第3章

图像与视频编码技术

本章视频资料

本章学习目标

- 熟练掌握图像与视频压缩编码的基本方法。
- 熟悉 JPEG、H.264、H.265 等图像与视频国际编码标准和编码方法。
- 了解码流分析工具的使用方法。

本章首先介绍图像与视频压缩编码的基本方法，再介绍 JPEG、H.264、H.265 等图像与视频国际编码标准，最后介绍码流分析工具的使用方法。

3.1 图像与视频编码基础

3.1.1 图像与视频编码原理概述

人们主要通过视觉感知外部世界。心理学实验证实，人类获取的信息 83%来自视觉。因此，汉语中有“眼见为实”的说法，英语中也有“Seeing is believing”的说法，这两句话都符合科学事实。正因为如此，在这个信息时代，与视觉相关的应用往往受到用户的广泛欢迎。涉及图像视频的应用已经渗透到现代人类社会的方方面面，如数字电视、视频会议、电子游戏、拍照摄像、网络直播、车载传感、医疗设备甚至是最近的热门话题——元宇宙等，数字视频是这些应用中的主角。

但是，人们在实际应用中接触到的视频都是压缩视频。这是因为未压缩的原始视频的数据量非常大，根本无法直接用于实际传输或存储。因此，视频应用的关键技术是视频编码（video coding），也称为视频压缩（video compression），其目的是尽可能去除视频数据中的冗余成分，减小视频的数据量。

视频编码主要是通过去除视频中的空间冗余、时间冗余和编码冗余实现的。具体地讲，视频编码器中包括很多编码算法，这些算法在视频编码器中被有效地组合在一起，使整个视频编码器具有较高的压缩效率。目前主流的视频编码器采用的技术主要有预测、变换、量化、熵编码等，这些技术在视频编码器中的基本次序关系如图 3.1 所示。

1. 预测编码

预测编码（prediction coding）是视频编码的核心技术之一。对于视频序列来

图 3.1 视频编码关键技术的基本次序关系

说,其空域和时域有着很强的相关性。这样就可以根据已编码的一个或几个样本值,利用某种模型或方法对当前的样本值进行预测,并对样本真实值和预测值之间的差值进行编码。预测编码最早的系统模型是 1952 年贝尔实验室的 Culter 等实现的差值脉冲编码调制(Differential Pulse Code Modulation,DPCM)系统,其基本思想是:不直接对信号进行编码,而是用前一信号对当前信号作出预测,对当前信号与预测值的差值进行编码传送。同年,Oliver 和 Harrison 将 DPCM 技术应用到视频编码中。DPCM 技术在视频编码中的应用分为帧内预测技术及帧间预测技术,分别用于消除空域冗余及时域冗余。

帧内预测利用图像在空间上相邻像素之间具有相关性的特点,由已编码的相邻像素预测当前块的像素值,可以有效地去除块间冗余。1952 年,Harrison 首先对帧内预测技术进行了研究,其方法是用已编码像素的加权和作为当前像素的预测值,这一基本思想在无损图像编码标准 JPEG-LS 的 LOCO-I 算法中得到了应用。该方法虽然简单易行,但是难以获得更高的压缩率。随着离散余弦变换(Discrete Cosine Transform,DCT)在图像与视频编码方面的广泛应用,帧内预测转变为在频域实现。DCT 技术被广泛应用于早期的一些图像与视频编码标准当中,如 JPEG、H.261、MPEG-1 等。但是 DCT 技术的性质决定了它只能反映当前块内像素的取值,无法体现出图像、视频的纹理信息。在现代视频编码中,采用了基于块空域帧内预测技术,包含多个预测模式,每个预测模式对应一种预测方向,按照图像本身的特点选择一个最佳的预测方向,最大限度地去除空间冗余。多方向空间预测技术与 DCT 技术相结合,可以弥补 DCT 技术只能去除块内冗余的缺点,获得较高的编码性能。基于块的帧内预测技术在现代视频编码标准中的应用有 MPEG-4 标准中相邻块的频域系数预测,(如 DC 预测及 AC 预测)以及 H.264/AVC、H.265/HEVC、AVS 标准中的多方向空间预测技术。

帧间预测是消除运动图像时间冗余的技术。Seyler 在 1962 年发表的关于帧间预测编码的研究论文奠定了现代帧间预测编码的基础。他提出视频序列相邻帧间存在很强的相关性,因此只需要对视频序列相邻帧间的差异进行编码,并指出相邻帧间的差异是由于物体的移动、摄像机镜头的摇动及场景切换等造成的。在此之后,帧间预测技术的发展经历了条件更新、3D-DPCM、基于像素的运动补偿等几个阶段,最终从有效性及可实现性两方面综合考虑,确定了基于块的运动补偿方案。现代视频编码系统都采用了基于块的运动补偿的帧间预测技术,用于消除时域冗余。

由于视频相邻帧中的场景存在着一定的相关性,因此可为当前块搜索出在相邻帧中最相似的参考块,该过程被称为运动估计(Motion Estimation,ME),并可根据参考块的位置得出两者在空间位置上的相对偏移量,即通常所说的运动矢量(Motion Vector,MV),当前块与参考块的像素差值被称为预测残差(Prediction Residual,PR)。根据 MV 得到的参考块对当前块进行预测的过程称为运动补偿(Motion Compensation,MC),MC 得到的预测值加上预测残差,就得到了最终的重建值。

2. 变换

变换技术对图像进行正交变换以去除空间像素之间的相关性,也就是变换后的频域系

数使图像信息的表示更加紧凑，这有利于编码压缩。另一方面，正交变换使得原先分布在每个像素上的信息集中到频域的少数几个低频系数上，这代表了图像的大部分信息；而高频系数值较小，这是与大多数图像的高频信息较少相一致的。频域系数的这种性质有利于采用基于人类视觉特性的量化方法，例如，对低频系数采用小的量化步长以保持大部分信息不丢失，而对高频系数量化步长大一些，虽然信息损失较多，但人的视觉系统对这部分信息损失不敏感。

K-L(Karhunen-Loeve)变换是均方误差标准下的最佳变换，但其计算复杂度高，需要针对每个输入图像计算特征向量，从而获得变换矩阵，不适合对实时性要求较高的视频编码系统，很难在实际应用中被采用，并且变换矩阵需要转送到解码端，这额外增加了传输的开销。后来人们退而求其次，寻找不必每次都要计算变换矩阵的正交变换方法。使用正交变换的原因是，正交变换的转置矩阵和逆矩阵是相等的，这在逆变换(解码)时非常方便。人们开始尝试使用快速傅里叶变换(Fast Fourier Transform，FFT)，后来发现，对于图像数据压缩这个特定问题，由于图像数据是非负的，在傅里叶空间中只有第一象限被涉及，表现效率低。

随后，人们采用离散余弦变换(DCT)代替 K-L 变换，取得了很好的效果。DCT 不依赖于输入信号的统计特性，而且有快速算法，因此得到了广泛应用。考虑到实现的复杂性，不是对整幅图像直接进行变换，而是把图像分成不重叠的固定大小块，对每个图像块进行变换。MPEG-2、H.263 以及 MPEG-4 都采用了 8×8 DCT。这些标准中的 DCT 技术采用了浮点 DCT 实现，浮点计算会引入较高的运算量，同时如果对浮点精度不作规定，解码器会出现误差漂移。人们又提出了用整数 DCT 技术解决这个问题，同时整数 DCT 只需加法和移位操作即可实现，计算复杂度低。最新的 H.264/AVC 及 AVS 标准都采用了整数 DCT 技术。DCT 技术的另一个重要进展是 H.264/AVC 标准制定过程中出现了自适应块大小变换技术(Adaptive Block-size Transforms，ABT)。ABT 的主要思想是用与预测块相同尺寸的变换矩阵对预测残差去相关，这样不同块尺寸的预测残差系数的相关性都可以被充分地利用。ABT 技术可以使编码效率提高 1dB。

