

H.264/AVC编解码器

H.264/AVC的结构

帧内预测

帧间预测

变换与量化

H.264/AVC编解码器

H.264/AVC的结构

帧内预测

帧间预测

变换与量化

H.264/AVC编解码器

H.264编解码器特点

H.264并不明确地规定一个编解码器如何实现，而是规定了一个编码的视频比特流的句法，和该比特流的解码方法，各个厂商的编码器和解码器在此框架下应能够互通，在实现上具有较大灵活性，而且有利于相互竞争。

H.264/AVC编解码器

H.264编码器

- 编码器采用的仍是变换和预测的混合编码法。输入的帧或场 F_n 以宏块为单位被编码器处理。首先，按帧内或帧间预测编码的方法进行处理。
- 如果采用帧内预测编码，其预测值PRED（图中用P表示）是由当前片中前面已编码的参考图像经运动补偿（MC）后得出，其中参考图像用 F'_{n-1} 表示。为了提高预测精度，从而提高压缩比，实际的参考图像可在过去或未来（指显示次序上）已编码解码重建和滤波的帧中进行选择。
- 预测值PRED和当前块相减后，产生一个残差块 D_n ，经块变换、量化后产生一组量化后的变换系数X，再经熵编码，与解码所需的一些边信息（如预测模式量化参数、运动矢量等）一起组成一个压缩后的码流。
- 正如上述，为了提供进一步预测用的参考图像，编码器必须有重建图像的功能。因此必须使残差图像经反量化、反变换后得到的 D_n' 与预测值P相加，得到 uF_n' （未经滤波的帧）。为了去除编码解码环路中产生的噪声，为了提高参考帧的图像质量，从而提高压缩图像性能，设置了一个环路滤波器，滤波后的输出 F_n' 即重建图像可用作参考图像。

H.264/AVC编解码器

H.264解码器

- 由图1可知，由编码器的NAL输出一个压缩后的H.264压缩比特流。由图2，经熵解码得到量化后的一组变换系数 X ，再经反量化、反变换，得到残差 Dn' 。利用从该比特流中解码出的头信息，解码器就产生一个预测块PRED，它和编码器中的原始PRED是相同的。当该解码器产生的PRED与残差 Dn' 相加后，就产生 uFu' ，再经滤波后，最后就得到滤波后的 Fn' ，这个 Fn' 就是最后的解码输出图像。

H.264/AVC编解码器

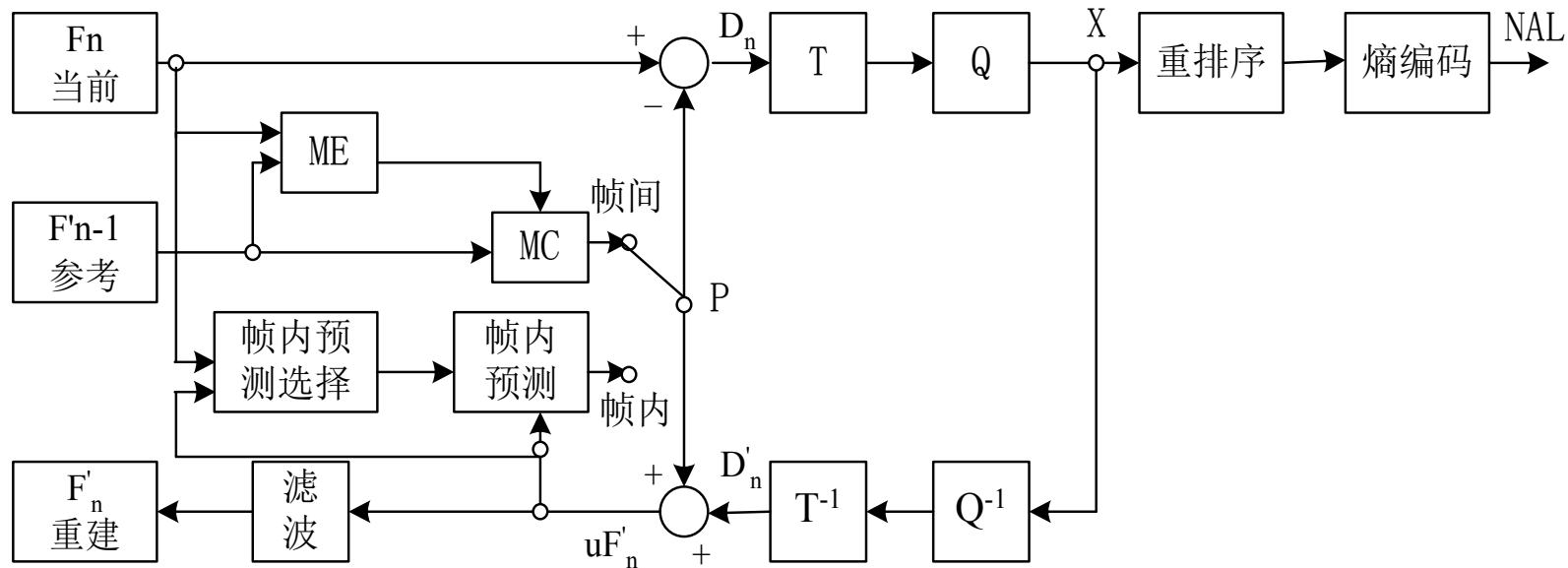


图1 H.264编码器

H.264/AVC编解码器

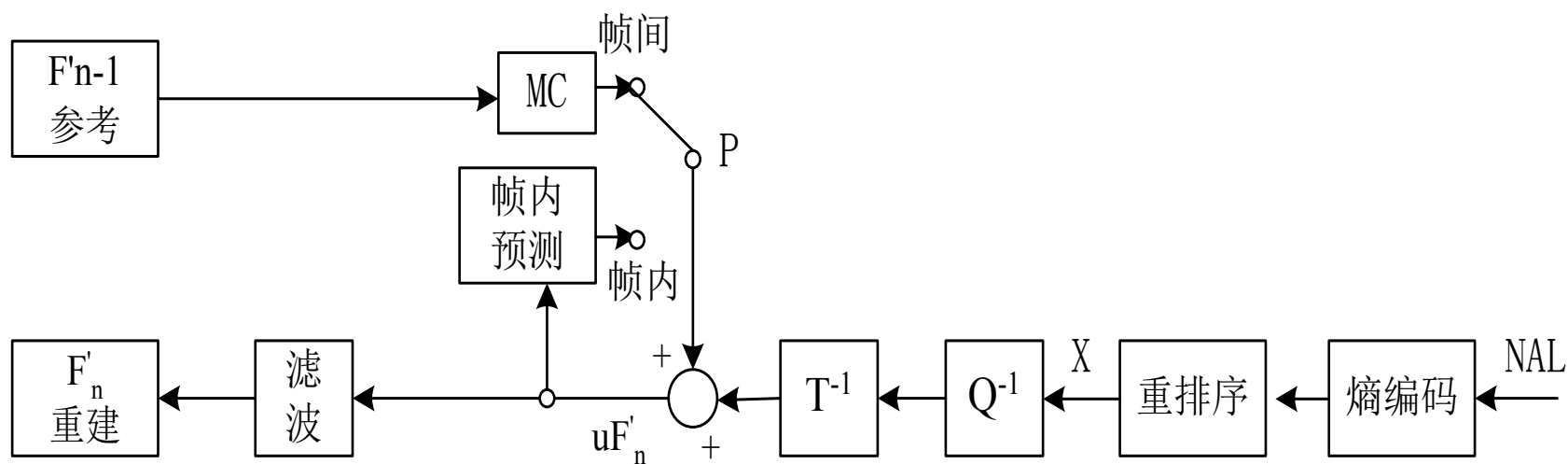


图2 H.264解码器

H.264/AVC编解码器

H.264/AVC的结构

帧内预测

帧间预测

变换与量化

H.264/AVC的结构

名词解释

— 场和帧

视频的一场或一帧可用来产生一个编码图像。在电视中，为减少大面积闪烁现象，把一帧分成两个隔行的场。

H.264/AVC的结构

名词解释

- **宏块、片**：一个编码图像通常划分成若干宏块组成，一个宏块由一个 16×16 亮度像素和附加的一个 8×8 Cb和一个 8×8 Cr彩色像素块组成。每个图象中，若干宏块被排列成片的形式。
- I片只包含I宏块，P片可包含P和I宏块，而B片可包含B和I宏块。
- I宏块利用从当前片中已解码的像素作为参考进行帧内预测。
- P宏块利用前面已编码图象作为参考图象进行帧内预测，
- 一个帧内编码的宏块可进一步作宏块的分割：即 16×16 、 16×8 、 8×16 或 8×8 亮度像素块（以及附带的彩色像素）；如果选了 8×8 的子宏块，则可再分成各种子宏块的分割，其尺寸为 8×8 、 8×4 、 4×8 或 4×4 亮度像素块（以及附带的彩色像素）。
- B宏块则利用双向的参考图象（当前和未来的已编码图象帧）进行帧内预测。。

H.264/AVC的结构

档次和级

- H.264规定了三种档次，每个档次支持一组特定的编码功能，并支持一类特定的应用。
- 1) 基本档次：利用I片和P片支持帧内和帧间编码，支持利用基于上下文的自适应的变长编码进行的熵编码（CAVLC）。
主要用于可视电话、会议电视、无线通信等实时视频通信；
- 2) 主要档次：支持隔行视频，采用B片的帧间编码和采用加权预测的帧内编码；支持利用基于上下文的自适应的算术编码（CABAC）。
主要用于数字广播电视与数字视频存储；
- 3) 扩展档次：支持码流之间有效的切换（SP和SI片）、改进误码性能（数据分割），但不支持隔行视频和CABAC。
主要用于网络的视频流，如视频点播
- 图3为H.264各个档次具有的不同功能，可见扩展档次包括了基本档次的所有功能，而不能包括主要档次的。每一档次设置不同参数（如取样速率、图像尺寸、编码比特率等），得到编解码器性能不同的级。

