H.264 概览

1. 引言

数字电视和 DVD-video 的出现使得广播电视和家庭娱乐发生了彻底的变革.越来越多的这些应用成为可能随着视频压缩技术的标准化.MPGE 系列的下一个标准,MPEG4,正使得新一代的基于因特网的视频应用成为可能.而现在视频压缩的 ITU-T H.263 标准被广泛的应用于视频会议系统.

MPEG4(视频)和H.263都是基于视频压缩(视频编码)技术的标准(大约从1995年开始).运动图像专家组和视频编码专家组(MPEG 和 VCEG)致力于开发一个比 MPEG4 和 H.263 有更好性能的新标准,有着高品质,低比特视频流的特性一个更好的视频图像压缩方法.新标准"高级视频编码"(AVC)的历史可追溯到 7 年前.

1995 年,为了通过电话线传输视频信号而制定的 H.263 标准定稿以后.ITU-T 视频编码专家组(VCEG)就开始工作在两个更深入的发展领域:一个是"短期"的努力去增加 H.263 的额外特性(制定出标准的版本 2),还有一个"长期"的努力,去开发一个适用于低比低率下可视通信的新标准,提供比之前的 ITU-T 标准更有效,明显更好的视频压缩方法.2001 年,ISO 运动图像专家组(MPEG)意识到H.26L的潜在优点,就组成了视频联合工作组(JVT),包括MPEG和VCEG的的专家.JVT 的主要任务就是将 H.26L"模式"草案发展成为一个完全的国际标准.实际上,结果产生了两个标准:ISO MPEG4 第 10 部分和 ITU-T H.264. 新标准的官方命名是"高级视频编码"(AVC);然而,旧的命名 H.26L 和以 ITU 文档号命名的 IH.264[1]更广为人知.

2. H.264 编解码器

和之前的标准一样(如 MPEG1,MPEG2 和 MPEG4),H.264 标准草案并没有明确定义一个编解码器.在一定程度上,标准定义了视频比特流编码和与之相对应的解码方法的语法.然而实际上,一个符合的编码和解码器一般包括如图 Figure 2-1 和 Figure 2-2 中所示的功能模块.同时这些图中所示功能通常是必须的,但编解码器还是可以有相当多的变种.基本的功能模块(预测,传输,量化,熵编码)与之前的标准(MPEG1,MPEG2,MPEG4,H.261,H.263)差不多.H.264 的最重要的变化是在这些功能模块的实现细节上.

编码器包括两个数据流路径.一个"前向"路径(从左到右,以蓝色表示)和一个"重构"路径(从右到左,以洋红色表示).解码器的数据流路径以从右到左的方式表示,以此来说明编码器和解码器之间的相同点.

2.1 编码器（前向路径）

当一个输入帧 Fn 被提交编码。该帧以宏块（相当于 16X16 像素的原始图像）为单位来进行处理。 每个宏块被编码成帧内模式或帧间模式。在这两种情况下，会产生一个基于重建帧的预测宏块 P。在帧内模式下，P 根据之前已经编码，解码，重建的当前帧 n 中的采样产生（图中以 uF’n 表示。注意是未经过滤的采样用来产生 P）。在帧间模式下，P 根据采用一个或多个参考帧的运动补偿预测来产生。在图中，参考帧表示为之前已经编码的帧 F’n-1；然而，每个宏块的预测可能根据过去或将来（以时间为序）的一或多个已经编码并重构的帧来产生。

预测 P 被从当前宏块中减去来产生一个残留的或差异宏块 Dn。它以量化变换系数集 X 变换（使用块变换）并量化.这些系数被重新排序并进行熵编码。在宏块解码时需要的熵编码系数和边信息（如宏块预测模式，量化步长，描述宏块如何运动补偿的运动矢量等等）组成了压缩的比特流。它被传输到了网络抽象层（NAL）进行传输或保存。

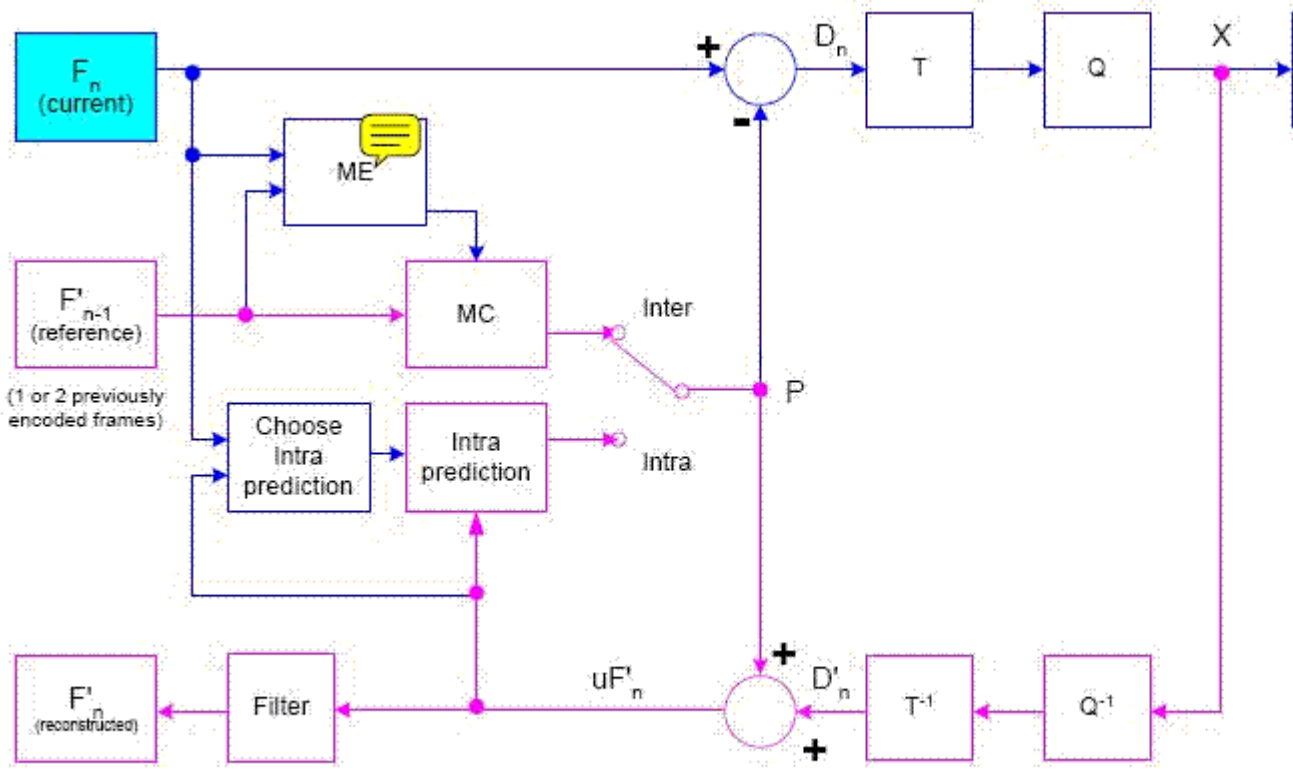


Figure 2-1 AVC Encoder

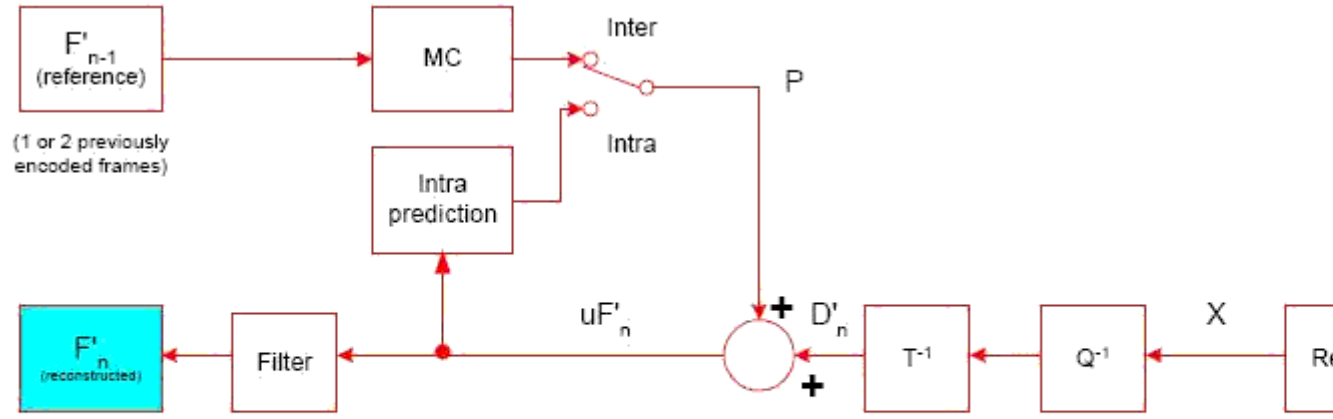


Figure 2-2 AVC Decoder

2.2 编码器（重建路径）

为了编码更进一步的宏块，需通过解码宏块量化系数 X 来重建一帧。系数 X 被重新调整（Q-1）并且进行逆变换（T-1）来产生一个不同的宏块 Dn'，这与原始的差异宏块 Dn 不同；它在量化过程中有了损耗，所以 Dn'是 Dn 的一个失真版本。