变换技术的另一个重要进展是离散小波变换(Discrete Wavelet Transform，DWT)技术，DWT 具有多分辨率多频率时频分析的特性，信号经 DWT 分解为不同频率的子带后更易于编码，并且采用适当的熵编码技术，码流自然地具有嵌入式特性。JPEG2000 图像编码标准建立在 DWT 技术之上，MPEG-4 标准也采用 DWT 技术对纹理信息进行编码。此外，采用 DWT 技术的视频编码方案也得到了深入研究。

除了 DCT 和 DWT 外，视频编码标准中常用的变换还有哈达玛(Hadamard)变换，主要用于空域去相关编码。理论上，哈达玛变换比快速傅里叶变换更利于小块的压缩，但会产生更多的块效应。由于哈达玛变换的计算复杂度较低，仅仅需要加减操作就可以实现，因此常在运动估计或模式决策中被用来替代 DCT，得到和 DCT 相近的决策结果，再根据决策结果用 DCT 进行编码。在 H.265/HEVC 标准中，针对帧内预测残差系数的相关性分布，离散正弦变换(Discrete Sine Transform，DST)比 DCT 具有更好的去相关性能。

3. 量化

量化是降低数据表示精度的过程，通过量化可以减少需要编码的数据量，达到压缩数据的目的。量化可分为矢量量化和标量量化两种。矢量量化是对一组数据联合量化。标量量化独立量化每一个输入数据，标量量化也是一维的矢量量化。根据香农提出的信息率失真

理论,对于无记忆信源,矢量量化编码总是优于标量量化编码,但设计高效的矢量编码码本却是十分复杂的问题,因此当前的编码标准通常采用标量量化。由于 DCT 具有能量集中的特性,变换系数的大部分能量都集中在低频范围,只有很少的能量落在高频范围。利用人的视觉系统对高频信息不敏感的特点,通过量化可以减小高频的非零系数,提高压缩效率。

量化是一种有损压缩技术,量化后的视频图像不能进行无损恢复,因此导致源图像与重建图像之间的误差,称为失真。编码图像的失真主要是由于量化引起的,失真是量化步长的函数。量化步长越大,量化后的非零系数越少,视频压缩率越高,但重建图像的失真也越大。从这里可以看出,图像质量和压缩率是一对矛盾。通过在量化阶段调整量化步长,可控制视频编码码率和编码图像质量,根据不同应用的需要,在两者之间进行选择和平衡。

4. 熵编码

变换量化系数在熵编码之前通常要通过 Z 形(Zigzag)扫描将二维变换量化系数重新组织为一维系数序列,经过重排序的一维系数再经过有效的组织能够被高效编码。如图 3.2 所示,给出了 8×8 变换量化系数块的 Z 形扫描的顺序。扫描的顺序一般根据待编码的非零系数分布,按照空间位置出现非零系数的概率从大到小排序。排序的结果是使非零系数尽可能出现在整个一维系数序列前面,而后面的系数尽可能为零或者接近零,这样排序非常利于提高系数的熵编码效率。基于这一原则,在 H.265/HEVC 标准中,针对帧内预测块的系数分布特性,还专门设计了垂直、水平和对角等新的扫描方式。

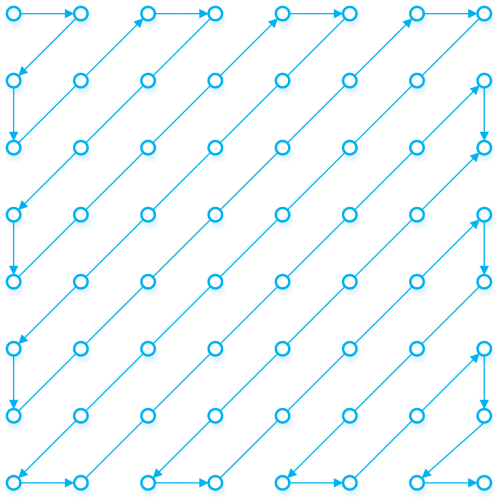


图 3.2 8×8 变换量化系数块的 Z 形扫描顺序

利用信源的信息熵进行码率压缩的编码方式称为熵编码,它能够去除经预测和变换后依然存在的统计冗余信息。视频编码常用的熵编码方法有两种:变长编码(Variable Length Coding,VLC)和算术编码(Arithmetic Coding,AC)。变长编码的基本思想是:为出现概率大的符号分配短码字,为出现概率小的符号分配长码字,从而达到总体平均码字最短。1971 年,Tasto 和 Wintz 首次将熵编码应用到图像编码中。在他们的方案中,对量化后的每个系数进行变长编码。1976 年,Tescher 在他的自适应变换编码方案中首次提出了 DCT 系数的高效组织方式,即 Z 形扫描。Chen 在 1981 年利用哈夫曼码构造了两个变长码表,分别用于扫描产生的非零系数和连续零系数游程的编码。1986 年,Chen 又采用变长码

联合编码非零系数与零系数游程，这一方法被称为 2D-VLC。这是利用联合熵提高熵编码效率的一个实例，这一技术被应用到 H.261、MPEG-1 及 MPEG-2 标准中。在 H.263 及 MPEG-4 标准中，采用了 3D-VLC，非零系数与零系数游程以及是否是最后一个非零系数的信息进行联合编码。对于给定的信源及其概率分布，哈夫曼编码是最佳编码方法。哈夫曼编码用于视频编码有两个缺点：一个是编码器建立哈夫曼树的计算开销巨大；另一个是编码器需要向解码器传送哈夫曼码字表，解码器才能正确解码，这会降低压缩效率。因此，实际应用中常使用有规则结构的指数哥伦布码（Exp-Golomb Code，EGC）代替哈夫曼编码。对于服从一般高斯分布的符号编码，指数哥伦布码的编码性能不如哈夫曼编码，但因为指数哥伦布码的码字结构对称，编解码复杂度较低，容易在编解码器中实现，所以被广泛采用。

算术编码是另一类重要的熵编码方法，在平均意义上可为单个符号分配码长小于 1 的码字，通常算术编码具有比变长编码更高的编码效率。算术编码和变长编码不同，不是采用一个码字代表一个输入信息符号的方法，而是采用一个浮点数代替一串输入符号。算术编码计算输入符号序列的联合概率，将输入符号序列映射为实数轴上的一个小区间，区间的宽度等于该序列的概率值，然后在此区间内选择一个有效的二进制小数作为整个符号序列的编码码字。可以看到，算术编码是对输入符号序列进行操作，而非单个符号，因此在平均意义上可以为单个符号分配长度小于 1 的码字。算术编码的思想在香农信息论中就已提出，但直到 1979 年才由 Rissanen 和 Langdon 将算术编码系统化。由于算术编码对当前符号的编码需要依赖前一个编码符号的信息，因此很难并行实现，计算复杂度较高。