H.264/AVC的结构

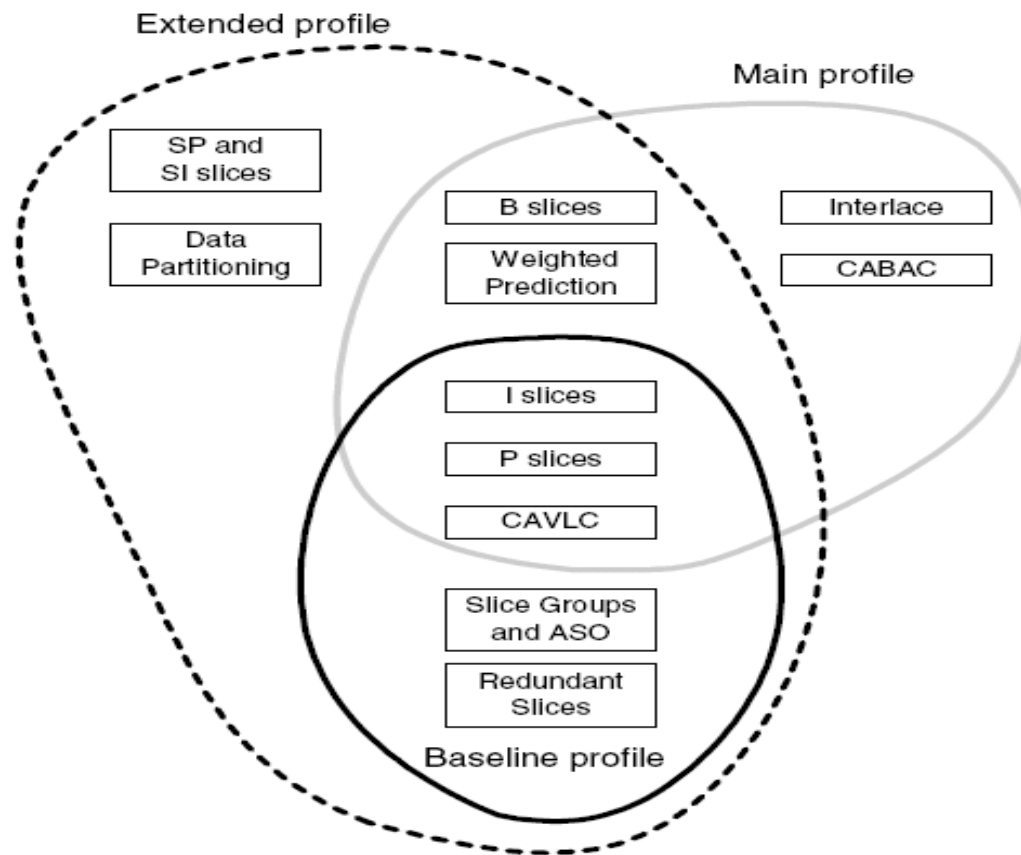


图3 H.264档次

H.264/AVC的结构

编码数据格式

- H.264支持4:2:0的逐行或隔行视频的编码和解码。

H.264/AVC的结构

参考图像

- 可从一组前面或后面已编码图像中选出一个或两个与当前最匹配的图像作为帧间编码间的参数图像，
- H.264中最多可从15个参考图像中进行选择，选出最佳的匹配图像。
- 对于P片中帧间编码宏块可从表“0”中选择参数图像；对于B片中的帧间编码宏块和宏块分割的预测，可从表“0”和“1”中选择参考图像。

H.264/AVC的结构

片和片组

— 片

- 一个视频图像可编码成一个或更多个片，每片包含整数个宏块（MB），即每片至少一个MB，最多时每片包含整个图像的宏块。
- 设片的目的是为了限制误码的扩散和传输，使编码片相互间是独立的。某片的预测不能以其它片中的宏块为参考图像，这样某一片中的预测误差才不会传播到其它片中去。
- 编码片共有5种不同类型，I片、P片、B片外，还有SP片和SI片。其中SP（切换P）是用于不同编码流之间的切换。

H.264/AVC的结构

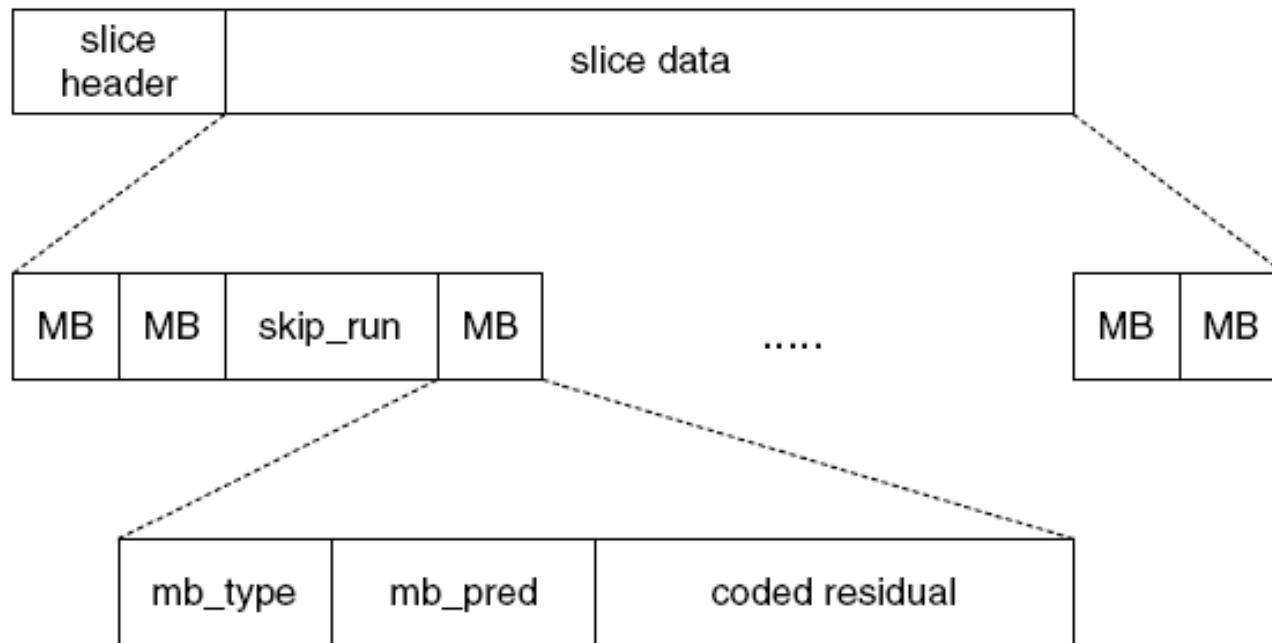


图4 片的句法结构

H.264/AVC的结构

片和片组

— 片组

片组是一个编码图象中若干MB的一个子集，它可包含一个或若干个片。

- 在一个片组中，每片的MB按光栅扫描次序被编码，如果每幅图象仅取一个片组，则该图象中所有的MB均按光栅扫描次序被编码。
- 还有一种片组，叫灵活宏块次序（FMO），它可用灵活的方法，把编码MB序列映射到解码图象中MB的分配用MB到片组之间的映射来确定，它表示每一个MB属于哪个片组。表1 为MB到片组的各种映射类型。

H.264/AVC的结构

表1 MB到片组的映射

类型	名称	描述
0	交错	MB游程被依次分配给每一块组（图8）
1	散乱	每一片组中的MB被分散在整个图象中（图9）
2	前景和背景	例见图10
3	Box-out	从帧的中心开始，产生一个箱子，其MB属于片组0，其它MB属于片组（图11）
4	光栅扫描	片组0包含按光栅扫描次序从顶-左的所有MB，其余MB属片组1（图11）
5	手绢	片组0包含从顶-左垂直扫描次序的MB，其余MB属片组1（图11）
6	显式	每一Mbslice_group_id,用于指明它的片组（即MB映射完全是用户定义的）

H.264/AVC的结构

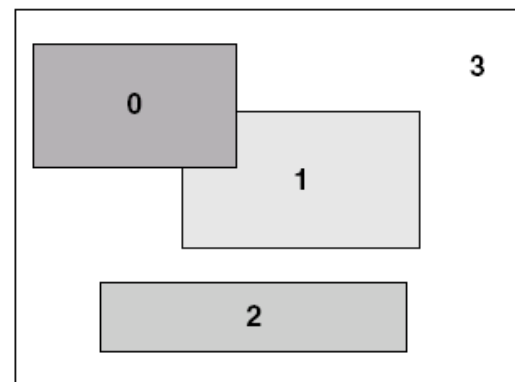
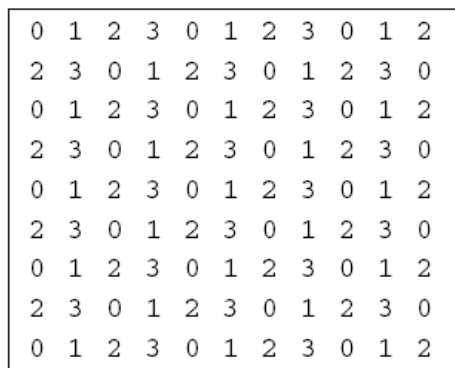
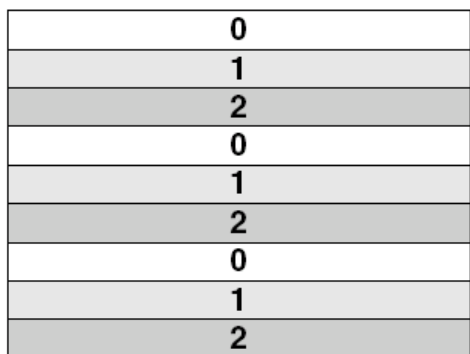