预测宏块 P 被加到 Dn'中来创建一个重建宏块 uF'n（原始宏块的一个失真版本）。为了减少阻断失真的影响使用了一个滤镜，重建参考帧从一系列的宏块 F'n 中创建。

2.3 解码器

解码器从 NAL（网络抽象层）接收压缩的比特流。数据元素被熵解码并且重新排列来产生一个量化系数集 X。它们被重新调整并进行逆转换来生成 Dn'（与编码器中所示的 Dn'相同）。使用比特流中解码出的头部信息，解码器生成一个预测宏块 P，与在编码器中生成的原始预测帧 P 相同。P 被加到 Dn'中来生成 uFn'，uF'n 经过过滤生成了解码的宏块 Fn'。

编码器的重建路径应该从图示和上面的讨论中清除，它实际上是为了确保编码器和解码器使用相同的参考帧来生成预测帧 P。如果不这样做，编码器和解码器中的预测帧 P 就不会相同，导致编码器和解码器之间存在一个越来越大的误差或是“偏移”。

3．参考资料

1 ITU-T Rec. H.264 / ISO/IEC 11496-10, “Advanced Video Coding”, Final Committee Draft, Document JVTE022,September 2002

帧内宏块预测

1. 引言

联合视频工作组（JVT）正在定案一个新的自然视频图像编码（压缩）标准。新标准被称为H.264或称作MPEG-4 Part 10、“高级视频编码（AVS）”。这篇文档描述了H.264编解码器中宏块帧内编码的方法。

如果一个块或宏块按帧内模式编码，会基于之前已经编码并重构（未过滤）的块生成一个预测宏块。这个预测块P被从之前已经编码的当前宏块中减去。对于亮度（luma）采样，P可能为每个4X4子块或16X16宏块产生。对于每个4X4的亮度块总共有9种可选的预测模式；对于16X16亮度块有4种可供选择的模式；而每个4X4的色度块只有一种预测模式（注：此处当理解成只有一种8X8的色度分块模式？）。

2. 4X4亮度块预测模式

Figure 1显示在一个QCIF帧中的一个亮度宏块和一个需要进行预测的4X4的亮度块。该块左边和上边的采样（即像素对应值）已经被编码、重建并且因而能够在编码器和解码器中用来生成预测块。基于Figure 2中标识为A-M的采样计算出预测块P。注意在有些情况下，不是所有的采样A-M都在当前切片（即分块）中可用：为了保持切片解码的独立性，只有在当前切片中可用的采样才会用来进行预测。直流(DC)预测（模式2）根据A-M中的哪些采样可用来修改；其它模式（1-8）或许只能使用在所有预测中用到的采样都可用的情况（除了E，F，G，H不可用的情况。这些值可从D中复制）。

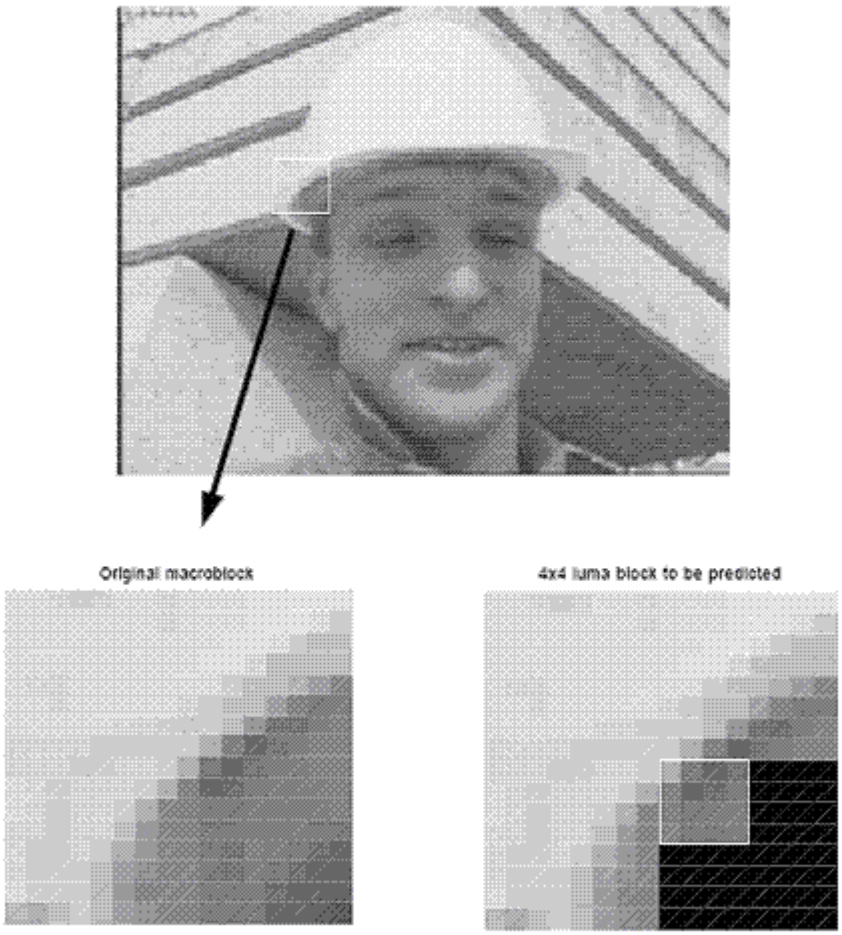


Figure 1 Original macroblock and 4x4 luma block to be predicted

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				

Figure 2 Labelling of prediction samples (4x4)

Figure 3 中的箭头标出了每种模式下预测的方向。在模式 3-8 中，预测值根据预测采样 A-Q 的加权平均来产生。编码器可能为每个块选择一种预测模式来减小 P 和被编码块之间的误差。

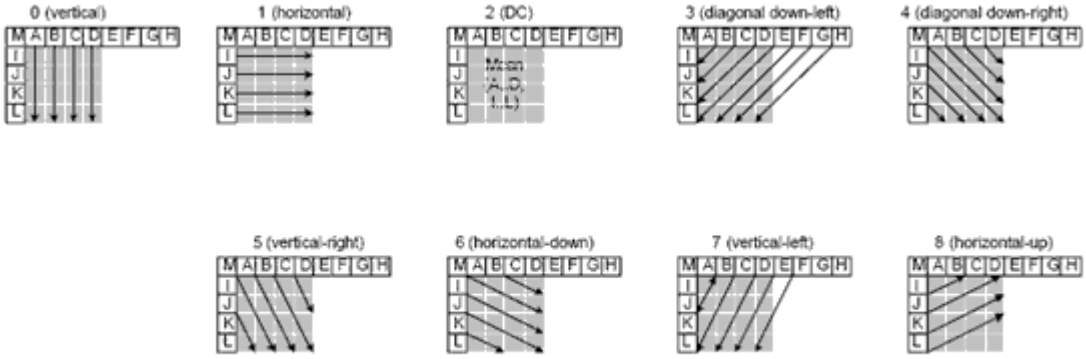


Figure 3 4x4 luma prediction modes

例：对 Figure 1 中所示的 4X4 块使用 9 种预测模式（0-8）计算。Figure 4 显示了每种预测生成的预测块 P。每种预测模式的绝对误差和（SAE）表明了预测误差的规模。在这种例子中，与实际当前块匹配最好的是模式 7(垂直向左)，因为这种模式给出了最小的 SAE；经过视觉上的比较可以看出 P 块与原始的 4X4 块非常相似。