熵编码技术应用于视频编码的一次技术革新是在 H.264/AVC 标准的制定过程中引入了上下文自适应技术。在编码过程中，熵编码器利用上下文信息自主切换码表或更新符号的条件概率，这较好地解决了以往熵编码技术中全局统计概率分布与编码符号局部概率分布不一致的问题，因此编码效率进一步提高。基于上下文的熵编码由上下文建模与编码两个技术模块构成。上下文建模挖掘了高阶条件熵，因此提高了编码效率。比较典型的基于上下文的熵编码方法包括无损图像编码中的 LOCO-I 与 CALIC、JPEG2000 标准中的 EBCOT、AVS 视频编码标准中的 C2DVLC 与 CBAC、H.264/AVC 标准中的 CAVLC 与 CABAC 等。编码可通过变长编码或算术编码实现。

3.1.2　视频编码框架与基本概念

1. 视频编码数据组织

下面介绍视频编码中常见的基本概念与常见术语。

（1）帧组（Group of Pictures，GOP）。在视频编码中，GOP 指定了帧内和帧间的排列顺序。GOP 是编码视频流中的一组连续图像，每个编码视频流由连续的 GOP 组成，从其中包含的图像生成可见帧。

（2）I 帧、B 帧、P 帧。编码标准通常将画面（即帧）分为 I 帧、P 帧、B 帧 3 种，I 是内部编码帧，P 是前向预测帧，B 是双向内插帧。简单地讲，I 帧是一个内部独立编码的画面，而 P 帧和 B 帧记录的是相对于 I 帧的变化。没有 I 帧，P 帧和 B 帧就无法解码。GOP 越长，B 帧所占比例越高，编码的率失真（rate distortion）越大。

（3）宏块和块。宏块是运动补偿的基本单元，块是 DCT 的基本单元。

（4）量化参数（Quantization Parameter，QP）。量化参数通常在编码器参数设定时选

用。量化参数和量化步长息息相关。量化参数越小,量化步长越小,代表精度越高;反之精度越低。

(5) 码率(bit rate)。数据传输时单位时间传输的数据位数,一般采用的单位是 kb/s,即千位每秒。

(6) 压缩比。视频文件压缩前和压缩后文件大小的比值。

(7) 传输流(transport stream)。它是将一个节目的多个组成部分按照其相互关系进行组织,加入各组成部分关系描述和节目组成信息,并进一步封装成传输包后的码流。传输流是将视频、音频、PSI(Program Specific Information,节目特定信息)等数据打包进行传送。传输流主要用于节目传输。传输流的传输包长度固定,一般为 188B。

2. 视频编码器结构(以 MPEG-2 为例)

MPEG-2 视频编码标准支持对不同格式的数字视频进行不同复杂度的压缩编码处理,因此它的应用范围十分广泛。针对不同的应用要求,MPEG-2 标准规定了 4 种输入视频格式,称之为级(level),分别为低级(LL,格式为 352×288×25 帧或 352×248×30 帧)、主级(ML,格式为 720×576×25 帧或 720×480×30 帧)、1440 高级(H1440L,格式为 1440×1080×25 帧或 1440×1080×30 帧)和高级(HL,格式为 1920×1080×25 帧或 1920×1080×30 帧)。

针对不同复杂度的压缩编码处理要求,MPEG-2 标准定义了 5 种不同的压缩编码处理类型,简称为类(Profile),分别为简单类(Simple Profile,SP)、主类(Main Profile,MP)、信噪比可分级类(SNR Scalable Profile,SNRP)、空间可分级类(Space Scalable Profile,SSP)和高类(High Profile,HP)。

MPEG-2 视频编码器也采用了基于运动补偿和变换编码的混合型压缩编码结构,其基本模块包含了采用 DCT 的变换编码、非线性量化器、相邻帧的运动预测以及采用哈夫曼编码和游程编码的熵编码等基本单元。图 3.3 是 MPEG-2 视频编码器的工作原理框图。实践证明,对于主级和主类(720×576×25 帧),在压缩比为 30∶1 或更低时,可以提供广播质量的编码图像。

经过压缩编码后的视频信号形成视频基本码流(Elementary Stream,ES),MPEG-2 标准的视频基本码流可分成 6 个层次,从高往低依次是视频序列层(Sequence)、图像组层(GOP)、图像层(Picture)、像条层(Slice)、宏块层(Macro Block)和像块层(Block)。MPEG-2 标准的视频基本码流的帧结构如图 3.4 所示。图 3.4 中的 SC 是起始码(Start Code),PIC 是图像。

视频编码器和音频编码器输出的码流分别为视频基本流和音频基本流,即 ES。ES 再经过打包后输出的是包基本流,即 PES (Packet Elementary Stream)。包基本流的包长度是可变的,视频通常是一帧(即一幅图像)一个包;音频包长度通常为一个音频帧,不超过 64KB。

为了把同一个电视节目的视频、音频和其他数据合成为一路节目流进行传输,需要将视频基本流、音频基本流和其他数据流进行合成,这一过程称为单节目复用。单节目复用的结果可以形成两种不同结构的码流:一种称为节目流,即 PS (Program Stream);另一种称为传输流,即 TS (Transport Stream)。

