

## 2.4 H.264 视频压缩编码标准

### 2.4.1 H.264 标准简介

H.264 是继 MPEG 和 H.263 后由 ISO/IEC 与 ITU-T 联合制定的新一代视频压缩编码标准。1996 年制定 H.263 标准后, ITU-T 的视频编码专家组 (VCEG) 开始了两个方面的研究: 一个是在 H.263 标准的基础上增加选项, 随之产生了 H.263<sup>+</sup>和 H.263<sup>++</sup>; 另一个是制定一种支持低码率视频通信的新标准, 产生了压缩效率有明显优势的 H.26L 标准草案。随后的几年里 ISO/IEC 的 MPEG 组织也认识到 H.26L 的潜在优势, 2001 年 ISO/IEC MPEG 与 ITU-T VCEG 组建了联合视频组 (JVT), 共同将 H.26L 标准草案发展成为一个国际性标准。2003 年 5 月, JVT 组织推出了 H.264 标准。这个标准在 ITU-T 中命名为 H.264 标准; 而在 ISO/IEC 中命名为 AVC (Advanced Video Coding), 作为 MPEG-4 的第 10 部分。该标准常被称为 H.264/AVC。

H.264/AVC 以其高效的压缩编码性能和良好的网络适应性, 在数字电视广播、视频通信、网络流媒体、视频监控等领域得到了广泛应用。H.264/AVC 技术具有四个方面的特点:

(1)、压缩编码效率高: 在保证图像质量相同的情况下, 采用 H.264/AVC 技术压缩后的码率只有 MPEG-2 的一半。

(2)、容错能力强: H.264 码流有较强的抗误码特性, 能适应干扰严重、丢包率高的无线信道和 IP 网络应用。

(3)、网络适应性强: H.264/AVC 提供了网络适配层, 使得 H.264 码流能够通过不同网络进行传输。

(4)、计算复杂度高: H.264/AVC 的优越性能是通过高复杂度的计算得到的。H.264/AVC 的计算复杂度大约相当于 MPEG-2 的 2 倍~3 倍, 相对来说 H.264/AVC 编码器的实现难度有所增加。

随着数字集成电路技术和工艺技术的迅猛发展, 目前在数字信号处理芯片上完全可以实现 H.264/AVC 编码器。

### 2.4.2 H.264/AVC 的编解码架构

与 MPEG-2 标准类似, H.264/AVC 也不明确规定如何实现编解码器, 只规定了已编码视频比特流的句法以及解码该比特流的方法。在这一框架下, 各个厂商可以用灵活的方式实现各自的编码器和解码器, 同时能保证不同厂商的编码器和解码器的互连互通。这样做可以形成相互竞争的专业产品市场, 有效促进技术进步, 保证产品价格的合理性。

从编、解码器的框架来看, H.264/AVC 与之前的标准 (如 H.261、H.263、MPEG-1、MPEG-2、MPEG-4) 中的编、解码器功能模块的组成没有什么区别, 都是基于混合编码框架, 包括预测、变换、量化和熵编码等基本功能模块。具体差别主要体现在各功能模块的实现细节方面。

## 1、H.264/AVC 编码器框架

H.264/AVC 编码器采用基于变换和预测的混合编码框架。

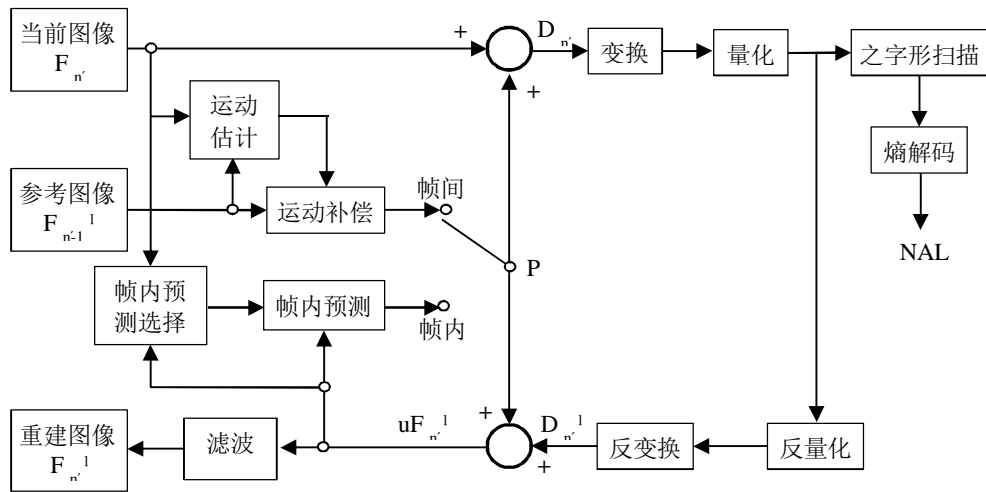


图 2-4-1 H.264 编码框图

在图 2-4-1 中，编码器以宏块为单位对输入的帧或场  $F_n$  进行处理。通常按帧内或帧间预测编码的方法进行处理。

采用帧内预测编码时，其预测值  $P$  (PRED) 是由当前片 (Slice) 中已编码的参考图像经运动补偿 (MC) 后得到的，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，实际的参考图像可以选择已经编码重建和滤波的帧 (显示次序上的过去或未来帧)。

预测值  $P$  (PRED) 和当前块相减后，产生一个残差块  $D_n$ ，经过变换和量化后产生一组变换量化后的变换系数  $X$ ，再经过熵编码，与解码所需的预测模式、量化参数、运动矢量等信息一起复用成一个压缩后的码流，最后经过网络适配层处理后形成适合传输和存储用的数据流。

为了进行帧间预测编码，需要提供预测用的参考图像，编码框架中包含了重建图像的环路。各残差图像变换系数块  $X_n$  经过反量化和反变换得到的  $D_{n-1}$ ，与预测值  $P$  相加后，可恢复出重建未滤波的图像帧 ( $uF_{n-1}$ )，为了提高参考帧的图像质量，在编码框架中设置了一个环路滤波器，去除编解码环路中产生的噪声，滤波之后输出的图像  $F_{n-1}$  可用作参考图像。

H.264/AVC 的帧间预测编码过程与 MPEG-2 类似，这里不再重复。

## 2、H.264/AVC 解码器框架

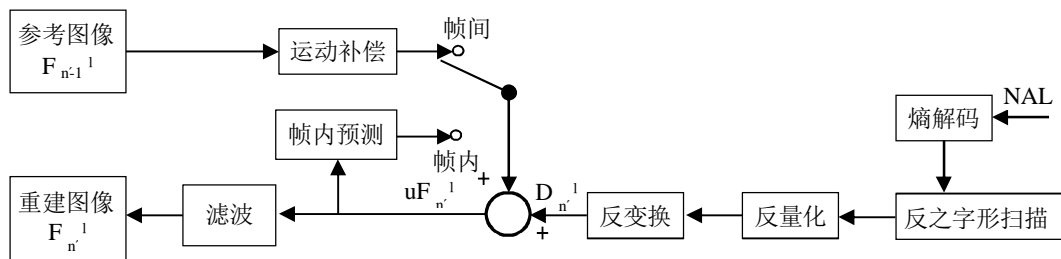


图 2-4-2 H.264 解码框图

解码器处理的是由编码器网络适配层(NAL)输出的 H.264 压缩比特流。在图 2-4-2 中, H.264 压缩比特流熵解码得到量化后的一个图像块的变换系数  $X$ , 再经过反量化、反变换, 得到残差  $D_n^{-1}$ 。利用从该比特流中解出的解码附加信息, 解码器可以恢复出图像块的预测值  $P$  (PRED), 它与编码器中的预测值是相同的。预测值  $P$  与残差  $D_n^{-1}$  可以重建出图像块, 进而得到未滤波的重建图像帧 ( $\mu F_n^{-1}$ ), 经过滤波后就得到重建的图像  $F_n^{-1}$ , 也就是最后的解码输出图像。

### 2. 4. 3 H. 264/AVC 的概念

#### 1. H.264 的型(Profile)和级(Level)

H.264 在第一版标准中规定了三种基本的编码工具集, 每一类支持一种特定的编码功能, 并支持一类特定的应用。这三个型分别如图 2-4-3 所示。

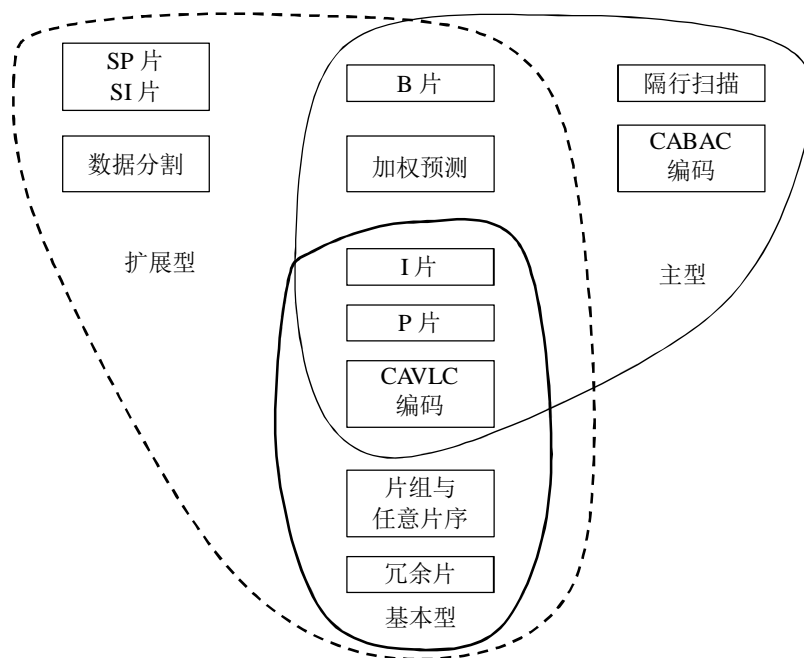


图 2-4-3 H.264 标准第一版定义的三个型

#### (1)、基本型 (Baseline Profile)

这是 H.264 标准规定的最简单的型, 采用 I 片和 P 片, 支持帧内和帧间编码, 采用基于上下文的自适应可变长编码。广泛应用在可视电话、电视会议、无线通信等实时视频通信领域。

#### (2)、主型 (Main Profile)

在基本型部分主要编码工具的基础上, 主型支持隔行扫描, 采用基于 B 片的帧间预测和加权预测的帧内编码, 支持基于上下文的自适应算术编码。主要应用在数字电视广播和数字存储领域。

#### (3)、扩展型 (Extended Profile)

在基本型和主型编码工具的基础上，扩展类支持码流之间的切换，利用数据分割技术改进抵抗误码性能，但不支持隔行扫描视频和基于上下文的自适应算术编码。主要应用在网络流媒体。图 2-4-3 给出了 H.264 的三个型，每个型具有不同的功能。扩展型包括了基本型的所有功能，但不包括主型的全部功能。每一型可以设置不同的取样率、图像尺寸、编码比特率等参数，从而得到对应编码器性能的不同级。

2. H.264/AVC 语法有关的术语

(1)、片 (Slice)

在 H.264 中一帧视频图像可划分成一个或多个片进行编码，每个片包含整数数量的宏块 (MB)，每片至少有一个宏块，最多包含整个图像的全部宏块。一帧图像中的各个片所包含宏块的数量并不固定。

对图像帧进行分片的主要目的是为了限制预测编码引起的误码扩散和传播。由于编码片之间相互保持独立，任何一个片的预测都不以其他片中的宏块作为参考，这样一个片中因传输引起的误码就不会传播到后面其它的片中。

编码片共有 5 种不同的类型，它们分别是 I 片、P 片、B 片、SP (Switch P) 和 SI (Switch I) 片。I 片，称为片内编码片，只包含 I 宏块，只利用当前片中已解码的像素作为参考图像进行帧内预测；P 片，称为前向预测片，可包含 I 宏块和 P 宏块，P 宏块则利用前面已编码图像作为参考图像进行预测；B 片，称为双向预测片，可包含 I 宏块和 B 宏块，B 宏块则利用当前和未来已编码图像作为参考图像进行预测；SP 片，称为切换 P 片，它包含 P 和/或 I 宏块，支持流媒体应用，用于码流之间切换；SI 片，称为切换 I 片，只包含 I 宏块，也用于码流之间切换。

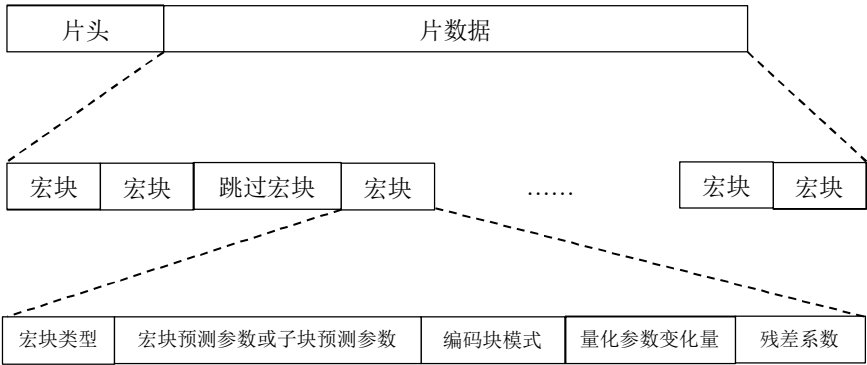


图 2-4-4 H.264 片的语法结构

在 H.264 定义的码流中，序列、图像、片、宏块和子宏块被组织成有层次的语法结构。其中，片的语法结构是码流中一个比较重要的部分，如图 2-4-4 所示，一个片包括片头和片数据。片头含有片的类型、该片属于哪个图像以及有关参考图像等信息；片数据包含一系列编码宏块和跳过宏块，跳过宏块不携带任何编码数据；片中的每个宏块含有宏块类型 (mb-type)、宏块预测参数 (mb-pred) 或子块预测参数 (sub-mb-pred)、编码块模式 (coded-block-pattern)、量化参数变化量 (mb-qp-delta) 以及残差系数 (residual) 等信息。其中，宏块类型用于传送宏块的编码模式 (帧内或帧间) 信息，宏块预测参数用于传送帧内

