

# 第 22 讲: H.264/AVC 视频压缩

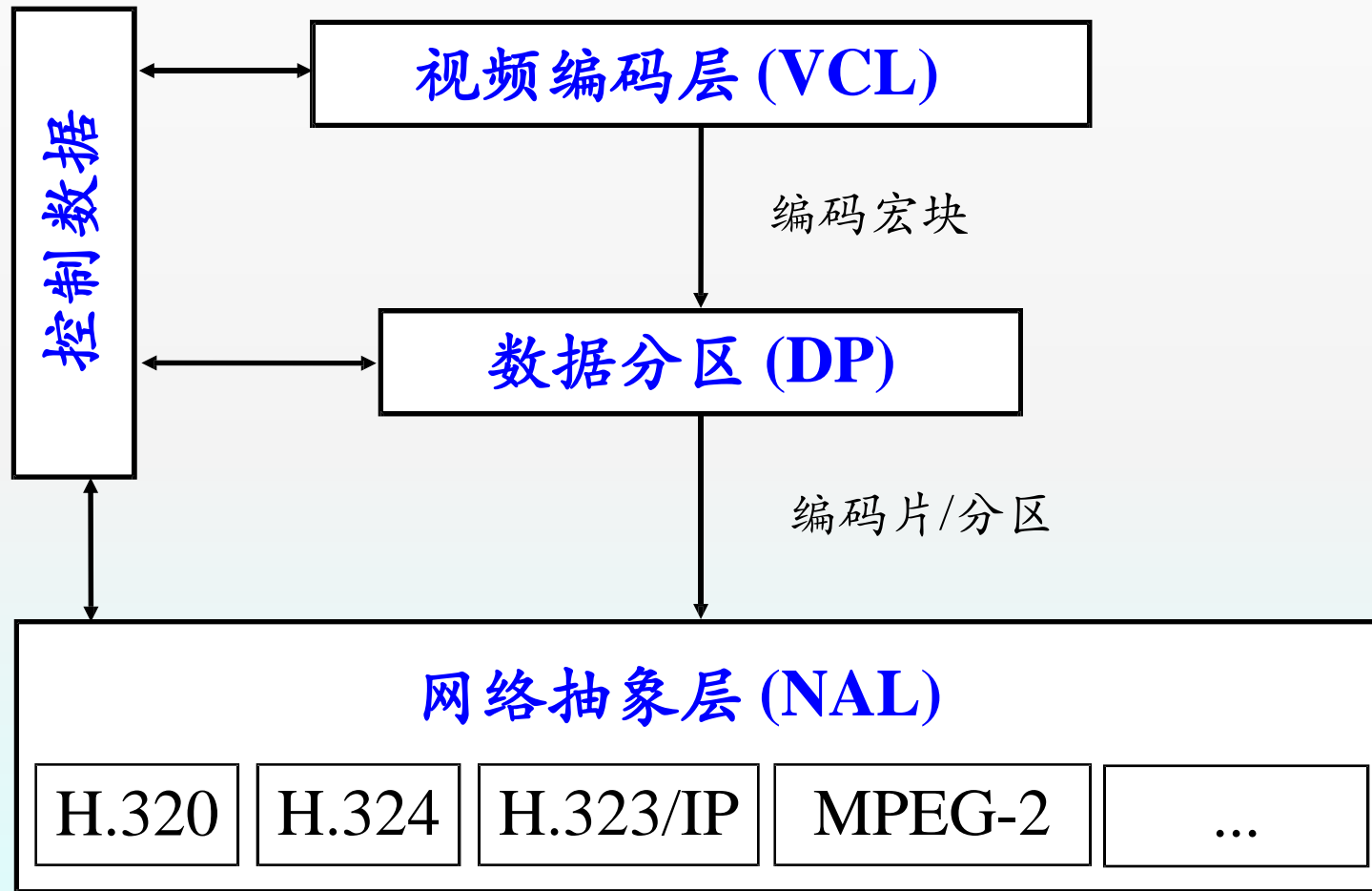
---

- H.264 / AVC 架构和数据结构
- 帧内预测
- 可变块尺寸运动补偿
- B 片和多参考运动补偿
- 运动矢量精度和编码
- 整数变换和量化
- 熵编码
- 隔行视频编码工具
- 容错编码工具
- 档次和级别
- 保真度范围扩展 (FRExt)

# H.264/MPEG-4 第10 部分: AVC

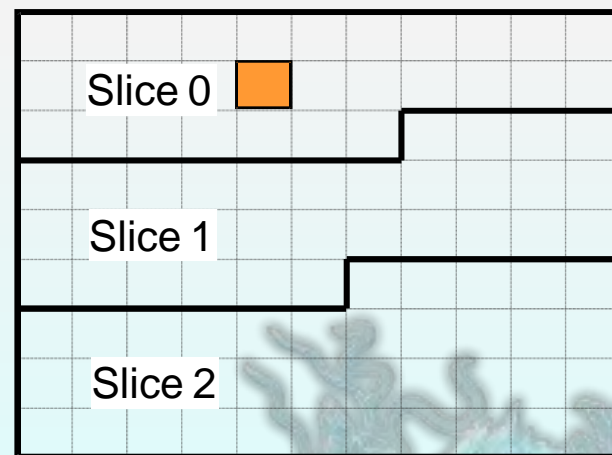
- ISO / IEC MPEG和ITU-T VCEG于2001年12月组建了**联合视频团队 (JVT)**
- 由JVT开发, MPEG命名为**MPEG-4 Part 10 / AVC**, ITU命名为**ITU-T H.264**建议书
- 对于包括隔行扫描视频的广泛应用, AVC的目标是取得**高于MPEG-2 (2倍) 的编码效率**
- **发展历史:**
  - 2001年12月: 工作草案 (WD-1.0)
  - 2002年5月 : 委员会草案 (CD)
  - 2002年8月 : 最终委员会草案 (FCD)
  - 2003年2月 : FCD研究
  - 2003年6月 : 批准标准
- H.264 / AVC是**当今最流行的视频压缩标准**
  - HDTV广播, 互联网上75%的视频流

# H.264/MPEG-4 AVC 架构



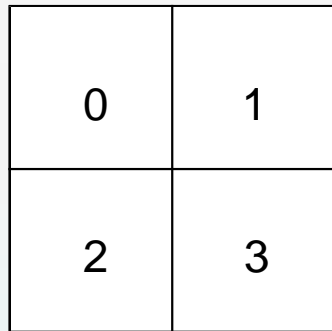
# H.264/AVC 视频数据结构

- H.264 / AVC将**类型**与**片**而不是图像相**关联**。
- 引入**IDR图像**以实现对比特流的随机访问
- 图像可以被划分成1个或多个片
- 三个主要片类型是：**I, P 和 B片**
- 两个次片类型是：**SI和SP片**
- 片是**自包含**的
- 片是**宏块序列**
- 宏块与以前的标准一样
- 片中的宏块相互依赖
- 宏块被划分成块



# 宏块到块的映射

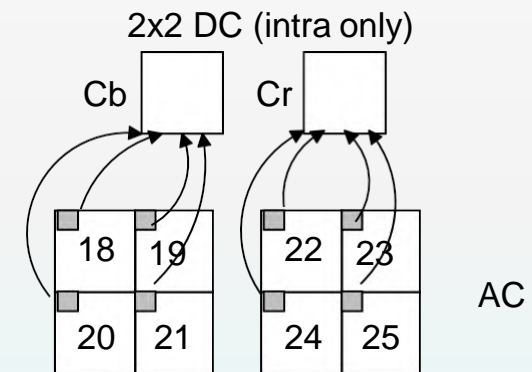
- 用于帧内预测和变换的MB由一系列 $4 \times 4$ 块组成，如下所示：