图5 交错型片组

图6 散乱型片组

图7 前景和背景型片组

H.264/AVC的结构

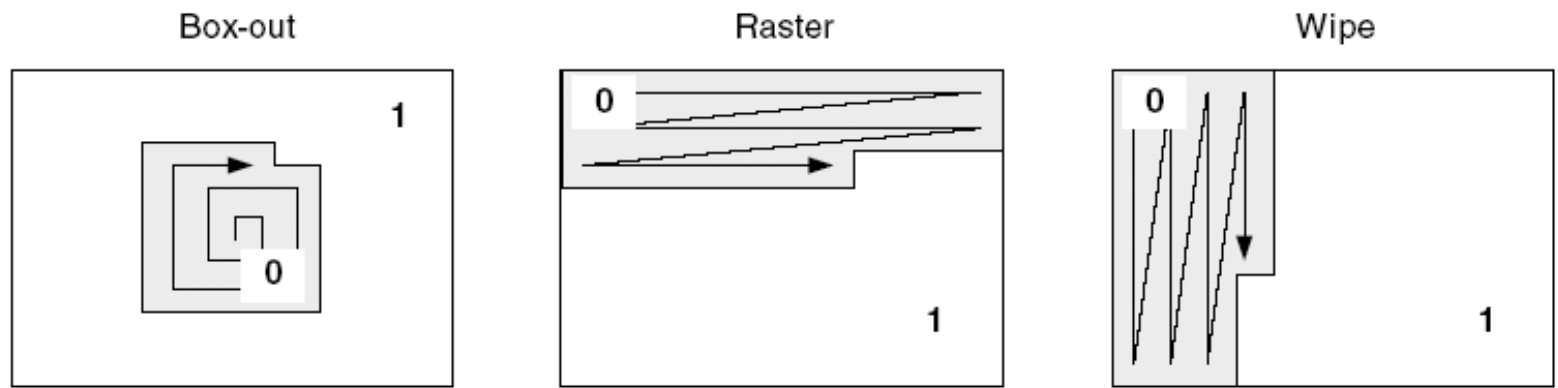


图8 片组

H.264/AVC编解码器

H.264/AVC的结构

帧内预测

帧间预测

变换与量化

帧内预测

- 在帧内预测模式中，预测块P是基于已编码重建块和当前块形成的。对亮度像素而言，P块用于 4×4 子块或者 16×16 宏块的相关操作。 4×4 亮度子块有9种可选预测模式，独立预测每一个 4×4 亮度子块，适用于带有大量细节的图像编码； 16×16 亮度块有4种预测模式，预测整个 16×16 亮度块，适用于平坦区域图像编码；色度块也有4种预测模式，类似于 16×16 亮度块预测模式。编码器通常选择使P块和编码块之间差异最小的预测模式。

帧内预测

4×4亮度预测模式

- 如图6.14所示，4×4亮度块的上方和左方像素A～M为已编码和重构像素，用作编解码器中的预测参考像素。a～p为待预测像素，利用A～M值和9种模式实现。其中模式2(DC预测)根据A～M中已编码像素预测，而其余模式只有在所需预测像素全部提供才能使用。图6.15箭头表明了每种模式预测方向。对模式3～8，预测像素由A～M加权平均而得。例如，模式4中， $d = \text{round}(B/4 + C/2 + D/4)$ 。

帧内预测

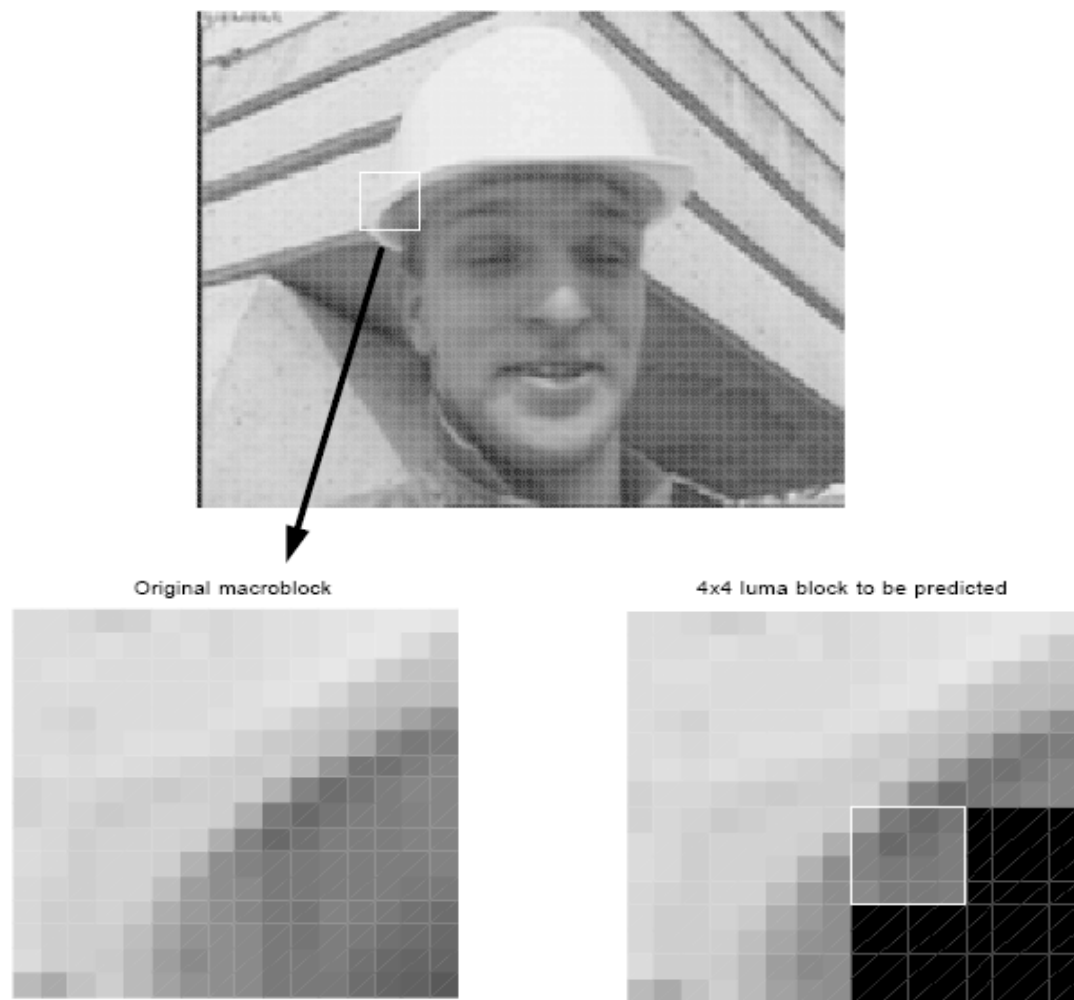
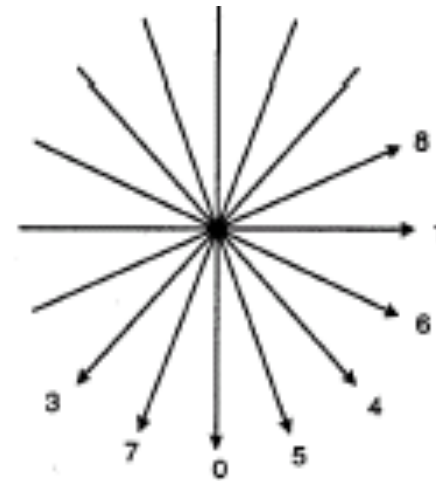