预测模式、参考图像以及差分编码的运动矢量等信息，编码块模式用来指明包含编码变换系数的编码块。

(2)、片组（Slice Group）

片组是一个编码图像中若干个 MB 组成的集合，可以包含一个或若干个片。H.264 为了提供不同的信道保护方案，在解码端更好地利用图像纹理特征进行错误隐藏，达到更好的主观图像质量。在图像分片之前先将图像映射成片组（Slice Group），即把宏块或者宏块对映射到不同的片组，如图 2-4-5a 所示的 Slice Group0、Slice Group1、Slice Group2 等。

在片组中再进行分成片（Slice），片中包含片组内部按照光栅扫描顺序排列的整数个宏块或宏块对。由于经过了图像到片组的映射，在同一片组中相邻的宏块在实际图像中不一定是相邻的，如图 2-4-5b 所示 Slice Group0 中的宏块分散在图像中，当 Slice Group0 内的片的宏块发生丢包等错误的时候，就可以利用 Slice Group1、2、3 内的 Slice 的宏块对其进行预测恢复或替代，因为这些片组中的宏块或宏块对与该丢失宏块或宏块对在空间位置上是相邻的，相关性最强，预测误差最小，从而可以取得很好的错误恢复效果。

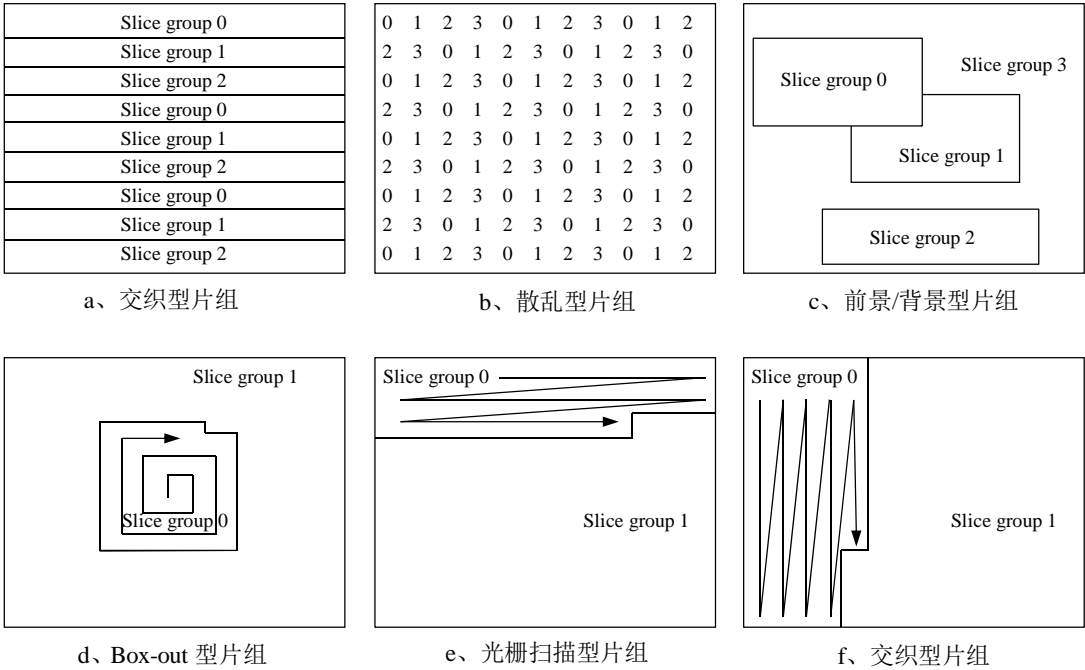


图 2-4-5、片组映射类型

在一个片组内部，每个片的宏块按照光栅扫描次序进行编码，如果每帧图像只取成一个片组，则该图像中的所有宏块均按照光栅扫描次序编码（除了使用任意片次序 Arbitrary Slice Ordering）。

在 H264 标准中还定义了灵活宏块次序（Flexible Macroblock Ordering，简称 FMO）的宏块到片组的映射模式，共有七种映射模式，如图 2-4-5 所示，各个 FMO 映射类型可以表示出每一个宏块属于哪个片组。

交织型片组（Interleaved），不同的片组交替出现，如果超过 2 个片组，按照图 2-4-5a 所示的形式交替出现，最多可以有 8 个片组。采用交织型片组可以避免在丢包环境中大块区

域图像信息的整体丢失。

散乱型片组(Dispersed), 空间相邻的宏块在片映射时被完全分开, 如图 2-4-5b 所示, 散乱型片组可以认为是交织型片组的一种拓展, 按照光栅扫描的顺序进行映射, 相同序号的宏块属于同一片组。这种模式由于其空间位置上的分散性, 数据压缩率有所下降。

前景/背景型片组 (Foreground/Background), 主要区分视频序列中的前景与背景区域, 通过设置开始和结束宏块位置 (按光栅扫描顺序) 来确定前景矩形区域的大小, 如图 2-4-5c 所示, 标准中允许不同片组的区域交叠, 交叠区域的宏块属于序号较小的片组。通过多个片组的划分, 可以将图像分为目标区域和背景区域。由于根据图像内容进行片组划分, 可以把相对静止的背景和运动主体分离, 对重要性不同的区域信息采用分级差错保护方案。

Box-out 型片组, 它将不同的片组按螺旋状嵌套排列, 编码时从中心位置的宏块开始, 如图 2-4-5d 所示按照顺时针 (或逆时针, 可通过参数设定) 方向螺旋向外进行, 中心位置的宏块属于序号较小的片组, 可通过参数设定每个片组包含的宏块数目。

光栅扫描型片组(Raster Scan), 只支持两个片组可进行正方向扫描 (左上角起始) 或反方向扫描 (右下角起始)。

划像型片组(Wipe), 类似于光栅扫描型片组, 只支持两个片组, 可从左边擦入或从右边擦入。

### (3)、SP/SI 的应用

H.264 标准主要包括 3 种类型帧: I 帧、P 帧和 B 帧。H.264/AVC 为了顺应视频流的带宽自适应性和抗误码性能的要求, 又定义了两种新的帧类型: SP 帧和 SI 帧。

SP 帧编码的基本原理同 P 帧类似, 基于帧间预测的运动补偿预测编码, 两者之间的差异在于 SP 帧能够参照不同的参考帧重构出相同的图像帧。SI 帧则是基于帧内预测的编码技术, 重构图像的方法与 SP 帧相同。SP/SI 帧技术有效满足了网络传输条件下的视频流切换要求, 大大改善了 H.264 的网络亲和性, 支持流媒体服务应用, 具有很强的抗误码性能, 广泛应用于流间切换、拼接、随机进入、快进快退以及错误恢复等。

#### ①SP 帧应用于码流间切换

在网络应用环境下, 视频服务器需要根据网络带宽变化调整编码码率, 实现带宽的自适应性, 这类应用包括流业务和会话业务。在流业务应用场合中, 通常进行非实时编码, 需要根据不同的质量和带宽要求, 预先设置多种不同的编码参数对同一视频序列分别进行编码, 从而生成适应不同质量和带宽要求的多组不同码率的码流。服务器只需在不同码率的码流之间切换, 以适应网络有效带宽的不断变化。

在 H.264 之前的视频压缩标准中, 主要通过码流中周期性地放置 I 帧实现码流间的切换, 但 I 帧的周期性插入会造成视频流数据量的增大, 增加了对传输带宽的要求。如果为了降低码率, 不放置 I 帧, 那么在切换点后的 P 帧会因为参考帧的改变而无法正确解码。为了解决这个矛盾, H.264 在视频流的切换点产生相应的 SP 帧, 如图 2-4-6 所示, 要从码流 A 的  $P_{A1}$  切换到  $P_{B2}$ , 该 SP 帧以 A 路视频序列  $P_{A1}$  为参考帧, 采用与 B 码流相同的编码参数

对 B 路视频序列  $P_{B2}$  进行帧间预测编码，得到相应的  $SP_{AB2}$ 。这样，在不同编码参数但内容相同的视频码流间进行切换时，发送切换点对应的 SP 帧，既降低了对传输带宽的要求，也保证了解码器对切换码流的连续正确解码。

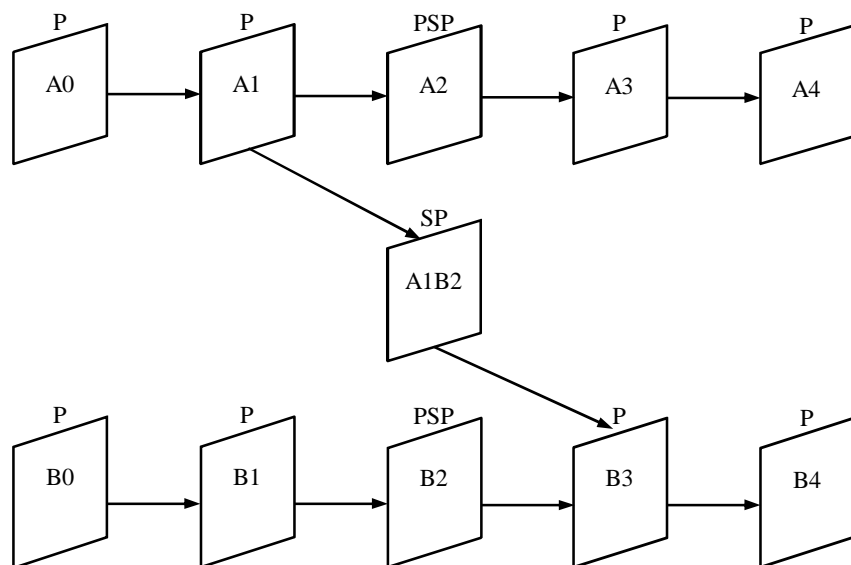


图 2-4-6、利用 SP 帧进行码流切换

## ②SI 帧应用于码流拼接和随机进入

实际应用中，还有是在不同内容的码流之间进行切换，如节目的切换组合，在节目中插入广告等，涉及不同内容图像序列合成码流的问题。如图 2-4-7，两个码流 A 和 B 分别是不同内容的序列编码生成的，两个码流之间的各帧缺少相关性，切换点之间的辅帧不能采用 SP 帧进行有效编码，只能采用空间预测的 SI 帧。

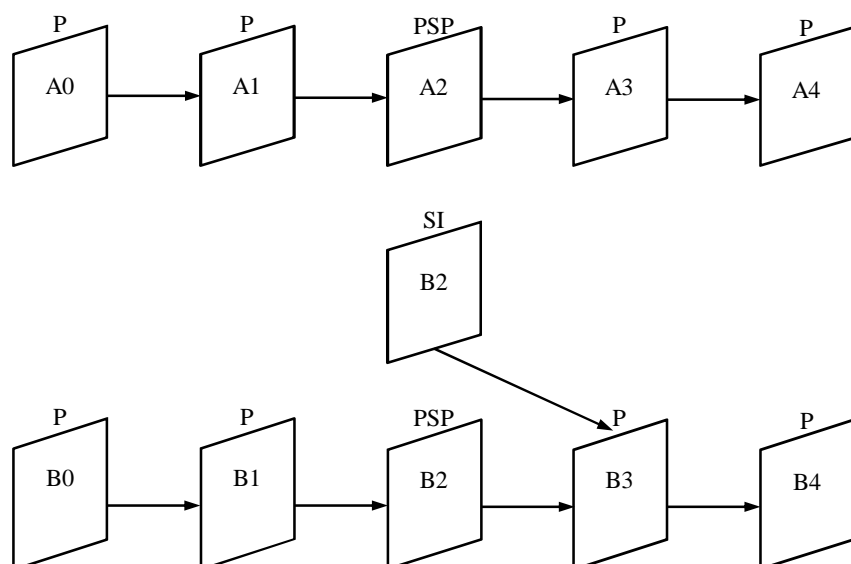


图 2-4-7、利用 SI 帧进行码流切换

可见，SP 和 SI 都可以应用于流切换，但应用的场合不同。当视频流的内容相同，编码参数不同时，采用 SP 帧。当视频流的内容差别很大时，则采用 SI 帧。

### ③错误恢复

采用不同的参考帧预测，可以获得同一帧的多个 SP 帧，利用这个特性可以防止误码扩散，从而增强码流的错误恢复能力。如图 2-4-8 所示，正在进行视频流传输的比特流中的一个帧无法正确解码。得到用户端反馈的错误报告后，服务器可以发送其后最邻近的主 SP 帧的一个辅 SP 帧或辅 SI 帧给客户端作为后续解码的参考帧，从而有效避免该错误影响后续的更多帧。