MPEG-2 传输流结构是为系统复用和传输所定义的,属于系统传输层结构中的一种。

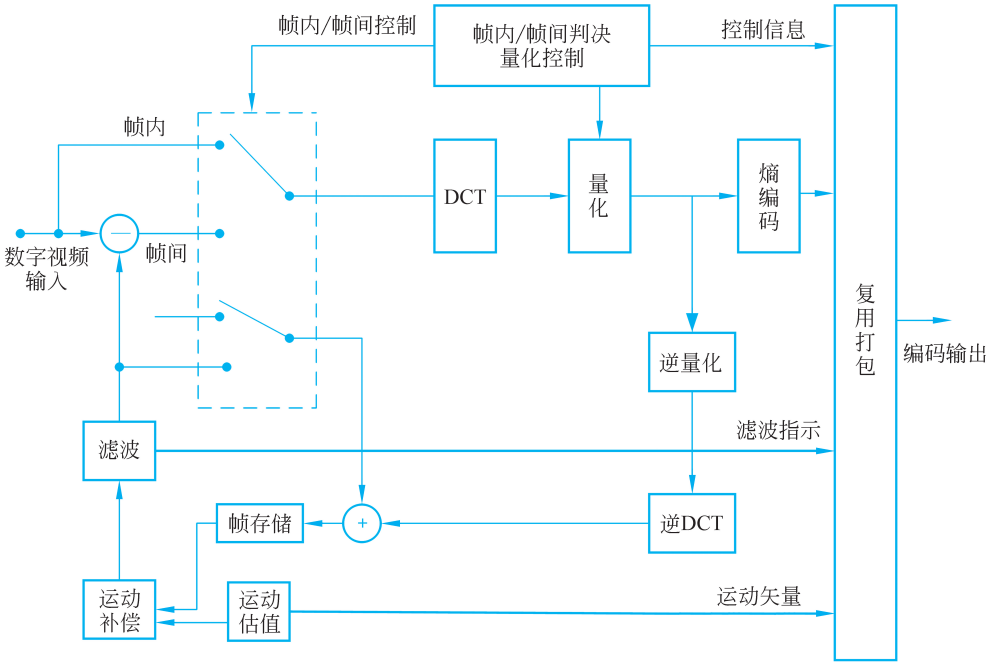


图 3.3　MPEG-2 视频编码器的工作原理框图

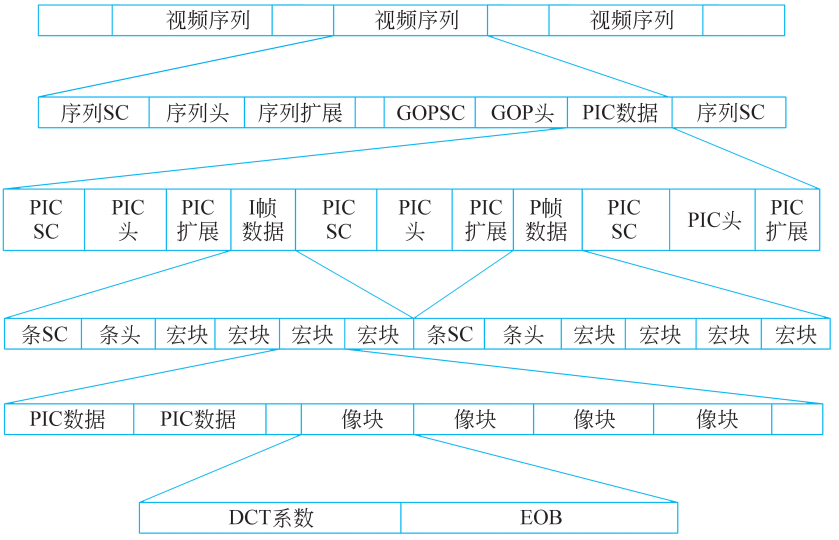


图 3.4　MPEG-2 标准的视频基本码流的帧结构

通过与 MPEG-2 系统时序模型的建立、节目特殊信息（PSI）及服务信息（Service Information，SI）共同作用实现在恶劣的信道环境中灵活可靠的复用、传输与解复用。MPEG-2 系统部分给出了多路音频、视频的复用和同步标准。MPEG-2 系统传输层的结构可以用图 3.5 描述。

数字视频和音频分别经过视频编码器和音频编码器编码之后，生成视频基本流和音频基本流。在视频基本流中还要加入一个时间基准，即 27MHz 时钟信息。然后，再分别通过各自的打包器将相应的基本流转换为包基本流。最后，节目复用器和传输复用器分别将视

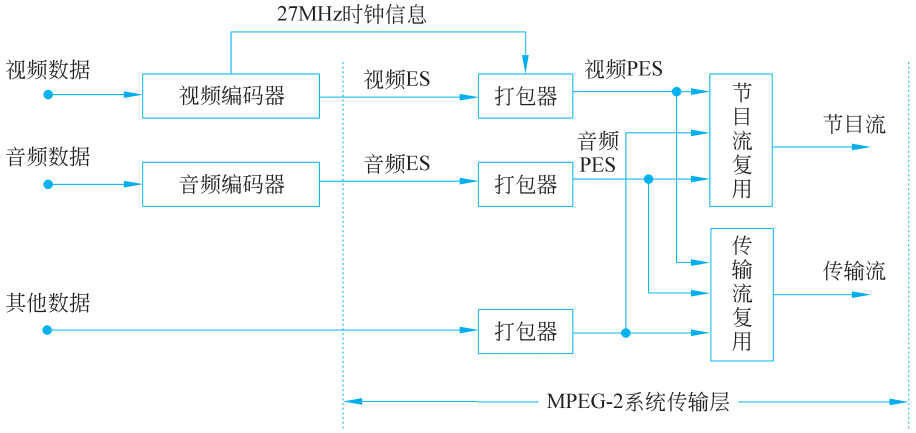


图 3.5 MPEG-2 系统传输层的结构

频 PES、音频 PES 及经过打包的其他数据组合成相应的节目流和传输流。

传输流的系统层可分作两个子层：一个对应特定数据流操作(PES 分组层，可变长度)，该层是为编解码的控制而定义的逻辑结构；另一个对应多路复用操作(TS 分组层，188B 固定长度)，该层是针对交换和互操作而定义的。在传输流的包头中加入同步信息，说明有无差错和加扰，并加入连续计数、不连续性指示、节目参考时钟(PCR)以及包 ID(PID)等。传输流的包结构如图 3.6 所示，由包头、调整字段(自适应区)和有效负载(包数据)3 部分组成。每个包长度为固定的 188B，包头占 4B，调整字段和有效负载共占 184B。

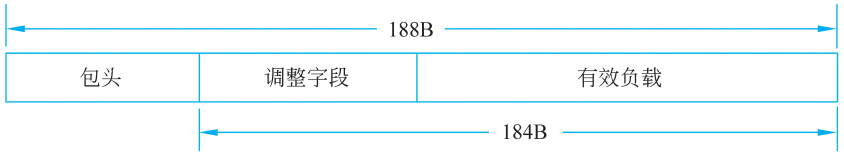


图 3.6 传输流的包结构

3.1.3 视频编码标准发展历程

自 20 世纪 80 年代起，一些国际组织就开始着手建立一套规范的、国际通用的视频编码标准。目前，国际上制定视频编码标准的组织主要包括国际电信联盟电信标准化部门(International Telecommunication Union-Telecommunication Standardization Sector，ITU-T)、国际标准化组织(International Organization for Standardization，ISO)、国际电工委员会(Intenational Electrotechnical Commission，IEC)。ITU-T 制定的视频编码标准就是 H.26x 系列，被广泛应用于基于网络传输的视频通信。人们熟知的 MPEG 系列视频编码标准是由 ISO/IEC 的动态图像专家组(Moving Picture Experts Group，MPEG)制定的，主要应用于视频存储(如 VCD/DVD)、广播电视、网络流媒体等。ITU-T 视频编码专家组(Video Coding Experts Group，VCEG)与 ISO/IEC 在视频编码标准制定中也有多次合作。例如，H.262/MPEG-2 标准，成为当时 DVD 的核心技术；H.264/AVC 视频编码标准，在视频广播、视频存储、交互式视频等领域得到了广泛的应用；更有后来著名的 H.265/HEVC 视频编码标准，获得了突出的压缩性能，正在被广泛应用。新一代通用视频编码标准 H.266/VVC 由双方联合制定。

2002 年，我国成立了数字音视频编解码技术标准工作组，也叫 AVS 工作组，制定了具有我国自主知识产权的音视频编码标准（AVS），并在 2006 年正式成为国家标准。随着广电高清数字广播的不断发展，2012 年，我国成立了 AVS 技术应用联合推进工作组，并在 2016 年完成了第二代视频编码标准（AVS2），在 2019 年完成了第三代视频编码标准（AVS3 Phase 1），在 2021 年完成了 AVS3 第二阶段标准（AVS3 Phase 2）。

视频编码领域的另一大标准制定者是由谷歌公司于 2015 年主导的开放媒体联盟（Alliance for Open Media，AOM），旨在建立一个开发开放式、无版权费的视频编码标准。在开源编解码器 VP9 的基础上，2018 年年底，AOM 完成了 AV1 视频编码格式（标准），其性能优于 x265 编码器，并且还在不断优化，已达到近似于 VP9 的实现复杂度要求。目前，AOM 正在组织开发下一代视频编码标准 AV2。

下面概述各大视频编码标准的发展历程。

1. H.26x 系列标准

H.261 标准是 ITU-T 在 1990 年制定的数字视频编码标准，针对的是基于综合业务数字网（Integrated Services Digital Network，ISDN）的视频通信应用，如可视电话、视频会议等。另外，H.261 还针对世界各国不同的电视制式提出了一种通用中间格式（Common Intermediate Format，CIF）以解决不同制式的格式转换问题。H.261 主要采用了基于运动补偿的帧间预测、DCT、量化、Z 形扫描和熵编码等。这些构成了混合编码（hybrid coding）框架，被认为是混合编码标准的鼻祖并沿用至今。

