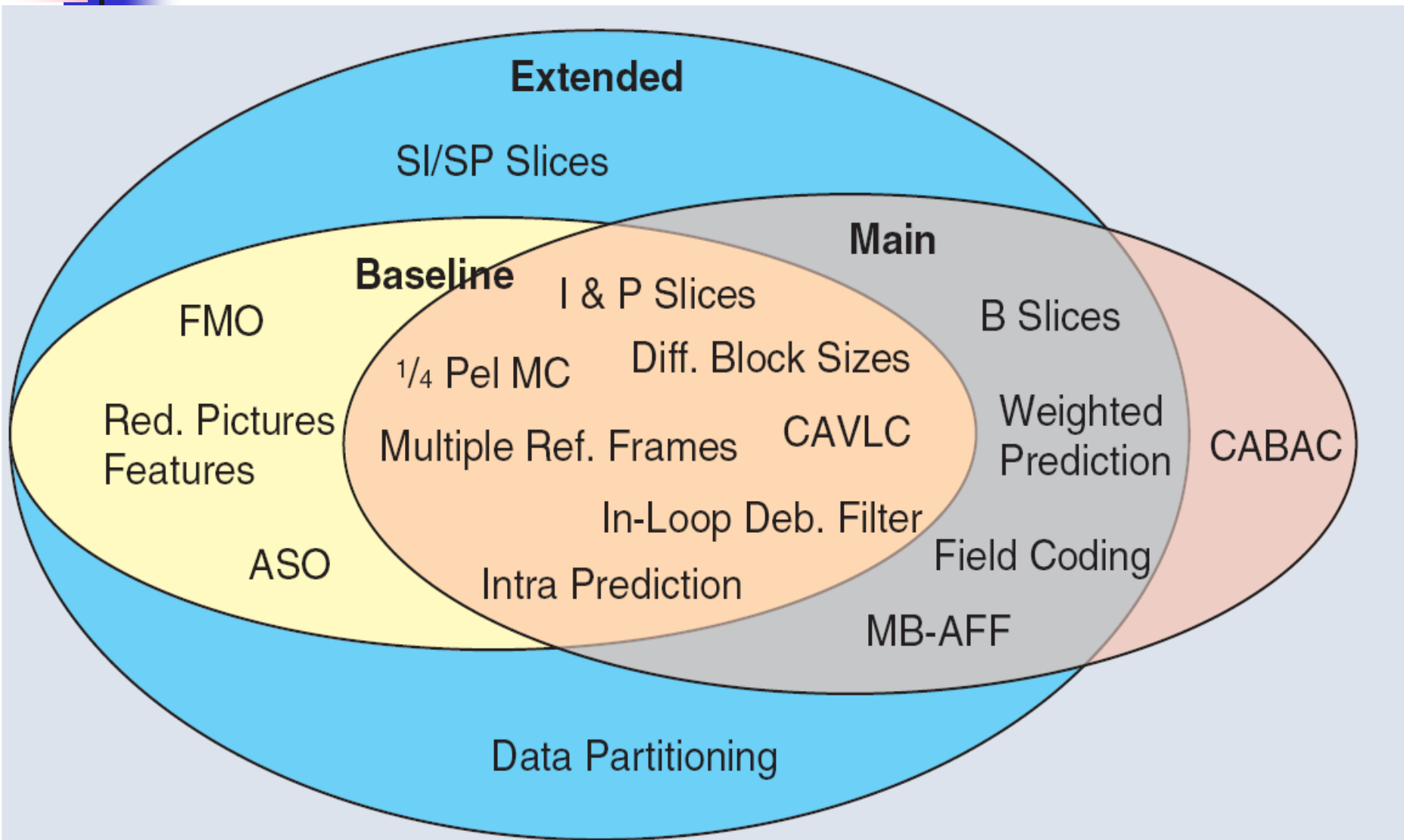
Slice Group 1



# H.264/AVC的档次与级

- H.264 可进行不同的性能配置，基于 “profiles” & “levels”：
  - 档次 (profile)：编码算法或工具的集合。
  - 级 (level)：对某些关键参数的限制（例如分辨率、码率等）
- H.264 分为 4个档次：Baseline、Main、Extended 和 High
  - 基本 (Baseline)：最小实现，逐行扫描、I-frame、P-frame 和 CAVLC，但没有 CABAC、B-frame 和 SP-Slices，从视频会议、无线视频通信到数字影院，大多数应用都是良好的。
  - 主 (Main)：增加了隔行扫描、B-Slices 和 CABAC等，广播应用
  - 扩展 (Extended)：SP/SI帧，帧数据分割，流媒体应用
  - 高 (High)：8\*8 DCT 变换
- H.264/AVC 有 15 级：
  - 分辨率、容量、比特率、缓存器、参考帧数、解码处理速度等
  - 图像尺寸：QCIF ~ 数字影院 D-Cinema
  - 匹配国际上通行的设备和发送格式。

# H.264/AVC的档次







# H.264 Baseline Profile (L2)

---

- I 和 P slice (没有B slice)
- 熵编码 CAVLC (Context-based Adaptive VLC, 基于上下文的自适应变字长码)
- 去块效应环路滤波器
- 逐行扫描
- 1/4 像素运动估计和补偿
- 增强误码适应性
  - 支持 ASO (Arbitrary slice ordering, ASO)
  - 支持 FMO (Flexible macroblock ordering, FMO)
  - 支持冗余片 (Redundant slices)
- 应用: IP 视频、视频会议等



# H.264 Main Profile

- 支持所有的 Baseline Profile 算法, 除了**不支持 Baseline Profile 中增强的误码适应性**
  - 不包括 Arbitrary slice order (ASO)
  - 不支持 Flexible macroblock ordering (FMO)
  - 不支持冗余片 Redundant slices
- I、P、B slice types
- CABAC 和 CAVLC 熵编码
- 隔行扫描 和逐行扫描
- MB 级帧/场切换
- B 和 P 帧加权预测
- $\frac{1}{4}$  像素运动估计和补偿
- 应用: 电视广播