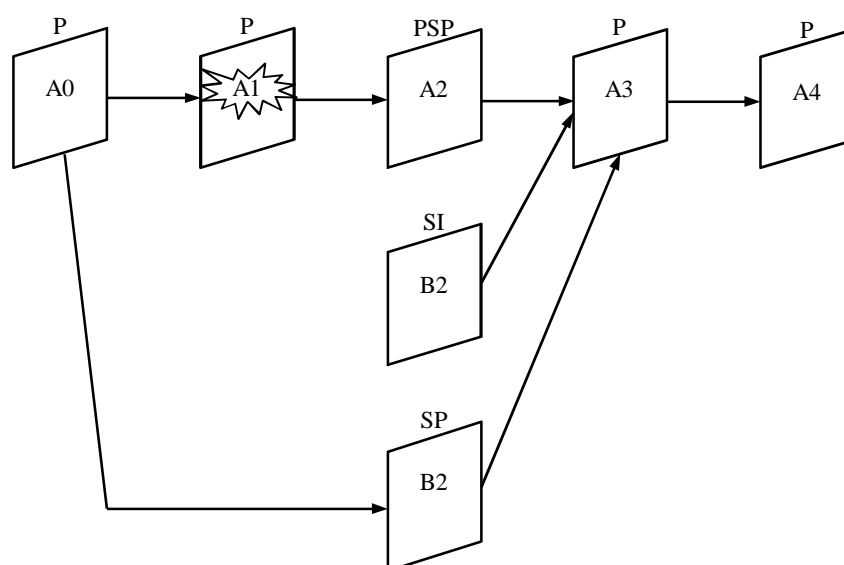


图 2-4-8、利用 SP、SI 帧进行码流错误恢复

## 2. 4. 4 H. 264/AVC 编码技术

### 1. 帧内预测编码

在 H.263<sup>+</sup>及 MPEG-4 以前的编码标准中，帧内编码都是直接对原始图像数据块进行离散余弦变换，然后量化和熵编码。实际上，不仅图像块内存在很强的相关性，相邻的图像块之间也存在相关性。基于这一点，H.264 的帧内编码并不直接对图像块进行变换、量化和编码处理，而是根据邻近块(通常是位于当前块上面和左边相邻的已编码块)的值来预测当前宏块的值，然后再对预测值和原始值的差值进行变换、量化和编码。由于是对预测误差进行编码，这种方法的编码效率更高。

在 H.264 的帧内预测模式中，预测块 P 是由已编码重建块和当前块形成的。对于亮度数据块而言，P 块采用 4×4 子块与 16×16 块的编码模式。4×4 亮度子块有 9 种帧内预测模式，适用于有大量细节的图像部分的帧内预测；16×16 亮度块有 4 种帧内预测模式，适用于平坦区域图像的帧内预测；8×8 色度块也有 4 种帧内预测模式，采用与 16×16 亮度块几乎相同的预测模式。编码器通常选择使得预测块和编码块差异最小的预测模式。

此外，还有一种帧内编码模式称为 I-PCM 编码模式，在该模式下，编码器不进行预测和变换，直接传输图像块的像素值。在图像内容不规则或者量化参数非常低的情况下，采用 I-PCM 模式可以比常规编码流程可以获得更高的编码效率。



为了防止误码扩散,应保证每个片的编码独立性,帧内预测是不允许跨越所在片的边界。

#### (1)、4×4 亮度预测模式

4×4 亮度预测模式共有 9 种不同的预测方法,如图 2-4-9 所示,4×4 亮度块的上方和左方的 A~Q 为已编码并重建的像素,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~Q 值和 9 种模式实现。其中,模式 2 (DC 预测) 根据 A~Q 中已编码像素进行预测,而其余模式只有在所需预测像素全部提供后才能使用。对模式 3~8,预测像素由 A~Q 加权平均而得,图 2-4-10 是 9 种 4×4 亮度预测模式,在表 2-1 中给出了相对应 9 种预测模式的具体描述。

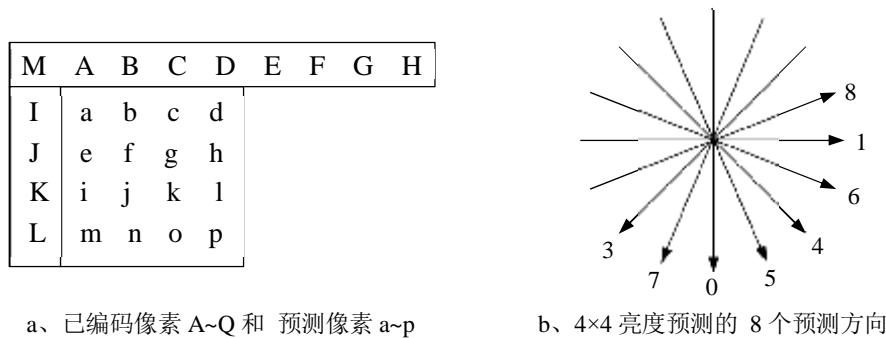


图 2-4-9、4×4 亮度预测

在编码过程中,对帧内预测的子块要采用上述 9 种方法分别进行预测,然后选取一种效果最好的方法。通常预测效果的好坏要根据预测块和当前块的绝对误差和 (SAE: Sum of Absolute Error) 的大小来判断。利用 9 种预测模式对图 2-4-11 所示图像的一个 4×4 亮度块计算产生的相应预测块如图 2-4-12 所示,同时图 2-4-12 还给出了各种预测模式下对应的 SAE 值,不难发现与当前块最匹配的预测模式是模式 8,因为该模式下的 SAE=203,是 9 种预测模式中 SAE 最小的一种。

表 2-1 H.264 的 4×4 亮度预测模式描述

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 平均推出所有像素值
模式 3 (下左对角线)	由 45° 方向像素内插得出相应像素值
模式 4 (下右对角线)	由 26.6° 方向像素内插得出相应像素值
模式 5 (右垂直)	由 26.6° 方向像素内插得出相应像素值
模式 6 (下水平)	由 45° 方向像素内插得出相应像素值
模式 7 (上水平)	由 45° 方向像素内插得出相应像素值

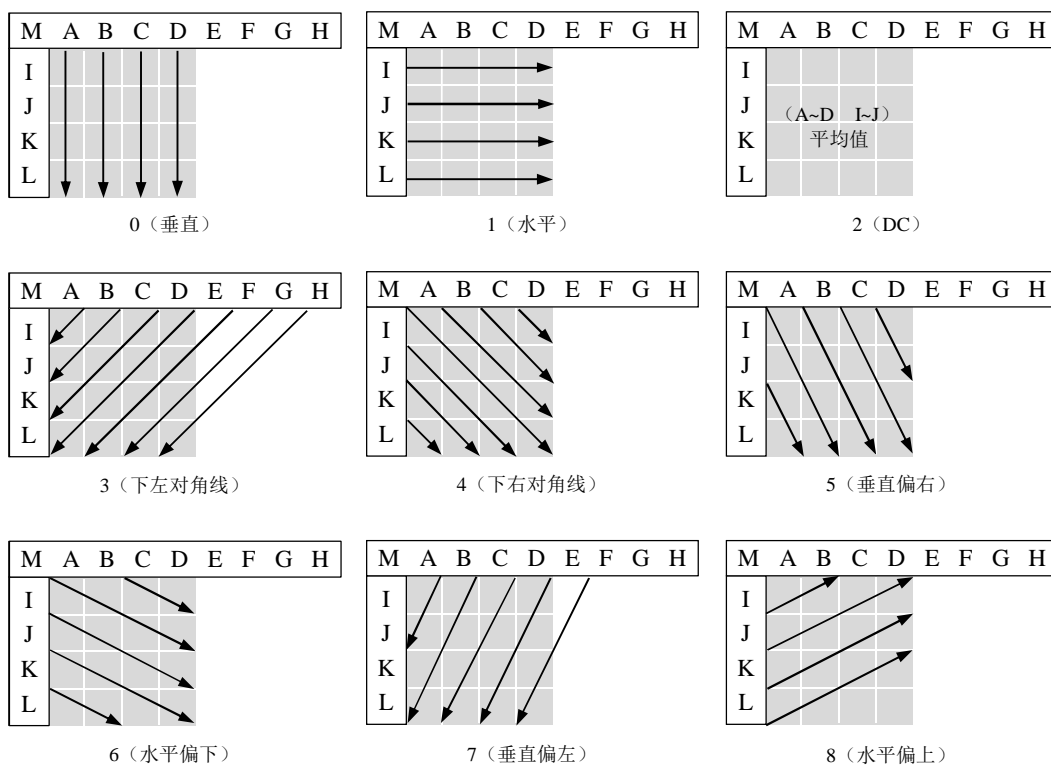


图 2-4-10 4×4 亮度块预测模式

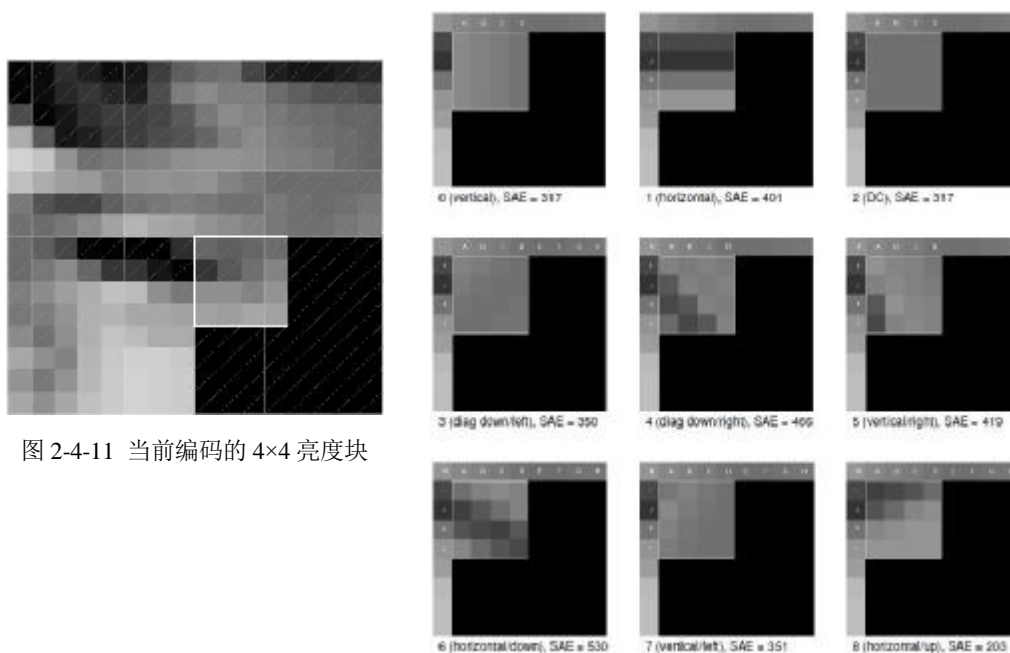


图 2-4-11 当前编码的 4×4 亮度块

图 2-4-12 9 种预测模式下的 4×4 亮度预测块

## 2、16×16 亮度预测模式

针对图像中的平坦区域，采用 16×16 亮度预测模式有利于提高处理速度和降低码率。H.264 提供了如图 2-4-13 所示的垂直预测、水平预测、直流预测和平面预测四种 16×16 预测模式。每一种预测模式的具体描述如表 2-2。

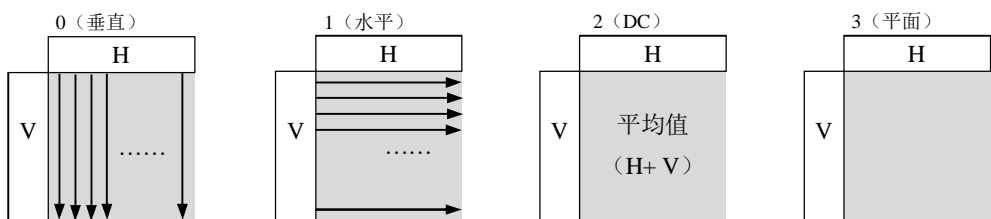


图 2-4-13 16×16 亮度块预测模式

表 2-2 16×16 亮度预测模式

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。

为了进一步说明 16×16 亮度预测模式的特点，采用图 2-4-14 所示的亮度块作为当前编码块，其左边和上边是已编码块的像素，采用四种预测模式对该块进行预测的结果及 SAE 值如图 2-4-15 所示，可见模式 3 与原始图像块最匹配。帧内 16×16 亮度预测模式适用于图像平坦区域的预测。

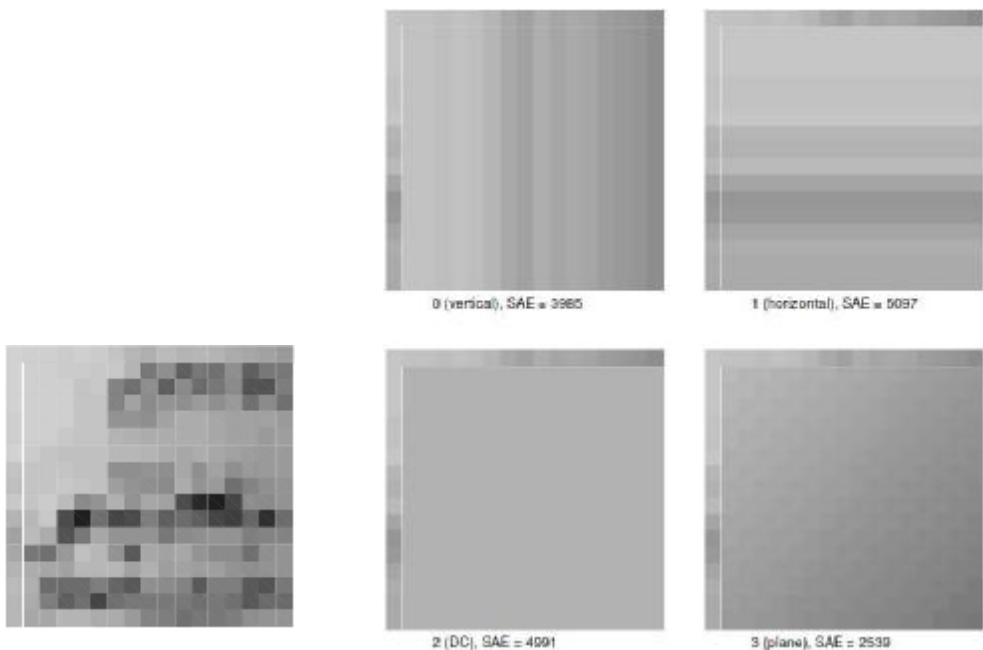


图 2-4-14 16×16 亮度块

图 2-4-15 16×16 亮度预测块

### (3)、8×8 色度块预测模式

每个帧内编码宏块的 8×8 色度分量由已编码左上方色度像素的预测而得，两种色度分量常用一种预测模式。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 为模式 0、水平为模式 1、垂直为模式 2、平面为模式 3。