H.263 标准也由 ITU-T 制定，最初是为低码率的视频会议应用而设计的，后期证明 H.263 不局限于低码率传输环境，还适用于很大范围的动态码率。H.263 标准仍以混合编码框架为核心，其基本原理、原始数据和码流组织都与 H.261 相差无几，但与此同时也吸纳了 MPEG 系列标准等其他一些国际标准技术。成功应用于基于 H.323 标准的视频会议系统以及基于 H.320、RTSP（Real Time Streaming Protocol，实时流协议）和 SIP（Session Initiation Protocol，会话初始协议）标准的视频通信系统。

2. MPEG 系列标准

MPEG-1 标准是 MPEG 制定的第一个视频和音频有损压缩标准，也是最早推出及在市场上应用的 MPEG 技术。当初，它主要是针对数字存储媒体（如 CD）记录活动图像及其伴音的编码方式。MPEG-1 标准后来成为影音光碟（即 VCD）的核心技术。

MPEG-2 标准是继 MPEG-1 标准制定之后由 MPEG 推出的音视频编码标准，于 1994 年面世。MPEG-2 标准的应用领域包括卫星电视、有线电视等，经过少量修改后，成为广为人知的 DVD 产品的核心技术。前面曾提到，MPEG-2 视频编码标准（MPEG-2 标准第 2 部分）事实上是由 MPEG 和 ITU-T 联合制定的，ITU-T 的 H.262 与 MPEG-2 视频编码标准是完全相同的。不过，MPEG-2 标准是人们更为熟悉的名称。MPEG-2 视频编码标准中开始引入了级和类的概念，能够针对不同应用要求进行编码模式的选择。MPEG-2 标准按编码图像的分辨率分为 4 级，按不同的编码复杂程度分为 5 类。级与类的若干组合构成 MPEG-2 视频编码标准在某种特定应用下的子集：对某一输入格式的图像，采用特定集合的编码工具，产生规定速率范围内的编码码流。

MPEG-4 标准在 1998 年 11 月被 ISO/IEC 正式批准。相比于 MPEG-1 标准和 MPEG-2 标准，MPEG-4 标准涵盖的内容非常丰富，它包括 31 部分（Part）。MPEG-4 标准的不同

部分分别定义了系统、音视频编码、多媒体传输集成框架、知识产权管理、动画框架扩展和3D图形压缩等内容,其中第10部分就是著名的H.264/AVC标准。

H.264/AVC标准是由ITU-T的VCEG和ISO/IEC的MPEG组成的联合视频组(Joint Video Team,JVT)共同开发的数字视频编码标准,也称ITU-T H.264建议和MPEG-4第10部分先进视频编码(Advanced Video Coding,AVC)标准。H.264/AVC标准仍然采用了混合编码框架的理念,此框架支持许多先进的编码技术,如具有方向性的帧内预测、多参考帧的运动补偿、灵活分块的运动补偿、可用于预测的B帧、4×4/8×8的整数DCT、环路去方块滤波和自适应熵编码等。H.264/AVC标准比H.263+、MPEG-4(SP)标准减少了约50%的码率,因此在视频存储、广播和流体等领域得到广泛应用。

H.265/HEVC标准也是由ITU-T的VCEG和ISO/IEC的MPEG联合组成的JVT共同开发的数字视频编码标准。该标准沿用了混合编码框架,支持许多先进的编码技术,如四叉树编码单元划分结构、35种帧内预测模式、运动信息融合技术、先进的运动矢量预测技术、自适应变换技术、像素自适应补偿技术等,在相同重建视频质量条件下,H.265/HEVC比H.264/AVC标准减少了约50%的码率。

3. AVS系列标准

音视频编码标准(Audio Video coding Standard,AVS)是我国具有自主知识产权的第二代信源编码标准。AVS1-P2是第一个AVS视频标准,针对标清和高清视频进行了编码工具的优化,在编码性能与编解码复杂度之间实现了较好的平衡。该标准采用了16×16的宏块结构。随着视频内容分辨率由高清向4K过渡,以及视频内容位宽由8位演进为10位,AVS标准工作组于2012年启动了第二代AVS视频编解码标准(简称AVS2-P2)制定工作,AVS2-P2与H.265/HEVC编码压缩性能相当。在监控场景中,AVS2-P2编码性能则超过H.265/HEVC。

第三代AVS视频编解码标准(AVS3-P2)于2021年4月完成第二阶段标准(AVS3-P2 Phase 2)的制定。AVS3-P2 Phase 2面向高编码压缩性能的应用,其目标性能与H.266/VVC基本持平。帧内编码的预测模式由33种扩展为65种,并采用非方形的帧内预测块划分技术。帧间预测和编码过程引入了解码端导出运动信息及修正运动信息的技术,并将滤波技术应用于帧间预测块的获取过程,采用了更灵活的帧间预测块划分形状(非方形),扩展了运动矢量和预测模式的编码方式。

4. AOM标准

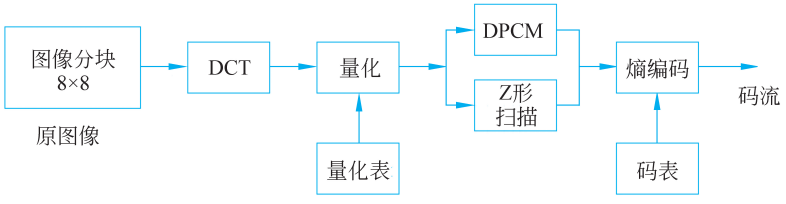
2015年9月,谷歌、微软、Netflix等多家科技公司创立了开放媒体联盟(AOM),旨在通过制定全新、开放、免版权费的视频编码标准和视频格式,创建一个持久的生态系统,为下一代多媒体体验创造新的机遇。2011年11月至2013年7月,谷歌公司研发了开源的VP9编码标准,在不同测试条件下VP9的性能与H.265/HEVC相当或低于H.265/HEVC。2014年起,谷歌公司开始了VP10编码标准的研发工作。随着2015年9月AOM的成立,谷歌公司停止了VP10的研发工作,而后AOM开始了AV1标准的研究。2018年6月,AOM发布了其首款免版权费、开源的视频编码格式AV1。AV1沿用了传统的混合视频编码框架,它始于同样免版权费、开源格式的VP9的衍生版本,同时采纳了谷歌公司的VP10、Mozilla公司的Daala、Cisco公司的Thor共3款开源编码项目中的技术成果。AV1共推出了100多个新的编码工具,在压缩效率方面显著优于同期的编码器,在实现方面也考虑了硬

件可行性和后续可扩展性。

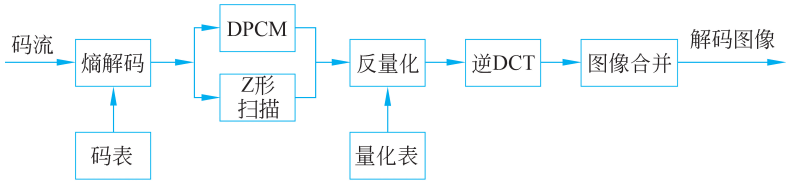
3.2　JPEG 静止图像编码标准

3.2.1　JPEG 编码标准

JPEG(Joint Photographic Experts Group,联合图像专家组)编码标准允许对静止图像进行有损和无损的编码。JPEG 有几个定义的模式,包括基本、渐进和分级模式,其中压缩算法主要分为两种,一种是以离散余弦变换为基础的有损压缩算法,另一种是以预测技术为基础的无损压缩算法。JPEG 算法基本编码框架如图 3.7 所示,在基于 DCT 块压缩的帮助下,可以实现平均 15∶1 的压缩比。利用预测编码压缩技术可以实现无损编码,包括差分编码、游程编码和哈夫曼编码。JPEG 进行按照 HVS 加权的均匀量化。对量化系数要进行 Z 形扫描,因为它允许以从低频分量到高频分量的顺序进行熵编码。JPEG 图像压缩算法能够在提供良好的压缩性能的同时具有比较好的重建质量,被广泛应用于图像、视频处理领域。JPEG 格式是最常用的图像文件格式。