# H.264 Extended Profile

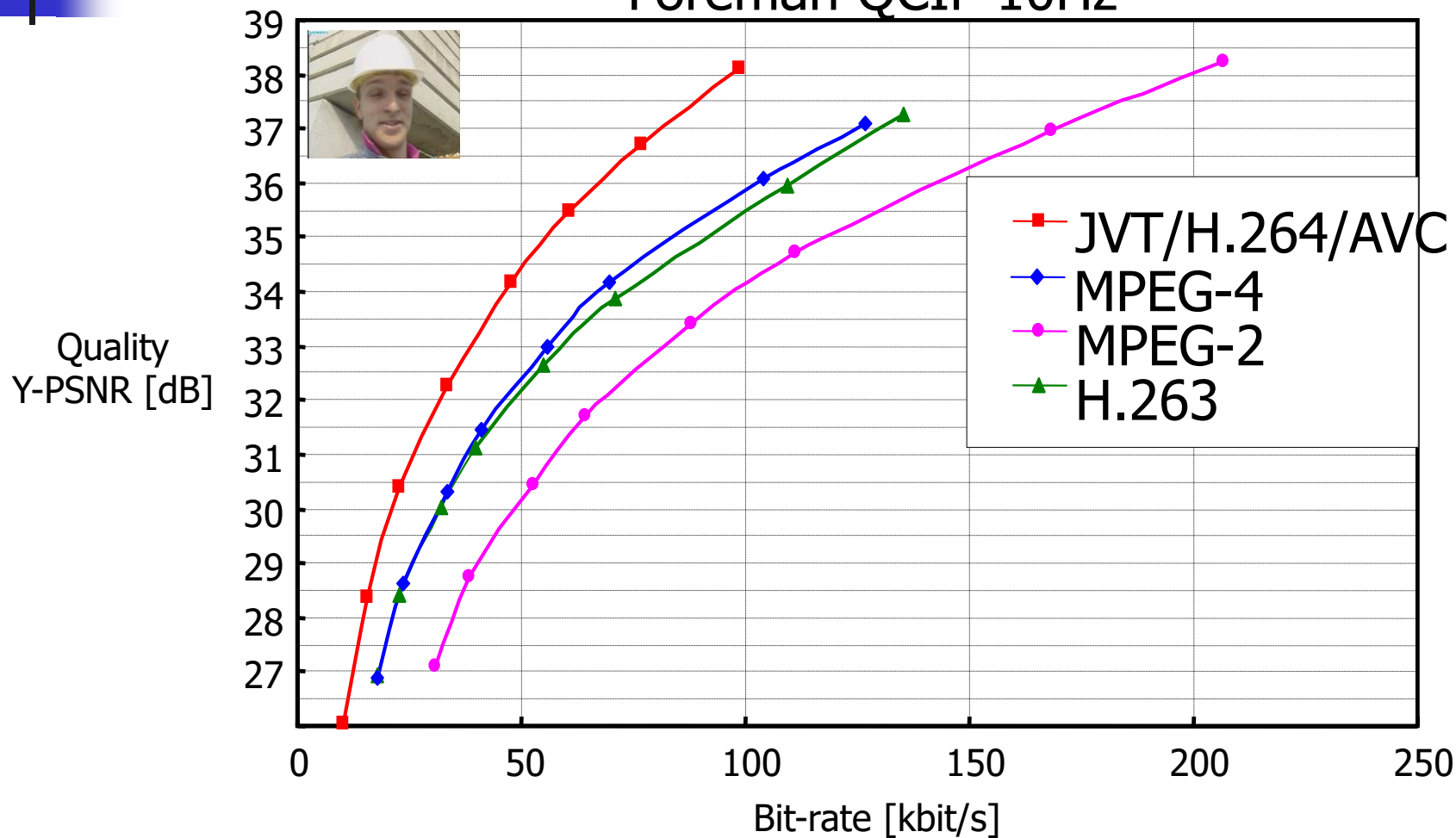
---

## ■ Extended Profile

- 所有的 Baseline profile 算法或工具
- B slice
- 加权预测
- SP/SI 切换帧
- 更多的误码可靠: 数据分割 (Data partitioning)