图1 Original macroblock and 4x4 luma block to be predicted

帧内预测

M	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				



a) 利用像素A-M对方块中a-p像素进行帧内4×4预测
b) 帧内4×4 预测的8个预测方向

帧内预测

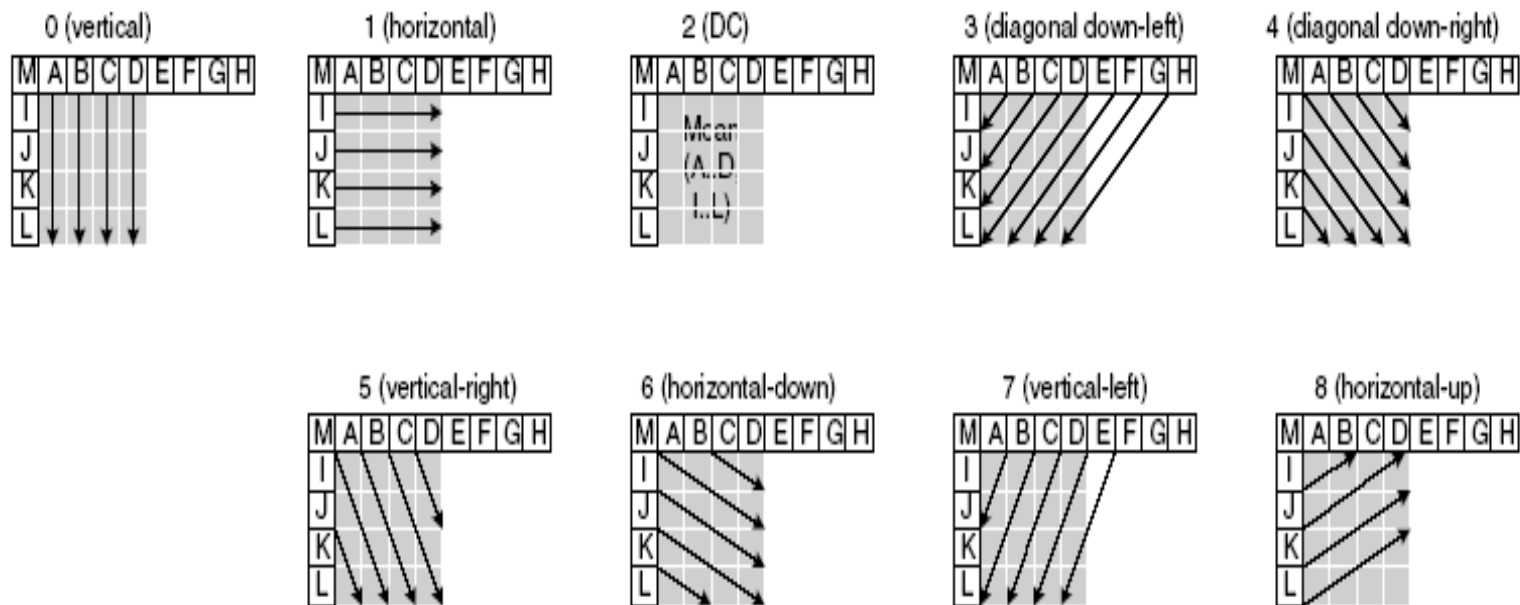


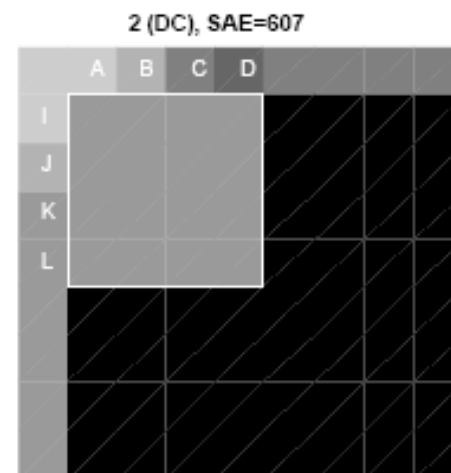
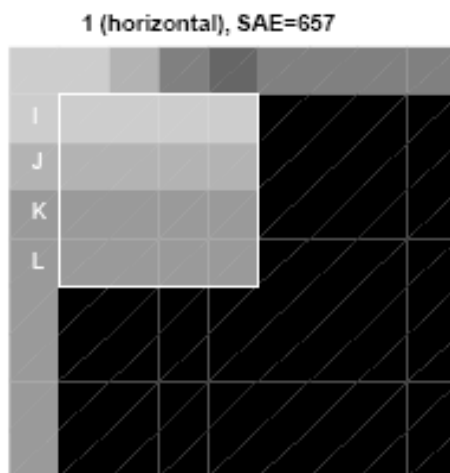
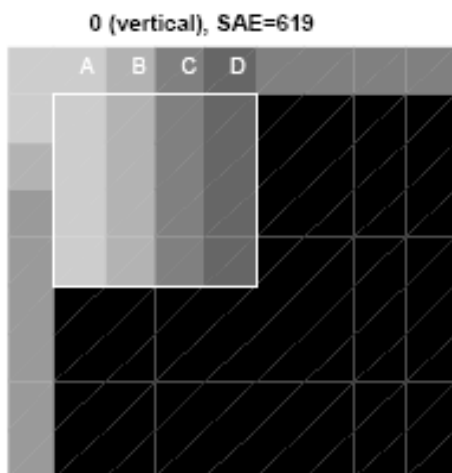
图3 4×4亮度块预测模式

帧内预测

Example:

4x4 块, 9种预测模式 (0-8)

对于当前块的最佳模式是: 模式 7 (vertical-right)



帧内预测

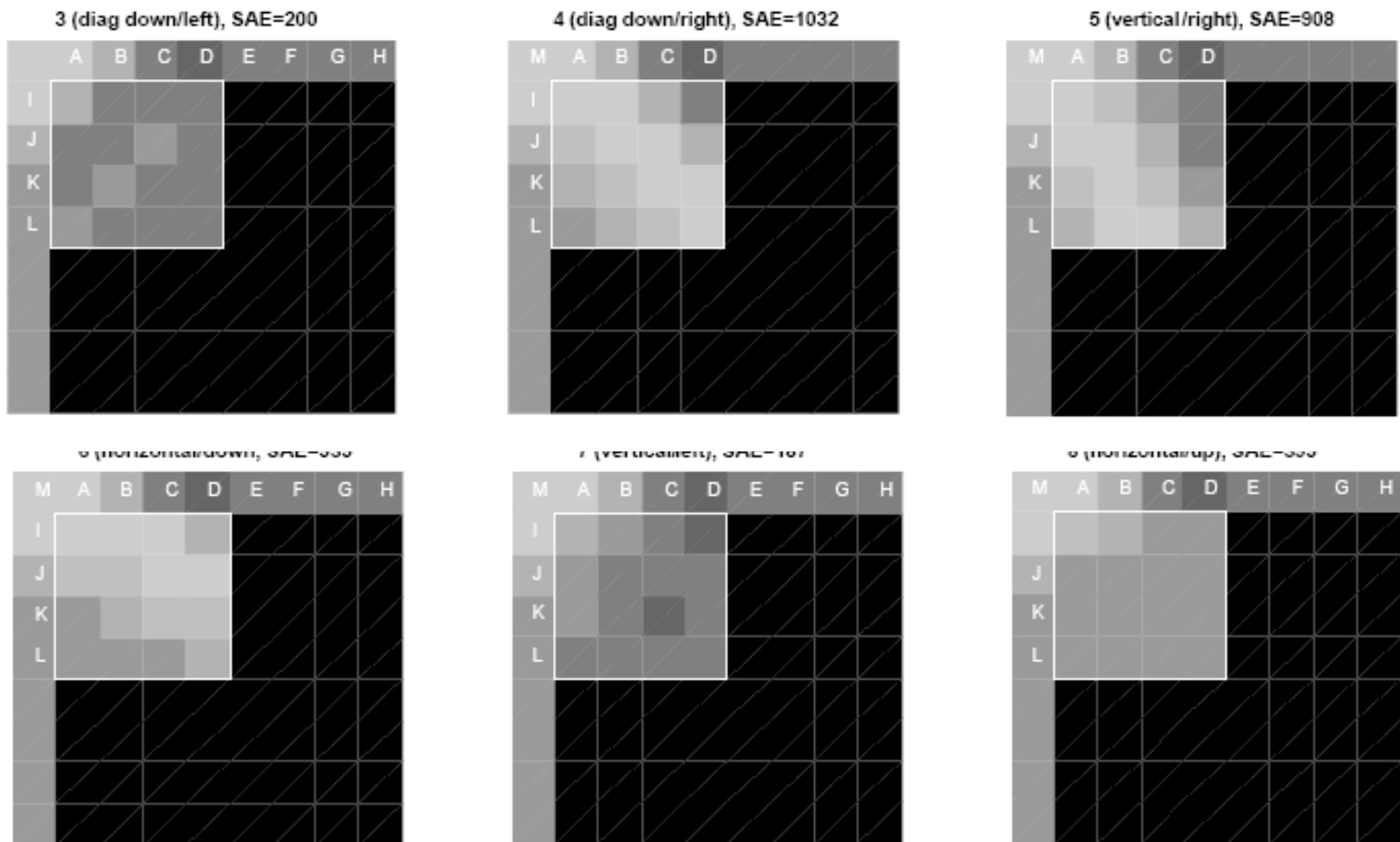


图4

帧内预测

16×16亮度预测模式

- 宏块的全部16×16亮度成分可以整体预测，有4种预测模式，如表2和图5所示。

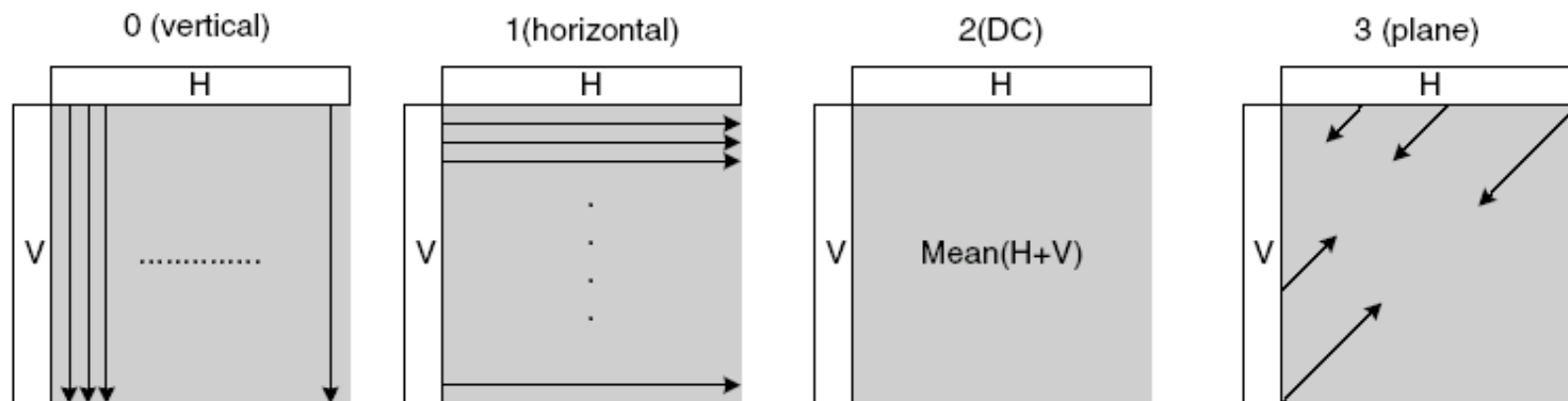


图5 16×16预测模式

帧内预测

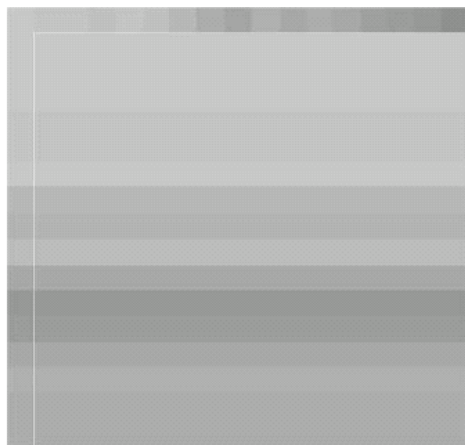
表2 16×16预测模式

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，适用于亮度变化平缓区域

帧内预测



0 (vertical), SAE = 3985



1 (horizontal), SAE = 5097

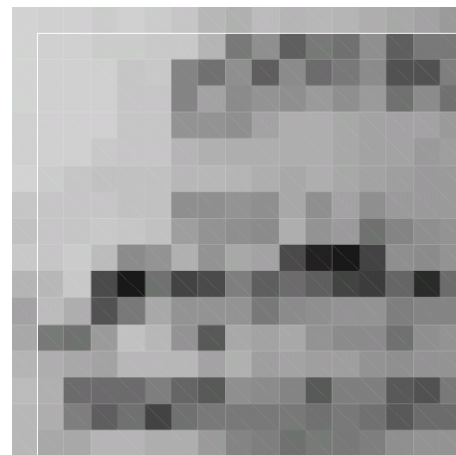
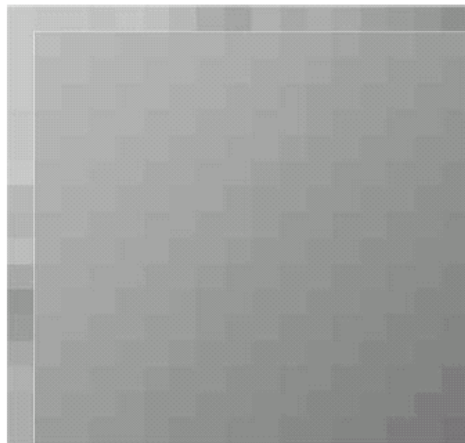