(a) 编码器



(b) 解码器

图 3.7　JPEG 算法基本编码框架

对于输入的图像,JPEG 编码要经过以下 6 个步骤。

1. 图像预处理

图像预处理包括色彩空间转换、采样与分块处理等。JPEG 采用的是 YCrCb 色彩空间,因此将输入图像转换为 YCrCb 色彩空间,并按一定的采样格式进行采样。在 YCrCb 色彩空间中,Y 代表亮度,Cr、Cb 则代表色度和饱和度(也有人将 Cb、Cr 两者统称为色度),三者通常以 Y、U、V 表示,即用 U 代表 Cb,用 V 代表 Cr。

研究发现,人眼对亮度变化的敏感度要比对色彩变化的敏感度高出很多。因此,可以认为 Y 分量要比 Cb、Cr 分量重要得多。在 BMP 图像中,R、G、B 3 个分量各采用一字节进行采样;而 JPEG 图像中,通常采用两种采样方式:YUV411 和 YUV422,它们所代表的意义是 Y、Cb、Cr 3 个分量的数据取样比例为 4∶1∶1 或者 4∶2∶2。这样的采样方式虽然损失了一定的精度,但在人眼不易察觉到的范围内减小了数据的存储量。当然,JPEG 格式也允

许将每个点的 U、V 值都记录下来。

分块处理将输入图像分成若干 8×8 的小块。在此过程中需要对图像的宽和高进行裁剪,使其都为 8 的倍数,不足的部分复制与其最邻近的像素值。编码时,按从左至右、从上至下的顺序依次读取一个 8×8 块,对其进行 DCT、量化、编码后,再读取下一个 8×8 块。

2. 零偏置

JPEG 编码将图像分为 8×8 的块作为数据处理的最小单位,对于灰度级为 256 的像素,通过减去 128,将无符号数变成有符号数。即对于原来图像的灰度范围 0～255 的像素,减去 128 后,范围变成了－128～127。经过零偏置后,像素灰度的绝对值被控制在较小的范围内,便于后续的编码。

3. DCT

将帧数据分成 8×8 的矩阵子块,每个块按从左到右、从上到下的顺序送入离散余弦变换器进行二维 DCT。一般 8×8 的二维数据块经 DCT 后变成 8×8 个变换系数,这些系数都有明确的物理意义。例如当 U、V 分量为 0 时,$F(0,0)$是原来 64 个样本值的平均值,相当于直流分量;随着 U、V 分量值的增加,相应系数分别代表逐步增加的水平空间频率和垂直空间频率分量的大小。

4. 量化

图像数据转换为 DCT 频率系数之后,还要经过量化阶段才能进入编码过程。JPEG 量化是事先建立 1 张 8×8 个数据的量化表,量化表内数据大小排布的规律是:数值随频率上升而上升,直流/低频位置上的数值小,高频位置上的数值大。量化阶段需要两个 8×8 量化矩阵数据,一个专门处理亮度的频率系数,另一个则针对色度的频率系数,将频率系数除以量化矩阵的值之后取整,即完成了量化过程。频率系数经过量化之后,由浮点数转换为整数,这才便于执行最后的编码。不难发现,经过量化阶段之后,所有的数据只保留了整数近似值,也就再度损失了一些数据内容。在 JPEG 算法中,由于对亮度和色度的精度要求不同,分别对亮度和色度采用不同的量化表,前者细粒度量化,后者粗粒度量化。

对亮度和色度分量的 DCT 频率系数进行量化,使用如表 3.1 和表 3.2 所示的标准亮度分量量化表和标准色度分量量化表,这两个量化表是从广泛的实验中得出的。当然,也可以自定义量化表。

表 3.1 标准亮度分量量化表

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	44	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

表 3.2　标准色度分量量化表

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

量化表去掉了很多高频量，对 DCT 频率系数进行量化后得到的结果中会出现大量的 0，使用 Z 形扫描可以将这些 0 集中到一起，减小编码后的视频大小。越偏离左上方，表示频率越高，通过量化将图像的高频信息去掉了。

5. Z 形扫描

从量化后的 DCT 频率系数表中读出数据和表示数据的方式也是减小码率的一个重要过程。读出的方式可以有多种选择，如水平逐行读出、垂直逐列读出、交替读出和 Z 形扫描读出。其中 Z 形扫描读出是最常用的一种方式，它实际上是按二维频率的高低顺序读出频率系数。

6. 熵编码

对直流分量进行 DPCM 编码，对交流分量进行 RLC 编码，这两种编码都有中间格式，以进一步减小存储量。在得到直流分量系数的中间格式和交流分量系数的中间格式之后，为进一步压缩图像数据，有必要对两者进行熵编码，通过对出现概率较高的字符采用较小的位数编码以达到压缩的目的。JPEG 标准具体规定了两种熵编码方式：哈夫曼编码和算术编码。JPEG 基本系统规定采用哈夫曼编码。

哈夫曼编码的基本思想是：对出现概率大的字符分配长度较小的二进制编码，对出现概率小的字符分配长度较大的二进制编码，从而使得字符的平均编码长度最短。哈夫曼编码的原理请参考数据结构课程中的哈夫曼树或者最优二叉树的内容。

在进行哈夫曼编码时直流分量系数与交流分量系数分别采用不同的哈夫曼编码表，对于亮度和色度也采用不同的哈夫曼编码表。因此，需要 4 张哈夫曼编码表才能完成熵编码的工作。具体的哈夫曼编码采用查表的方式高效地完成。然而，在 JPEG 标准中没有定义默认的哈夫曼编码表，用户可以根据实际应用自由选择，也可以使用 JPEG 标准推荐的哈夫曼编码表，或者预先自定义一个通用的哈夫曼编码表，还可以针对一幅特定的图像，在压缩编码前通过搜集其统计特征计算哈夫曼编码表的值。

3.2.2　JPEG 工作模式

JPEG 定义了以下 4 种工作模式：

(1) 顺序编码。其基本算法是将图像分成 8×8 的块，然后进行 DCT、量化和熵编码(哈夫曼编码或算术编码)。

(2) 渐近编码。采用的算法与工作模式(1)相类似,不同的是,首先传送部分 DCT 系数信息(例如低频带系数,或所有系数的近似值),使接收端尽快获得一个粗略的图像,然后再将剩余频带的系数(或所有系数的低比特数据)渐次传送,最终形成清晰的图像。

(3) 无失真编码。采用一维或者二维的空域 DPCM 和熵编码。由于输入图像已经是数字化的,经空域 DPCM 之后,预测误差值也是一个离散量,因此可以不再量化而实现无损编码。

(4) 分层编码。在这个工作模式中,将输入图像的分辨率逐层降低,形成一系列分辨率递减的图像。先对分辨率最底层图像进行编码,然后将经过内插的低层图像作为上一层图像的预测值,再对预测误差进行编码,以此类推,直至顶层图像。