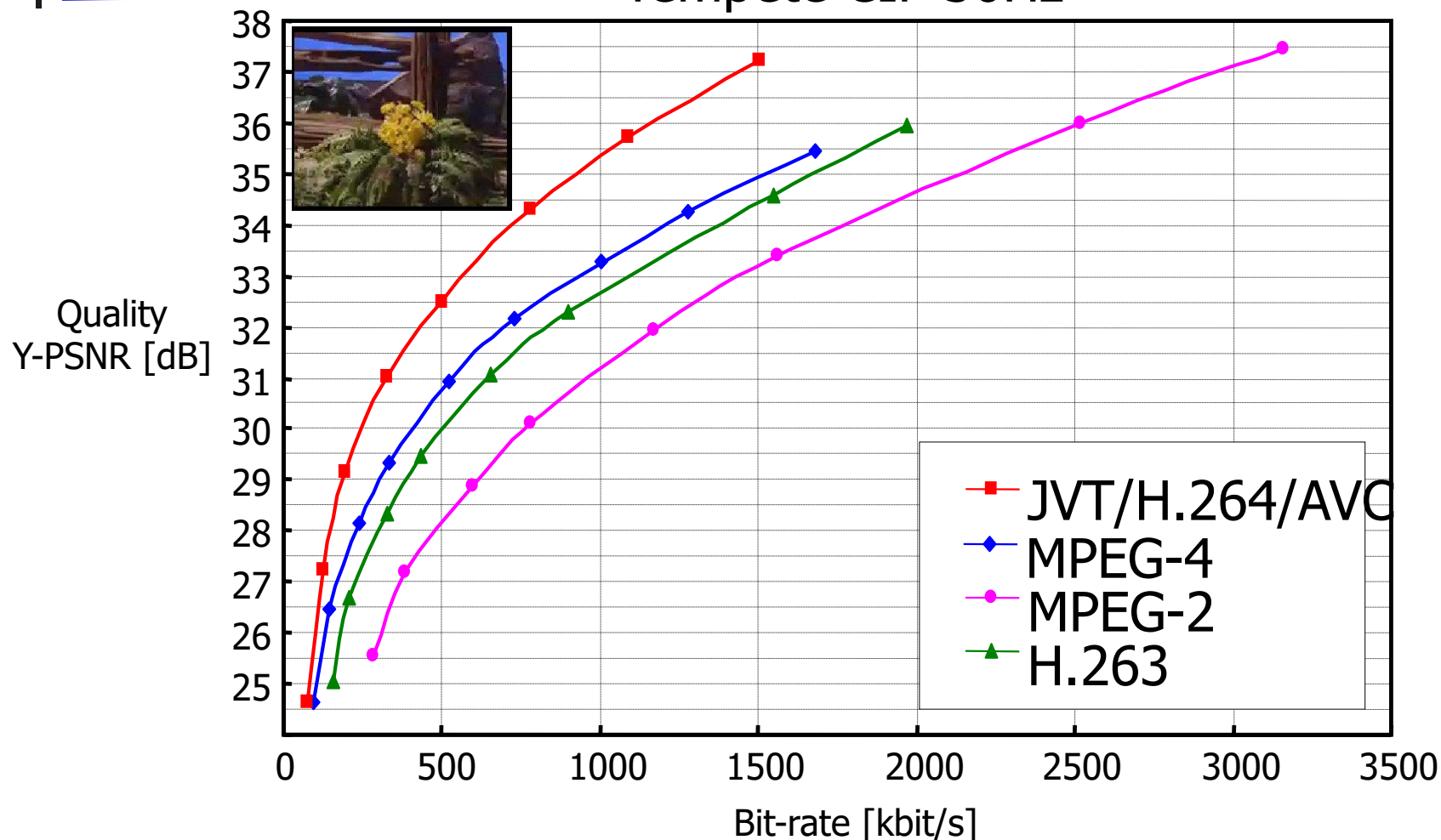
# H.264 与 MPEG-2/H.263/MPEG-4 的性能比较

Foreman QCIF 10Hz



# H.264 与 MPEG-2/H.263/MPEG-4 的性能比较

Tempete CIF 30Hz



# 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

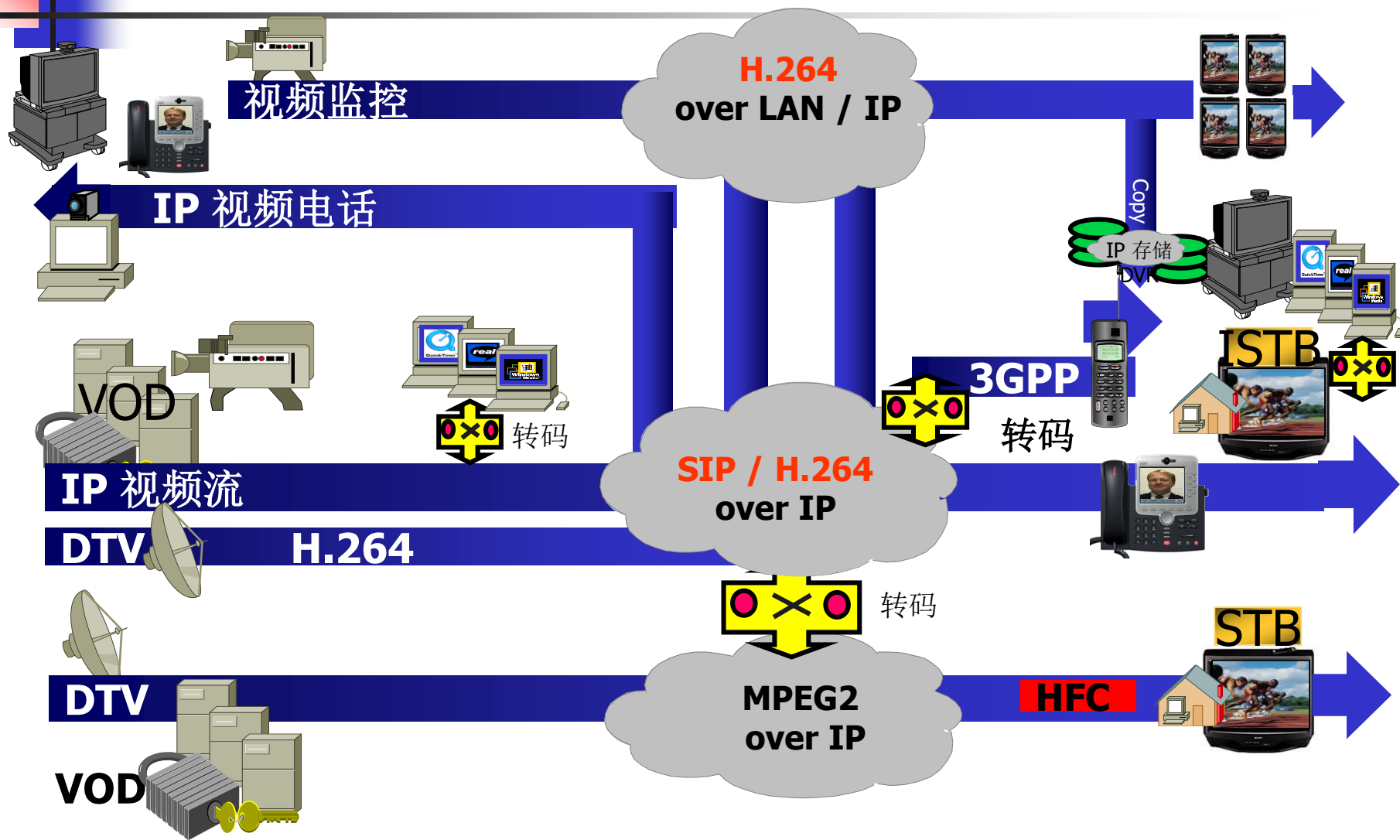


CIF, 30Hz : 340 & 1024 kbit/s





# H.264 视频应用环境





# 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

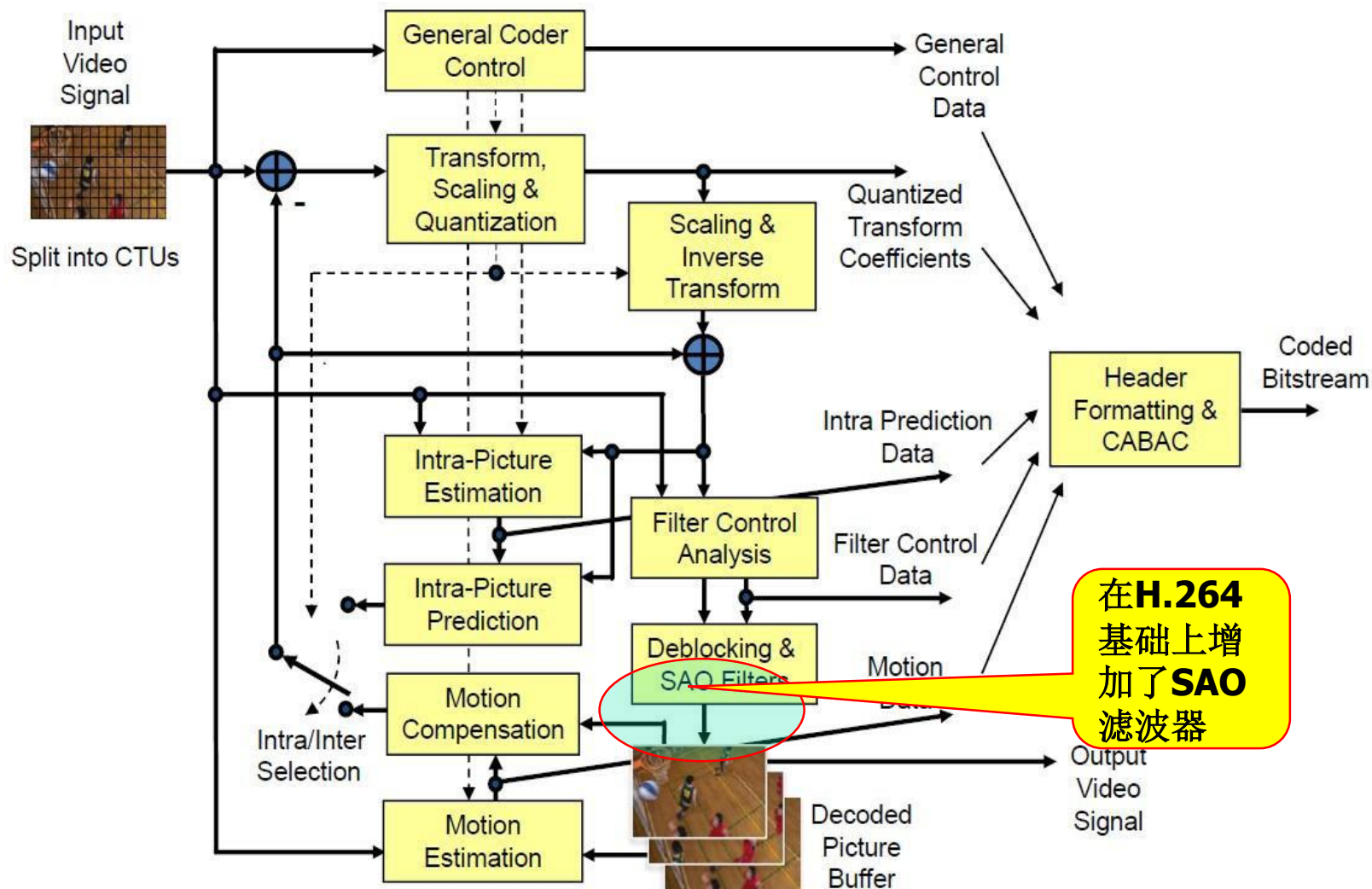


## 第6章 数字图像与视频压缩编码标准

---

- 6.1 静止图像编码标准
- 6.2 数字视频编码标准概述
- 6.3 H.264/AVC视频编码标准