图6 16×16 宏块

举例：图6给出了一个左上方像素已编码的亮度宏块。图7 给出了4种预测模式预测结果。其中模式 3最匹配原始宏块。帧内16×16模式适用于图像平坦区域预测。



2 (DC), SAE = 4991



3 (plane), SAE = 2539

图7 帧内 16×16 预测块

帧内预测

8×8色度块预测模式

- 每个帧内编码宏块的8×8色度成分由已编码左上方色度像素预测而得，两种色度成分常用同一种预测模式。
- 4种预测模式类似于帧内16×16预测的4种预测模式，只是模式编号不同。其中DC（模式0）、水平（模式1）、垂直（模式2）、平面（模式3）。

帧内预测

- 帧内预测模式帧内预测模式编码
- 例如：如果A块和B块的最优预测模式均是预测模式2，那么，最可能当前块C的最优预测模式是模式2

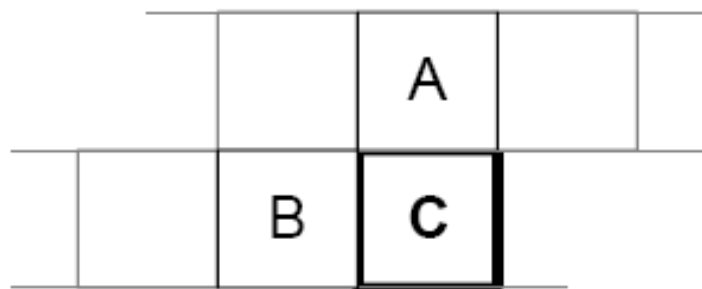


图8 帧内预测模式编码

帧内预测

对于当前块C, 编解码器按照如下方法计算

probable prediction mode =

$\min\{\text{prediction mode of A, prediction modes of B}\}$

当A (或者 B) 的预测模式不可用时,

prediction mode of A = 2.

例如

A 和 B块的预测模式分别为 3 和1

most probable mode for block C = 1

帧内预测

编码器为每个4x4 块发送一个标记 flag,解码器按照如下方式解码

If flag==1, prediction mode=most_probable_mode

If flag==0

 If rem_intra4x4_pred_mode< most_probable_mode

 prediction mode=rem_intra4x4_pred_mode

 else

 prediction mode=rem_intra4x4_pred_mode+1

- 这样表示9中预测模式只需要8个值 (0 to 7)

帧内预测

Choice of prediction mode (most probable
mode = 1)

rem_intra4×4_pred_mode	prediction mode for block C
0	0
1	2
2	3
3	4
4	5
5	6
6	7
7	8

H.264/AVC编解码器

H.264/AVC的结构

帧内预测

帧间预测

变换与量化

树状结构运动补偿

- 每个宏块（ 16×16 像素）可以4种方式分割：一个 16×16 ，两个 16×8 ，两个 8×16 ，四个 8×8 。其运动补偿也相应有四种。而 8×8 分割还可以有四种方式的分割：一个 8×8 ，两个 4×8 或两个 8×4 及四个 4×4 。这种分割下的运动补偿则称为树状结构运动补偿。

帧间预测

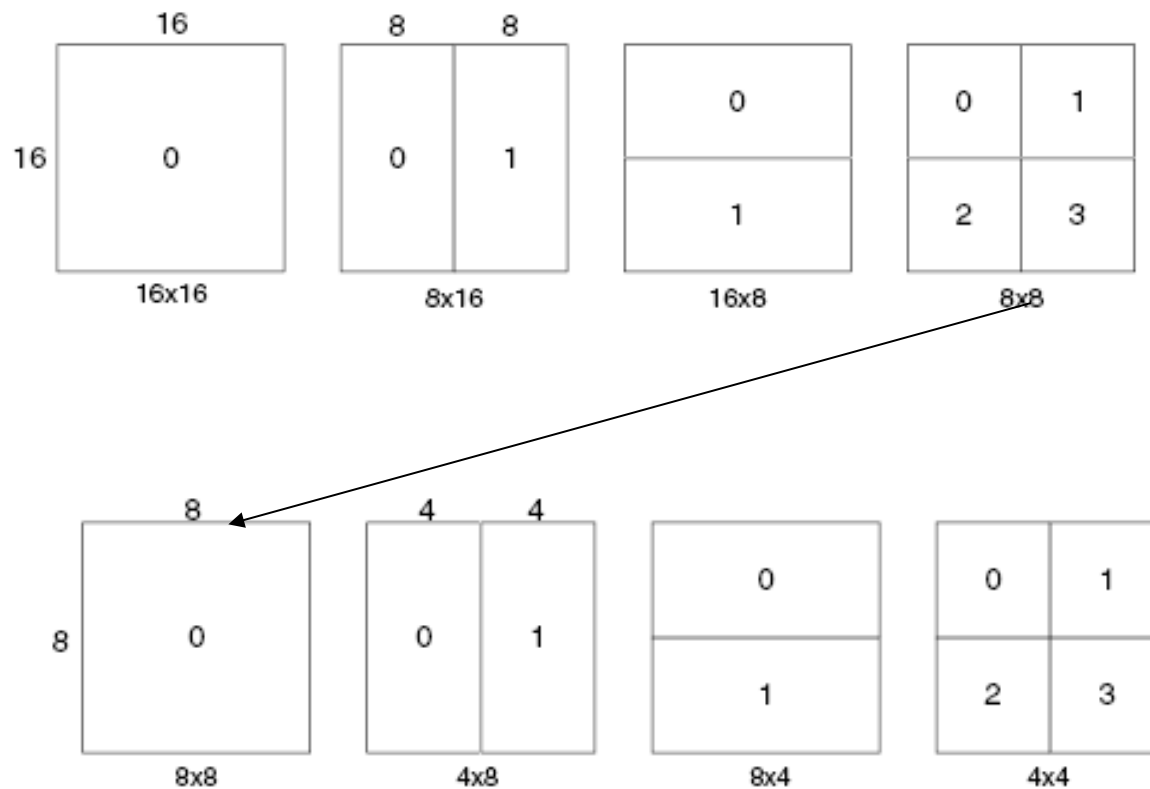


图1 宏块分割

树状结构运动补偿

- 每个分割或子宏块都有一个独立的运动补偿。每个MV必须被编码、传输，分割的选择也需编码到压缩比特流中。对大的分割尺寸而言，MV选择和分割类型只需少量的比特，但运动补偿残差在多细节区域能量将非常高。小尺寸分割运动补偿残差能量低，但需要较多的比特表征MV和分割选择。分割尺寸的选择影响了压缩性能。整体而言，大的分割尺寸适合平坦区域，而小尺寸适合多细节区域。

帧间预测

- 宏块的色度成分（Cr和Cb）则为相应亮度的一半（水平和垂直各一半）。色度块采用和亮度块同样的分割模式，只是尺寸减半（水平和垂直方向都减半）。例如， 8×16 的亮度块相应色度块尺寸为 4×8 ， 8×4 亮度块相应色度块尺寸为 4×2 等等。色度块的MV也是通过相应亮度MV水平和垂直分量减半而得。
- **举例：**如图：一个残差帧（没有进行运动补偿）。H.264编码器为帧的每个部分选择了最佳分割尺寸，使传输信息量最小，并将选择的分割加到残差帧上。在帧变化小的区域（残差显示灰色），选择 16×16 分割；多运动区域（残差显示黑色或白色），选择更有效的小的尺寸。

帧间预测

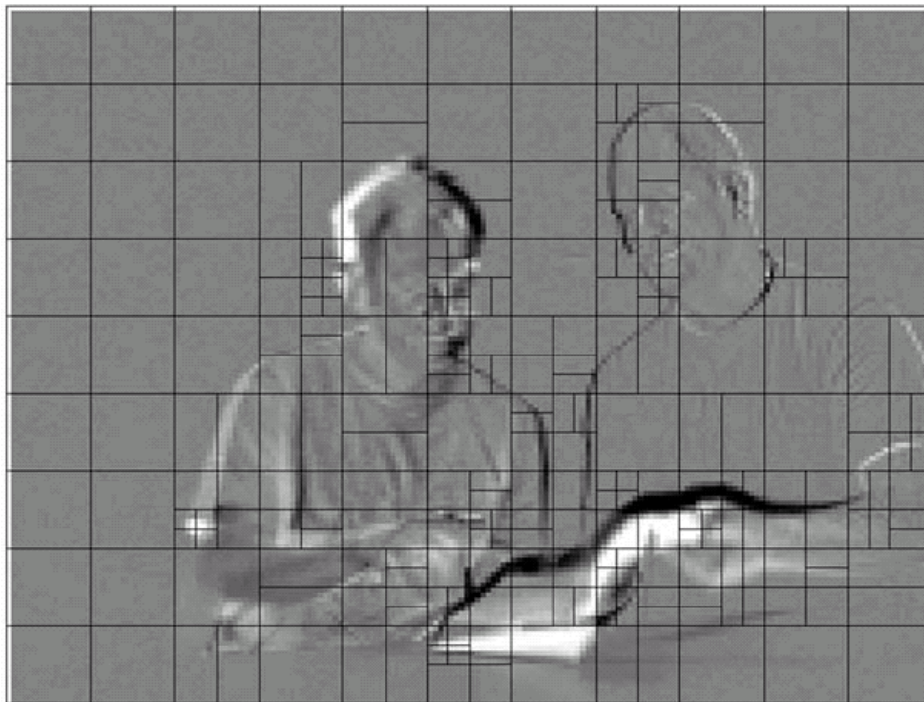


图2 残差帧

运动矢量

- 帧间编码宏块的每个分割或者子宏块都是从参考图像某一相同尺寸区域预测而得。两者之间的差异 (MV) 对亮度成分采用1/4像素精度，色度1/8像素精度。亚像素位置的亮度和色度像素并不存在于参考图像中，需利用邻近已编码点进行内插而得。图中，当前帧的4×4块通过邻近参考图像相应区域预测。如果MV的垂直和水平分量为整数，参考块相应像素实际存在（灰色点）。如果其中一个或两个为分数，预测像素（灰色点）通过参考帧中相应像素（白色点）内插获得。