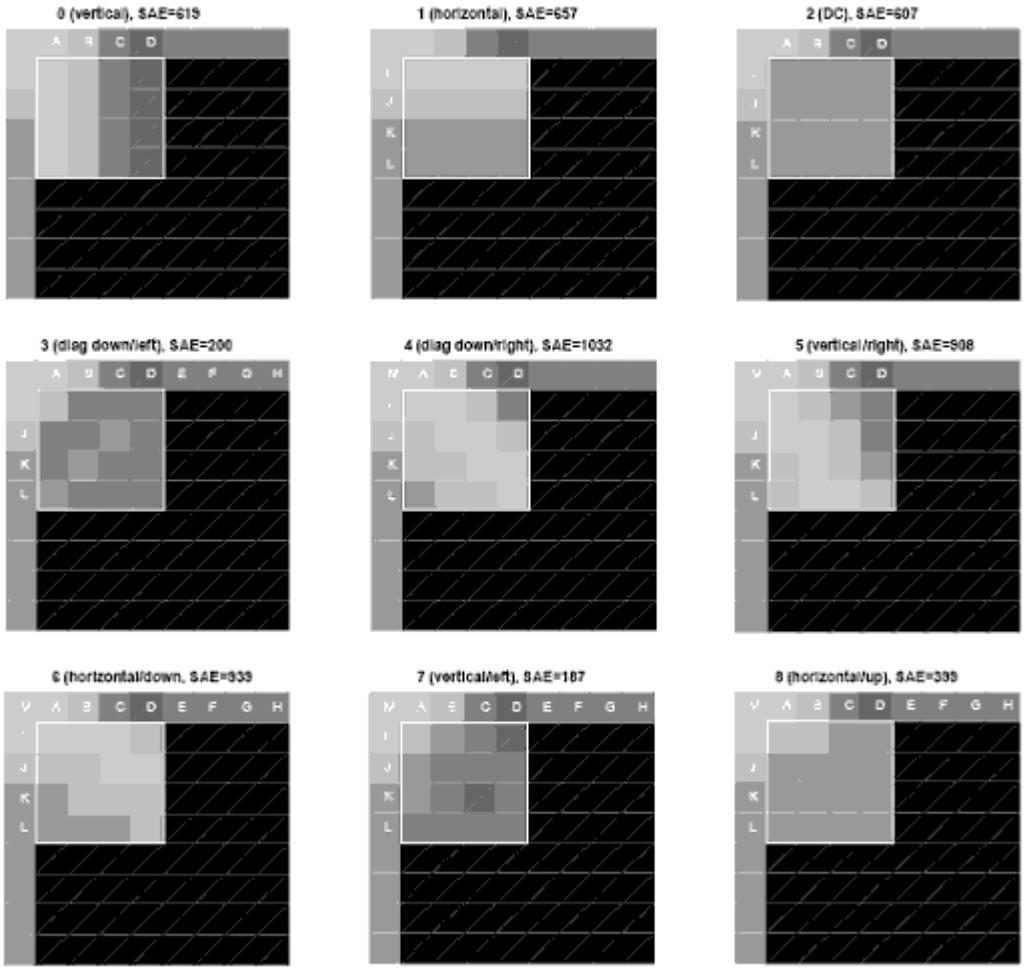


Figure 4 Predictio n blocks P (4x4)

3. 16X16 亮度块预测模式

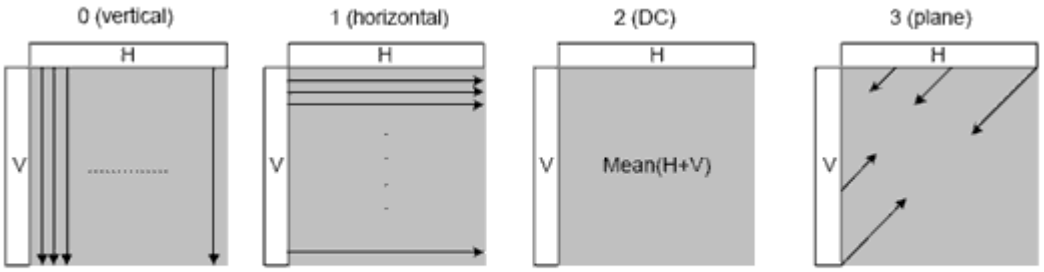
16X16 亮度块预测模式是一个上述 4X4 亮度块预测模式的一种替代方法，整个 16X16 的亮度块可以被预测。有四种模式可以使用。如 Figure 5 所示：

模式 0(垂直)：从块上部的采样推出（H）。

模式 1(水平)：从块左侧的采样推出（V）。

模式 2(直流)：从块上部和左侧的采样的均值推出（H+V）

模式 3(平面)：对块上部和左侧的采样 H 和 V 使用一个线性“平面”函数，这在平滑的亮度区域中效果最好。



H.26L Intra 16x16 prediction modes (all predicted from pixels H and V)

Figure 5 Intra 16x16 prediction modes

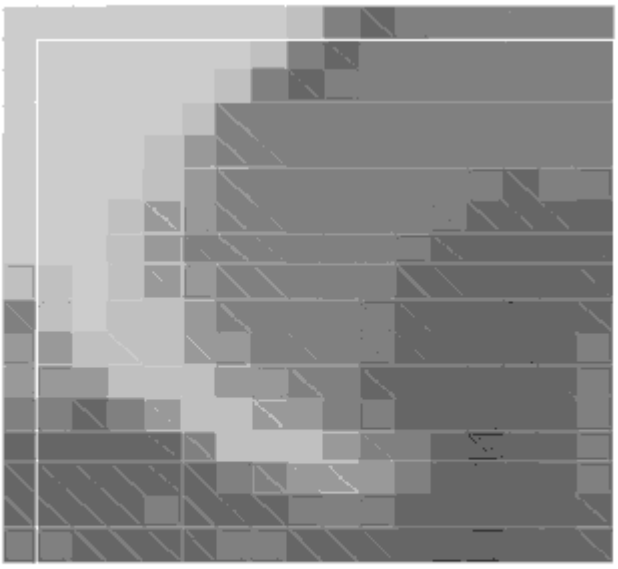


Figure 6 16x16 macroblock

例：

Figure 6 显示了一个左侧和上部采样已经编码的亮度宏块。预测结果（Figure 7）表明最佳匹配由模式 3 给出。帧内 16X16 模式在分布均匀的图像区域中效果比较好。

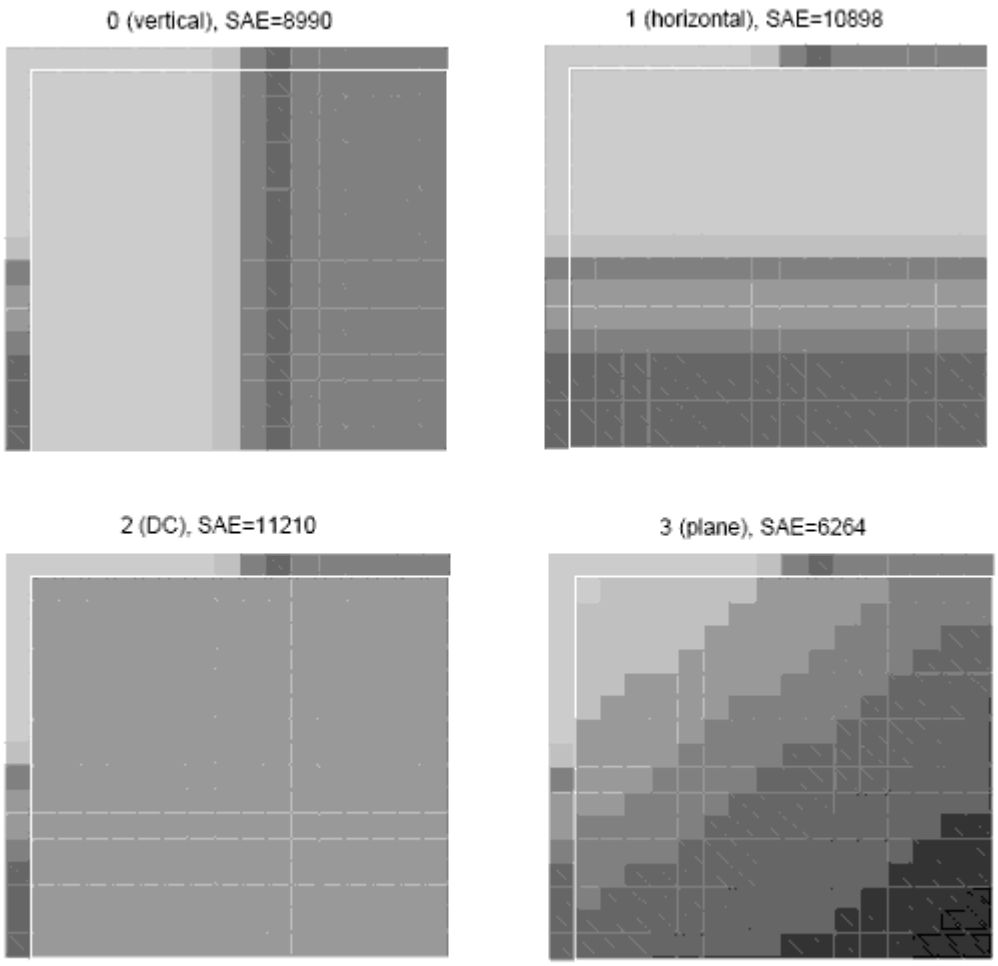


Figure 7 Intra 16x16 predictions

4. 8X8 色度块预测模式

一个宏块的每个 8X8 的色度分量从之前已编码重构的上部和（或）左侧的色度采样中预测。色度块的四种预测模式和第 3 部分中描述（Figure 5 所示）的 16X16 亮度块预测模式非常相似，除了模式号不同：直流（模式 0），横向（模式 1），垂直（模式 2）和平面（模式 3）。相同的预测模式在色度块中也始终适用（色度和亮度的相同预测模式的预测方法相同）。