4 亮度  $8 \times 8$  块及在 MB 内的顺序



16 亮度  $4 \times 4$  块及在 MB 内的顺序



4 个Cb色度和4个Cr色度  $4 \times 4$  块及其顺序



# H.264/AVC 视频编码层

---

- **IDR图像，I，P和B片**
- **帧内空间预测**
  - $4 \times 4$ 块预测，9种模式，8种预测方向和均值
  - $16 \times 16$ 块预测，4种模式，3种预测方向和均值
- **运动补偿预测**
  - 多种块尺寸，例如 $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$ ,  $8 \times 4$ ,  $4 \times 8$ ,  $4 \times 4$
  - 用于预测的多个解码参考帧
  - 加权预测
  - 四分之一像素精度的运动矢量
  - 运动矢量差分编码与预测（中值或分割）
- **帧内/帧间变换编码**
  - $4 \times 4$ 块整数变换，DCT的近似
- **量化和扫描**
  - 量化器可以在MB基础上进行调整，量化参数在0-51范围内
  - 之字形扫描和双重扫描

# H.264/AVC 视频编码层 (cont'd)

---

- 环路自适应去块滤波器
- B图像/具有灵活性的片
  - 一般B图像以及预测其他 (P-或B-) 图像的B图像
  - 一般B图像以及使用过去或将来参考图像的B图像
- 熵编码
  - 上下文自适应VLC编码 (CAVLC)
  - 上下文自适应二进制算术编码 (CABAC)
- 隔行视频编码
  - 帧图像, 包括自适应帧/场 (AFF) 编码
  - 场图像
- 容错编码工具
  - 灵活的宏块顺序
  - 冗余片
  - 任意的片顺序
  - 数据分区
- 切换图像, SI和SP片

# H.264 / AVC与MPEG-2中的视频工具

Visual Tools	MPEG-2	H.264/MPEG-4 AVC
I, P and B pictures	Yes	No. IDR pictures and slice types
Intra prediction	DC only.	Spatial domain. Adaptive.
Variable block-size MC	No. 16×16 only.	Yes. 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4
Multiple-reference MC	No.	Yes. Up to 16 reference frames in memory.
MV prediction	Simple	Adaptive
Unrestricted MV		Yes. (Main Profile)
¼ pixel MV	No. ½ pixel accuracy	Yes.
In-loop deblocking filter	No.	Yes.
Interlace support	Yes.	Yes. (Main Profile)
Transform size	8×8	4×4
Adaptive arithmetic coding	No.	Yes. (Main Profile)



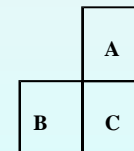
# 帧内空间预测

- 16x16 宏块帧内空间预测模式

- 模式0: 垂直预测
- 模式1: 水平预测
- 模式2: 直流预测
- 模式3: 平面预测

- 4×4块空间预测模式

- 模式0: 垂直预测
- 模式1: 水平预测
- 模式2: 直流预测
- 模式3: 对角左下预测
- 模式4: 对角右下预测
- 模式5: 垂直向左预测
- 模式6: 水平向下预测
- 模式7: 垂直向右预测
- 模式8: 水平向上预测



0	0	2	2
1	1	3	3
4	4	6	6
5	5	7	7

- 4x4帧内预测模式编码

# 帧内空间预测 (cont'd)

## • 16x16 帧内空间预测方向

- 直流:  $(A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P)/16$
- 垂直: 使用A,B,C,D,E,F,G,H去预测对应列 a,b,c,d,...
- 水平: 使用 I,J,K,L,M,N,O,P 去预测对应行 a,e,i,m,...
- 平面

## • 4x4 帧内空间预测方向

- 直流:  $(A+B+C+D+I+J+K+L)/8$
- 8 个方向:

### • 水平

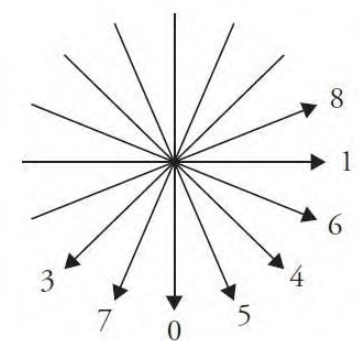
$a=b=c=d=I$   
 $e=f=g=h=J$   
 $i=j=k=l=K$   
 $m=n=o=p=L$

### • 垂直

$a=e=i=m=A$   
 $b=f=j=n=B$   
 $c=g=k=o=C$   
 $d=h=l=p=D$

### • 2- 直流

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								



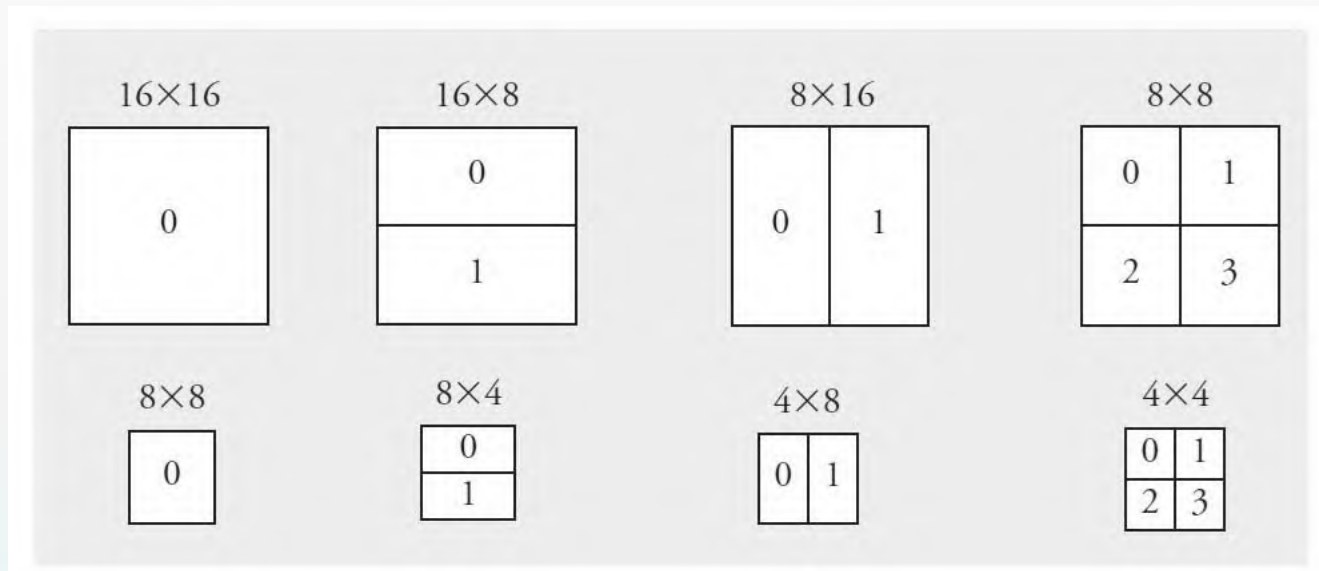
# 运动补偿

---

- 可变块大小运动估计和补偿
- 对于MB，可使用 $16 \times 16$  亮度块的运动估计/补偿
- 对于MB的亮度块，运动估计/补偿可以为 $16 \times 8$ ,  $8 \times 16$ 和 $8 \times 8$ 亮度块
- 对于 $8 \times 8$ 亮度块，运动估计/补偿可以为 $8 \times 4$ ,  $4 \times 8$ 和 $4 \times 4$ 块
- 图像边界可以扩展以进行运动补偿
- 1/4像素运动补偿精度
- 多参考运动补偿
- B片可作为运动补偿的参考