- **6.4 H.265/HEVC视频编码标准**
- 6.5 AVS/AVS+视频编码标准

## 5.6.1 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 $4 \times 4$ 块：9种模式 亮度 $8 \times 8$ 块：9种预测模式 亮度 $16 \times 16$ 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

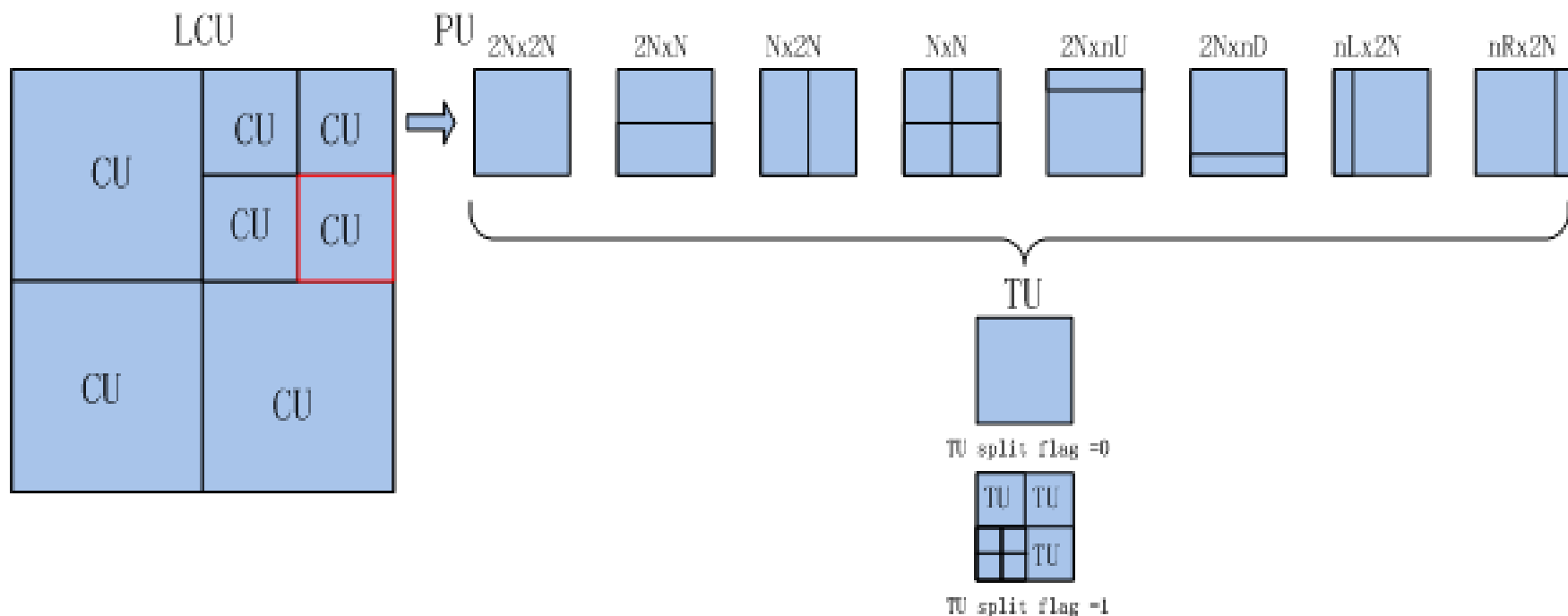


## H.265与H.264各模块技术差异(续)

变换	DCT $4 \times 4/8 \times 8$	DCT $4 \times 4/8 \times 8/16 \times 16/32 \times 32$ DST $4 \times 4$
滤波器	$4 \times 4$ 和 $8 \times 8$ 边界去块滤波	● $8 \times 8$ 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