注意：如果亮度分量中的 8X8 块被编码成帧内预测模式，色度块也要编码成帧内预测模式。

5. 帧内预测模式编码

每个 4X4 块所选择的帧内预测模式必须传递给解码器，这就要求一个大量的位来存储。然而，相邻 4X4 块帧内预测模式高度相关。例如，如果 Figure 8 中的之前已经编码的 4X4 块 A 和 B 使用模式 2，很可能块 C（当前编码块）的最佳预测模式也是模式 2。

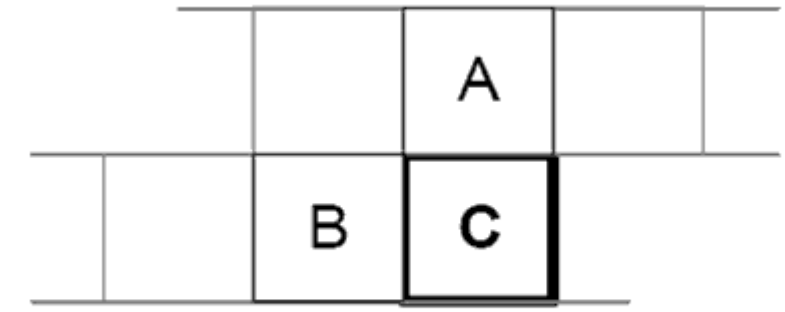


Figure 8 Adjacent 4x4 intra coded blocks

对每个当前块 C，编码器和解码器计算出最可能模式（most_probable_mode）. 如果 A 和 B 都使用了 4X4 帧内预测模式并且都包含在当前切片中，最可能模式就是 A 和 B 中模式号最小的预测模式；否则最可能模式设置为 2(直流预测)。

编码器给每个 4X4 块发送标志 use_most_probable_mode，如果标志为”1”，参数 most_probable_mode 被使用。如果标志为”0”，另一个参数 remaining_mode_selector 被发送，表明需改变模式。如果 remaining_mode_selector 比当前 most_probable_mode 小，预测模式就设置成 remaining_mode_selector; 否则预测模式设置成 remaining_mode_selector+1. 采用这种方法，remaining_mode_selector 只需使用 8 个值（0 到 7）来标志当前帧内预测模式（0 到 8）。

6. 参考资料

1 ITU-T Rec. H.264 / ISO/IEC 11496-10, “Advanced Video Coding”, Final Committee Draft, Document JVTF100,

December 2002

2 Iain E G Richardson, “H.264 and MPEG-4 Video Compression”, John Wiley & Sons, to be published late

2003

P 片帧间预测

1. 引言

联合视频工作组（JVT）正在定案一个新的自然视频图像编码（压缩）标准。新标准[1]被称为H.264或称作MPEG-4 Part 10、“高级视频编码（AVS）”。这篇文档描述了H.264中以P-片来进行帧间预测编码的方法。

帧间预测从一个或多个之前已经编码的视频帧中生成一个预测模型。这个模型由对参考帧中的采样进行漂移产生（运动补偿预测）。AVC CODEC 使用基于块的运动补偿，与从H.261以来的主要编码标准中采用的规则相同。然而，它与早期标准有着重要的区别：（1） 支持多种块大小（最小4X4）的预测；（2） 细粒度的次像素运动矢量（量度分量可精确到1/4像素）。

2 树结构的运动补偿

AVC对亮度采样支持16X16到4X4之间多种块大小的运动补偿。每个宏块（16X16采样）的亮度分量能以Figure 2-1中所示的4种方式分割：16X16，16X8，8X16或8X8.每个细分区域是一个宏块分区。如果选择8X8分割模式，宏块内的四个8X8宏块分区都能够以4种方式再次分割，如图Figure 2-2所示：8X8，8X4，4X8或4X4（称作宏块子分区）。这些分区和子分区使得每个宏块内都可以有很多种组合方式。这种将宏块分割成不同大小运动补偿子块的方法称作树结构运动补偿。

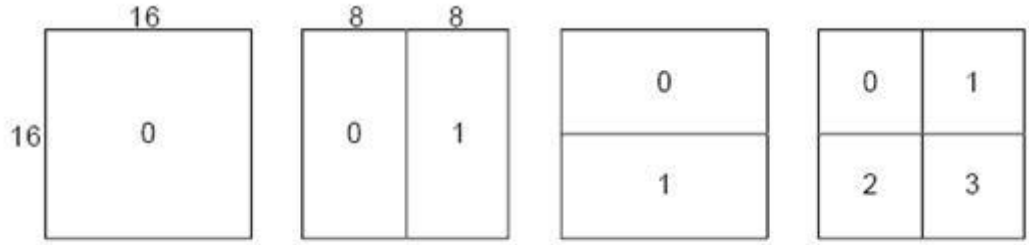


Figure 2-1 Macroblock partitions: 16x16, 8x16, 16x8, 8x8

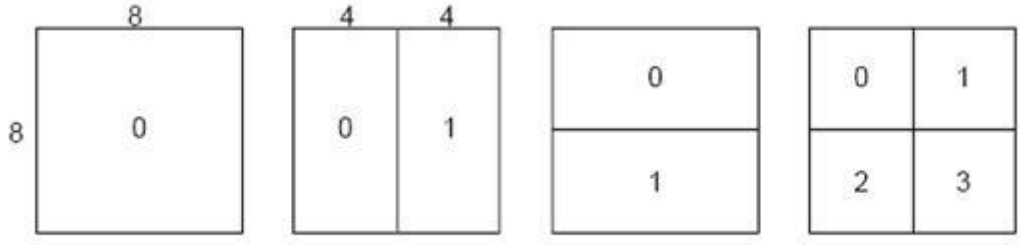


Figure 2-2 Macroblock sub-partitions: 8x8, 4x8, 8x4, 4x4

每个分区或子分区都有一个独立的运动矢量。每个运动矢量必须编码并传输；另外，分区（分割方式）的选择也必须被编码到压流的比特流中。选择大的分区（如16X16,16X8,8X16）意味着只需很少的比特来标记运动矢量的选择和分区的类型；然而，运动补偿残差可能包含了大量的帧中高细节部分信息。选择一个小的分区大小（8X4，4X4 等等）可以在运动补偿后得到一个较小的残差，但它需要更多的比特来标记运动矢量和分区的类型。因此分区大小的选择对压缩效果有着很大的影响。通常情况下，一个大的分区适用于帧内分布均匀的区域而一个小的分区将有利于细节区域的描述。

宏块（Cr 和 Cb）中每个色度分量的分辨率为亮度分量的一半。每个色度块和亮度块的分割方式相同，只是分区大小是水平和垂直方向的分辨率的一半（如一个 8X16 亮度分区对应一个 4X8 的色度分区；一个 8X4 亮度分区对应一个 4X2 的色度分区）。每个运动矢量（每个分区一个）的水平和垂直分量运用到色度块上时减半。

例：Figure 2-3 显示了一个残留帧（未经过运动补偿）。AVC 参考编码器为帧的每个部分选择了“最好的”分区大小。这种分区大小尽量减小编码残差和运动矢量。为每个区域选择的宏块分区显示在残留帧上（图示）。在帧间变化很小的区域（残留帧上显示为灰色）采用 16X16 分区方式；在运动的细节描述部分（残留帧上显示为黑色或白色）采用较小的分区效果更好。

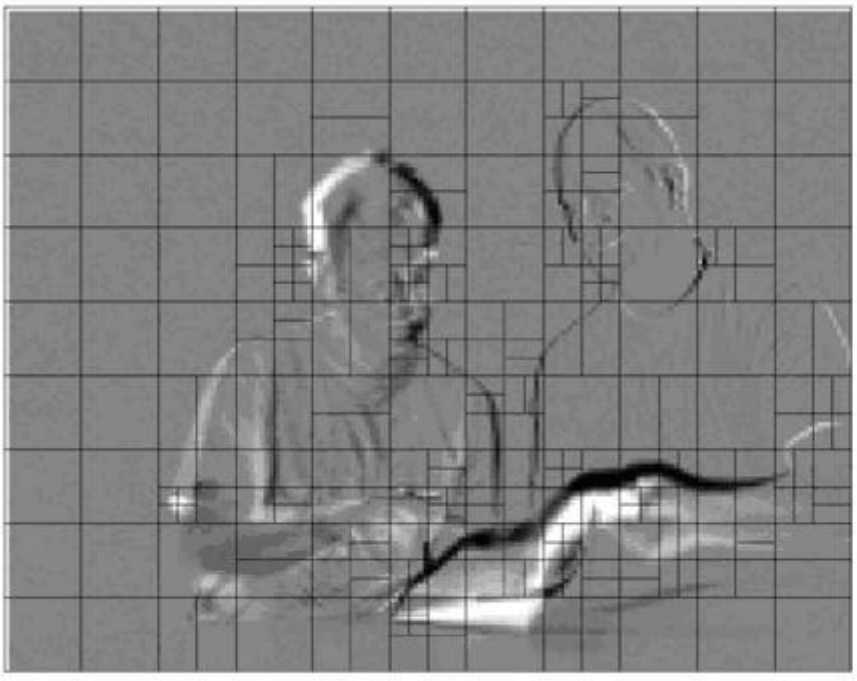


Figure 2-3 Residual (without MC) showing optimum choice of partitions

3. 次像素运动矢量

帧间编码宏块的每个分区根据参考图片中的一个相同大小的区域预测。两个区域间的偏移（运动矢量）的分辨率（亮度分量）为 1/4 像素。次像素位置的亮度和色度采样在参考图片中不存在，所以它们必须通过对附近的图像采样进行插补来创建。Figure 3-1 给出了一个例子。当前帧（a）中的一个 4X4 的子分区根据参考图像的一个邻近区域来预测。如果运动矢量的水平和垂直分量为整数（b），则参考块中的相关采样实际已经存在（灰点）。如果一个或两个向量分量都为分数值（c），预测采样（灰点）根据参考帧中相邻采样（白点）间的插补产生。