# 运动补偿 (cont'd)

- 基于**宏块分割**的运动估计/补偿

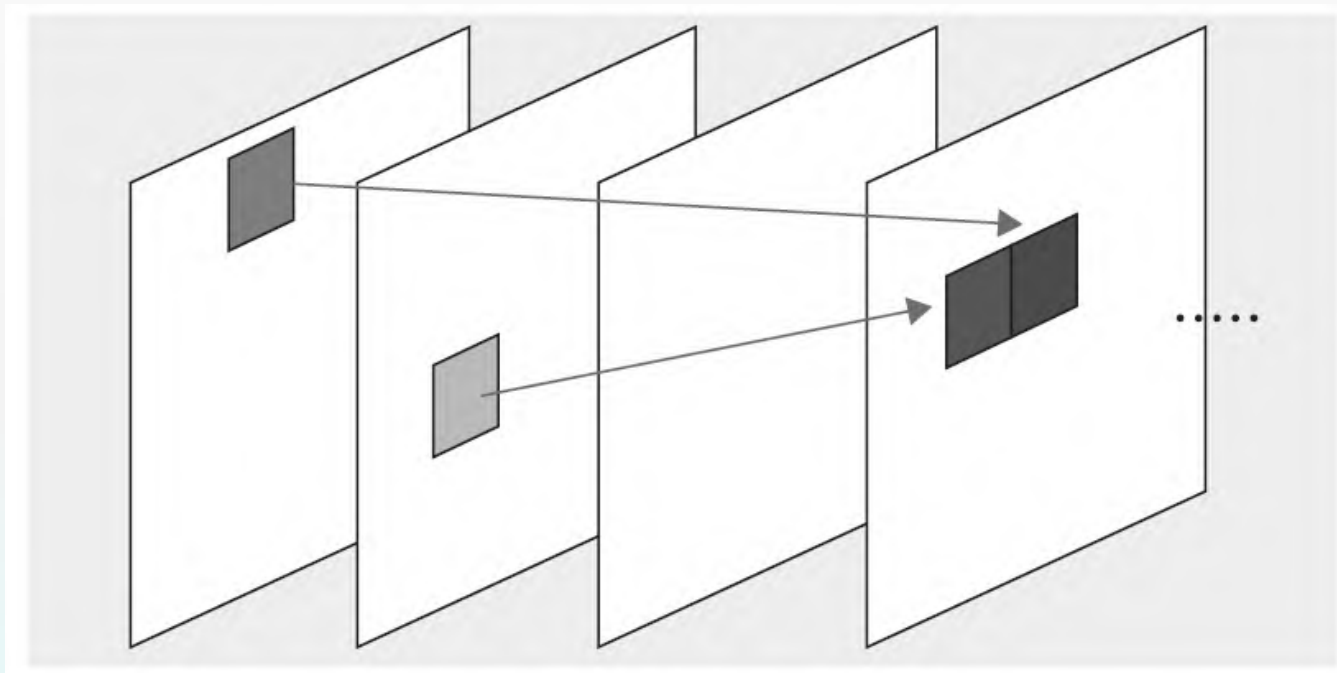


- MC预测参考帧**

- 可以在 $16 \times 16$ ,  $16 \times 8$ 或 $8 \times 16$ 的基础上选择
- 如果每个块以 $8 \times 8$ 的子分区模式编码，则可以为每个 $8 \times 8$ 块再选择一次

# 多参考帧运动补偿

- 可以使用 **多于1个过去的解码帧** 进行预测



**3个过去的解码帧**

- 编码器通知解码器保存哪些帧
- 用作参考的实际帧的信令需要被发送到解码器端，用于宏块（或基于亚宏块）的预测



# 参考图像缓冲区管理

- 条目

- 参考图像缓冲区：用于参考图像的缓冲区，最多16个参考帧
- 参考索引列表：分配给参考图像缓冲区中的参考图像的索引列表
- 参考索引列表0：在P，B或SP片中用于帧间预测的参考索引列表
- 参考索引列表1：用于B片的帧间预测的参考索引列表
- 图像顺序计数：以解码顺序相对于最新的IDR图像的输出顺序的图像位置
- 图片重新排序：解码顺序与输出顺序不同时对图像重新排序
- **ref\_pic\_buffering\_mode**指定当前解码图像的缓冲模式，并指定当前图像被解码后如何修改图像缓冲区

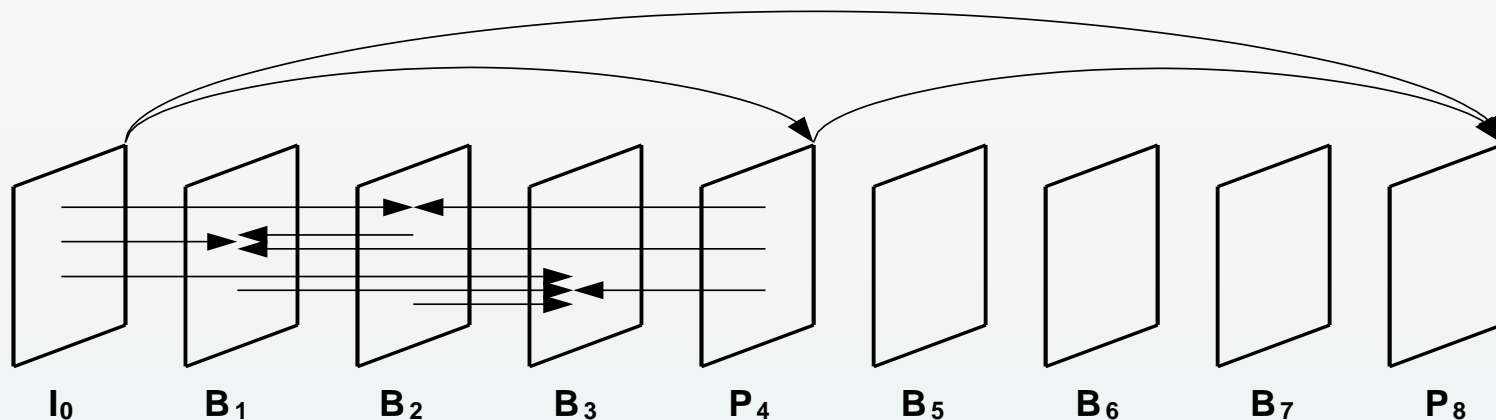
## 值      参考图像缓冲区

- 0      滑动窗缓冲；一个简单的FIFO，用于没有长期索引的图像
- 1      自适应缓冲；将图像标记为未使用，分配长期索引，重置参考图像缓冲区

- **memory\_management\_control\_operation**是发送到解码器的一个指令，以指定要应用于管理解码器参考图像缓冲区的操作
- - 如果memory\_management\_control\_operation复位，则参考图像缓冲区中的所有帧都应标记为未使用。