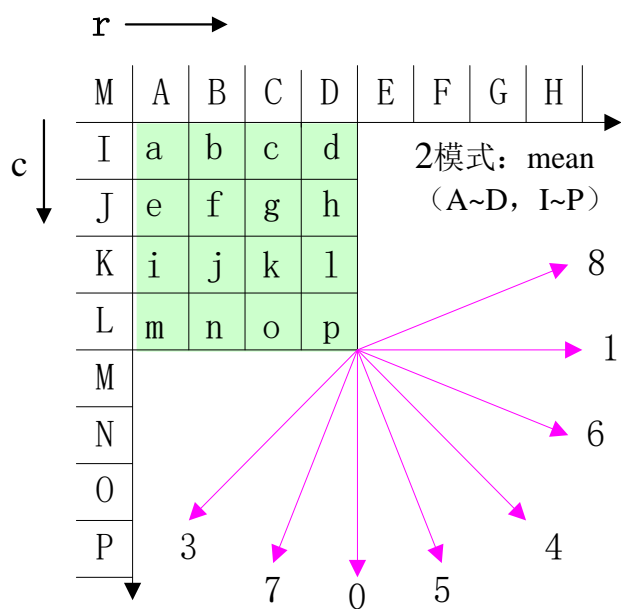
## 5.6.2 基于四叉树结构的编码单元划分

- 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

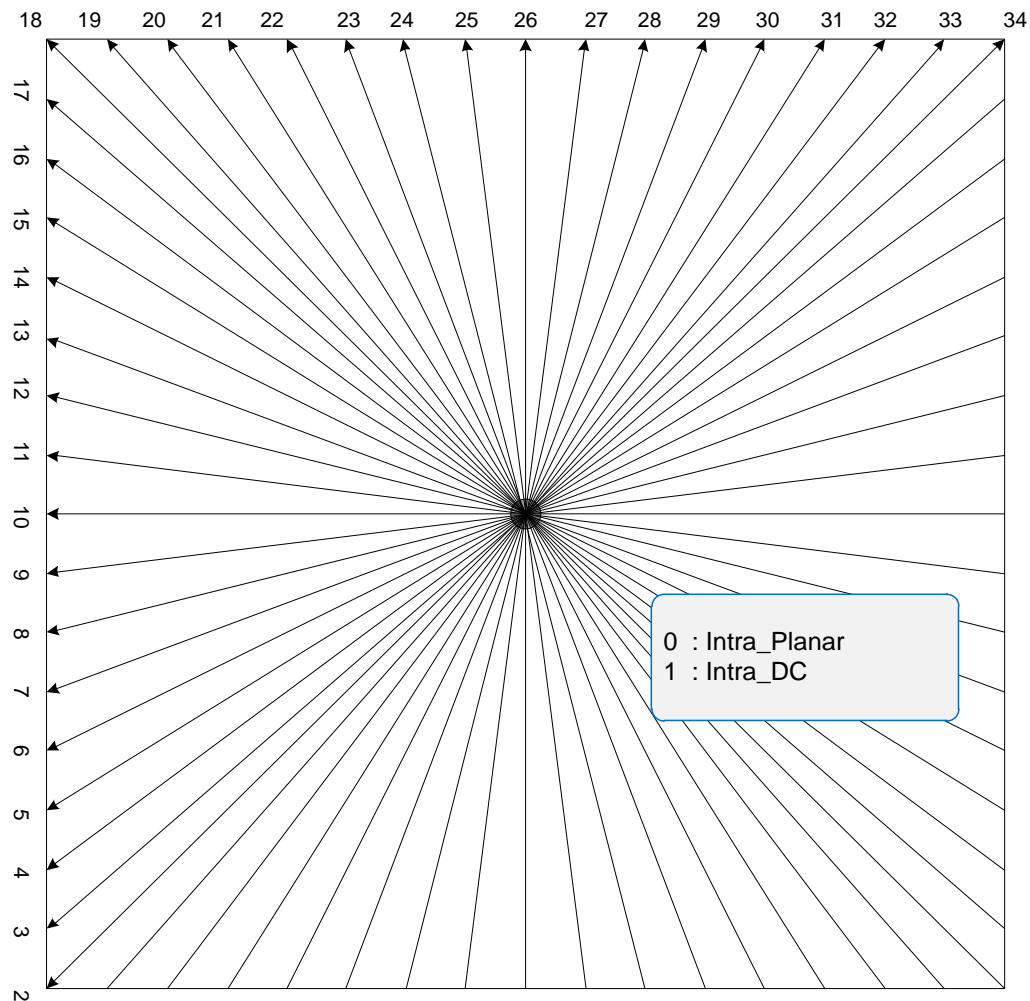


## 5.6.3 帧内预测

### H.264帧内预测方向



### H.265帧内预测方向

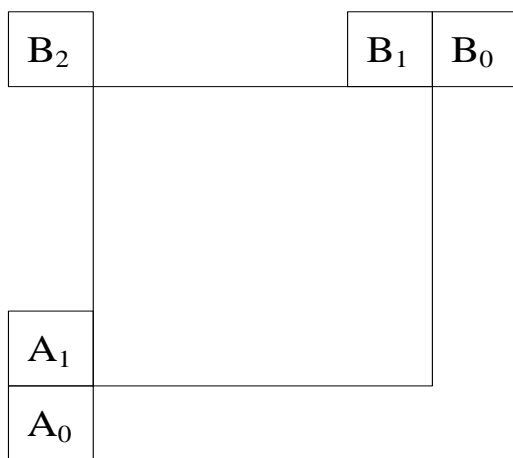




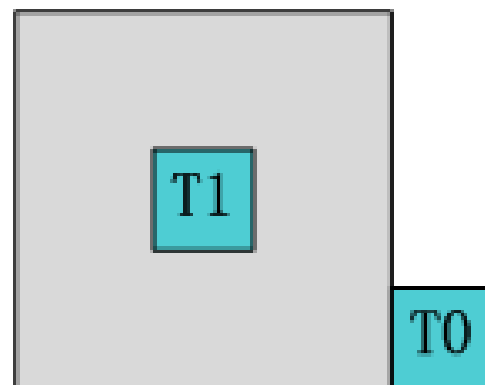
## 5.6.4 帧间预测

- 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



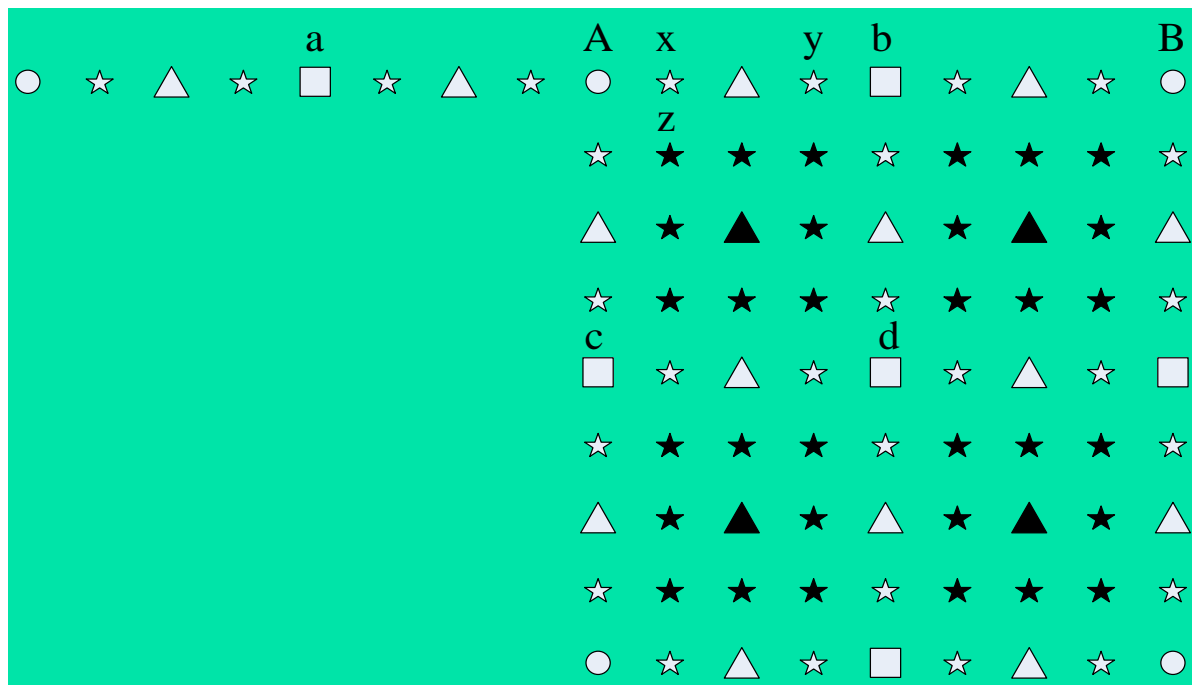