### (4)、预测模式信息的编码方式

为了解码的需要，除了需要传送  $4 \times 4$  亮度块的预测差值外，还必须传送每个块所采用的预测模式信息给解码器。如果直接传送这一信息的二进制编码值将会占用比较多的比特数，H.264 则采取了更加有效的方法来传送预测模式信息，这种方法基于每个子块所采用的预测方式与其相邻的子块所采用的预测方式的紧密相关性。如图 2-4-16a 所示，当 A 块和 B 块的预测模式已知，则可以知道当前块 C 最有可能采用的预测方式，其次有可能采用的方式等等。为了更有效的进行编码，H.264 中把两个  $4 \times 4$  子块的帧内预测模式信息放在一起进行传送，一个宏块共需要 8 个码字进行传送，其顺序如图 2-4-16b 所示。

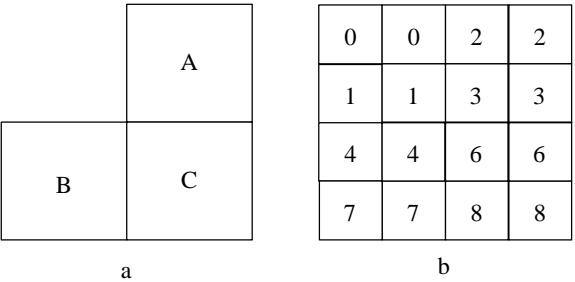


图 2-4-16 帧内预测模式信息的传送

## 2. 帧间预测编码

H.264 编解码器采用已编码视频帧/场和基于块的运动补偿进行帧间预测。除了保留以前编码标准帧间预测的主要特性外，H.264 还采用了一些新的特性来提高编码效率。其主要方法包括：①使用不同大小和形状的块进行运动搜索；②使用  $1/4$  像素精度搜索，即使用高精度的运动矢量来表示图像块的运动方向和位移；③使用多个参考帧进行帧间预测；④使用块分离滤波器消除块效应。这里将重点讨论 H.264 帧间预测编码新技术。

### (1)、树状结构运动补偿

H.264 支持从  $16 \times 16$  到  $4 \times 4$  多种大小的运动补偿块。如图 2-4-17 所示，亮度宏块（ $16 \times 16$  像素）可以按四种方式进行分割：1 个  $16 \times 16$  或 2 个  $16 \times 8$  或 2 个  $8 \times 16$  或 4 个  $8 \times 8$ ，每个分割区域称作一个宏块分区（Macroblock Partition）。如果选择  $8 \times 8$  划分模式，那么每个  $8 \times 8$  子宏块可以进一步以 4 种形式进行分割：1 个  $8 \times 8$  或 2 个  $8 \times 4$  或 2 个  $4 \times 8$  或 4 个  $4 \times 4$ ，称作宏块子分区（Macroblock Subpartition）。这种大小可变的宏块分割有利于运动估计时寻找最佳匹配区域。这种宏块分割下的运动补偿方法称为树型结构化运动补偿（Tree Structured Motion Compensation）。

每个分割或子宏块分割都有一个独立的运动补偿。运动矢量和分割类型都必须被编码和传输。对于比较大的分割尺寸如  $16 \times 16$ 、 $16 \times 8$ 、 $8 \times 16$  等，由于分割区域数量比较少，分割类型和运动矢量只需要少量比特，但含有丰富细节的区域因匹配不理想而导致运动补偿残差值比较大。小尺寸分割如  $8 \times 4$ 、 $4 \times 8$ 、 $4 \times 4$  等的残差值会比较小，但分割区域和运动矢量会增多，为此需要更多的比特数表征运动矢量和分割类型。分割尺寸的选择会影响压缩性能。整体而言，大的分割尺寸适用图像中的平坦区域，小的分割尺寸适用图像中的细节丰富区域。

宏块的色度块（Cr 和 Cb）大小是亮度块的一半，采用与亮度块相同的分割模式，只是尺寸减半，在水平和垂直方向上分别为亮度块的一半。例如亮度块中 8×16 分割对应色度块中 4×8 的分割，亮度块中 8×4 分割对应色度块中 4×2 的分割。

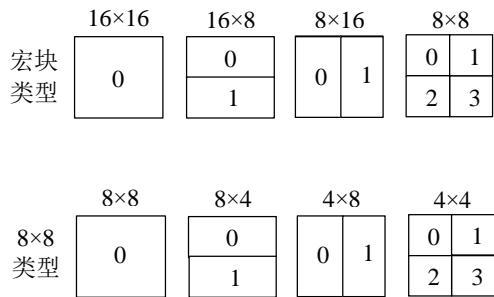


图 2-4-17 宏块及子宏块分割

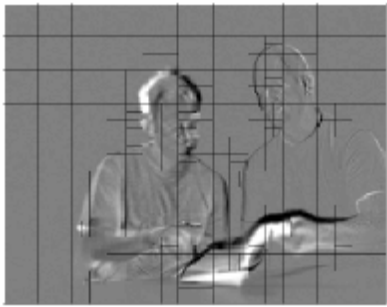


图 2-4-18 残差帧及分割

因此，应该根据图像内容特征选择分割尺寸。如图 2-4-18 所示的一个残差帧（没有进行运动补偿）中，为了尽量减小编码残差值和运动矢量数量，H.264 编码器为图像帧的每个区域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。在帧间变化小的区域（残差显示灰色）选择 16×16 分割，在多运动细节丰富的区域（残差显示黑色或白色）则选择更有效小尺寸分割。

### (2)、1/4 像素精度运动矢量

与以前的编码标准相比，H.264 算法由于采用了更高的精度来表示运动矢量的大小，因此能够更准确的得到预测块相对于原始块的位移，从而提高了预测精度，以达到压缩码率的效果。目前，H.264 最高支持 1/4 像素精度，与整像素预测相比，它能够节省 20% 的比特开销。

1/4 像素精度的运动搜索是指用以 1/4 像素为基本单位来定义运动矢量的大小。为了保证编解码端的一致，双方的像素内插方法必须一样。因为在参考帧中子像素位置上的样值并不存在，所以必须利用相邻样值进行插值运算得到。

子像素精度运动补偿提高了编码复杂度，但所达到的压缩效果远远优于整像素精度运动补偿（子像素精度比使用整数像素精度可以节省 20% 的码率）。使用 1/4 像素精度的效果也相对于使用半像素精度有所提高。在亮度块中，1/4 像素位置的样值是通过插值运算得到的：将整像素位置的样值通过 6 抽 FIR 滤波器，得到半像素位置的样值，然后对相应位置上的半像素样值或整像素样值作双线性内插运算，得到 1/4 像素位置的样值。

### (3)、运动矢量预测（Motion Vector Prediction）

每个宏块分割的运动矢量的编码需要相当数目的比特数，特别是在选择了小尺寸分割时运动矢量编码需要更多的比特数。为了减少传输运动矢量比特数，可以利用相邻宏块分割区域的运动矢量的相关性，由邻近已编码分割的运动矢量进行预测编码。进行编码并传输的是当前运动矢量和其预测值（MVP）的差值 MVD。运动矢量预测值取决于运动补偿尺寸和邻近分割的有无。

当存在已传送分割块时，如图 2-4-19 所示，E 为当前宏块或分割子宏块，A、B、C 分别为 E 的左、上、右上方的三个已传送块。如果 E 的左边不止一个分割，取最上的一个作为 A；如果上方不知一个分割块时，取最左边一个为 B。图 2-4-19a 为所有分割尺寸相同时的邻近分割选择，图 b 为分割尺寸不相同时的邻近分割选择。

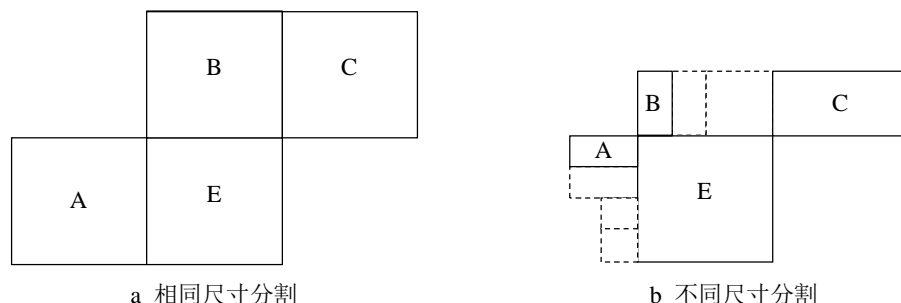


图 2-4-19 用来预测当前分割的和邻近分割

当传输分割不包括  $16 \times 8$  和  $8 \times 16$  时，运动矢量预测值为 A、B、C 运动矢量的中值；传输分割是  $16 \times 8$  时，上部运动矢量预测值由 B 分割运动矢量来预测，下部运动矢量预测值由 A 分割运动矢量来预测；当传输分割是  $8 \times 16$  时，左部运动矢量预测值由 A 分割运动矢量来预测，右部运动矢量预测值由 C 分割运动矢量来预测。

如果已传送块不存在时，运动矢量预测值的选择需重新进行调整。在解码端，运动矢量预测值以编码端相同的方式形成并加到运动矢量差值上。由于跳过宏块不存在运动矢量差值，其运动补偿宏块又运动矢量直接生成。

#### (4)、多个参考帧预测

在 H.264 标准中，帧间编码使用多个参考帧进行预测。编码器可以在多个参考帧中进行运动搜索，选择与当前编码帧最相似的一帧作为参考帧。这样，可以得到更好的主观图像质量及编码效率。与采用一个参考帧的方法相比，使用多个参考帧进行预测能够节省 5~10% 的比特开销。

### 3. 整数变换

为了进一步节省图像的传输码率，需要对图像信号进行压缩，为了去除图像信号中的相关性并减少图像编码的动态范围，通常采用变换编码及量化技术。变换编码将图像时域信号变换成变换域信号。在变换域，图像信号的能量大部分集中在低频区域，相对于时域信号，为码率下降提供了基础。H.264 对图像或预测残差采用  $4 \times 4$  整数离散余弦变换技术，有效避免了采用通用  $8 \times 8$  离散余弦变换、反变换引起的失配问题。

H.264 的“baseline”型（profile）根据所压缩的数据类型不同而采用三种不同的变换方式。它们分别是：适用于帧内预测宏块  $4 \times 4$  亮度 DC 系数块的整数变换，适用于任何色度宏块  $2 \times 2$  DC 系数块的整数变换和适用于其他  $4 \times 4$  残差数据块的整数变换。

一个宏块的数据按照如图 2-4-20 所示的顺序传输的，如果宏块采用  $16 \times 16$  帧内预测编码模式，那么首先传输的是序号为“-1”的块，它包含的是宏块中每个  $4 \times 4$  亮度块的 DC 系数。然后按照 0~15 的顺序传输亮度残差数据块（每个块的 DC 系数置零）。块 16 和块 17

是包含色度 DC 系数的 2×2 块。最后传输的是色度残差数据块 18~25（每块 DC 系数置零）。

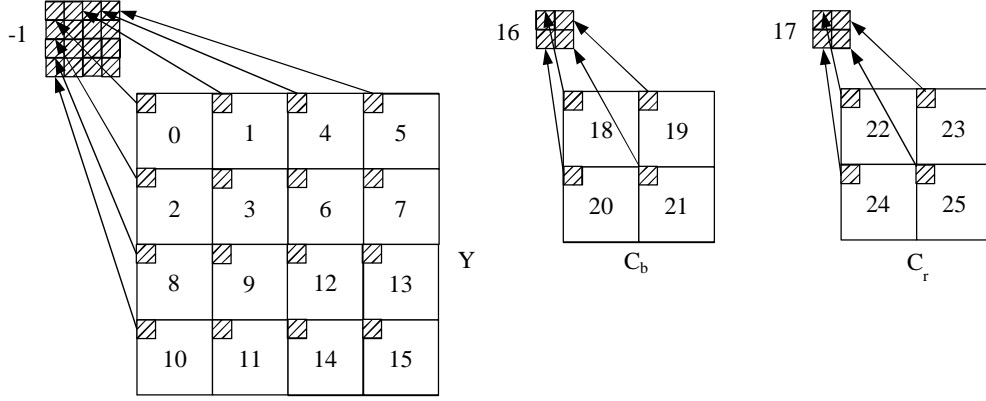


图 2-4-20 宏块各子块编码传送顺序

4×4 残差整数变换和量化适用于块 0~15 以及 18~25。虽然 4×4 整数离散余弦变换的基础是 DCT 变换，但二者之间存在着基本差别：

- ①4×4 整数离散余弦变换所有的运算都是整数运算并且没有精度损失。
- ②H.264 标准中对该变换的反变换有详细的说明，如果完全按照说明正确执行，编解码器之间不会出现误匹配。
- ③不需要乘法，仅仅需要加法和移位运算就可以实现整数变换的核心算法。