# B片/图像

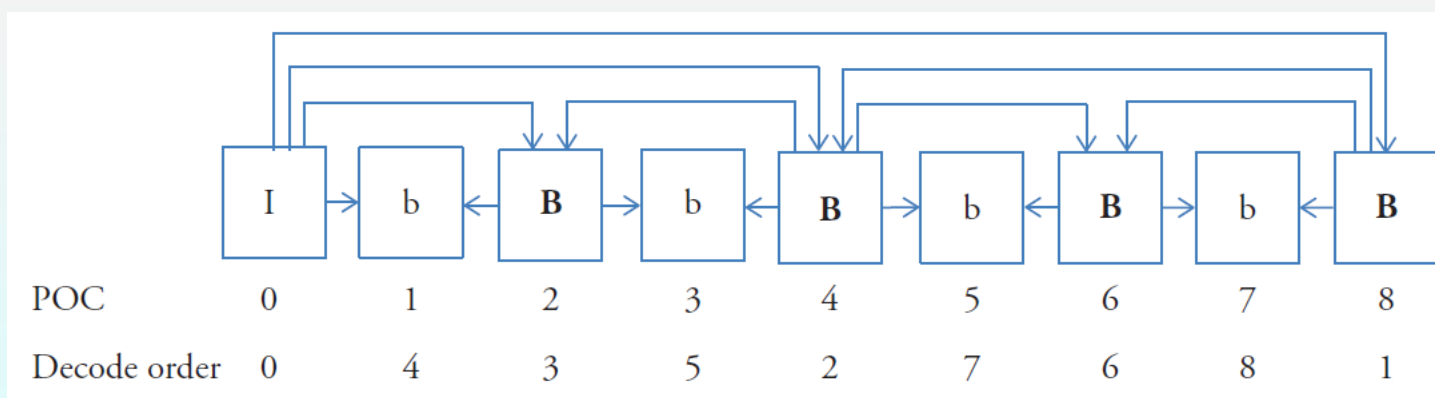
- **预测参考帧**是过去的一帧和未来的一帧（如MPEG-1,2,4），或者都是过去帧，甚至都是未来帧。



- 5种预测类型：**直接，正向，反向，双向和帧内**
- **双向模式**发送显式运动矢量（正向/反向模式），而**直接模式**使用缩放的相同位置宏块的运动矢量
- 每个16x16, 16x8, 8x16以及每个8x8子分区的参考帧参数。此外，对于每个16x16, 16x8, 8x16和8x8子分区，预测类型可以单独选择

# 分层 B图像

- B-图像
- b-图像
- 提高压缩效率
- 时间可扩展性



# 四分之一像素运动补偿

- 用于1/4像素精度运动补偿的插值

- ‘A’ 是原始像素

- ‘b’ 为半采样位置像素，可通过在水平方向上应用滤波器{1, -5, 20, 20, -5, 1}到最近的“A像素”，作为中间值。然后，通过除以32，四舍五入和截断到0-255范围计算出最终值

A	d	b	d	A
e	h	f	h	
b	g	c	g	b
e	h	f	i	
A		b		A

- ‘c’ 通过在垂直或水平方向上最接近的1/2个样本位置的中间值‘b’上应用{1, -5, 20, 20, -5, 1}获得，以得到中间值。然后通过除以1024，四舍五入和截断到0-255范围计算出最终值。

- ‘d’, ‘g’, ‘e’ 和 ‘f’ 通过将两个整数或1/2样本位置上的样值进行平均和截断获得，如： $d=(A+b)/2$ ,  $g=(b+c)/2$ ,  $e=(A+b)/2$ ,  $f=(b+c)/2$

- 1/4 样本位置 ‘h’ 通过将两个在对角线方向最近的 ‘b’ 样值进行平均和截断获得

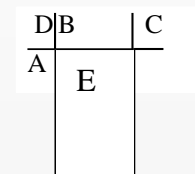
- ‘i’ 通过使用4个最近的原始像素的均值  $(A1+A2+A3+A4+2)/4$  来获得。



# 自适应运动矢量预测

## • MV的中值预测

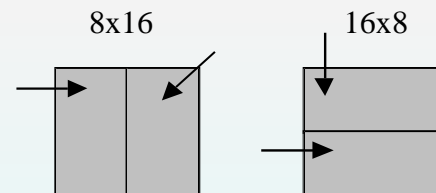
- A, B, C, D, E的预测可以使用不同的参考图像
- 如果C在图像或片之外, 则它假定为D的mv和参考图像索引
- 如果B, C, D在图像或片之外, 则它们假定为A的mv和参考图像索引
- 如果以上没有指定的预测变量被编码为帧内, 则mv为0, 并且参考图索引与E不同。
- 如果A, B和C中只有一个具有与E相同的参考图像, 则其mv用于E的预测, 否则E的预测mv的每个分量是A, B的mv分量的中值。



## • 分割方向预测

矢量块大小 8x16

- 左块, 如果具有与E相同的参考图像, 则A用作预测, 否则使用中值预测
- 右块, 如果具有与E相同的参考图像, 则C用作预测, 否则为中值预测



矢量块大小 16x8

- 上块, 如果与E使用相同的参考图像, 则A用作预测, 否则使用中值预测
- 下块, 如果具有与E相同的参考图像, 则C用作预测, 否则使用中值预测



# 2D 整数变换

- **(斜) 4x4 整数变换**，近似于DCT，但是系数为2的幂，以消除乘法运算。

**DCT**

$$T = \begin{bmatrix} .50 & .50 & .50 & .50 \\ .65 & .27 & -.27 & -.65 \\ .50 & -.50 & -.50 & .50 \\ .27 & -.65 & .65 & -.27 \end{bmatrix}$$

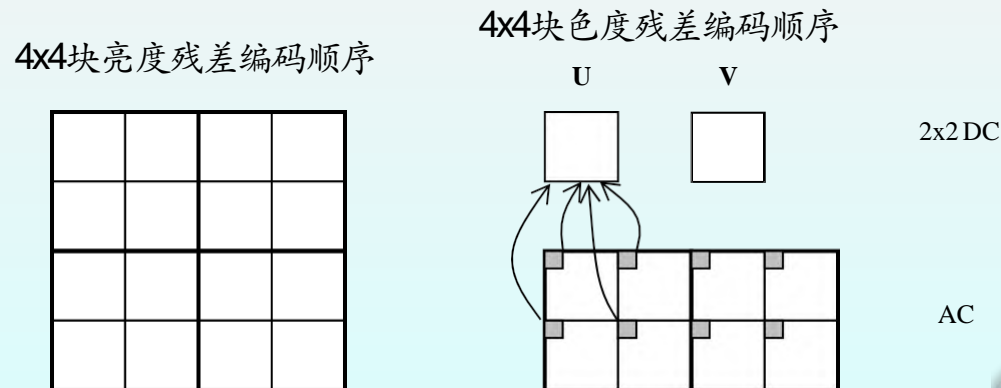
**斜变换**

$$T = \frac{1}{2} \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 1 & -1 & -3 \\ 1 & -1 & -1 & 1 \\ 1 & -3 & 3 & -1 \end{bmatrix}$$

**AVC 整数变换**

$$T = \frac{1}{\sqrt{10}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

- 每行的缩放常数被**吸收到量化步骤**中。



# 量化

## ■ 量化特征

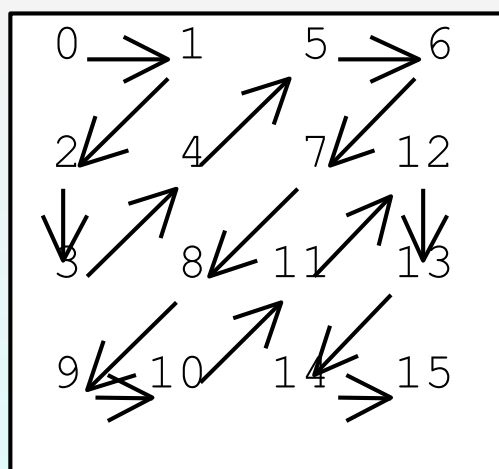
- 对数步长控制
  - 步长范围扩大 (52个值而不是31个)
  - 提供对色度更小的步长
  - 步长可以在宏块级别更改为任意值
  - 步长尺寸预测差值被编码
- 在范围[0-51]中有**52个QP**值。缩放方程式的设计使得参数在Qp每增加6时翻倍。从一个Qp到另一个Qp的缩放比例大约增加了12%
- **色度QP**,  $QP_C$ 可由亮度Qp,  $QP_Y$ 获得:

$QP_Y$ : <30    30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51

$QP_C$ : =  $QP_Y$     29 30 31 32 32 33 34 34 35 35 36 36 37 37 37 38 38 38 39 39 39 39

# 量化与系数扫描

- 正向/反向变换的**缩放与量化/去量化**一起执行。整个操作被设计为16位比特整数运算
- 4x4块的变换系数进行**之字形扫描**



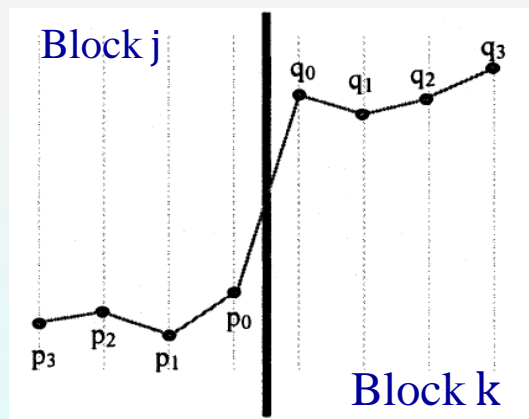
之字形扫描

# 去块滤波器



- 滤波器对 $4 \times 4$ 或 $8 \times 8$ 变换块的边界像素进行操作。
- 滤波器强度可预先设定或由解码器确定。

## 滤波器强度的确定



Block modes and conditions	Bs
One of the blocks is Intra and the edge is a macroblock edge	4
One of the blocks is Intra	3
One of the blocks has coded residuals	2
Difference of block motion luma sample distance $\geq 1$	1
Motion compensation from different reference frames	1
Else	0

Strong filter

Weak filter

Filter off

- **弱滤波器** 仅在 $p_0$ 和 $q_0$ 上工作
- **强滤波器** 在每侧3个像素上运行

# VLC 编码 - Golomb 码

- **Golomb码 (UVLC)** 具有规则的结构，根据以下规则生成：

1  
0 1 x<sub>0</sub>  
0 0 1 x<sub>1</sub> x<sub>0</sub>  
0 0 0 1 x<sub>2</sub> x<sub>1</sub> x<sub>0</sub>  
0 0 0 0 1 x<sub>3</sub> x<sub>2</sub> x<sub>1</sub> x<sub>0</sub>  
.....

其中  $x_n = 0$  或  $1$

## ■ VLC 码表

Table A: Unsigned Symbols

Code_number	Code word
0	1
1	0 1 0
2	0 1 1
3	0 0 1 0 0
4	0 0 1 0 1
5	0 0 1 1 0
6	0 0 1 1 1
7	0 0 0 1 0 0 0
8	0 0 0 1 0 0 1
9	0 0 0 1 0 1 0
10	0 0 0 1 0 1 1
.....	.....

Table B: Mapping of Signed Symbols

Code_number	Signed Symbol
0	0
1	1
2	-1
3	2
4	-2
5	3
6	-3
7	4
8	-4
9	5
10	-5
.....	.....



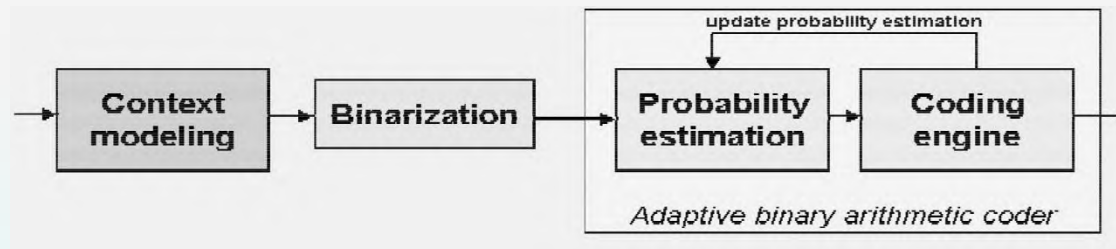
# 上下文自适应VLC (CAVLC)- 系数编码

- **变换系数使用自适应VLC编码**
  - 没有EOB被解码，系统向后扫描，使用上下文
  - 非零系数的个数，所有非零系数的电平值和符号，最后一个非零系数之前所有零的个数，每个非零系数前的游程
- **系数的个数/拖尾的‘1’**
  - 最后非零系数具有|电平| 为 1
  - 非零系数的个数 (例如, 6) 和拖尾1的个数 (2) 作为一个联合符号进行编码，通常有 50% 的系数可被编码为拖尾1及其符号，不需要其他电平级别。
  - 选择的VLC 码表由相邻块系数的个数决定
- **逆向扫描和电平编码**
  - 最后非零系数（拖尾1）的统计特性比第一个（电平值较高）非零系数要更加稳定
  - 逆向扫描，对第一个系数使用缺省的 VLC, 使用电平值上下文选择下一个码表
- **游程信息 - TotalZeros 与 RunBefore**
  - TotalZeros: 在正向扫描中最后一个非零系数之前零的总个数
  - RunBefore: 在逆向扫描中对每个非零系数之前的游程进行编码

# 上下文自适应二进制算术编码

- 基于上下文的自适应二进制算术编码 (CABAC)

- 上下文建模以提供对符号条件概率的估计
- 允许非整数比特分配给符号表中每个符号的算术代码
- 自适应算术编码，以自适应非平稳符号统计特性



- 运动和模式信息编码的上下文建模

- 宏块类型的上下文模型
- 运动矢量数据的上下文模型
- 参考帧参数的上下文模型

# CABAC 算术编码 (cont.)

---

- 纹理数据编码的上下文建模
  - 编码块模式的上下文模型
  - 帧内预测的上下文模型
  - 游程/电平和系数个数的上下文模型
    - 基于上下文的系数个数编码
    - 基于上下文的游程编码
    - 基于上下文的电平信息编码
- 双重扫描始终用于CABAC帧内模式
- Dquant编码的上下文建模
- 非二值信号的二值化
- 自适应二进制算术编码

# 附加工具

- 隔行视频编码效率工具

- 静态帧编码：两幅图像（场）组合在一起进行编码。
- 静态场编码：两场独立地编码为两个图像。
- 序列级自适应帧场编码
- 图像自适应帧场（PAFF）：编码器在每幅图像决定使用帧编码或场编码。
- 宏块自适应帧场（MBAFF）：编码器决定每MB的编码模式。