### H.265时域共同位置候选项



## 5.6.4 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 $8 \times 8$ 块所需取点数分别为 $11 \times 11$ ,  $13 \times 13$





## 5.6.5 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。



## 5.6.6 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

## 5.6.6 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模式

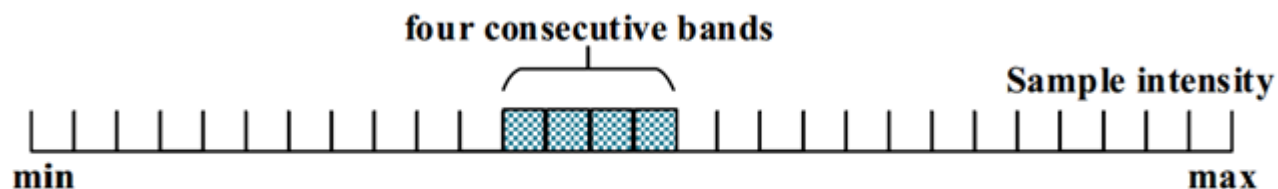
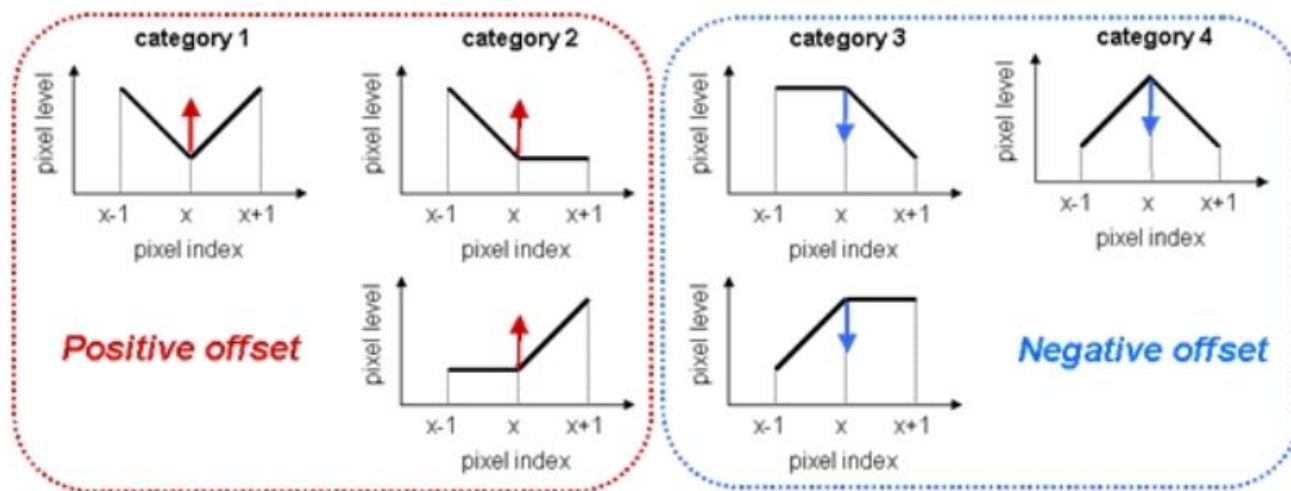
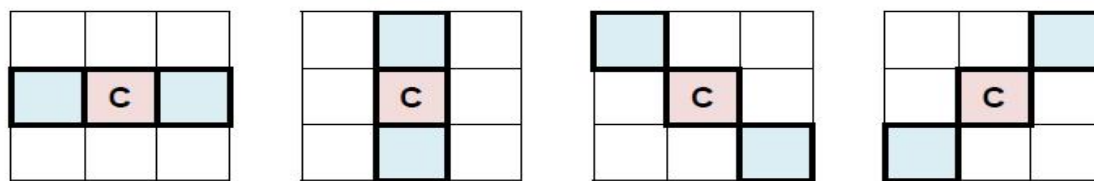