帧间预测

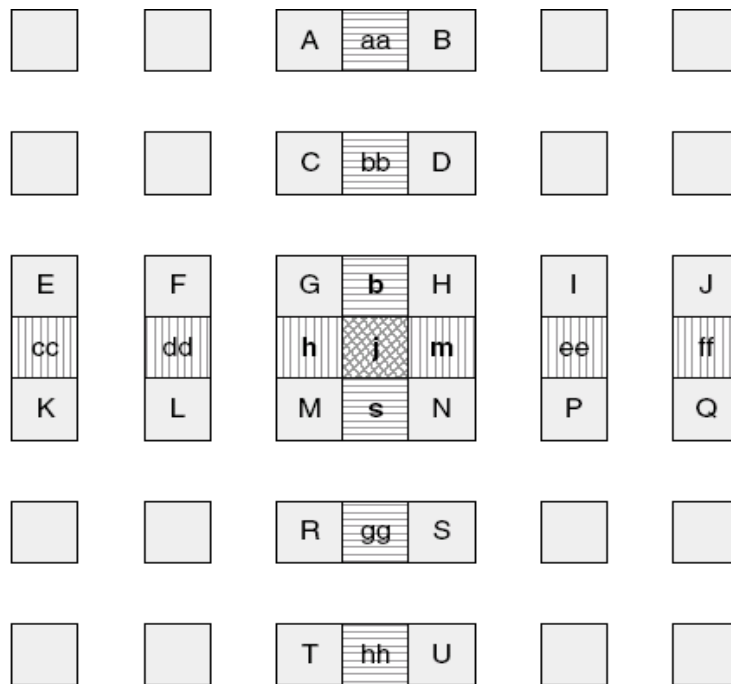


图3 亮度半像素位置内插

$$b = \text{round}((E - 5F + 20G + 20H - 5I + J) / 32)$$

内插像素生成：

- 生成参考图像亮度成分半像素像素。半像素点（如 b,h,m）通过对相应整像素点进行6抽头滤波得出，权重为 $(1/32, -5/32, 5/8, 5/8, -5/32, 1/32)$ 。

$$b = \text{round}((E - 5F + 20G + 20H - 5I + J) / 32)$$

- 类似的，h由A、C、G、M、R、T滤波得出。一旦邻近（垂直或水平方向）半像素点的所有像素都计算出，剩余的半像素点便可以通过对6个垂直或水平方向的半像素点滤波而得。例如，j由cc, dd, h,m,ee,ff滤波得出。这里说明的是，6抽头滤波器比较复杂，但可明显改善运动补偿性能。

帧间预测

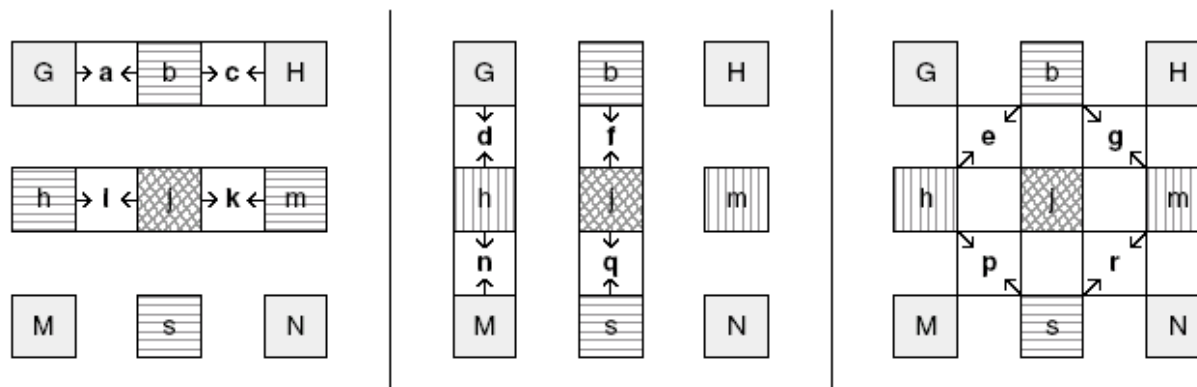


图4 亮度1/4像素内插

半像素点计算出来以后，1/4像素点就可通过线性内插得出，如图4所示。

1/4像素点（如a, c, i, k, d, f, n, q）由邻近像素内插而得，如

$$a = \text{round}((G + b) / 2)$$

剩余1/4像素点（p, r）由一对对角半像素点线性内插得出。如，e由b和h获得。

帧间预测

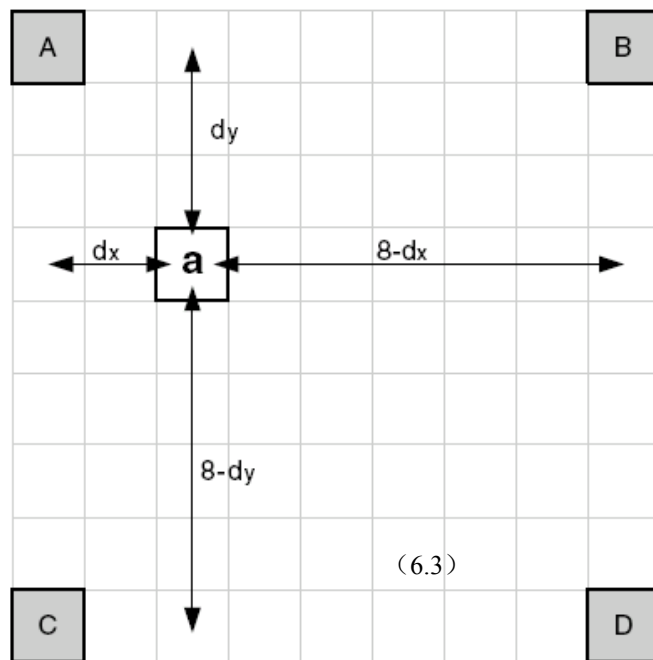


图5 色度1/8像素内插

相应地，色度像素需要1/8精度的MV，
也同样通过整像素地线性内插得出，如图5所示。

$$a = \text{round}([(8 - d_x) \cdot (8 - d_y)A + d_x \cdot (8 - d_y)B + (8 - d_x) \cdot d_y C + d_x \cdot d_y D]/64)$$

MV预测

- 每个分割MV的编码需要相当数目的比特，特别是使用小分割尺寸时。为减少传输比特数，可利用邻近分割的MV较强的相关性，MV可由邻近已编码分割的MV预测而得。

B片预测

- B片中的帧间编码宏块的每个子块都是由一个或两个参考图像预测而得。该参考图像在当前图像的前面或者后面。参考图像存储于编解码器中，其选择有多种方式。图6显示了三种方式：一个前向和一个后向的；两个前向；两个后向。

帧间预测

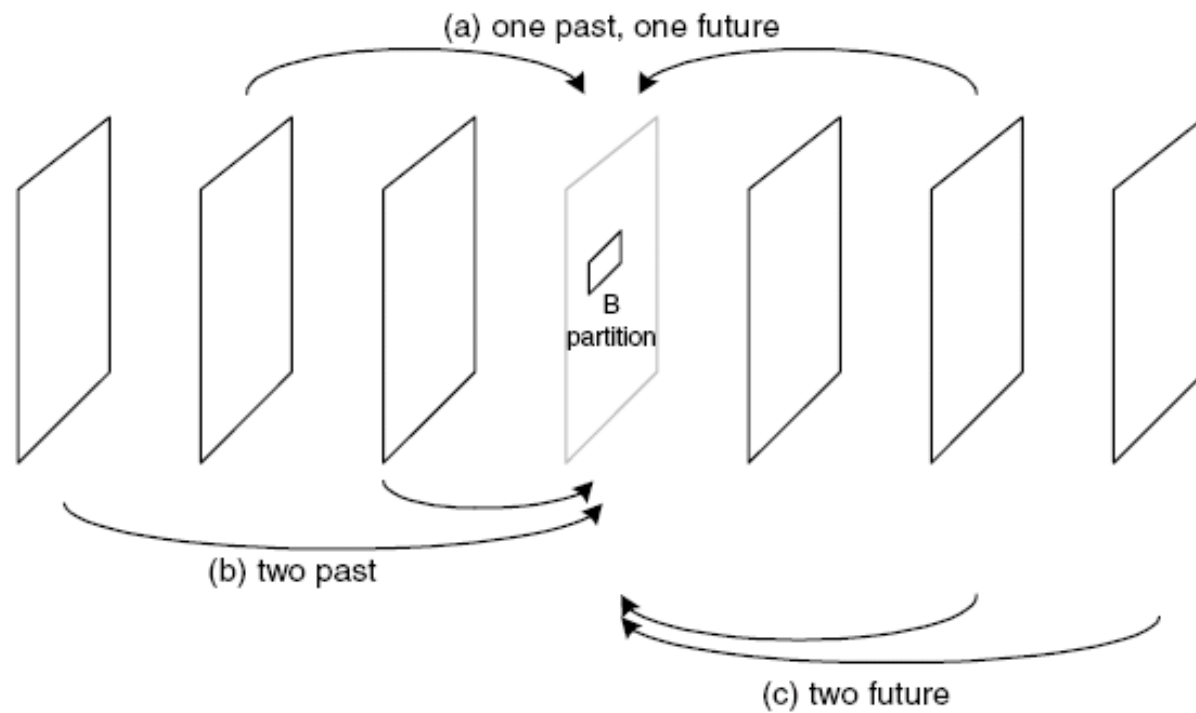


图6. 预测

帧间预测

参考图像

- B片用到了两个已编码图像列表：list0和list1，

帧间预测

- **举例：**一个H.264解码器存储了6幅短期参考图像。其POC分别为：123，125，126，128，129和130。当前图像为127。所有6幅短期参考图像在list0和list1中都标为“作用参考”，如表所示。