- 容错编码工具

- 数据分区
- 灵活的宏块顺序（FMO）
- 任意片顺序（ASO）
- 冗余片

- S图像，流媒体切换工具，SI和SP片

- 高效的内存管理工具（MMCO）



# 隔行视频编码

---

- 场编码

- 将每场视为单独图像进行编码

- 帧编码

- 将每帧视为单独图像进行编码

- 根据运动剧烈程度选择

- 低运动图像适合帧编码，快速运动适合场编码

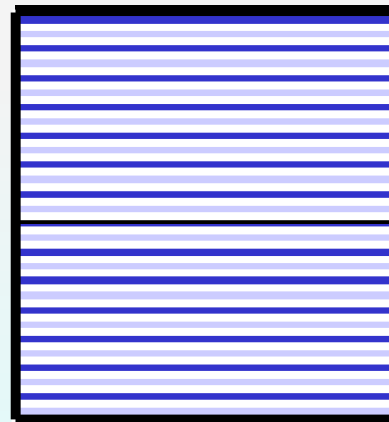
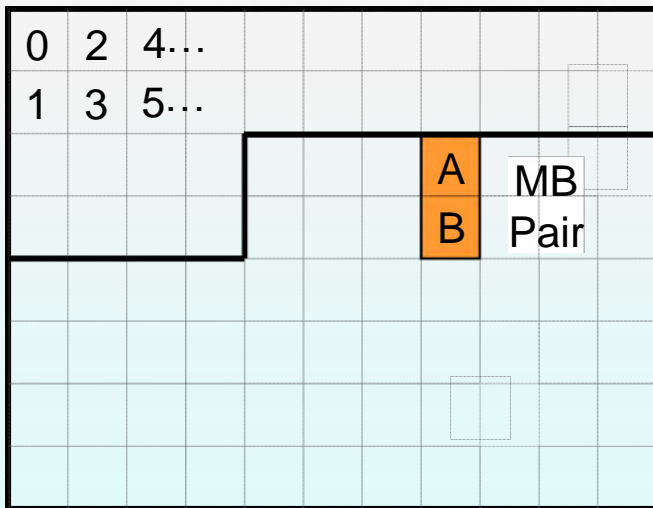
- 根据视频内容来使用不同编码模式。有三个选项：

- 在序列级别（出现运动的变化）
- 在图像级别或
- 在宏块级别（在空间变化的情况下）

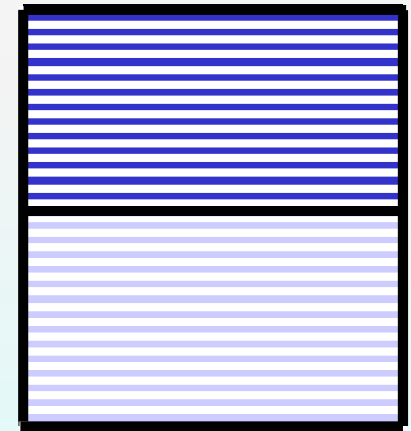


# 隔行帧编码: MBAFF 编码

- 基于宏块的自适应帧/场 (MBAFF) 编码
- 在帧编码模式中支持:
  - 在每帧中扫描宏块对, 对于每个宏块对, 可以选择帧或场编码。



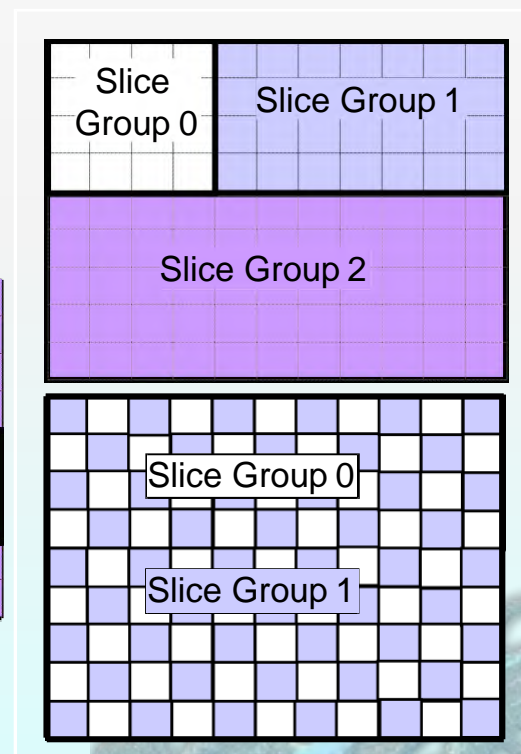
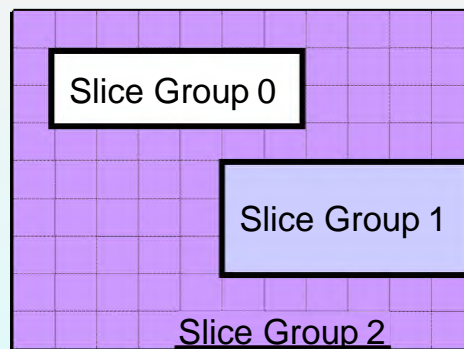
帧组织的 MB 对



场组织的 MB 对-上/下场

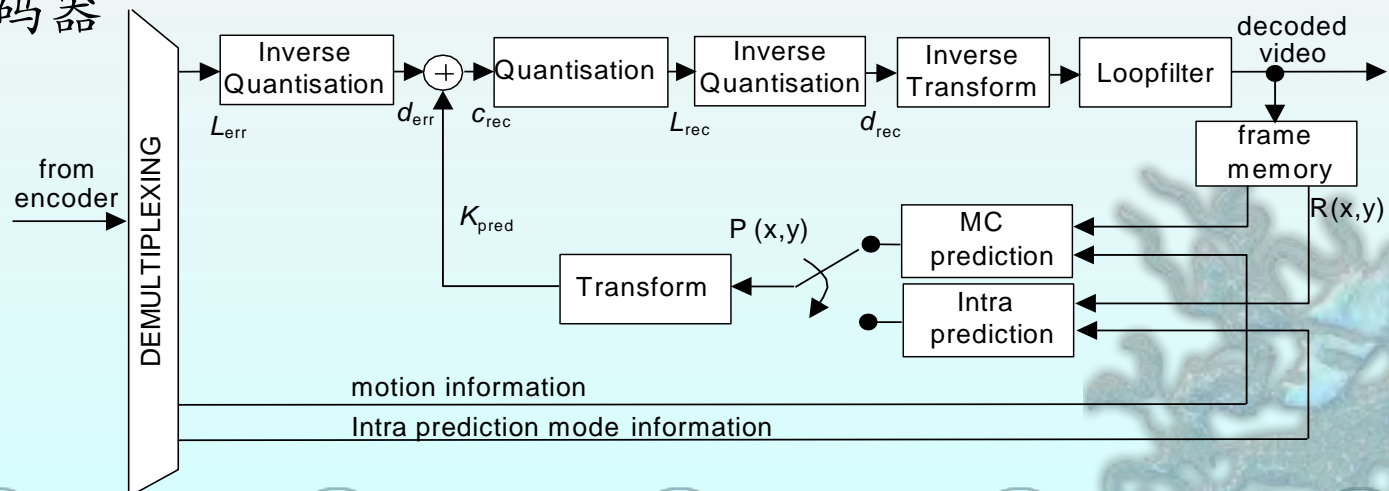
# 灵活的宏块顺序 (FMO)