Fig. 1 Band offset

## 5.6.6 环路滤波

### 样值自适应偏移 (SAO)

EO模式



不属于以上四个类别的像素值不进行补偿。编码器只需要传输补偿值的绝对值，在解码端会根据当前像素点属于什么类别，给补偿值加正负符号



## 5.6.7 上下文自适应的熵编码

---

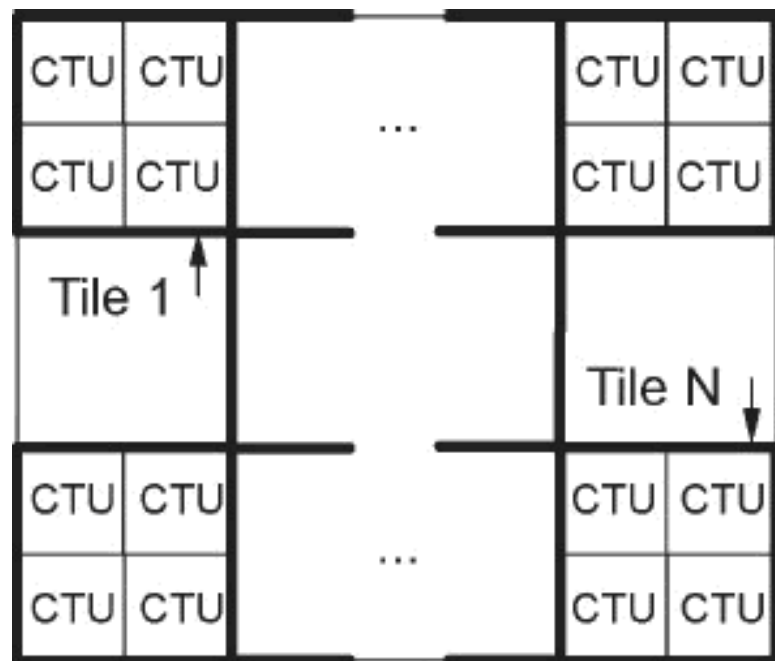
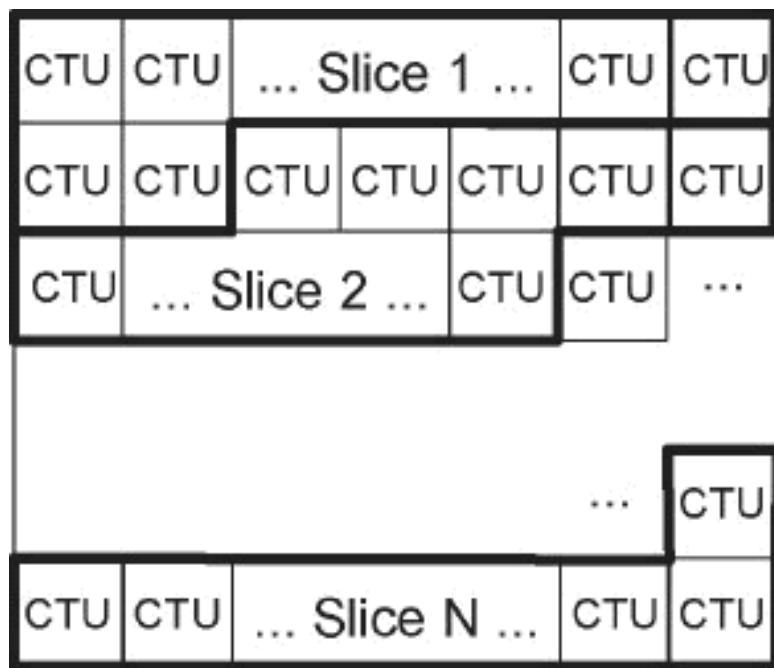
H.265/HEVC标准中使用的上下文自适应的二进制算术编码（CABAC）与H.264/AVC中使用的CABAC基本类似，除了上下文建模过程中概率码表需要重新布置以外，在算法上并没有什么变化。

**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

## 5.6.8 并行化处理

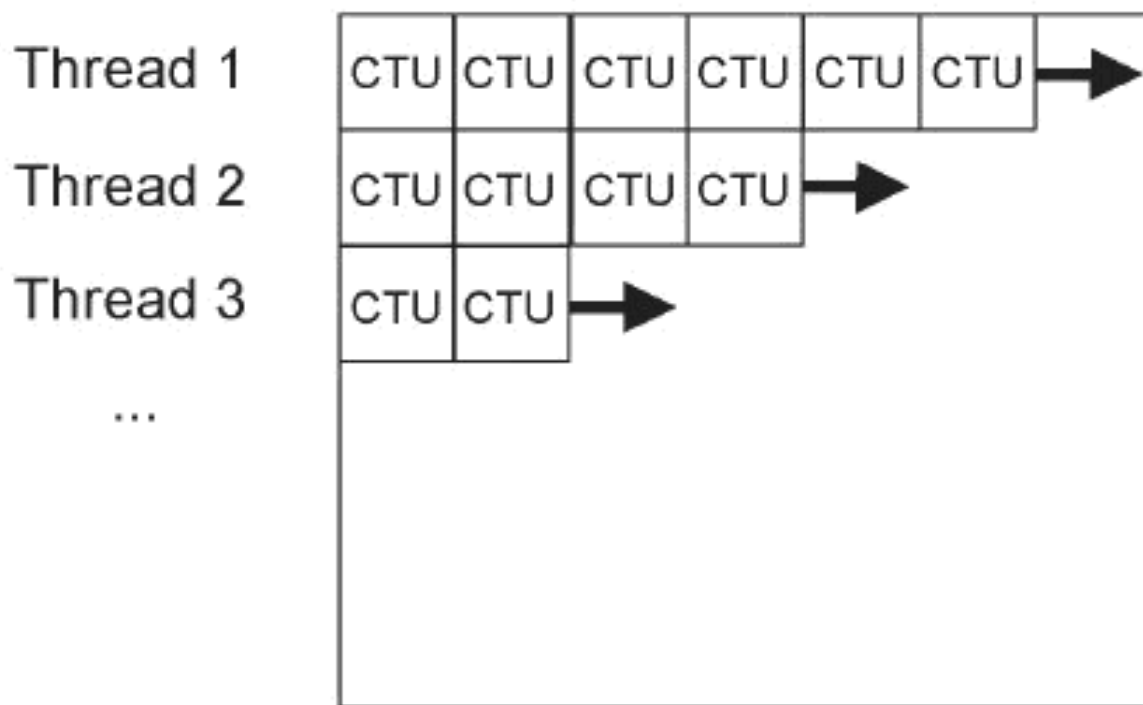
- **Tile:** 将图像分割为**矩形区域**。其主要**目的**增强**并行处理**性能。每个**Tile**区域相当于一幅子图像，可以独立的以LCU块为单位进行编解码。一个**Tile**块为基本的并行单元，每个**Tile**为一个子码流