4×4 整数变换是在 4×4DCT 变换的基础推导而成，下面说明推导过程。

对于一个大小为 N×N 像素的图像块，像素点样值的分布函数为 X(x, y)（其中 x, y=1、2...N-1）的二维离散余弦变换可表示为：

$$Y(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} X(x, y) \cos \frac{(2x+1)u\mathbf{p}}{2N} \cos \frac{(2y+1)v\mathbf{p}}{2N}; \quad (\text{式2-1})$$

其中， $u, v = 1, 2, \mathbf{L}, N-1; k = 0, C(k) = 1/\sqrt{2}; k \neq 0, C(k) = 1;$

二维离散余弦变换的反变换可表示为：

$$X(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)Y(u, v) \cos \frac{(2x+1)u\mathbf{p}}{2N} \cos \frac{(2y+1)v\mathbf{p}}{2N}; (\text{式2-2})$$

其中， $x, y = 1, 2, \mathbf{L}, N-1; k = 0, C(k) = 1/\sqrt{2}; k \neq 0, C(k) = 1;$

在式 2-1 和式 2-2 中，X(x,y) 是图像块 X 中第 x 行第 y 列的图像或残差值，Y(u,v) 是变换结果矩阵 Y 相应频率点上的变换系数。因此，公式 2-1 和公式 2-2 可以用矩阵表示为：

$$\mathbf{Y} = \mathbf{A} \cdot \mathbf{X} \cdot \mathbf{A}^T \quad (\text{式 2-3})$$

$$\mathbf{X} = \mathbf{A}^T \cdot \mathbf{Y} \cdot \mathbf{A} \quad (\text{式 2-4})$$

这里，变换矩阵 A 中的系数相当于一维离散余弦变换的核函数。

$$A(u, x) = C(u) \cos \frac{(2x+1)u\mathbf{p}}{2N}; \quad x = 0, 1, \mathbf{L}, N-1; (\text{式2-5})$$

在 H.264 标准中，进行 DCT 变换的图像取 4×4 像素块(N=4)，因此，4×4 DCT 变换的变换矩阵 A 为：

$$\begin{aligned}
A &= \begin{bmatrix} \frac{1}{2} \cos 0 & \frac{1}{2} \cos 0 & \frac{1}{2} \cos 0 & \frac{1}{2} \cos 0 \\ \frac{1}{\sqrt{2}} \cos \frac{\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{5\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{7\pi}{8} \\ \frac{1}{\sqrt{2}} \cos \frac{2\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{6\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{10\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{14\pi}{8} \\ \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{9\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{15\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{21\pi}{8} \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} \cos \frac{\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & -\frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{\pi}{8} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} & -\frac{1}{\sqrt{2}} \cos \frac{\pi}{8} & \frac{1}{\sqrt{2}} \cos \frac{\pi}{8} & -\frac{1}{\sqrt{2}} \cos \frac{3\pi}{8} \end{bmatrix} \quad (\text{式 2-6})
\end{aligned}$$

$$\text{设 } a = \frac{1}{2}, b = \frac{1}{\sqrt{2}} \cos \frac{p}{8}, c = \frac{1}{\sqrt{2}} \cos \frac{3p}{8},$$

那么,

$$A = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \quad (\text{式 2-7}) \quad A^T = \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \quad (\text{式 2-8})$$

矩阵 A 以及矩阵  $A^T$  中的 a、b、c 是实数，而图像块中的各像素是整数。对于实数的离散余弦变换，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。相对于之前其它视频编码标准，H.264 使用了更多的预测编码技术，在帧内编码模式也采用了空间预测。因此，H.264 对预测漂移十分敏感。为此，H.264 对 4×4 DCT 中的变换矩阵 A 进行了改造，采用整数 DCT 变换技术，既不损失图像的准确度又有效减少变换的运算量。

A 矩阵可以表示为

$$A = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & c \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ b/c & 1 & -1 & -b/c \\ 1 & -1 & -1 & 1 \\ 1 & -b/c & b/c & -1 \end{bmatrix} \quad (\text{式 2-9})$$

由于 b/c=2.41，令 b/c=2，则得到改造后的 A 矩阵：

$$A = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & c \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad (\text{式 2-10})$$

同理，矩阵 A 的转置矩阵  $A^T$  可以表示为：

$$A^T = \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & c \end{bmatrix} \quad (\text{式 2-11})$$

由此，式 2-3 可以表示成：

$$Y = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & c \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \cdot X \cdot \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & c \end{bmatrix} \quad (\text{式 2-12})$$



由于，

$$E = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & c \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & a & 0 \\ 0 & 0 & 0 & c \end{bmatrix} = \begin{bmatrix} a^2 & ac & a^2 & ac \\ ac & c^2 & ac & c^2 \\ a^2 & ac & a^2 & ac \\ ac & c^2 & ac & c^2 \end{bmatrix} \quad (\text{式 2-13})$$

那么，式 2-12 可以表示为矩阵的叉乘：

$$Y = CXC^T \otimes E = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \cdot X \cdot \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & ac & a^2 & ac \\ ac & c^2 & ac & c^2 \\ a^2 & ac & a^2 & ac \\ ac & c^2 & ac & c^2 \end{bmatrix} \quad (\text{式 2-14})$$

因为  $b/c=2$ ，式 2-14 中的  $c=b/2$ ，可以表示为：

$$Y = CXC^T \otimes E = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \cdot X \cdot \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} \quad (\text{式 2-15})$$

这样，通过一次叉乘运算（ $\otimes$ ）将原来的 DCT 变换分成两部分运算。其中， $CXC^T$  中只包括整数的加法、减法和移位（乘 2）运算，这部分矩阵运算被称为整数 DCT 变换。而叉乘运算部分与量化过程相似，我们可以将它归纳到量化运算中。

对于  $4 \times 4$  亮度 DC 整数变换，根据对应像块在宏块中的排列顺序，将亮度宏块中的各个图像块的直流系数组成新的直流矩阵  $W_D$ ，然后，采用式 2-16 所示的 Hadamard 变换。

$$Y_D = H \cdot W_D \cdot H^T \quad (\text{式 2-16})$$

其变换矩阵  $H$  是将残差变换矩阵  $C$  中的 2 置 1 后得到，如式 2-17 所示。

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (\text{式 2-17})$$

$C_b$  和  $C_r$  色差块的直流系数块是  $2 \times 2$  矩阵，采用  $2 \times 2$  Hadamard 变换类似亮度直流块的变换，所用的变换矩阵如式 2-18。

$$H = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (\text{式 2-18})$$

#### 4. 量化

量化是在不降低视觉效果的前提下减少图像编码长度，以减少图像中不必要的信息。

H.264 标准采用标量量化技术，将每个图像块的变换系数值映射成较少的数值。一般，基本的标量量化公式如式 2-19 所示。

$$FQ = \text{round}\left(\frac{y}{Q_{\text{step}}}\right) \quad (\text{式 2-19})$$

其中， $y$  是输入样本数值， $Q_{\text{step}}$  是量化步长， $FQ$  为量化后的量化值， $\text{round}()$  为取整函数，该函数的输出值为与输入实数最接近的整数；反量化后的值  $y' = FQ \cdot Q_{\text{step}}$ 。

在量化和反量化过程中，量化步长  $Q_{\text{step}}$  决定了量化器的编码压缩率及图像精度。如果

量化步长  $Q_{step}$  比较大, 则量化值  $FQ$  的动态范围较小, 其相应的编码长度较短, 而反量化时损失较多的图像细节信息; 如果量化步长  $Q_{step}$  比较小, 则量化值  $FQ$  的动态范围较大, 相应的编码长度较长, 而反量化后图像细节损失较小。编码器根据图像值实际的动态范围自动地改变量化步长  $Q_{step}$ , 在编码长度和图像精度之间进行折衷, 达到最佳的整体效果。

H.264 标准支持 52 个量化步长, 每个量化步长有一个对应的序号, 被称为量化参数(QP)。如表 2-3 所示, 当量化参数取最小值(0)时, 代表最精细的量化, 当量化参数取最大值(51)时, 代表最粗糙的量化。量化参数 QP 值每增加 6, 量化步长  $Q_{step}$  值增加一倍。应用时可以在量化步长范围内结合图像质量和压缩效率进行灵活选择。对于色度信号的量化, 一般采用与亮度相同的量化步长。为了避免较大量化步长引起的颜色量化失真, H.264 标准把色度的量化参数 QP 最大值限制在亮度 QP 最大值的 80% 范围内, 即色度 QP 最大值是 31。

表 2-3 H.264/AVC 编解码器量化步长表

QP	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
QP	12	13	14	15	16	17	18	19	20	21	22	23
$Q_{step}$	2.5	2.75	3.25	3.5	4	4.5	5	5.5	6.5	7	8	9
QP	24	25	26	27	28	29	30	31	32	33	34	35
$Q_{step}$	10	11	13	14	16	18	20	22	26	28	32	36
QP	36	37	38	39	40	41	42	43	44	45	46	47
$Q_{step}$	40	44	52	56	64	72	80	88	104	112	128	144
QP	48	49	50	51								
$Q_{step}$	160	176	208	224								

在 H.264 标准中, 量化过程是对前面(式 2-15)的变换结果按式 2-20 进行操作。

$$Z_{uv} = \text{round}\left(\frac{Y_{uv}}{Q_{step}}\right) \quad (\text{式 2-20})$$

其中,  $Y_{uv}$  是矩阵  $Y$  中的变换系数,  $Z_{uv}$  是输出的量化系数,  $Q_{step}$  是量化步长。

而实际的整数变换是对式 2-15 中的叉乘左边的  $CXC^T$  进行操作, 将其结果记做矩阵  $W$ 。这样, 矩阵  $W$  中的整数变换系数  $W_{uv}$  与矩阵  $E$  中的对应元素  $E_{uv}$  相乘后可得到  $Y_{uv}$ 。量化过程按式 2-21 进行操作。

$$Z_{uv} = \text{round}\left(\frac{W_{uv} \cdot E_{uv}}{Q_{step}}\right) \quad (\text{式 2-21})$$

其中, 根据矩阵  $E$  的数值分布特性, 在不同的图像块空间位置  $(u,v)$  上  $E_{uv}$  只有 3 种数值, 如式 2-22 所示。

$$E_{uv} = \begin{cases} a^2 & (0,0)、(2,0)、(0,2)、(2,2); \\ b^2/4 & (1,1)、(1,3)、(3,1)、(3,3); \\ ab/2 & \text{其它情况;} \end{cases} \quad (\text{式 2-22})$$

式 2-21 中有实数的乘法和除法, 为了简化量化过程中的运算, H.264 标准在算法的实

现过程中利用量化步长随量化参数每增加 6 而增加一倍的性质，同时结合三个  $E_{uv}$  元素的数值，重新设计了新的只有 6 个值的量化表，量化表中的数值记作 MF，由式 2-23 可得到表 2-4。

$$MF = \frac{E_{uv} \cdot 2^{15}}{Q_{step} \cdot 2^{\text{floor}(QP/6)}} \quad (\text{式 2-23}) \quad \text{同时可以得出式 2-24。}$$

$$\frac{E_{uv}}{Q_{step}} = \frac{MF}{2^{15+\text{floor}(QP/6)}} \quad (\text{式 2-24})$$

其中， $\text{floor}()$  为取整函数，输出值为不大于输入实数的最大整数。将式 2-24 代入式 2-21 可以得到式 2-25。

$$Z_{uv} = \text{round} \left( \frac{W_{uv} \cdot MF}{2^{15+\text{floor}(QP/6)}} \right) \quad (\text{式 2-25})$$

这样，在表 2-4 中可以查得整数的 MF 值，表 2-4 只列出了对应量化参数 QP 值为 0~5 的 MF 值，对于量化参数 QP 值大于 5 的情况，利用 QP/6 的余数查表得到对应的 MF 值，该值保持与量化参数 QP 值小于等于 5 时的值相同，只是在分母 2 的指数中的  $\text{floor}(QP/6)$  函数值随 QP 每增加 6 而增加 1，从而产生量化步长增加一倍的效果。根据式 2-25 对变换系数进行量化，由于参与量化运算的变换系数  $W_{uv}$  和 MF 都是整数，两者相乘可以用移位和加法实现。同时除数为 2 的整数次幂，可以利用右移实现。因此，可以在量化过程中避免使用除法和乘法，利用简单移位和加法进行量化，大大简化运算的复杂度。

表 2-4 H.264 中的 MF 值

QP \ 位置	(0,0) (2,0) (2,2) (0,2)	(1,1) (1,3) (3,1) (3,3)	其它位置样点
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

具体的量化运算如式 2-26 所示。

$$|Z_{uv}| = (|W_{uv}| \cdot MF + f) \gg (15 + \text{floor}(QP/6)), \quad \text{sign}(Z_{uv}) = \text{sign}(W_{uv}) \quad (\text{式 2-26})$$