- 片组可以包含一个或多个片
- 宏块分配图可以定义将宏块分配给片的模式
- 宏块分配图的类型
  - 交织的
  - 分散的
  - 指定的
  - 剩余的

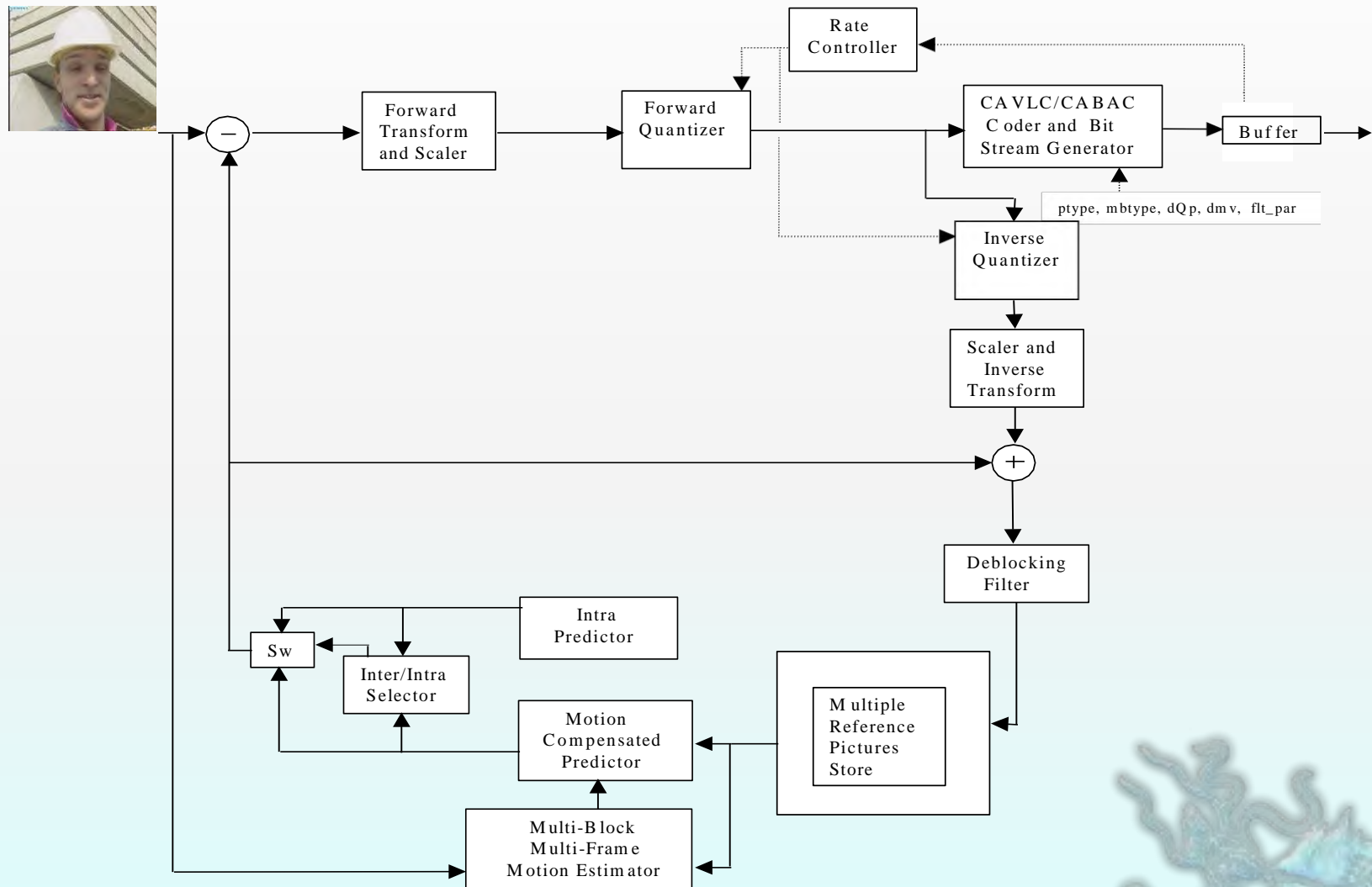


# 切换片/图像 (SI和SP)

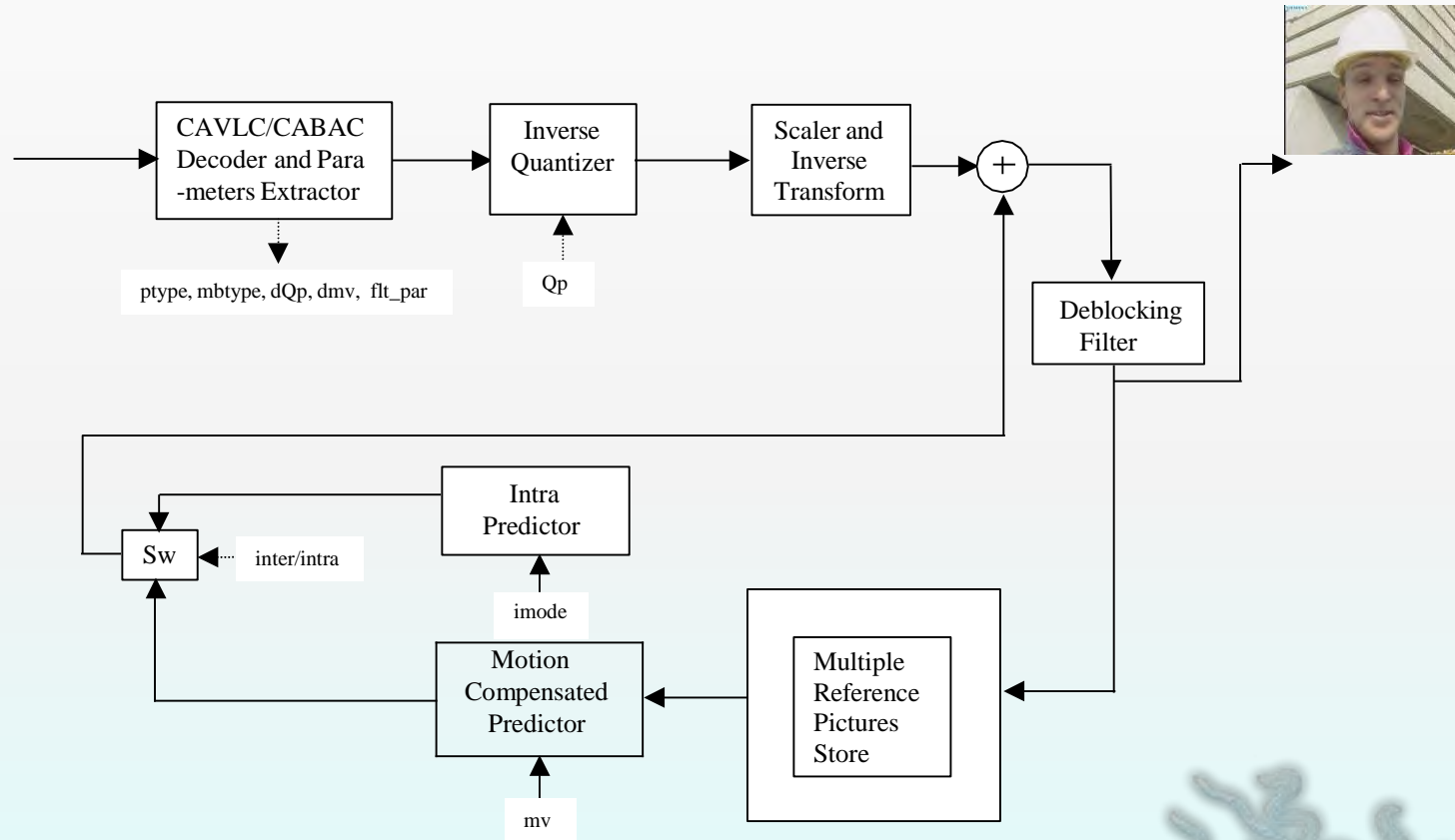
- 切换图像用于比特流切换，拼接，随机访问，VCR功能，如快进，容错/错误恢复
- SI图像使用空间预测（如I图像），并且SP图像使用运动补偿预测以利用时间冗余（如同P图像）
- 与P图像不同的SP图像即使在使用不同的参考帧时也允许对帧进行相同的重建
- S图像解码器



# H.264/AVC 编码器{非规范}



# H.264/AVC 解码器





# H.264/AVC 视频编码

---

- I, P, B图像结构选择
- B帧
  - B帧个数和结构选择
  - 存储或不存储的B帧
  - 空间直接与时间直接
- 运动估计/补偿和参考帧
  - 快速或穷举搜索
  - MB分区选择
  - 运动补偿参考帧数
  - 运动补偿参考帧选择
  - SAD与SATD
- 模式确定（重要!!）
  - 基于非RD（低复杂度）
  - 使用拉格朗日的RD优化（高复杂度）