## 5.6.8 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





## 第6章 数字图像与视频压缩编码标准

---

- 6.1 静止图像编码标准
- 6.2 数字视频编码标准概述
- 6.3 H.264/AVC视频编码标准
- 6.4 H.265/HEVC视频编码标准
- **6.5 AVS/AVS+视频编码标准**

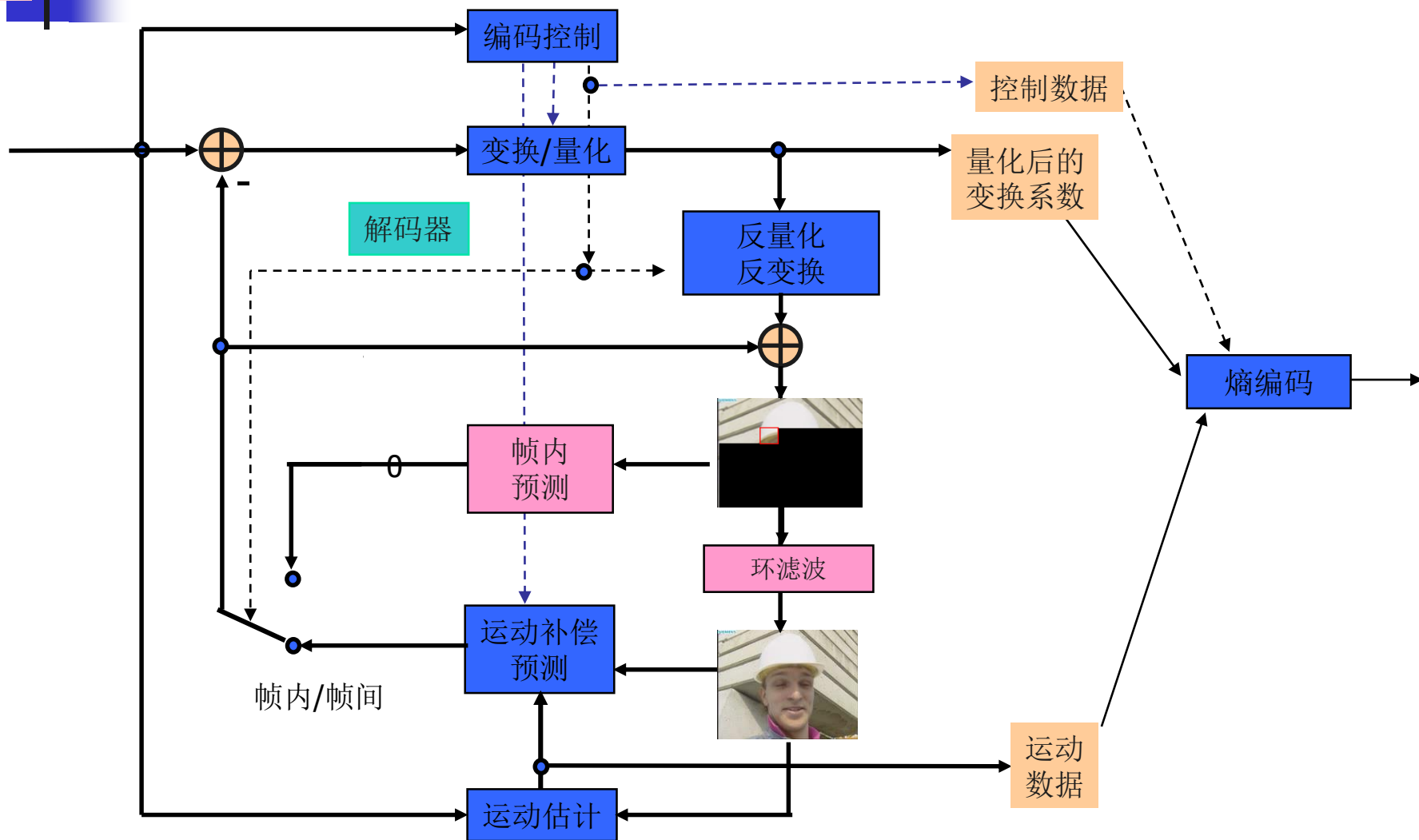


## 6.5 AVS和AVS+视频编码标准

---

- 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。**

## 6.5 AVS和AVS+视频编码标准





# 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的意义

---

- **直接产业化成果：未来10年我国需要的4 ~ 5亿颗解码芯片，辐射超万亿的国民经济产业**
- **构建完整的产业链：技术→专利→标准→芯片→系统→制造业→运营业→文化产业**
- **我国高清电视、直播卫星等音视频产业跨越发展的难得契机**



# AVS产品形态

---

## ■ 芯片

- 高清/标清AVS解码芯片，国内需求量在未来十多年的时间内年均将达到4000多万片。

## ■ 软件

- AVS节目制作与管理系统，Linux和Window平台上基于AVS标准的流媒体播出、点播、回放软件

## ■ 整机

- 机顶盒、播出服务器、编码器、高清激光视盘机、高清数字电视机顶盒和接收机、手机、便携式数码产品等

## ■ 媒体运营产业



# 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。