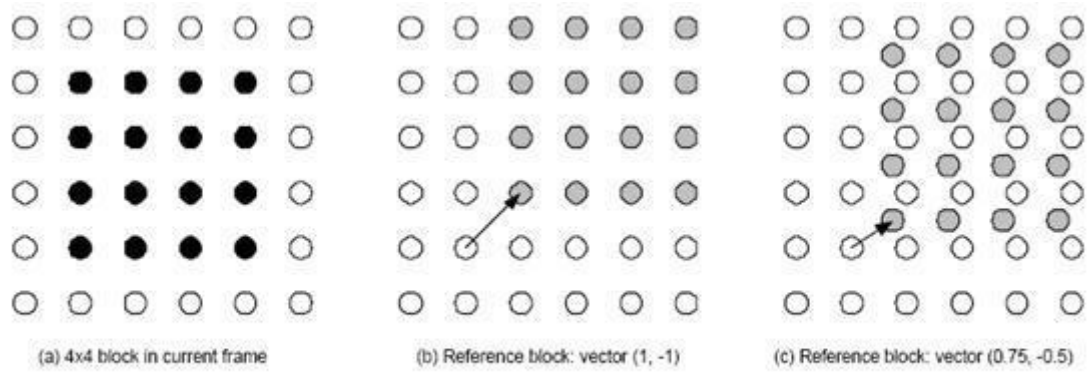


Figure 3-1 Example of integer and sub-pixel prediction

次像素运动补偿的效果明显优于整数像素补偿，但是却增加了复杂性。1/4 像素精度效果优于半像素精度。

在亮度分量中，半像素位置的次像素采样首先产生，它使用一个 6 头有限脉冲响应滤波器（6-tap Finite Impulse Response filter）从邻近的整数像素采样中插值。这意味着每个半像素采样是 6 个邻近整数采样的加权和。一旦所有的半像素采样可用，每个 1/4 像素采样通过对相邻的半像素和整数像素使用双线性插值得到。如果视频源采样比是 4:2:0，色度分量需使用 1/8 像素采样（与亮度的 1/4 像素采样对应）。这些采样通过对整数像素的色度采样进行插值（线性插值）得到。

4. 运动矢量预测

为每个分区编码一个运动矢量需要大量的比特，特别是在选择小的分区的时候。通常情况下邻近分区的运动矢量都高度相关，所以每个运动矢量可以通过相邻并且之前已经编码的分区的矢量预测。预测矢量 MVp 基于之前已计算出的运动矢量产生。当前矢量和预测矢量间的差异 MVD 被编码并传输。预测矢量 MVp 的生成方法取决于运动补偿块的大小和相邻矢量是否可用。“基本”预测值是当前分区或子分区相邻上部和对角线方向右下和相邻左侧的宏块分区或子分区的运动矢量的中间值。如果（a）选择 16X8 或 8X16 的分区并且（或者）(b)有些邻近分区不可用作预测值，则预测值会作出改变。如果当前宏块被跳过（未传输），则会产生一个预测矢量，即使宏块以 16X16 分区模式编码。

在解码器中，预测矢量 MVp 以相同的方式形成并加到解码后的矢量差异 MVD 中。若有跳过宏块，则这里没有解码矢量（MVD），这时会根据 MVp 的规模产生一个运动补偿宏块。

5. 参考资料

1 ITU-T Rec. H.264 / ISO/IEC 11496-10, “Advanced Video Coding”, Final Committee Draft, Document JVTG050,March 2003

变换和量化

1. 引言

联合视频工作组（JVT）正在定案一个新的自然视频图像编码（压缩）标准。新标准[1]被称为 H.264 或称作 MPEG-4 Part 10、“高级视频编码（AVS）”。这篇文档描述了标准所定义或隐含的变换和量化过程。

每个残差宏块被传输，量化并编码。之前的标准如 MPEG-1，MPEG-2，MPEG-4 和 H.263 使用了 8X8 离散余弦变换（DCT）作为基本变换。H.264 的基本规范使用三种变换，采用何种变换取决于被编码的残差数据：（1）对宏块内部（以 16X16 模式预测）的亮度直流系数 4X4 数组进行变换。（2）对（任何宏块中的）色度直流系数 2X2 数组进行变换（3）对所有其它的 4X4 块的残差数据进行变换。如果使用了“自适应块大小变换”选项，则根据运动补偿块大小（4X8，8X4，8X8，16X8 等等）选择更进一步的变换。

宏块中的数据以图 Figure 1-1 中所示顺序进行传输。如果宏块以 16X16 帧内模式编码，则标注为“-1”的块（包含每个 4X4 亮度块的直流系数）首先被传输。然后，亮度残差块 0-15 按所示顺序传输（16X16 宏块内部直流系数被设为 0）。块 16 和块 17 各自包含一个来自色度分量 Cb 和 Cr 的 2X2 排列。最后，色度残差块 18-25（直流系数为 0）被发送。

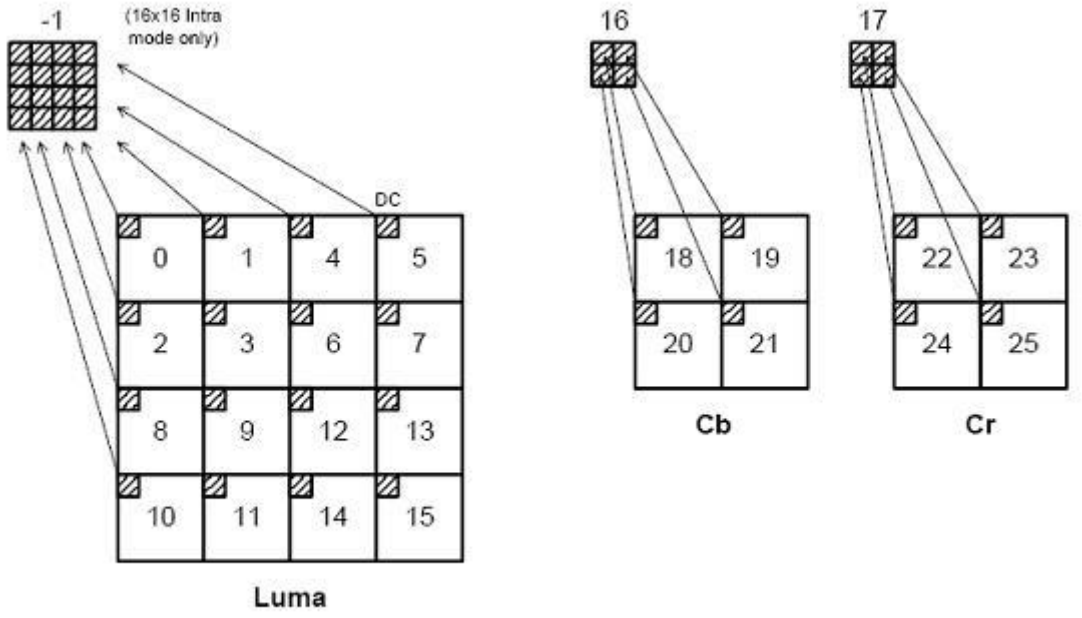


Figure 1-1 Scanning order of residual blocks within a macroblock

2. 4X4 残差变换和量化（块 0-15,18-25）

在运动补偿预测和帧内预测后对 4X4 残差数据块（在 Figure 1-1 中标志为 0-15 和 18-25）使用这个变换。这个变换基于 DCT 但是有一些基本的区别：

（1） 它是一个整数变换（所有的操作可以用整数算术进行，没有降低精度）。

（2） 逆变换在 H.264 标准中已经被充分描述，如果这个描述被正确的继承，那么编码器与解码器之间不应该存在不匹配。

（3） 变换的核心部分是没有乘法的，即它只需进行加、减和移位。

（4） 一个缩放乘法（完整变换的一部分）被融入到量化器中（减少了总的乘法次数）。

变换和量化的整个过程可以使用 16 位整数算术来进行，并且每个系数只乘一次，没有降低精度。

2．1 从 4X4 DCT 推出的整数变换

对一个输入阵列 X 进行 4X4 DCT 变换：

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

Equation 2-1

这里

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

这个矩阵乘法可以被分解成下面的等价形式(等式 Equation 2-2)

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

Equation 2-2

CXC^T(即 CXC’)是一个“核心”2-D 变换。E 是一个比例因子矩阵，并且符号⊕（圆圈中实为 X 号)表示(CXC’)的每个元素与 E 中对应位置的比例因子相乘(用点乘代替矩阵乘法）。常量 a 和 b 与之前值相等，d=c/b(近似于 0.414).