其中“ $\gg$ ”为右移运算，右移一次完成整数除以 2； $\text{sign}()$  符号函数；f 为偏移量，用于改善恢复图像的视觉效果，对帧内预测图像块 f 取  $2^{(15+\text{floor}(QP/6))}/3$ ，对帧间预测图像块 f 取  $2^{(15+\text{floor}(QP/6))}/6$ 。

例题：已知帧内预测的残差数据块 X 如式 2-27 所示，量化参数 QP=21。求整数变换、量化后的变换系数矩阵 Z，求不经过量化及经过量化的反变换矩阵。

$$X = \begin{bmatrix} 7 & 3 & -8 & -8 \\ 12 & 9 & -1 & -2 \\ 7 & 5 & -7 & -8 \\ -5 & -8 & -21 & -22 \end{bmatrix} \quad (\text{式 2-27})$$

解：首先，对残差数据块进行整数变换。

$$W = CXC^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \cdot \begin{bmatrix} 7 & 3 & -8 & -8 \\ 12 & 9 & -1 & -2 \\ 7 & 5 & -7 & -8 \\ -5 & -8 & -21 & -22 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix}$$

$$= \begin{bmatrix} -47 & 168 & 9 & -31 \\ 121 & -16 & 5 & 7 \\ -77 & 8 & 3 & -1 \\ 8 & -2 & 0 & -4 \end{bmatrix}$$

然后，按照式 2-2 和表 2-4 对矩阵 W 的各系数  $W_{uv}$  进行量化。由于  $QP=21$ ，用  $QP=MOD(21/6)=3$  查表 2-4。可以查到  $MF=9362$ 、 $MF=3647$ 、 $MF=5285$ 。

$MF=9362$  时，可以求得以下各位置的变换系数。

$$Z_{00} = -(|W_{00}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = -(47 \times 9362 + 2^{18/3}) \gg 18 = -2;$$

$$Z_{20} = -(|W_{20}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = -(77 \times 9362 + 2^{18/3}) \gg 18 = -3;$$

$$Z_{22} = (|W_{22}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = (3 \times 9362 + 2^{18/3}) \gg 18 = 0;$$

$$Z_{02} = -(|W_{02}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = (-9 \times 9362 + 2^{18/3}) \gg 18 = 0;$$

$MF=3647$  时，可以求得以下位置的变换系数。

$$Z_{11} = -(|W_{11}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = -(16 \times 3647 + 2^{18/3}) \gg 18 = 0;$$

$$Z_{13} = (|W_{13}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = (7 \times 3647 + 2^{18/3}) \gg 18 = 0;$$

$$Z_{31} = -(|W_{31}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = -(2 \times 3647 + 2^{18/3}) \gg 18 = 0;$$

$$Z_{33} = -(|W_{33}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = -(4 \times 3647 + 2^{18/3}) \gg 18 = 0;$$

$MF=5285$  时，可以求得以下位置的变换系数。

$$Z_{01} = (|W_{01}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = (168 \times 5285 + 2^{18/3}) \gg 18 = 4;$$

$$Z_{03} = -(|W_{03}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = -(31 \times 5285 + 2^{18/3}) \gg 18 = -1;$$

$$Z_{10} = (|W_{10}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = (121 \times 5285 + 2^{18/3}) \gg 18 = 3;$$

$$Z_{12} = (|W_{12}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = (5 \times 5285 + 2^{18/3}) \gg 18 = 0;$$

$$Z_{21} = (|W_{03}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = (8 \times 5285 + 2^{18/3}) \gg 18 = 0;$$

$$Z_{23} = -(|W_{23}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = -(1 \times 5285 + 2^{18/3}) \gg 18 = 0;$$

$$Z_{30} = (|W_{30}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = (8 \times 5285 + 2^{18/3}) \gg 18 = 0;$$

$$Z_{32} = (|W_{32}| \cdot MF + 2^{(15+\text{floor}(21/6))/3}) \gg (15+\text{floor}(21/6)) = (0 \times 5285 + 2^{18/3}) \gg 18 = 0;$$

这样，可以得到式 2-28 变换、量化后的矩阵 Z。

$$Z = \begin{bmatrix} -2 & 4 & 0 & -1 \\ 3 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{式 2-28})$$

计算不经过量化的反变换矩阵，先求变换矩阵 Y:

$$Y = C \cdot X \cdot C^T \otimes E = W \otimes E = \begin{bmatrix} -47 & 168 & 9 & -31 \\ 121 & -16 & 5 & 7 \\ -77 & 8 & 3 & -1 \\ 8 & -2 & 0 & -4 \end{bmatrix} \otimes \begin{bmatrix} 1/4 & 1/\sqrt{40} & 1/4 & 1/\sqrt{40} \\ 1/\sqrt{40} & 1/10 & 1/\sqrt{40} & 1/10 \\ 1/4 & 1/\sqrt{40} & 1/4 & 1/\sqrt{40} \\ 1/\sqrt{40} & 1/10 & 1/\sqrt{40} & 1/10 \end{bmatrix}$$

再求反变换矩阵 X

$$X = A^T \cdot Y \cdot A = C^T \cdot Y \otimes E \cdot C = C^T \cdot W \otimes E \cdot C = C^T \cdot \begin{bmatrix} -47 & 166 & 7 & -31 \\ 121 & -12 & 2 & 2 \\ -76 & 10 & 4 & -5 \\ 6 & -4 & 3 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1/4 & 1/\sqrt{40} & 1/4 & 1/\sqrt{40} \\ 1/\sqrt{40} & 1/10 & 1/\sqrt{40} & 1/10 \\ 1/4 & 1/\sqrt{40} & 1/4 & 1/\sqrt{40} \\ 1/\sqrt{40} & 1/10 & 1/\sqrt{40} & 1/10 \end{bmatrix} \cdot C$$

$$= C^T \cdot \begin{bmatrix} -47 & 166 & 7 & -31 \\ 121 & -12 & 2 & 2 \\ -76 & 10 & 4 & -5 \\ 6 & -4 & 3 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1/16 & 1/40 & 1/16 & 1/40 \\ 1/40 & 1/100 & 1/40 & 1/100 \\ 1/16 & 1/40 & 1/16 & 1/40 \\ 1/40 & 1/100 & 1/40 & 1/100 \end{bmatrix} \cdot C$$

$$= \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} -2.9375 & 4.2 & 0.5625 & -0.775 \\ 3.025 & -0.16 & 0.125 & -0.07 \\ -4.8125 & -0.2 & -0.1875 & -0.025 \\ 0.2 & -0.02 & 0 & -0.04 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 7 & 3 & -8 & -8 \\ 12 & 9 & -1 & -2 \\ 7 & 5 & -7 & -8 \\ -5 & -8 & -21 & -22 \end{bmatrix}$$

可见，反变换后的矩阵与原始图像残差数据一致。

求经过量化后的反变换矩阵 X<sup>ˆ</sup>，先由量化后的矩阵 Z 求反量化后的变换系数矩阵 W<sup>ˆ</sup>。

$$W^{\hat{}} = \begin{bmatrix} -46 & 165 & 0 & -30 \\ 120 & 0 & 0 & 0 \\ -74 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

再求反变换矩阵 X<sup>ˆ</sup>

$$X^{\hat{}} = C^T \cdot W^{\hat{}} \otimes E \cdot C = C^T \cdot \begin{bmatrix} -46 & 165 & 0 & -30 \\ 120 & 0 & 0 & 0 \\ -74 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1/16 & 1/40 & 1/16 & 1/40 \\ 1/40 & 1/100 & 1/40 & 1/100 \\ 1/16 & 1/40 & 1/16 & 1/40 \\ 1/40 & 1/100 & 1/40 & 1/100 \end{bmatrix} \cdot C$$

$$= \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} -2.875 & 4.125 & 0 & -0.75 \\ 3 & 0 & 0 & 0 \\ -4.625 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 6 & 4.125 & -7.125 & -9 \\ 12.5 & 10.375 & -0.875 & -2.75 \\ 6.25 & 4.375 & -6.875 & -8.75 \\ -6 & -7.875 & -19.125 & -21 \end{bmatrix}$$

经量化和反量化后反变换的矩阵 X<sup>ˆ</sup>与原始图像残差数据存在差异。

## 5. 去块效应滤波

在 H.264/AVC 标准中，编解码器量化、反量化、反变换后的图像会出现块效应。一个原因是变换系数量化使得反量化恢复的变换系数出现误差，从而造成恢复图像块边缘的不连续；第二个原因是运动补偿块的匹配不可能绝对准确，造成运动补偿后图像块边界上数据的

不连续。

去块效应滤波器的作用就是去除 H.264 编码解码算法带来的块效应。如图 2-4-21 所示，图 2-4-21a 是采用块效应滤波器处理的编解码器输出的图像效果，在 DCT 变换边界上有明显的呈方块形的瑕疵痕迹。图 2-4-21b 是采用了去块效应滤波器的编解码器对同一幅图像的处理效果，相比较而言，图 2-4-21b 的块效应有所减轻。

在视频编解码器中加入去块效应滤波器有两种方法：后置滤波和环路滤波。后置滤波只处理编码环路以外的显示缓冲器中的数据，在 H.264 标准中只作为可选项。环路滤波则处理编码环路中的数据，在编码器中，滤波的图像作为后续编码帧运动补偿的参考帧；在解码器中，滤波后的图像除了作为后续解码帧运动补偿的参考帧外，还被输出显示。在编码环路中使用去块效应滤波比后置滤波有三个方面的优点：可以保证不同水平的图像质量；解码端无需额外的帧缓存；主客观评价的图像质量更高。缺点是环路滤波的算法复杂度更高。



a、无去块效应滤波

b、采用去块效应滤波

图 2-4-21 有无去块效应滤波效果比较

实践表明，去块效应滤波处理可以有效地改善图像的主观质量。保持同样主观质量的前提下，使用去块效应滤波比无滤波能够节省大约 9% 的码率。

## 6. 熵编码

H.264 标准提供的熵编码有通用变长码编码(UVLC)、基于上下文的自适应变长码编码(CAVLC)和基于上下文的自适应二进制算术编码(CABAC)。UVLC 编码和 CAVLC 编码都遵循变长编码的思路，即根据概率统计分布制定码表，通过信源符号与码字相对应实现编码。算术编码的基本思想是基于概率的区间划分，CABAC 通过上下文建模使基本概率模型能适应随视频帧变化的统计特性，利用合适的上下文概率模型可以有效去除符号间的相关性，提高算术编码的效率。同 CAVLC 相比，在以同样的质量对视频信号进行编码时，CABAC 节省的码率通常是 10~15% 之间。

UVLC 编码方法使用统一变字长编码表。其优点是简单，缺点是单一的码表是从概率统计分布模型得出的，没有考虑编码符号间的相关性，在中高码率时效果不是很好。

CAVLC 编码方法对除变换系数量化值以外的其他语法元素使用同一码表(指数哥伦布码, Exp-Golomb code)，而对变换系数量化值则根据概率统计制定了若干不同码表。根据上下文自适应地选择码表。这里重点介绍 CAVLC 的基本原理以及 H.264/AVC 中 CAVLC 编码过程。

## (1)、CAVLC 的基本原理

在 H.264 中，基于上下文自适应的变长码编码（CAVLC）根据已编码句法元素的情况动态调整编码中使用的码表，去除相关性，从而获得极高的相关性。

CAVLC 用于亮度和色度残差数据的编码。经过整数变换、量化后的残差变换系数数据具有相应的特性：非零系数主要集中在低频部分，高频系数大部分为零；经过之字形(zig-zag)扫描，DC 系数附近的非零系数数目较大，而高频系数位置上的非零系数大部分是+1 和-1；相邻块的非零系数的数目有一定的相关性。CAVLC 利用这些特性进行压缩处理，可以进一步去除数据中的冗余信息，提高编码效率。

CAVLC 主要利用相邻已编码符号所提供的相关性，为将要编码的符号选择合适的上下文模型。在 H.264 中，CAVLC 有两个上下文模型：根据相邻已编码块的使用的表格来确定当前块非零系数数目和绝对值为 1 的系数数目编码所用表格；根据已编码非零系数的幅度决定当前非零系数幅度编码所使用的表格。利用这两个上下文模型，可以有效去除符号之间的相关性。

## (2)、H.264/AVC 中 CAVLC 编码过程

利用之字形扫描对量化后的 4×4 数据块进行重排序成一维数组。

### ①对非零系数数目（TotalCoeffs）和绝对值为 1 的系数数目（TrailingOnes）进行编码

统计系数中非零系数数目和绝对值为 1 的系数数目，非零系数数目范围从 0~16，绝对值为 1 的系数数目的范围是从 0~3。如果绝对值为 1 的系数数目大于 3 个，只将最后 3 个视作绝对值为 1 的系数，将其余的系数作为普通的非零系数。通过查表方式对非零系数数目和绝对值为 1 的系数数目进行编码，共有 4 个变长表格和 1 个定长表格可供选择。其中的定长表格的码字是 6 个比特，高 4 位表示非零系数的个数（Totalcoeffs），最低两位表示绝对值为 1 的系数的个数（TrailingOnes）。

表格的选择体现了基于上下文的思想，在对非零系数数目（TotalCoeffs）和绝对值为 1 的系数数目（TrailingOnes）进行编码时，根据变量 NC（Number Current，当前表格选择序号）从 5 个表格中选择合适的表格，见附表 2-1。除了色度的直流系数外，其他系数块的 NC 值是由当前块的左边块的表格选择序号（NA）和上边块的表格选择序号（NB）求得。当同一片内存在左边和右边块时， $NC=(NA+NB) \gg 1$ （右移 1 位）；当片内只存在有左边块和上边块时， $NC=NA$  或  $NC=NB$ ；当片内不存在左边和上边块时， $NC=0$ 。