3.2.3 JPEG 编码实现与算能平台

libjpeg 使用 C 语言实现,这个库由 JPEG 工作组维护。算能平台的 JPEG 编解码程序是基于 OpenCV 实现的,OpenCV 是一个基于 Apache 2.0 许可(开源)发行的跨平台计算机视觉和机器学习软件库,可免费用于学术和商业用途,可以运行在 Windows、Linux、macOS、iOS 和 Android 操作系统上。算能平台提供了 JPEG 编解码模块 FnEncode 和 FnDecode 实现图像编码,这两个模块通过调用 OpenCV 库的图像编解码函数 imencode 和 imdecode 完成编码,除此之外还包括图像的读取、写入和保存等功能,具体实现过程见本书实验部分。

3.3 H.264 视频编码标准

3.3.1 H.264 编码标准概述

与 MPEG-2 类似,H.264 也有类和级的概念。H.264 中常见的有 3 个类,即基线类(baseline profile),主类(main profile)和扩展类(extended profile),每一类支持一组特定的编码功能。其中,基线类主要用于视频会话,如视频会议、可视电话、远程医疗、远程教学等;主类用于要求高画质的消费电子等应用领域,如数字电视广播、媒体播放器等;扩展类主要用于各种网络流媒体传输等方面。JVT 在 2004 年对高级类涵盖的范围做了进一步的扩充,新增了 4 个高级类:High(HP)、High10(Hi10P)、High4∶2∶2(H422P)、High4∶4∶4(H444P)。

下面列出这 3 个类所支持的特定的编码工具以及其主要应用领域。如图 3.8 所示,H.264 还对所有类规定了一组相同的级。级的选择一般都是根据计算机的运算能力和内存容量决定的,通过设置不同参数(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 视频编码原理如图 3.9 所示。输入的帧或场以宏块为单位进行处理。如果采用帧内预测编码,首先要选择最佳的帧内预测模式进行预测,然后对残差进行变换、量化和熵编码。量化后的残差系数经过反量化和反变换之后与预测值相加得出重建图像。为了去除环路中产生的噪声,提高参考帧的图像质量,设置了一个去块(去除块效应)滤波器,滤波后的输出图像可用作参考图像。如果采用帧间预测编码,当前块在已编码的参考图像中进行

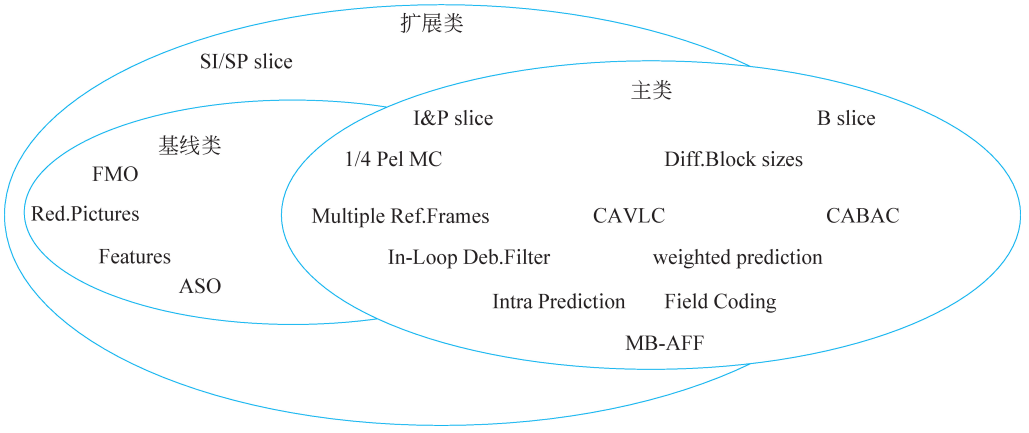


图 3.8　H.264 主要的 3 个类和对应编码工具

运动估计和运动补偿后得出预测值，预测值和当前块相减后，产生残差数据。残差图像块经过变换、量化和熵编码后与运动矢量一起送到信道中传输。同时，残差系数经逆量化、逆变换后与预测值相加并经过去块滤波器滤波后得到重建图像。

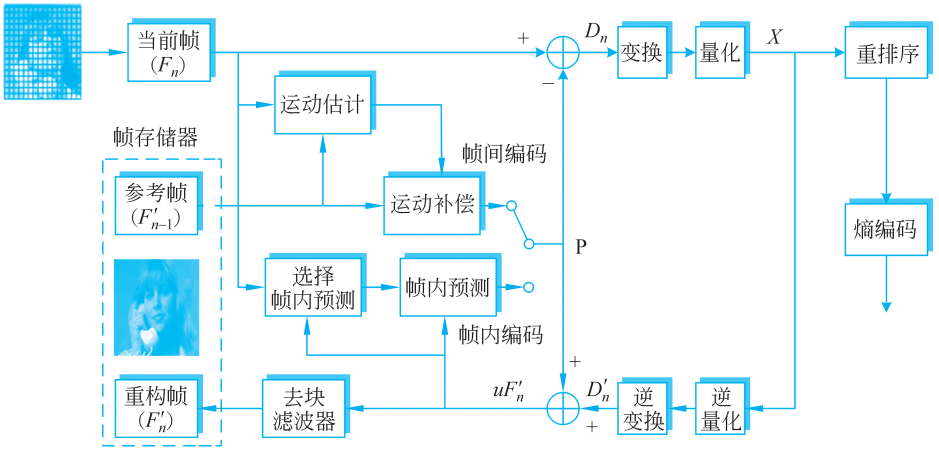


图 3.9　H.264 视频编码原理

解码是编码的逆过程，解码器接收到 NAL 包后，从 NAL 包中剥离出宏块压缩后的码流，然后经过熵解码和重排序，得到量化后的宏块系数，再经过逆量化、逆变换。这一过程和编码器重建码流生成的过程一致。使用码流中解出的预测块信息，解码器从参考帧中得到预测宏块，它和编码器形成的预测宏块相同。最后对重建图像进行滤波，去除块效应后可得到解码宏块。当前图像的所有宏块解码完成后，就得到当前重建图像用于显示输出。同编码过程一样，当前重建图像将用于未来的解码参考。

3.3.2　H.264 编码方法

H.264 视频编码器采用了与 MPEG-2 相同的基于运动补偿和变换编码的混合结构，其基本模块仍然包含了变换、量化、预测和熵编码等单元，但在技术上采纳了许多新的研究成果，使其在压缩编码效率上有了很大的提高。这些新技术包括帧内预测、帧间预测、可变块

大小的运动补偿、整数 DCT 与量化、1/8 像素精度的运动估计、去块滤波器以及基于上下文的自适应熵编码等。下面对部分新技术加以介绍。

1. 帧内预测

帧内预测编码是 H.264 采用的新技术之一。对视频图像进行分块后,同一个物体常常由相邻的许多宏块或者子块组成,这些块之间的像素值相差不大,而且纹理也往往高度一致。图像中的前景与背景也通常具有一定的纹理特性。图像在空域上的方向特性及块像素间的相关性为帧内预测创造了条件,因此,可以利用帧内预测去除相邻块之间的空间冗余。

对于 I 帧编码,H.264 使用了基于空间像素值的预测方法。编码时,根据已编码重建块和当前块形成预测块,然后对实际值和预测值的残差图像进行整数 DCT、量化和熵编码。为了保证对不同纹理方向图像的预测精度,H.264 定义了多种不同方向的预测选项,以尽可能准确地预测不同纹理特性的图像子块。预测时,每个块依次使用不同的选项进行编码,计算得到相应的代价,再根据不同的代价值确定最优的选项。