# H.264/AVC 视频编码 (cont'd)

---

- 量化和系数阈值化
  - 基于有或没有感知的
  - 消除帧间 MB 中的单个系数：亮度和色度
- 码率控制
  - 基于图像的
  - 基于MB的
- 环路滤波器
  - 缺省
  - 发送控制参数
- 容错和流媒体切换编码
  - FMO, ASO,..
  - 参考帧更新和再发送
  - 切换图像(SI 与 SP)
- 其他由档次选择的特定工具 (例如 CABAC 初始化)

# 运动估计/模式确定: 低复杂度

- 预测模式确定的步骤:

- SA(T)D0: 初始偏移, 有利于选择需要较少比特的预测模式
- 块差异: 原始和预测之间的块差异
- SAD或SATD (使用Hadamard 变换): 对于帧内和分数像素精度, 使用SATD
- 选择可使SA(T)D<sub>min</sub> 最小化的预测模式。

- 帧内和帧间模式

- 帧内模式

- 以帧内模式及4x4块与预测对MB进行编码, 计算 SATD<sub>intra</sub>
- 编码器使用的帧内预测模式表: 解码器使用表的逆

- 帧间模式

- 搜索7个块结构和 5 个过去的解码帧, 共有35种关于块尺寸和参考帧的组合方式
- 整数像素搜索: 围绕预测向量使用螺旋搜索。对16x16的块使用全范围, 但其他MC块尺寸的范围为一半。对于先前图像使用四分之一范围
- 分数像素精度搜索: 首先, 搜索最佳整数向量周围的8个半像素位置, 然后搜索最佳半像素位置周围的8个1/4像素位置。计算 SATD<sub>inter</sub>

- 帧内和帧间模式的确定: 如果 SATD<sub>intra</sub> < SATD<sub>inter</sub>, 使用帧内编码

# 运动估计/模式确定: 高复杂度

## • 运动估计步骤

### ■ 整数像素搜索

- 同低复杂度模式, 在预测矢量周围使用螺旋搜索
- 对于所有块大小和参考帧使用全范围
- 为了加速搜索, 使用16x16块MC作为螺旋搜索的中心

### ■ 分数像素搜索

- 同低复杂度模式

### ■ 最佳运动矢量: 整数/子像素通过最小化 $SA(T)D(s, c(\mathbf{m})) + \Lambda_{\text{motion}} R(\mathbf{m} - \mathbf{p})$ 进行细化

### ■ 寻找最佳参考帧: 最小化 $SATD(s, c(\text{ref}, \mathbf{m}(\text{ref}))) + \Lambda_{\text{motion}} (R(\mathbf{m}(\text{ref}) - \mathbf{p}(\text{ref})) + R(\text{ref}))$

s: 原始视频  
c: 编码视频  
m: 运动矢量  
p: 预测运动矢量  
R(): 码率项  
 $\Lambda$ : 拉格朗日乘子  
SSD: 方差和

## • 模式确定

### ■ 宏块模式确定: 最小化 $SSD(s, c, \text{mode}|Q_p) + \Lambda_{\text{mode}} R(s, c, \text{mode}|Q_p)$

### ■ Intra16x16: 选择产生最小SATD值的模式

### ■ Intra4x4: 同MB 模式确定, 最小化 $SSD(s, c, \text{Imode}|Q_p) + \Lambda_{\text{mode}} R(s, c, \text{Imode}|Q_p)$

## • 运动估计和预测模式确定步骤

### ■ 给定先前解码帧, $\Lambda_{\text{mode}}$ , $\Lambda_{\text{motion}}$ , 和 MB 量化器 $Q_p$

### ■ INTRA4x4 MB 模式, 为每个4x4块从9个可能的选项中选择预测模式

### ■ INTRA16x16 MB模式, 为每个16x16块从4个可能的选项中选择预测模式

### ■ 通过最小化运动代价函数来执行运动估计和参考帧选择

### ■ 通过最小化模式代价函数来选择MB预测模式



# H.264/AVC 档次

- 基本档次：低延迟实时通信
- 扩展档次：流媒体
- 主要档次：TV 和 DVD

基本档次中的工具	主要档次中的工具
<ul style="list-style-type: none"><li>• Only I and P slice types may be present</li><li>• No interlace support</li><li>• No CABAC</li><li>• Arbitrary slice order (ASO)</li><li>• Flexible macroblock ordering (FMO)</li><li>• Redundant pictures allowed</li></ul>	<ul style="list-style-type: none"><li>• I, P, B slices</li><li>• CABAC</li><li>• Support for interlaced video</li><li>• No ASO</li><li>• No FMO</li><li>• No redundant pictures</li></ul>



# H.264/AVC 档次比较

<i>Tool</i>	<i>Baseline</i>	<i>Extended</i>	<i>Main</i>
I, P pictures/slices	yes	yes	yes
1/4 sample MC	yes	yes	yes
MC up to 4x4 Block size	yes	yes	yes
Multiple Reference Pictures	yes	yes	yes
4x4 Block Transform	yes	yes	yes
In Loop Deblocking Filter	yes	yes	yes
Adaptive VLC Entropy Coding	yes	yes	yes
Redundant Slices	yes	yes	no
Arbitrary Slice Ordering	yes	yes	no
Flexible Macroblock Ordering	yes	yes	no
B-pictures/slices	no	yes	yes
Weighted Prediction	no	yes	yes
CABAC Arithmetic Entropy Coding	no	no	yes
SI/SP Slices	no	yes	no
Data Partitioned Slices	no	yes	no
Interlaced Frame/Field Video Coding	no	yes	yes

# NAL 与文件格式

- 网络适配层 (NAL)

- 定义字节流NAL格式，包含一系列数据包，其前缀为：零填充字节，3个字节的起始码前缀，有效载荷类型指示符字节，由有效载荷类型指示的一个或多个EBSP
- NAL层从临时文件格式获取信息，并将其直接转换为可以通过RTP直接传输的数据包。必须智能地排列数据包，分割/重组分区以匹配MTU的大小约束，避免RTP报头的冗余，并在丢包的情况下采取指定操作

- 文件格式

- 基于ISO MPEG-4的文件格式定义的文件格式
- 文件包含多个盒子，每个盒子可以包含其他盒子
- 盒子可能包含成员属性，如果一个盒子包含属性，则其他盒子应在属性之后
- 许多盒子包含索引值到其他盒子中的序列
- 文件格式在MPEG-4语法描述语言中指定

# H.264/AVC 保真度范围扩展 (FRExt)

- 2004年7月，作为修正添加到MPEG-4 part10以支持扩展像素位深和色度格式。
- 对高保真视频而言，与MPEG-2相比，可实现高达3倍的提高
- **高档次系列**，包括High Profile (HP) ， High 10 Profile (Hi10P) ， High 4: 2: 2 (Hi422P) 和High 4: 4: 4 Profile (Hi444P) 均已被添加，其中除了支持主要档次的所有特征以外，还支持 $8 \times 8$  帧内预测，自适应变换块大小 ( $4 \times 4$  或  $8 \times 8$ ) 以及感知量化缩放矩阵。
- Hi444P还支持RGB视频的**整数残差颜色变换**和**无损编码功能**（旁路变换和量化步长）。
- H.264 / MPEG-4 AVC HP比JPEG 2000平均高0.5 dB。

# H.264/MPEG-4 AVC 资源

---

- H.264 software  
VideoLAN x.264 Free Software Library and Application  
<http://www.videolan.org/developers/x264.html>  
  
<https://trac.ffmpeg.org/wiki/Encode/H.264>
- Ian Richardson's H.264 notes  
<http://www.vcodex.com/h264.html>