### ②对每个绝对值为 1 的系数的符号进行编码

按照反向扫描的顺序，从高频开始对每个绝对值为 1 的系数的符号进行编码。用一个比特表示，0 表示+，1 表示-。

### ③对剩余的非零系数的幅度值（Levels）进行编码

按照反向扫描的顺序，从高频开始对剩余的非零系数进行编码，如果遇到第一个编码的系数为负数，应将该系数加 1 后进行编码。在编码之前需要将带符号的非零系数（Levels[i]）按照式 2-29 转换成无符号的数值（LevelCode[i]）。

$$\text{LevelCode}[i] = \begin{cases} -\text{Level}[i] \times 2 - 1, & \text{Level}[i] < 0 \\ \text{Level}[i] \times 2 - 2, & \text{Level}[i] \geq 0 \end{cases} \quad (\text{式 2-29})$$

无符号数值  $\text{LevelCode}[i]$  分为两部分分别进行编码，即无符号数值  $\text{LevelCode}[i]$  除以  $2^{\text{suffixlength}}$  得到商和余数，其中余数的长度  $\text{suffixlength}$  是一个变量，根据上下文模式自适应更新，余数长度  $\text{suffixlength}$  是否更新取决当前值以及已经编码的无符号数值  $\text{LevelCode}[i]$ 。其更新过程：一般情况下，余数长度  $\text{suffixlength}$  初始化为 0，只有在块中有多于 10 个非零系数并且其中绝对值为 1 的系数的数目少于 3 个的情况下，长度  $\text{suffixlength}$  初始化为 0；对最高频率位置上的非零系数编码；如果当前已编码好的无符号数值  $\text{LevelCode}[i]$  大于规定的阈值，变量  $\text{suffixlength}$  的值加 1。

决定是否将变量  $\text{suffixlength}$  的值加 1 的阈值如表 2-5 所示。通常第一个非零系数被编码后， $\text{suffixlength}$  的值总是加 1。如果  $\text{suffixlength}$  要从 1 变成 2，已经编码的无符号数值  $\text{LevelCode}[i]$  必须达到阈值 6。

表 2-5 决定余数长度的已编码无符号数值的阈值

当前 $\text{suffixlength}$	$\text{LevelCode}[i]$ 的阈值
0	0
1	3
2	6
3	12
4	24
5	48
6	N/A

#### ④对最后一个非零系数前的零的总数（TotalZeros）进行编码

$\text{TotalZeros}$  指的是在最后一个非零系数前零的数目，此非零系数值的是按照正向扫描的最后一个非零系数。因为非零系数数目（ $\text{TotalCoffs}$ ）是已知的，这就决定了非零系数前零的数目可能的最大值。根据这个特性，H.264 标准在编排 CAVLC 的  $\text{TotalZeros}$  码表时进行了优化，如附表 2-1 所示，需要根据  $\text{TotalCoffs}$  来决定编码  $\text{TotalZeros}$  使用的编码表。

#### ⑤对零的游程进行编码

每个非零系数前零的数目（ $\text{RunBefore}$ ）是按照反序进行编码的，从最高频的非零系数开始。在两种情况下不需要编码：最后一个非零系数前零的个数；如果没有剩余的零需要编码，就没有必要进行编码。

在 H.264 中，CAVLC 对每个非零系数前零的个数的编码依赖于  $\text{ZerosLeft}$  的值， $\text{ZerosLeft}$  表示当前非零系数左边的所有零的个数。当  $\text{ZerosLeft} = \text{RunBefore}$  时，结束编码。

**例题：**对下面经过整数变换、量化后的系数块 Z 进行 CAVLC 编码和解码，该系数块左边块的  $\text{NA}=2$ ，上边块的  $\text{NB}=1$ 。

$$Z = \begin{bmatrix} -2 & 4 & 0 & -1 \\ 3 & 0 & 0 & 0 \\ -3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$



利用之字形扫描将系数块 Z 转换成一维数据：-2, 4, 3, -3, 0, 0, -1, 0...

①对非零系数数目 (TotalCoeffs) 和绝对值为 1 的系数数目 (TrailingOnes) 进行编码, 在该一维数组中共有 5 个非零系数和 1 个绝对值为 1 的系数, 即 TotalCoeffs=5, TrailingOnes=1。对该数据查表前, 先计算当前表格选择序号  $NC=(NA+NB) \gg 1 = (2+1) \gg 1 = 1$ 。查附表 2-1 绝对值为 1 的系数数目和非零系数数目组成的数组编码可得数组 (1, 5) 的码字为: 0000 0001 10。

②按照反向扫描的顺序, 从高频开始对每个绝对值为 1 的系数的符号进行编码。该一维数组中只有 -1, 编码的码字为: 1。

③按照反向扫描的顺序对剩余的非零系数的幅度值 (-3, 3, 4, -2) 进行编码。

由于第一个编码的幅度值为 -3, 按 -2 来进行编码, 因此编码的幅度值 (Levels[i]) 为 -2, 3, 4, -2。

编码前还需要将带符号的幅度值按照式 2-29 转换成无符号的幅度值 (LevelCode[i]): 3, 4, 6, 3。

先对第一个无符号幅度值 3 进行编码, 编码的余数位长初始化为 suffixlength=0, 即第一个数值只有商 3, 无余数。对商 3 通过查附表 2-2 Level\_Prefix 码表可得编码值: 0001。

再对第二个无符号幅度值 4 进行编码, 编码的余数位长自动加 1 变为 suffixlength=1。 $4/2^{\text{suffixlength}} = 4/2^1 = 2$  (0), 即商为 2, 余数为 0, 对商 2 通过查附表 2-2 Level\_Prefix 码表可得编码值: 001, 对余数 0 进行自然二进制编码为 0。最终的编码码字为 001 0。

接着, 对第三个无符号幅度值 6 进行编码, 因为第二个无符号幅度值 4 没有达到门限 6, 编码的余数位长保持不变, suffixlength=1。 $6/2^{\text{suffixlength}} = 6/2^1 = 3$  (0), 即商为 3, 余数为 0, 对商 3 通过查附表 2-2 Level\_Prefix 码表可得编码值: 0001, 对余数 0 进行自然二进制编码为 0。最终的编码码字为 0001 0。

最后, 对第四个无符号幅度值 3 进行编码, 因为第三个无符号幅度值 6 达到门限 6, 编码的余数位长加 1, suffixlength=2。 $3/2^{\text{suffixlength}} = 3/2^2 = 0$  (3), 即商为 0, 余数为 3, 对商 0 通过查附表 2-2 Level\_Prefix 码表可得编码值: 1, 对余数 3 进行自然二进制编码为 11。最终的编码码字为 1 11。

④对最后一个非零系数前的零的总数 (TotalZeros) 进行编码, 根据 TotalZeros=2, TotalCoeffs=5, 查附表 2-3 零总数编码, 可以得到编码码字: 0011。

⑤对零的游程进行编码, 根据 Zerosleft=2, RunBefore=2, 查附表 2-5 零游程编码得编码码字为: 00。由于 Zerosleft=RunBefore=2, 编码结束。

因此, 该数据块的编码码字为: **0000000110 1 0001 0010 00010 111 0011 00**。

CAVLC 编码字的解码过程:

①根据可变长编码表解码唯一性的特点, 利用附表 2-1 可以查得码字 0000000110 对应的 TotalCoeffs=5, TrailingOnes=1; (**0000000110 1 0001 0010 00010 111 0011 00**)

②根据 TrailingOnes=1, 得知下一比特 1 对应的绝对值为 1 的非零系数为 -1; (**1 0001 0010**

00010 111 0011 00)

③根据可变长编码表解码唯一性特点，利用附表 2-2 可以查得码字 0001 对应的商 3，转换成无符号幅度值为  $3 \times 2^0 + 0 = 3$ ，再按式 2-29 转换成带符号幅度值  $(-3-1)/2 = -2$ ，因第一个非零负系数，减 1 后恢复原始系数为 -3；(0001 0010 00010 111 0011 00)

④根据可变长编码表解码唯一性特点，利用附表 2-2 可以查得码字 001 对应的商 2，此时有 1 比特余数为 0，转换成无符号幅度值为  $2 \times 2^1 + 0 = 4$ ，再按式 2-29 转换成带符号幅度值  $(4+2)/2 = 3$ ；(0010 00010 111 0011 00)

⑤根据可变长编码表解码唯一性特点，利用附表 2-2 可以查得码字 0001 对应的商 3，此时有 1 比特余数为 0，转换成无符号幅度值为  $3 \times 2^1 + 0 = 6$ ，再按式 2-29 转换成带符号幅度值  $(6+2)/2 = 4$ ；(00010 111 0011 00)

⑥根据可变长编码表解码唯一性特点，利用附表 2-2 可以查得码字 1 对应的商 0，此时因上一步的无符号幅度值达到 6，则存在 2 比特余数为 11，转换成无符号幅度值为  $0 \times 2^2 + 3 = 3$ ，再按式 2-29 转换成带符号幅度值  $(-3-1)/2 = -2$ ；(111 0011 00)

⑦根据可变长编码表解码唯一性特点和  $\text{TotalCoeffs}=5$ ，利用附表 2-3 可以查得码字 0011 对应的零的总数目  $\text{TotalZeros}=2$ ；(0011 00)

⑧根据可变长编码表解码唯一性特点，利用零游程编码附表 2-5 可以查得码字 00 对应的  $\text{Zerosleft}=2$ ， $\text{RunBefore}=2$ ；(00)

由于  $\text{Zerosleft}=\text{RunBefore}=2$ ，解码结束，输出的系数为：-3，3，4，-2，0，0，1。

## 7. 分层设计

H.264 标准为了实现高效压缩和良好的网络亲和性，将其功能分为两个层：视频编码层 VCL(video coding layer)和网络适配层 NAL(network abstraction layer)。它们之间的关系如图 2-4-22 所示。

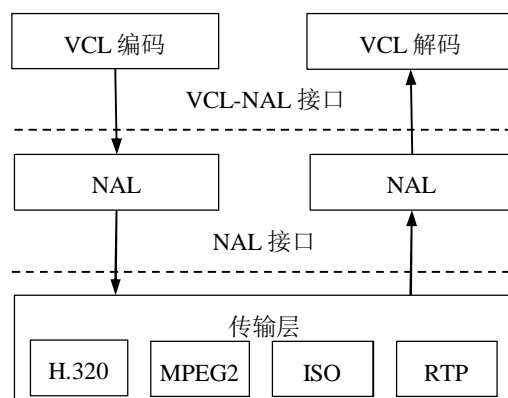


图 2-4-22 网络环境下的 H.264 的分层结构

图 2-4-22 网络传输环境中的 H.264/AVC 功能分层

### (1)、视频编码层

VCL 即视频编码层，包括视频编码器与视频解码器，主要功能是视频数据压缩编码和解码，包括运动补偿、变换编码、熵编码等处理单元。H.264 标准对视频编码层的句法元素

的分层结构进行了精心设计,相比传统标准的分层结构,当 H.264 码流在传输中出现错误时,新的分层结构可以有效提高重建视频图像的性能。

以往标准的分层结构如图 2-4-23 所示,在每一层都有头部和数据。在这样的结构中,每一层头部和它的数据之间形成管理与被管理的强依赖关系,头部的句法元素是该层的核心,而一旦头部丢失,数据部分的信息几乎不可能被解码出来。尤其是在序列层及图像层,由于网络中最大传输单元大小的限制,一个分组不可能传输整个层的句法元素,如果头部所在的分组丢失或严重错误,该层的其他分组即使能被正确接收也无法解码。

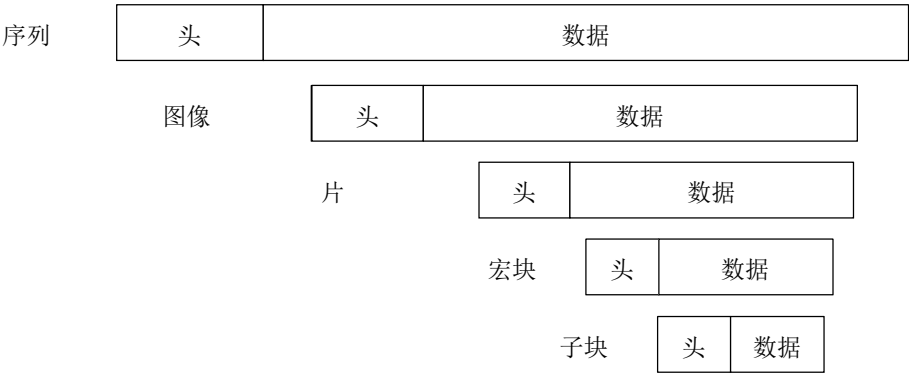


图 2-4-23 以往标准的语法元素分层结构

H.264 标准在设计句法元素分层时取消了序列层和图像层,将属于序列和图像头部的大部分句法元素游离出来形成序列和图像两级参数集,其余则放入片层。参数集作为一个独立的数据单位,不依赖于参数集以外的其他句法元素,只是在片层句法元素需要的时候被引用,而且一个参数集并不对应于某个特定的图像或序列,同一序列参数集可以被多个序列中的图像参数集引用。同样,同一图像参数集也可以被多个图像引用。只有在编码器认为需要更新参数集内容时,才会发送新的参数集。这样,由于参数集是独立的,既可以与其他句法元素分开通过彼此独立的通道传输,也可以采用多次重发方式或特殊纠错机制保证参数集的正确传输。