为简化变换实现过程，d 被近似成 0.5。为了确保变换仍然正交，b 也需要进行改变。所以

a=1/2　　b=(2/5)^(1/2)　　d=1/2

矩阵 C 的第 2 行和第 4 行和矩阵 C’的第 2 列和第 4 列以一个 2 为比例因子缩放（乘以 2）并且后缩放矩阵 E 也被按比例缩小来进行补偿。（这就避免了核心变换 CXC’中乘 1/2 的运算，使得使用整数算术运算不会降低精度）。最终的变换变成：

$$\mathbf{Y} = \mathbf{C}_f \mathbf{X} \mathbf{C}_f^T \otimes \mathbf{E}_f = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \mathbf{X} \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right) \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}$$

Equation 2-3

这个变换是 4X4 DCT 变换的一个近似变换。因为改变了因子 d 和 b,所以新的变换的输出不会与 4X4 DCT 变换完全相同。

逆变换的形式为：

$$\mathbf{X}' = \mathbf{C}_i^T (\mathbf{Y} \otimes \mathbf{E}_i) \mathbf{C}_i = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \left(\begin{bmatrix} \mathbf{Y} \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}$$

Equation 2-4

这次，Y 的每个系数与矩阵 Ei 中对应位置的合适加权因子相乘，以此来进行预缩放。注意矩阵 C 和 C’ 中的+/-（1/2）因子；它们可以通过一个右移来实现并且不会造成明显的精度损失，因为 Y 已经经过了预缩放。

正变换和逆变换都是正交的，即 T^-1(T(X))=X.

2.2 量化

H.264 使用了一个纯量量化器。它的定义和实现因为实际要求而变得复杂。它有以下要求：

（1）避免除法和（或）浮点算术运算。

（2）使上述的后缩放和预缩放矩阵 Ef，Ei 中的因子统一。

基本的正变换量化器操作如下：

Z_{ij} = round(Y_{ij}/Qstep)

Y_{ij} 是上述变换的一个系数，Qstep 是量化步长，Z_{ij} 是一个量化系数。

标准支持的量化步长有 52 种，量化步长根据量化参数 QP 建立索引。每个量化参数对应的量化步长的值如表 Table 2-1 所示。注意 QP 每增加 6，量化步长约增加一倍；QP 每增加 1，量化步长增加 12.5%.各种各样的量化步长使得编码器可以精确，灵活的在比特率和质量之间权衡。亮度和色度的 QP 值可能不同，虽然

两个参数的变化范围都是 0-51，但 QP_{Chroma} 是从 QPy 中得来的，QPc 比 QPy 小，QPy 的值大于 30。一个用户定义的 QPy 和 QPc 之间的偏移量或许可以从一个图像参数集中得到。

Table 2-1 Quantization step sizes in H.264 CODEC

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	....
QStep	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5	....
QP	...	18	...	24	...	30	...	36	...	42	...	48	...	51
QStep		5		10		20		40		80		160		224

后缩放因子 a^2,ab/2 或 b^2/4（Equation 2-3)是正量化器的一部分。首先，输入块 X 经过变换产生一个系数未经缩放的块 W=CXC’。然后，每个系数 Wij 被量化并且缩放（在单一操作中）。

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

PF 根据位置(i,j)来决定是为 a^2,ab/2 或是 b^2/4(如式 Equation 2-3 所示)：

Position	PF
(0,0),(2,0),(0,2),(2,2)	a^2
(1,1),(1,3),(3,1),(3,3)	b^2/4
其它	ab/2

因子（PF/Qstep）在 H.264 参考模型程序[3]中的实现方法是乘以 MF（一个乘法因子）再进行一次右移，这样就避免了除法操作。

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

这里 MF/2^qbits=PF/Qstep,qbits=15+floor(QP/6)

用整数算术，Equation 2-6 可以用下述方法实现：

$$|Z_{ij}| = (|W_{ij}|.MF + f) >> qbits$$
$$sign(Z_{ij}) = sign(W_{ij})$$

Equation 2-7

在这里>>代表二进值右移。在参考模型程序中，f 在帧内块中等于 2^qbits/3，在帧间块中等于 2^qbits/6.

例：

设 QP=4，则 Qstep=1.0

(i,j)=(0,0),则 PF=a^2=0.25

qbits=15,则 2^qbits=32768

MF/2^qbits=PF/Qstep,则 MF=（32768 X 0.25）/1=8192

根据 QP 和系数位置（i,j）值，MF 的一组值（每组 6 个）如表 Table 2-2 所示：

Table 2-2 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

表中第二列和第三列（因子为 b^2/4 和 ab/2 的位置）的值对 Equaton 2-6 的结果做了些小的修改（因为只有逆量化过程被标准化，所以为提高解码器的可感质量而改变一个前置量化器是可以接受的）。

QP>5 的时候，MF 因子保持不变（以 6 为周期重复上表）但是 QP 每增加 6，除数 2^qbits 增加一倍。例如，当 6<=QP<=11 时，qbits=16；12<=QP<=17 时，qbits=17

等等。

2.3 改变标度（逆量化）

逆量化操作如下：

$Y'_{ij} = Z_{ij}.Qstep$

Equation 2-8

逆变换中的预缩放因子（矩阵 Ei,对应于系数位置值分别为 a^2,ab 和 b^2）也是这个操作的一部分，同时增加了一个为 64 的常量比例系数来避免舍入错误。

$W'_{ij} = Z_{ij}.Qstep.PF.64$

Equation 2-9

比例系数 W'_{ij} 随后使用“核心”逆变换（Ci’WCi: 如 Equation 2-4）进行变换。逆变换的输出值被除以 64 来除去比例因子（这可以通过一次加法和一次右移来实现）。

H.264 标准没有直接指明 Qstep 或 PF，而 0<=QP<=5 时每个系数位置的参数 V=（Qstep.PF.64）被定义。逆量化操作为：

$W'_{ij} = Z_{ij}.Vij.2^{floor(QP/6)}$

Equation 2-10

例：

设 QP=3，则 Qstep=0.875,2^floor(Qp/6)=1

(i,j)=(1,2),则 PF=ab=0.3162

V=(Qstep.PF.64)=0.875X0.3162X65 =18(近似值）

W’ij=Zij X 18 X 1

0<=QP<=5 时 V 的值在标准中的定义如下表：

Table 2-3 Rescaling factor V

QP	Positions (0,0),(2,0),(2,2),(0,2)	Positions (1,1),(1,3),(3,1),(3,3)	Other positions
0	10	16	13

1	11	18	14
2	13	20	16
3	14	23	18
4	16	25	20
5	18	29	23

Equation 2-10 中的因子 2^floor(QP/6)使得重缩放输出以 2 为因子递增，QP 每增加 6,输出增加一倍。

3. 4X4 亮度 DC 系数变换和量化（只在 16X16 帧内模式中使用）

如果宏块以 16X16 帧内预测模式编码（全部的 16X16 亮度分量根据邻近像素预测），每个 4X4 残留块首先经过上述的核心变换（CfXCf^T）.然后每个 4X4 块的直流系数使用 4X4 Hadamard 变换进行再一次变换。

$$\mathbf{Y}_D = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \mathbf{W}_D \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \right) /2$$

Equation 3-1

$\mathbf{W}_D$是 4X4 直流系数块，$\mathbf{Y}_D$是变换后的块。输出系数 $Y_{D(i,j)}$被除以 2(四舍五入)。

然后对输出系数 $Y_{D(i,j)}$进行量化，产生一个量化的直流系数块：

$$|Z_{D(i,j)}| = (|Y_{D(i,j)}|.MF_{(0,0)} + 2f) >> (qbits+1)$$

$$sign(Z_{D(i,j)}) = sign(Y_{D(i,j)})$$

Equation 3-2

其中，MF，f 和 qbits 在之前定义，并且 MF 值和之前一样取决于它在 4X4DC 块中的位置（i,j）

在解码器中，首先使用逆 Hadamard 变换，随后进行逆量化（注意顺序并没有像想像中那样进行反转）。

$$\mathbf{W}_{QD} = \left(\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad \mathbf{Z}_D \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \right)$$