表1

Index	List 0	List 1
0	126	128
1	125	129
2	123	130
3	128	126
4	129	125
5	130	123

预测模式选择

- B片的预测方式包括：宏块分割方式、双向选择方式、参考列表选择方式等等。具体说，B片中宏块分割可由多种预测方式中的一种实现，如直接模式、利用list0的运动补偿模式、利用list1的运动补偿模式或者利用list0和list1的双向运动补偿模式。每个分割可选择各自的不同的预测模式（如表2所示）。如果 8×8 分割被使用，每个 8×8 分割所选则的模式适用于分割中的所有亚分割。图7给出了例子，左边的两个 16×8 分割分别使用List0和双向预测模式，而右边的4个 8×8 分割分别采用直接、list0、list1和双向预测四种模式。

帧间预测

表2 B片宏块预测选则

分割	选 择
16×16	直接、list0、list1、双向
16×8/8×16	list0、list1、双向（每个分割独立选择）
8×8	直接、list0、list1、双向（每个分割独立选择）

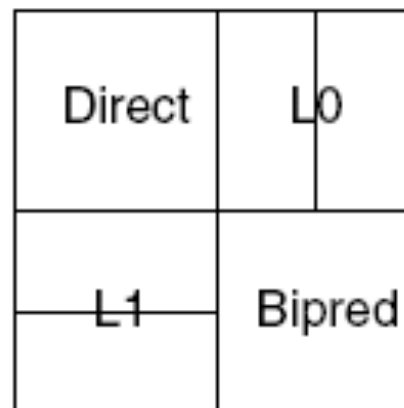
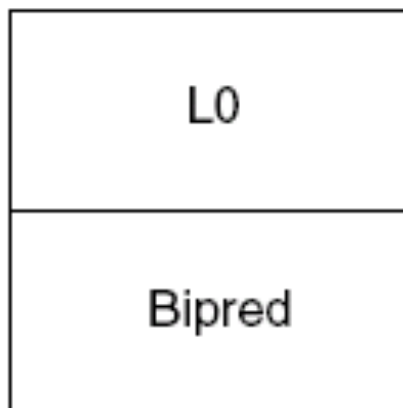


图7 B片中分割预测模式举例

帧间预测

双向预测

- 从list0和list1分别得出两个运动补偿参考区域（需要两个MV），而预测块的像素取list0和list1相应像素的平均值。当不用加权预测时，用下列等式：

$$\text{pred}(i,j) = (\text{pred0}(i,j) + \text{pred1}(i,j) + 1) >> 1$$

帧间预测

举例：

一宏块用B_i Bi 16×16模式预测。图和图6.30分别给出了基于list0和list1参考图像的运动补偿参考区域。图6.31给出了根据者两个参考区域的双向预测。

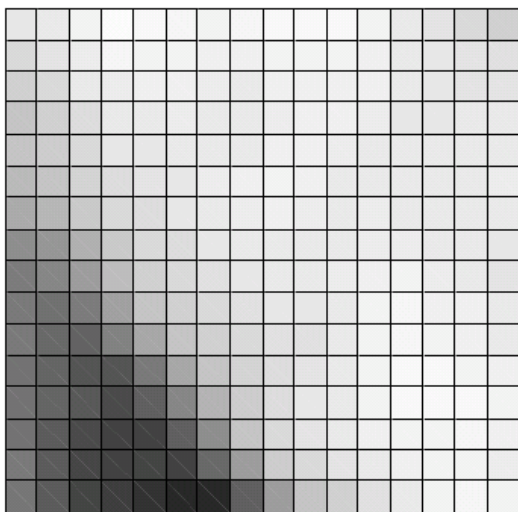


图8 参考区域(list0)

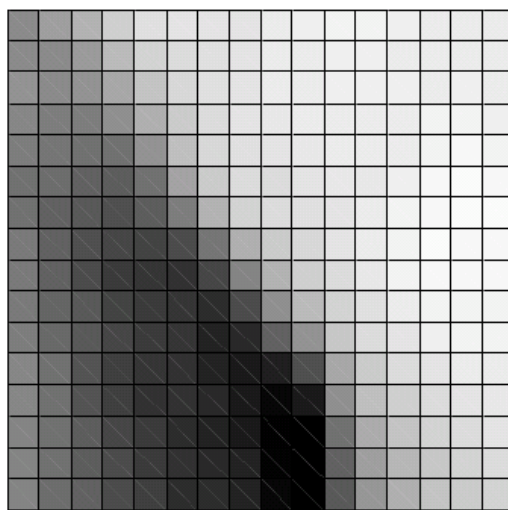


图9 参考区域(list1)

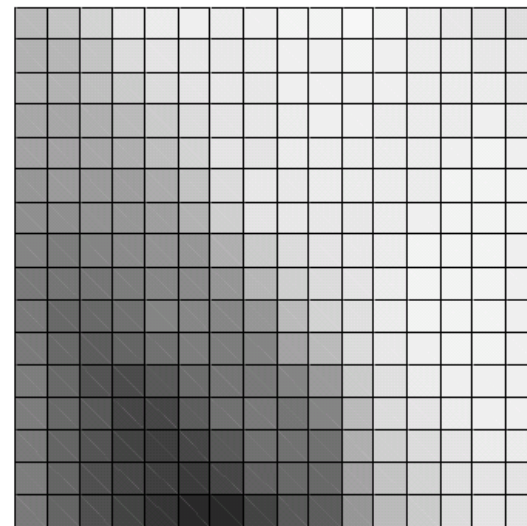
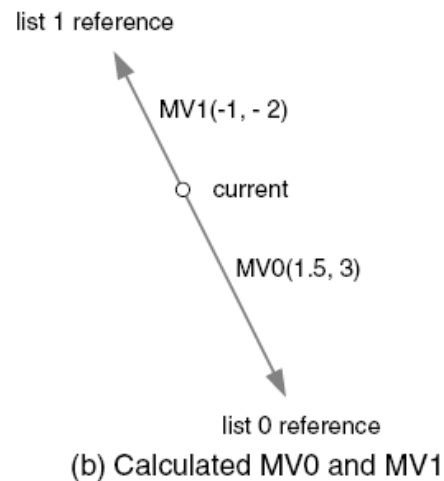
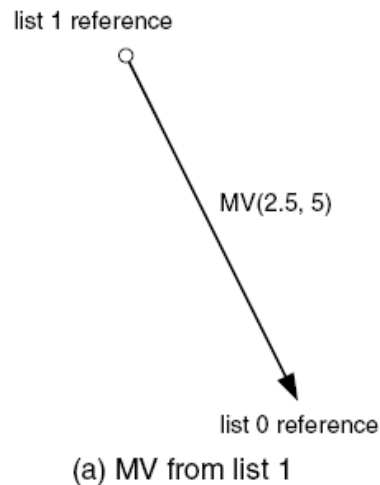


图10 双向预测（无加权）

帧间预测

直接预测

- 直接预测模式编码的B片宏块或宏块分割不传送MV。



加权预测

- 加权预测是一种用来修正P或B片中运动补偿预测像素方法。H.264中有3种加权预测
- 类型：
 - P片宏块“explicit”加权预测；
 - B片宏块“explicit”加权预测；
 - B片宏块“implicit”加权预测；
- 每个预测像素 $\text{pred0}(i,j)$ 和 $\text{pred1}(i,j)$ 在运动补偿之前通过加权系数 ω_0 和 ω_1 修正。在“explicit”类型中，加权系数由编码器决定并在片头中传输。在“implicit”类型中，系数 ω_0 和 ω_1 由相应list0和list1参考图像的时间位置推出。大的系数用于时间上接近当前图像的情况，小的则用于时间上远离当前图像的情况。