H.264 标准在参数集和片层结构上增加了数据的冗余度,一定程度上降低了编码效率。但是,带来的益处是信道的鲁棒性增强,当数据传输出现丢包时,能够将错误限制在最小的范围,防止错误扩散,并有利于解码后的错误隐藏和恢复。

在简单应用场合下,有了参数集和片就可以正常工作。但在复杂的通信环境下,除了片和参数集意外,还需要其他数据单位提供额外的信息,来保证系统的正常工作。复杂信道环境下可能出现的数据单位如图 2-4-24 所示,一个序列的第一个图像被叫做立即刷新图像,为 I 帧图像。引入立即刷新图像是为了解码的重同步,当解码器解码到立即刷新图像帧时,立即将参考帧队列清空,将已解码的数据全部输出或丢弃,重新查找参数集,开始一个新的序列解码。这样,如果在前一个序列中发生过重大错误,如严重丢包,或其他原因引起的数据错包,在此时可以获得重新同步。除了参数集和片以外还有其他的数据单元,这些单元可以提供由一系列句法元素构成的额外数据或同步信息,如分界符、SEI、冗余图像、序列结

束符等，这些信息不是解码过程必须的，但可以适当提高同步性能、定义图像的复杂特征。

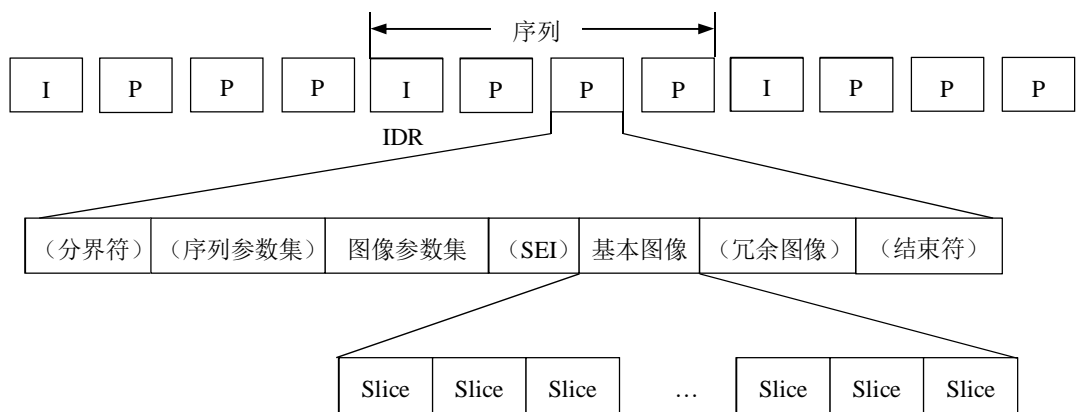


图 2-4-24 H.264 码流中的数据单位

(2)、网络适配层

NAL 用于定义适合于任何信道传输和存储的数据格式，并提供头信息，为视频编码提供与外界的统一接口。所有的数据都包含于 NAL 单元（NAL units）中，每个 NAL 单元包含整数字节的数据。H.264 定义了基于包和基于比特流两种数据格式，通过比特流传输层的一个起始代码可以区分这两种格式。

如图 2-4-25 所示，NAL 单元由一个字节的头部和原始字节序列有效载荷(RBSP)组成。头部字节由一个错误指示标志、一个处理的 NAL 单元标志和 NAL 单元类型组成。处理标志用于标识出 NAL 有效载荷的内容是否属于存储在多图像缓冲区中的图像，以及允许服务器、网络 and 网关在没有判断错误属性的情况下丢弃 NAL 单元。NAL 有效载荷类型包括定义 VCL 的 Slice 类型(I、P 或 B)，或者随机访问点、参数设置信息、附加的增强信息等。

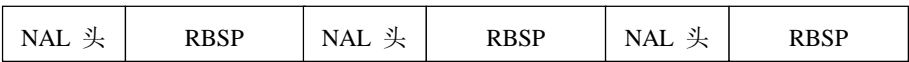


图 2-4-25 H.264 的 NAL 单元

8. 数据分割

组成片的编码数据存放在 3 个独立的数据分割（Data Partitioning）A、B、C 中，各自包含一个编码片的子集。分割 A 包含片头和片中每个宏块头的的数据。分割 B 包含帧内和 SI 片宏块的编码残差数据。分割 C 包含帧间宏块的编码残差数据。每个分割可放在独立的 NAL 单元中传输。分割 A 对传输误差非常敏感，如果分割 A 数据丢失，解码器便很难重建或不能重建片。解码器可根据要求只解 A 和 B 或者 A 或 C，以降低在一定传输条件下的复杂度。

CAVLC 编码表格

附表 2-4 色度 DC 2×2 块零系数总数编码表

total_zeros	TotalCoeff		
	1	2	3

0	1	1	1
1	01	01	0
2	001	01	
3	000		

附表 2-2 level\_prefix 编码表

level_prefix x	编码码字
0	1
1	01
2	001
3	0001
4	00001
5	000001
6	0000001
7	00000001
8	000000001
9	0000000001
10	00000000001
11	000000000001
12	0000000000001
13	00000000000001
14	000000000000001
15	0000000000000001

附表 2-5 零游程编码表

run_before	zeroLeft						
	1	2	3	4	5	6	> 6
0	1	1	11	11	11	11	111
1	0	01	10	10	10	000	110
2		00	01	01	011	001	101
3			00	001	010	011	100
4				000	001	010	011
5					000	101	010
6						100	001
7							0001
8							00001
9							000001
10							0000001
11							00000001
12							000000001
13							0000000001
14							00000000001

附表 2-1 非零系数总数和绝对值 1 的系数数目编码表

TrailingOnes	TotalCoeff	$0 \leq nC < 2$	$2 \leq nC < 4$	$4 \leq nC < 8$	$8 \leq nC$	$nC \geq -1$
0	0	1	11	1111	0000 11	01
0	1	0001 01	0010 11	0011 11	0000 00	0001 11
1	1	01	10	1110	0000 01	1
0	2	0000 0111	0001 11	0010 11	0001 00	0001 00
1	2	0001 00	0011 1	0111 1	0001 01	0001 10
2	2	001	011	1101	0001 10	001
0	3	0000 0011 1	0000 111	0010 00	0010 00	0000 11
1	3	0000 0110	0010 10	0110 0	0010 01	0000 011
2	3	0000 101	0010 01	0111 0	0010 10	0000 010
3	3	0001 1	0101	1100	0010 11	0001 01
0	4	0000 0001 11	0000 0111	0001 111	0011 00	0000 10
1	4	0000 0011 0	0001 10	0101 0	0011 01	0000 0011
2	4	0000 0101	0001 01	0101 1	0011 10	0000 0010
3	4	0000 11	0100	1011	0011 11	0000 000
0	5	0000 0000 111	0000 0100	0001 011	0100 00	—
1	5	0000 0001 10	0000 110	0100 0	0100 01	—

2	5	0000 0010 1	0000 101	0100 1	0100 10	—
3	5	0000 100	0011 0	1010	0100 11	—
0	6	0000 0000 0111 1	0000 0011 1	0001 001	0101 00	—
1	6	0000 0000 110	0000 0110	0011 10	0101 01	—
2	6	0000 0001 01	0000 0101	0011 01	0101 10	—
3	6	0000 0100	0010 00	1001	0101 11	—
0	7	0000 0000 0101 1	0000 0001 111	0001 000	0110 00	—
1	7	0000 0000 0111 0	0000 0011 0	0010 10	0110 01	—
2	7	0000 0000 101	0000 0010 1	0010 01	0110 10	—
3	7	0000 0010 0	0001 00	1000	0110 11	—
0	8	0000 0000 0100 0	0000 0001 011	0000 1111	0111 00	—
1	8	0000 0000 0101 0	0000 0001 110	0001 1110	0111 01	—
2	8	0000 0000 0110 1	0000 0001 101	0001 101	0111 10	—
3	8	0000 0001 00	0000 100	0110 1	0111 11	—
0	9	0000 0000 0011 11	0000 0000 1111	0000 1011	1000 00	—
1	9	0000 0000 0011 10	0000 0001 010	0000 1110	1000 01	—
2	9	0000 0000 0100 1	0000 0001 001	0001 010	1000 10	—
3	9	0000 0000 100	0000 0010 0	0011 00	1000 11	—
0	10	0000 0000 0010 11	0000 0000 1011	0000 0111 1	1001 00	—
1	10	0000 0000 0010 10	0000 0000 1110	0000 1010	1001 01	—
2	10	0000 0000 0010 01	0000 0000 1101	0000 1101	1001 10	—
3	10	0000 0000 0110 0	0000 0000 1001	0001 100	1001 11	—
0	11	0000 0000 0001 111	0000 0001 100	0000 0101 1	1010 00	—
1	11	0000 0000 0001 110	0000 0000 1000	0000 0111 0	1010 01	—
2	11	0000 0000 0010 01	0000 0000 1010	0000 1001	1010 10	—

续表

TrailingOnes	TotalCoeff	$0 \leq nC < 2$	$2 \leq nC < 4$	$4 \leq nC < 8$	$8 \leq nC$	$nC = -1$
3	11	0000 0000 0011 00	0000 0001 000	0000 1100	1010 11	—
0	12	0000 0000 0001 011	0000 0000 0111 1	0000 0100 0	1011 00	—
1	12	0000 0000 0001 010	0000 0000 0111 0	0000 0101 0	1011 01	—
2	12	0000 0000 0001 101	0000 0000 0110 1	0000 0110 1	1011 10	—
3	12	0000 0000 0010 00	0000 0000 1100	0000 1000	1011 11	—
0	13	0000 0000 0000 1111	0000 0000 0101 1	0000 0011 01	1100 00	—
1	13	0000 0000 0000 001	0000 0000 0101 0	0000 0011 1	1100 01	—
2	13	0000 0000 0001 001	0000 0000 0100 1	0000 0100 1	1100 10	—
3	13	0000 0000 0001 100	0000 0000 0110 0	0000 0110 0	1100 11	—
0	14	0000 0000 0000 1011	0000 0000 0011 1	0000 0010 01	1101 00	—
1	14	0000 0000 0000 1110	0000 0000 0010 11	0000 0011 00	1101 01	—
2	14	0000 0000 0000 1101	0000 0000 0011 0	0000 0010 11	1101 10	—
3	14	0000 0000 0001 000	0000 0000 0100 0	0000 0010 10	1101 11	—
0	15	0000 0000 0000 0111	0000 0000 0010 01	0000 0001 01	1110 00	—
1	15	0000 0000 0000 1010	0000 0000 0010 00	0000 0010 00	1110 01	—
2	15	0000 0000 0000 1001	0000 0000 0010 10	0000 0001 11	1110 10	—

3	15	0000 0000 0000 1100	0000 0000 0000 1	0000 0001 10	1110 11	—
0	16	0000 0000 0000 0100	0000 0000 0001 11	0000 0000 01	1111 00	—
1	16	0000 0000 0000 0110	0000 0000 0001 10	0000 0001 00	1111 01	—
2	16	0000 0000 0000 0101	0000 0000 0001 01	0000 0000 11	1111 10	—
3	16	0000 0000 0000 1000	0000 0000 0001 00	0000 0000 10	1111 11	—

思考题及练习题：

- 1、H.264 与它之前广泛应用的视频压缩标准相比有哪些突出的优点？
- 2、H.264 有哪些 Profiles？分别针对哪些应用？
- 3、什么是 H.264 的 Slices？Slices 有什么作用？H.264 的 Slices 有哪几种类型？
- 4、H.264 编码的分层设计分为哪几个层？分别有什么作用？
- 5、H.264 编码支持哪两种帧内编码模式？分别有什么特点？
- 6、在进行帧间预测编码处理时，H.264 为什么采用不同大小和形状的块进行运动补偿？
- 7、H.264 采用了哪些先进的技术措施？
- 8、H.264 的 SP 帧和 SI 帧有什么特点？分别应用在哪些场合？
- 9、H.264 编码标准的整数变换有什么特点？针对不同类型的数据，H.264 分别有哪几种变换方式？
- 10、H.264 的量化表有什么特点？具体的量化运算过程有什么特点？如何实现的？
- 11、H.264 的  $4 \times 4$  残差块系数矩阵为 X，求整数变换后的系数矩阵 Y。

$$X = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 3 & 1 & 2 & 2 \\ 1 & 2 & 4 & 1 \\ 1 & 4 & 3 & 3 \end{bmatrix}$$

- 12、H.264 的  $4 \times 4$  亮度直流系数矩阵为 X，求整数变换后的系数矩阵 Y。

$$X = \begin{bmatrix} 2 & 0 & 1 & 2 \\ 1 & 2 & 1 & 1 \\ 2 & 1 & 3 & 1 \\ 3 & 2 & 4 & 1 \end{bmatrix}$$

- 13、什么是 CAVLC？CAVLC 编码的基本思路是什么？它利用了残差数据块的哪些特性？
- 14、试对下面经过整数变换及量化运算后的残差块 Z 进行 CAVLC 编码（NC=1, 写出过程及结果）。

$$Z = \begin{bmatrix} 0 & -3 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

参考文献：

- [1] H.264 / MPEG-4 Part 10 White Paper
- [2] ITU-T Rec. H.264 / ISO/IEC 14496-10:2012 Information technology: Coding of audio-visual objects -- Part 10: Advanced Video Coding
- [3] 毕厚杰，新一代视频压缩编码标准—H.264/AVC，北京：人民邮电出版社，2005.5