如果 QP 大于等于 12，则逆量化过程如下：

$W'_{D(i,j)} = W_{QD(i,j)} . V_{(0,0)} . 2^{floor(QP/6)-2}$

如果 QP 小于 12，则逆量化过程如下：

$W'_{D(i,j)} = [W_{QD(i,j)} . V_{(0,0)} + 2^{1-floor(QP/6)}] >> (2-floor(QP/6))$

V 定义同前。之后重建的直流系数 $\mathbf{W'}_D$ 被分别插入到它们对应的 4X4 块中，并且每个 4X4 系数块使用核心 DCT-based 逆变换（Ci’W’Ci）。

在一个帧内编码宏块中，大部分能量集中在直流系数上，这个补充的变换起到对 4X4 亮度直流系数去相关的作用（即利用系数间的相关性）

4. 2X2 色度直流系数变换和量化

宏块中的每个色度分量由 4 个 4X4 采样块组成。每个 4X4 块进行第 2 部分所述的变换。每个 4X4 系数块的直流系数被组合到一个 2X2 块（$\mathbf{W_D}$）并且在量化前进行进一步的变换。

$$\mathbf{Y}_D = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{W}_D \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Equation 4-1

2X2 输出块 $\mathbf{Y_D}$ 的量化的实现如下：

$|Z_{D(i,j)}| = (|Y_{D(i,j)}| . MF_{(0,0)} + 2f) >> (qbits+1)$

$sign(Z_{D(i,j)}) = sign(Y_{D(i,j)})$

Equation 4-2

其中 MF，f 和 qbits 定义如上文相同。

在解码过程中，在逆量化前进行逆变换：

$$\mathbf{W}_{QD} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{Z}_D & \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Equation 4-3

若 QP 大于等于 6，则逆量化过实现如下：

$W'_{D(i,j)} = W_{QD(i,j)}.V_{(0.0)}.2^{floor(QP/6)-1}$

若 QP 小于 6，则逆量化实现如下：

$W'_{D(i,j)} = [W_{QD(i,j)}.V_{(0,0)}]>>1$

重建的系数分别替换到对应的 4X4 色度系数块中，然后做上文所述变换(Ci'W'Ci).和帧内亮度直流系数一样，这个补充的变换起到对 2X2 色度直流系数去相关的作用，以此来提高压缩效率。

5. 完整的变换，量化，逆量化和逆变换过程

输入残留块 X 到输出残留块 X'的整个过程如下所述，并在图 Figure 5-1 中描述。

编码：

1．输入：4X4 残留块： X

2. 正"核心"变换： W=CfXCf'

(随后对色度直流系数或 16X16 帧内模式亮度直流系数进行进一步变换)

3．后缩放和量化： Z=W.PF/(Qstep.2^qbits)

（对色度直流系数和 16X16 帧内模式亮度直流系数，此式做了修改）

解码：

（对色度直流系数和 16X16 帧内模式亮度直流系数做逆变换）

4．逆量化（融合了逆变换和预缩放）：

W'=Z.Qstep.PF.64

（对色度直流系数和 16X16 帧内模式亮度直流系数，此式做了修改）

5．逆核心变换 X'=Ci'W'Ci

6．后缩放：　　X’’=round(X’/64)

7．输出：4X4 残留采样块：　　X’’

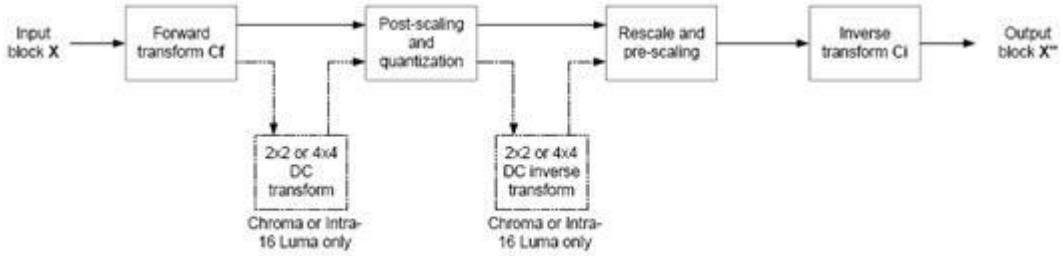


Figure 5-1 Transform, quantization, rescale and inverse transform flow diagram

例：（4X4 亮度残留块，帧间模式）：

QP=10

输入块 X 为：

	j=0	1	2	3
i=0	5	11	8	10
1	9	8	4	12
2	1	10	11	4
3	19	6	15	7

核心变换的输出 W 为：

	j=0	1	2	3
i=0	140	-1	-6	7
1	-19	-39	7	-92
2	22	17	8	31
3	-27	-32	-59	-21

MF=8192，3355 或 5243(取决于系数位置)，qbits=16。正量化器输出 Z 为：

	j=0	1	2	3
i=0	17	0	-1	0
1	-1	-2	0	-5
2	3	1	1	2
3	-2	-1	-5	-1

V=16,25 或 20（取决于系数位置），2^floor(QP/6)=2。逆量化输出 W'为：

	j=0	1	2	3
i=0	544	0	-32	0
1	-40	-100	0	-250
2	96	40	32	80
3	-80	-50	-200	-50

核心逆变换输出 X''（ 除 64 并取整后）：

	j=0	1	2	3
i=0	4	13	8	10
1	8	8	4	12
2	1	10	10	3
3	18	5	14	7

6. 参考资料

1 ITU-T Rec. H.264 / ISO/IEC 11496-10, "Advanced Video Coding", Final Committee Draft, Document JVTF100,December 2002

2 A. Hallapuro and M. Karczewicz, "Low complexity transform and quantization – Part 1: Basic Implementation", JVT document JVT-B038, February 2001

3 JVT Reference Software version 4.0, ftp://ftp.imtc-files.org/jvt-experts/reference_software/

重建滤镜

1. 引言

联合视频工作组（JVT）正在定案一个新的自然视频图像编码（压缩）标准。新标准被称为 H.264 或称作 MPEG-4 Part 10、"高级视频编码（AVS）"。这篇文档描述了 H.264 编码解码器中过滤重建块的方法。注意 H.264 草案标准现在尚未定稿，所以鼓励读者参考最新版本的标准。

2. 什么是重建滤镜

滤镜是为了减小块失真而应用到所胡已解码宏块上。抗块效应滤波器在下述两种情况下使用：（1）编码过程：逆变换之后使用（在重建之前并为了之后的预测而存储宏块）（2）解码过程：重建之前并显示宏块。重建滤镜有两个作用：（1）使块边缘平滑，提高解码图像质量（特别是在高压缩比情况下）；（2）过滤的

宏块用于编码器中另外的帧的运动补偿预测，在预测后产生一个更小的残留帧。（注意：帧内编码宏块被过滤，而帧内预测则是使用未经过滤的 d 重建的宏块产生预测。）图片边缘没有被过滤。

以下面的顺序对宏块中的 4X4 块的垂直和水平边缘使用过滤：

1. 对亮度分量的 4 个垂直边界进行过滤（以 Figure 1 中的 a,b,c,d 序）

2. 对亮度分量的 4 个水平边界进行过滤（以 Figure 1 中的 e,f,g,h 序）

3. 对每个色度分量的 2 个垂直边界进行过滤（i,j 序）

4. 对每个色度分量的 2 个水平边界进行过滤（k,l 序）

每个过滤操作最多对边界两边的三个像素起作用。Figure 2 显示了相邻块 p 和 q 间垂直或水平边界两边的各 4 个像素（p0,p1,p2,p3 和 q0,q1,q2,q3）。这可以有多种输出，从（a）没有像素被过滤到（b）p0,p1,p2,q0,q1,q2 都被过滤并产生像素 P0,P1,P2,Q0,Q1,Q2，这取决于当前的量化器、相邻块的编码模式和跨边界的图像采样的梯度。