H.264/AVC编解码器

H.264/AVC的结构

帧内预测

帧间预测

变换与量化

变换与量化

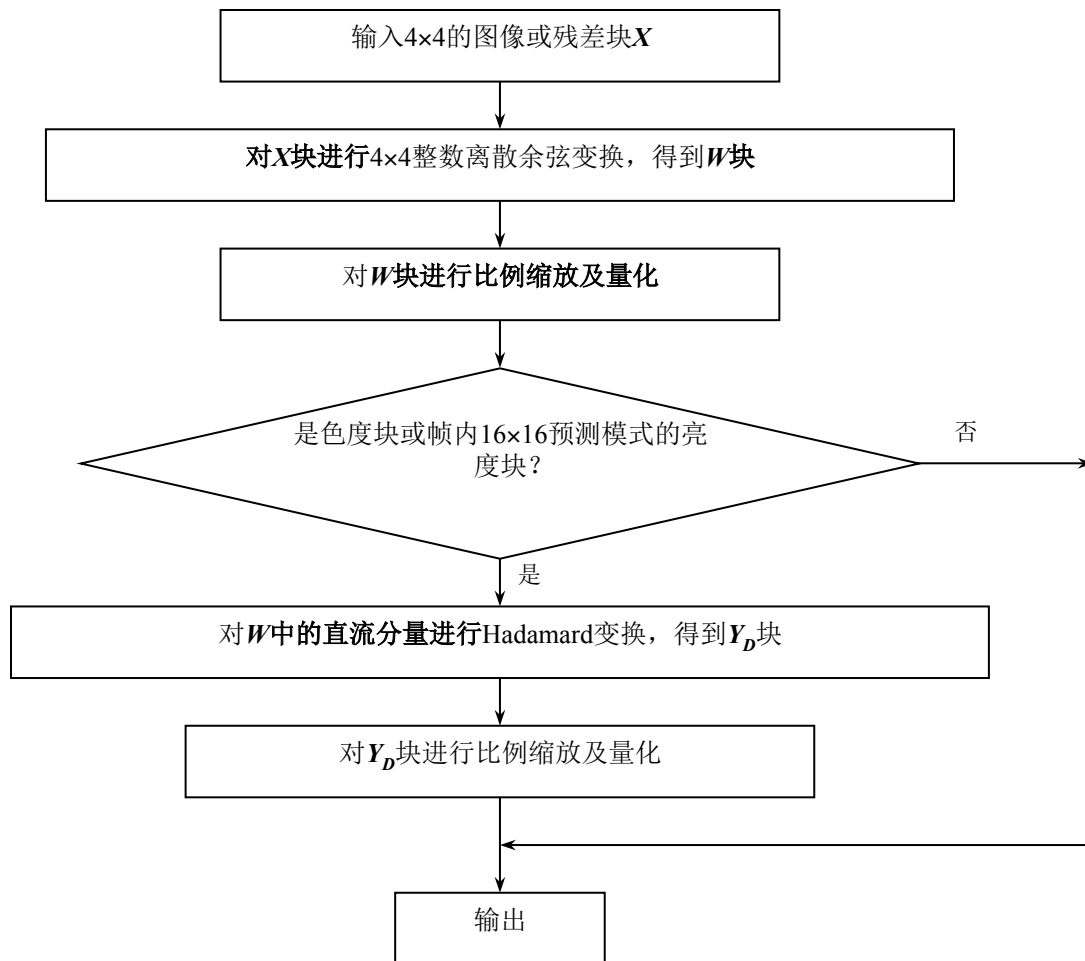


图1 编码器中变换编码及量化过程

变换与量化

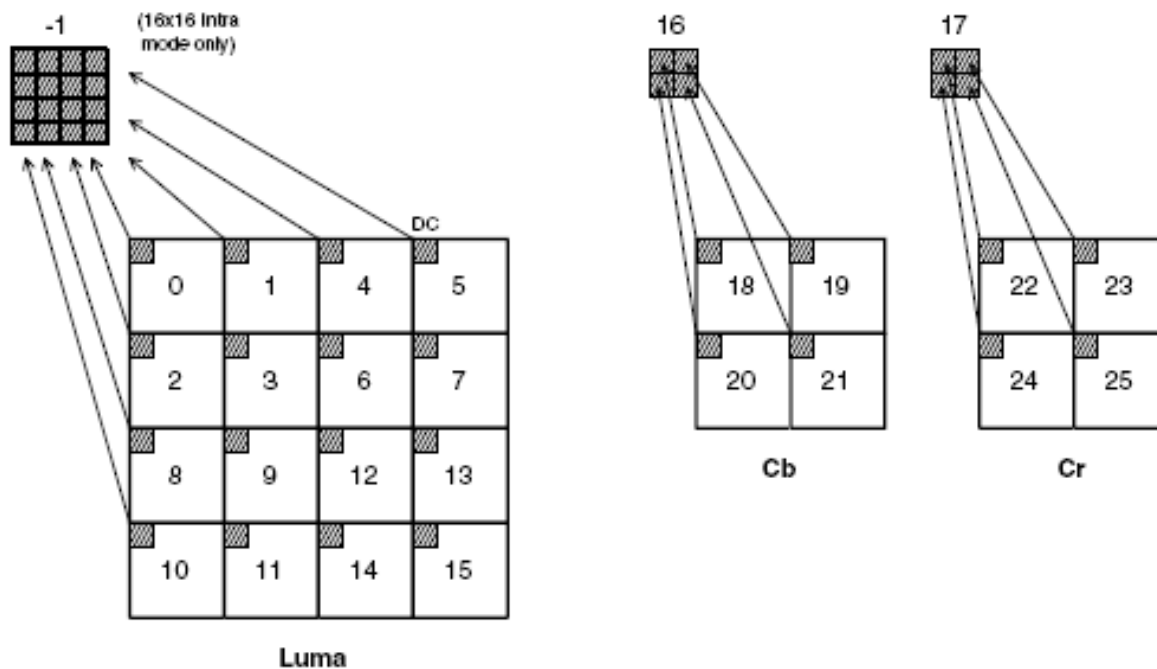


图2 Scanning order of residual blocks within a macroblock

变换与量化

- H.264对图像或预测残差采用了 4×4 整数离散余弦变换DCT技术
- H.264编码器中DCT变换及量化过程如图所示。其中，如果输入块是色度块或帧内 16×16 预测模式的亮度块，则将宏块中各 4×4 块的整数余弦变换的直流分量组合起来再进行Hadamard变换，进一步压缩码率

变换与量化

- 对实数的DCT，由于在解码端的浮点运算精度问题，会造成解码后的数据的失配，进而引起漂移。H.264较其它图像编码使用了更多的预测过程，甚至内部编码模式也依赖于空间预测。因此，H.264对预测漂移是十分敏感的。为此，H.264对4×4DCT 中的A进行了改造，采用整数DCT技术，有效地减少计算量，同时不损失图像准确度

变换与量化

- H.264对4×4的图像块（亮度块或Cr、Cb色度块）进行操作，则相应的4×4 DCT为（变换矩阵A）：

$$Y = AXA^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \end{bmatrix} \begin{bmatrix} X \end{bmatrix} \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \end{bmatrix}$$

$$a = \frac{1}{2}, \quad b = \sqrt{\frac{1}{2}} \cos\left(\frac{\pi}{8}\right), \quad c = \sqrt{\frac{1}{2}} \cos\left(\frac{3\pi}{8}\right)$$

$$Y = (CXC^T) \otimes E = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} \begin{bmatrix} X \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

其中， $d=c/b$ （ ≈ 0.414 ）。

符号“ \otimes ”表示（ CXC^T ）结果中的每个元素乘以矩阵E中对应位置上的系数值的运算。为了简化计算，取d为 0.5。

为了保持变换的正交性，对b进行修正，取

$$b = \sqrt{\frac{2}{5}}$$

变换与量化

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

$$Y = C_f X C_f^T \otimes E_f = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \begin{bmatrix} \\ \\ \\ \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix}$$

变换矩阵整数化处理

H.264将DCT融合到后面的量化过程中

实际的DCT输出为：

$$W = CXC^T$$

变换与量化

- 逆变换

$$\mathbf{X} = \mathbf{C}_i^T (\mathbf{Y} \otimes \mathbf{E}_i) \mathbf{C}_i = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \left(\begin{bmatrix} \mathbf{Y} \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}$$

变换与量化

表 H.264中编解码器的量化步长

QP	Qstep	QP	Qstep	QP	Qstep	QP	Qstep	QP	Qstep
0	0.625	12	2.5	24	10	36	40	48	160
1	0.6875	13	2.75	25	11	37	44	49	176
2	0.8125	14	3.25	26	13	38	52	50	208
3	0.875	15	3.5	27	14	39	56	51	224
4	1	16	4	28	16	40	64		
5	1.125	17	4.5	29	18	41	72		
6	1.25	18	5	30	20	42	80		
7	1.375	19	5.5	31	22	43	88		
8	1.625	20	6.5	32	26	44	104		
9	1.75	21	7	33	28	45	112		
10	2	22	8	34	32	46	128		
11	2.25	23	9	35	36	47	144		

1)QP是量化参数，
是量化步长Qstep的
序号；

2)当QP取最小值0
时,代表最精细的量
化，当QP取最大值
51时,代表最粗糙
的量化。

3) Q P 每 增 加 6,
Qstep
增加一倍。

变换与量化

- W_{ij} 是矩阵 W 中的转换系数, a_j 是矩阵 A 中的元素,

$$Z_{ij} = \text{round} \left(W_{ij} \frac{PF}{Qstep} \right)$$

Position	PF
(0,0), (2,0), (0,2) or (2,2)	a^2
(1,1), (1,3), (3,1) or (3,3)	$b^2/4$
Other	$ab/2$

- 利用量化步长随量化参数每增加6而增加一倍的性质, 可以进一步简化计算。设

$$qbits = 15 + \text{floor}(QP/6)$$

$$MF = \frac{PF}{Qstep} 2^{qbits}$$

$$Z_{ij} = \text{round} \left(W_{ij} \frac{MF}{2^{qbits}} \right)$$

变换与量化

- MF可以取整数，量化表只列出对应QP值为0到5的MF值。
- 对于QP值大于5的情况，只是qbits值随QP值每增加6而增加1，而对应的MF值不变。
- 量化过程则为整数运算，并且可以避免使用除法，并且确保用16位算法来处理数据，在基本保持PSNR性能的情况下，实现最小的运算复杂度。

变换与量化

Example:

QP = 4, hence Qstep = 1.0

(i,j) = (0,0), hence PF = $a^2 = 0.25$

qbits = 15, hence $2^{qbits} = 32768$

$$\frac{MF}{2^{qbits}} = \frac{PF}{Qstep}, \text{ hence } MF = (32768 \times 0.25) / 1 = 8192$$

Multiplication factor MF

QP	Positions (0,0),(2,0),(2,2),(0,2)	Positions (1,1),(1,3),(3,1),(3,3)	Other positions
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

变换与量化

DCT直流系数的变换量化

16×16的图像宏块中有4×4个4×4图像亮度块，所以亮度块的 W_D 为4×4矩阵，其组成元素为各图像块DCT的直流系数 W_{00} ，这些 W_{00} 在 W_D 中的排列顺序为对应图像块在宏块的位置。

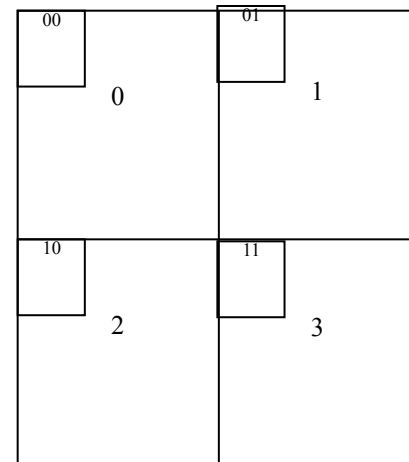
对亮度块 W_D 的Hadamard变换为：

$$Y_D = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} W_D \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \right) / 2$$

⁰⁰ 0	⁰¹ 1	⁰² 4	⁰³ 5
¹⁰ 2	¹¹ 3	¹² 6	¹³ 7
²⁰ 8	²¹ 9	²² 12	²³ 13
³⁰ 10	³¹ 11	³² 14	³³ 15

变换与量化

16×16的图像宏块中包含图像色度Cr及Cb块各2×2个，所以色度Cr或Cb块的 W_D 为2×2矩阵，其组成元素为各对应图像块色度信号DCT的直流系数 W_{00} ，这些 W_{00} 在WD中的排列顺序为对应图像块在宏块的位置



对各色度块 W_D 的Hadamard变换为：

$$Y_D = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} W_D \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$