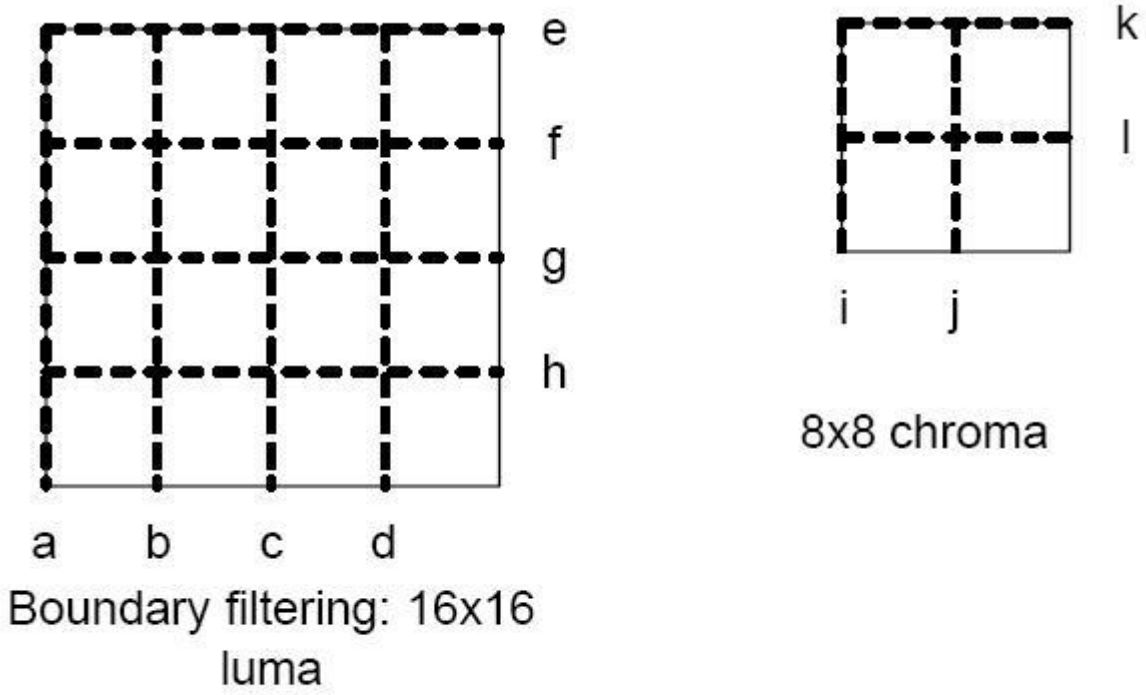


Figure 1 Edge filtering order in a macroblock

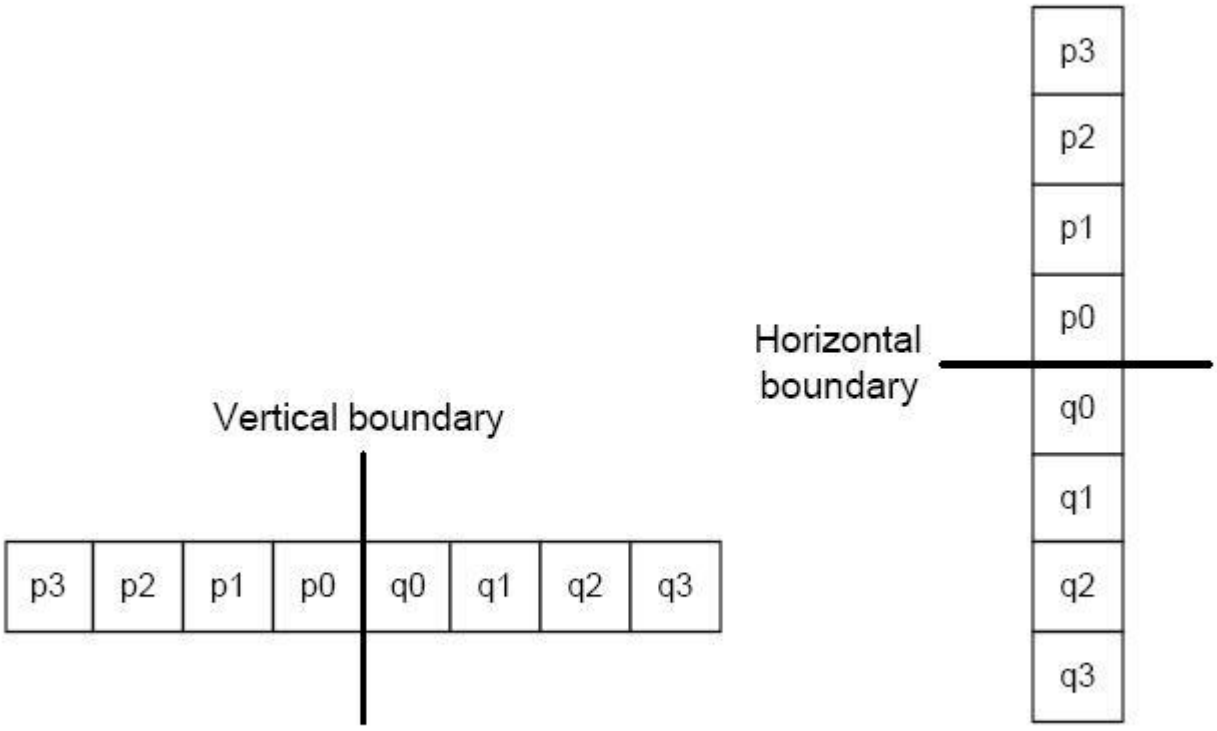


Figure 2 Pixels adjacent to vertical and horizontal boundaries

3 边界强度

过滤的输出取决于边界强度和跨边界的图像采样的梯度。根据下面规则来选择边界强度系数 Bs：

p 或 q 采用帧内编码并且边界是一个宏块边界（即宏块最外侧边界）	Bs=4（过滤强度最大）
p 或 q 采用帧内编码并且边界不是一个宏块边界	Bs=3
p 和 q 都不采用帧内编码但 p 或 q 包含编码系数	Bs=2
p 和 q 都不采用帧内编码且都不包含编码系数，p 和 q 有不同的参考帧或参考帧号不同或有着不同的运动矢量值。	Bs=1
p 和 q 都不采用帧内编码并且都不包含编码系数；p 和 q 有相同的参考 帧和相同的运动矢量。	Bs=0（未过滤）

在可能会产生很强失真的地方，过滤强度最大。如一些帧内编码宏块的边界或包含编码系数的块之间的边界。

5. 滤镜决策

集合（p2,p1,p0,q0,q1,q2）中的一组采样只有满足下列条件时才会被过滤：

（1） Bs>0 并且

（2） |p0-q0|,|p1-p0|和|q1-q0|都小于一个阀值？或？（？和 ？在标准[1]中定义）

阀值？和 ？随着块 p 和 q 的量化参数 QP 的平均值增加而增加。滤镜决策的目的是在原始图像跨越边界时，梯度没有明显改变的情况下关闭滤镜。明显改变的定义取决于 QP。当 QP 比较小的时候，除了非常微小的跨边界梯度，其它的都被看做是就应该保护的图像特征，这样阀值？和？就比较小。当 QP 比较大的时候，块失真可能会很明显，这样？和？就比较大，这样就会产生更多的过滤。

5. 滤镜的实现

（1） Bs ？{1,2,3};

对 p1,p0,q0 和 q1 使用一个 4-输入线性滤镜，过滤的的输出为 P0 和 Q0（0<Bs<4）

另外，如果|p2-p0|比阀值？小，则对 p2,p1,p0 和 q0 使用 4-输入线性滤镜进行过滤，过滤输出 P1。如果|q2-q0|比阀值？小，则对 q2,q1,q0 和 p0 使用 4-输入线性滤镜进行过滤，过滤输出 Q1。（在色度分量中 p1 和 q1 不会被过滤，只在亮度分量中使用）。

（2） Bs=4:

if |p2-p0|<?&&|p0-q0|<round(?/4) then：

使用一个 5-输入滤镜对 p2,p1,p0,q0 和 q1 进行过滤，输出 P0；

使用一个 4-输入滤镜对 p2,p1,p0 和 q0 进行过滤，输出 P1；

（只使用于亮度分量）使用一个 5 输入滤镜对 p3,p2,p1,p0 和 q0 进行过滤，输出 P2

else：

使用一个 3 输入滤镜对 p1,p0 和 q1 进行过滤，输出 P0；

if |q2-q0|<?&&|p0-q0|<round (?/4) then:

对 q2,q1,q0,p0 和 p1 使用一个 5 输入滤镜进行过滤，输出 Q0；

对 q2,q1,q0 和 p0 使用一个 4 输入滤镜进行过滤，输出 Q1;

(只使用于亮度分量)对 q3,q2,q1,q0 和 p0 使用一个 5 输入滤镜进行过滤，输出 Q2

else:

对 q1,q0 和 p1 使用一个 3 输入滤镜进行过滤，输出 Q0

6. 过滤实例

一个 QCIF 视频片段使用 AVC 参考程序以固定量化参数 32 编码。Figure 3 显示了视频片断的原始帧；Figure 4 显示了禁用环路滤镜时，帧间编码和重构后的相同帧。Figure 5 使用了环路滤镜时帧画面，显示效果大有改善。这里仍然有一些失真，但大部分块边缘已经消失或是减淡了。注意对比度明显的边界被滤镜保存，而同时图像平滑区域的块边界被平滑。

Figure 6 显示了一个有着更高 QP（36）的解码帧，此时禁用了环路滤镜（注意增加了大量伪像），Figure 7 显示了相同帧，使用环路滤镜。

Figure 3 Original frame

Figure 4 Reconstructed, QP=32 (no filter)

Figure 5 Reconstructed, QP=32 (with filter)

Figure 6 Reconstructed, QP=36 (no filter)

Figure 7 Reconstructed, QP=36 (with filter)

7. 参考资料

1 ITU-T Rec. H.264 / ISO/IEC 11496-10, “Advanced Video Coding”, Final Committee Draft, Document JVT G050, March 2003