H.264 对亮度和色度分量采用不同的预测方法。对于亮度,预测块可以有 4×4 和 16×16 两种尺寸。4×4 亮度块有 9 种预测模式,独立预测每一个 4×4 亮度块,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,适用于平滑区域图像编码。对于色度,类似于 16×16 亮度块,也有 4 种预测模式。编码器需要为当前待编码块选择一种使该块与预测块之间差别最小的预测模式。

此外,对于那些内容不规则或者量化参数非常小的图像,H.264 还提供了一种称为 I_PCM 的帧内编码模式,在这种模式下,不需要进行预测和变换,而是直接传输图像像素值,以获得更高的编码效率。

2. 帧间预测

在帧间预测方面,H.264 引入了多种技术以提高运动估计的准确性。它支持 7 种不同大小的匹配块,具有更精细的运动矢量,在主类和扩展类中,还包括了 B 分片和加权预测。

1) 树状结构运动补偿

H.264 以 16×16 宏块作为基本单位进行运动估计。但对细节较丰富的图像,同一个宏块内可能包含不同的物体,它们的运动方向也可能不同。把宏块进一步分解,可以更好地去除相关性,提高压缩效率。每个 16×16 宏块可以有 4 种分割方式:一个 16×16 的块、两个 16×8 的块、两个 8×16 的块和 4 个 8×8 的块,其运动补偿也相应有 4 种。8×8 的块被称为子宏块,每个子宏块还可以进一步分割为两个 4×8 的块或 8×4 的块,或者 4 个 4×4 的块。H.264 宏块的树状结构分割如图 3.10 所示。这种分割下的运动补偿称为树状结构运动补偿。

每个分割或者子宏块都要有一个独立的运动矢量,每个运动矢量以及分块方式也都要进行编码和传输。大的分区尺寸可能只需要较少的比特数表示运动矢量和分块方式,但残差将保存较大的能量;小的分区尺寸可以使运动补偿后的残差能量下降,但需要更多的比特数表示运动矢量和分块方式。因此,分区大小的选择对压缩性能有重要的影响。

2) 运动矢量精度

H.264 采用了 1/4 像素和 1/8 像素的运动估计。其中,亮度分量具有 1/4 像素精度,色度分量具有 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,需利用邻近的已编码样值进行内插后得到。

首先生成参考图像中亮度分量的半像素样值，如图 3.11 所示。半像素 b、h、m、s 样值通过对相应整像素进行 6 抽头滤波得出，6 抽头 FIR 滤波器的权重为 1/32、−5/32、5/8、5/8、−5/32、1/32。例如，b 可以由水平方向的整数样值 E、F、G、H、I、J 计算得到，h 可以由垂直方向的样值 A、C、G、M、R、T 计算得到。一旦邻近（垂直或水平方向）整像素点的所有像素值都计算出来，剩余的半像素便可以通过对 6 个垂直或水平方向的半像素点滤波得出。例如，j 可由 cc、dd、h、m、ee、ff 滤波得出。

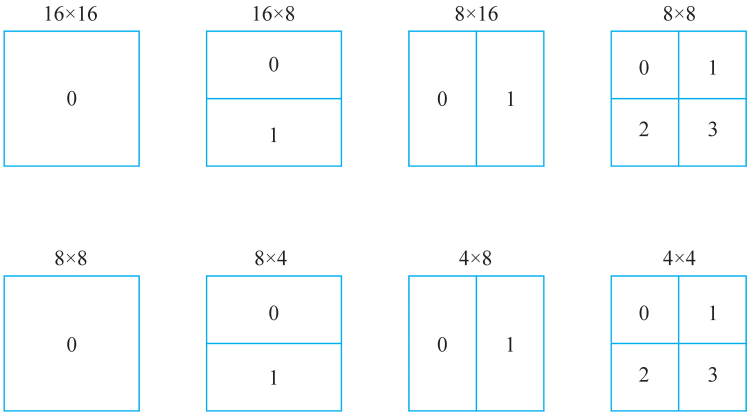


图 3.10　H.264 宏块的树状结构分割

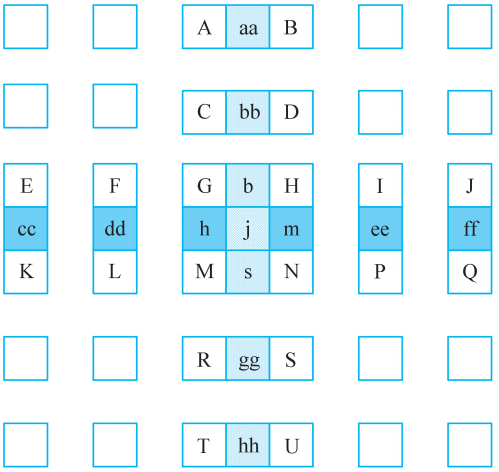


图 3.11　亮度分量的半像素插值

半像素样值计算出来以后，可线性内插生成 1/4 像素样值，如图 3.12 所示。1/4 像素 a、c、i、k、d、f、n、q 由邻近像素内插得出；水平或者垂直方向的 1/4 像素点由两个半像素或者整像素插值生成；剩余的 1/4 像素 e、g、p、r 由一对对角半像素点线性内插得出，例如 e 由 b 和 h 获得。色度像素需要 1/8 精度的运动矢量，也同样通过整像素线性内插得出。

3）运动矢量预测

每个块的运动矢量需要一定数目的比特表示，因此有必要对运动矢量进行压缩。由于邻近区域的运动矢量通常具有相关性，因此当前块的运动矢量可由邻近已编码块的运动矢量预测得到，最后传输的是当前矢量和预测矢量的差值。

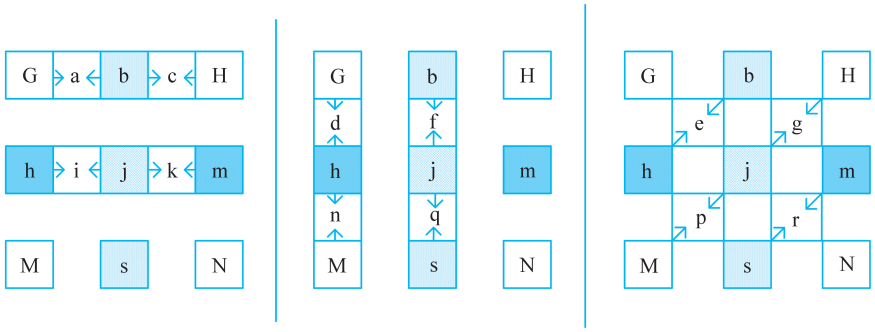


图 3.12 亮度分量的 1/4 像素插值

预测矢量 MVP 的生成取决于运动补偿分割的尺寸以及周围邻近运动矢量是否存在。图 3.13 给出了相同尺寸和不同尺寸分割时邻近块的选择方法。其中,E 为当前块,C、A、B 分别为 E 的左、上、右上方的 3 个邻近块。当 E 的左边不止一个分割时,取其中最上方的一个为 A;当上方不止一个分割时,取其中最左边的一个为 B。

当前运动矢量的预测值(Motion Vector Prediction,MVP)的确定方法如下:

(1) 若当前块尺寸不是 16×8 或者 8×16,则 MVP 为 A、B、C 块运动矢量的中值。

(2) 若当前块尺寸为 16×8,则上面部分 MVP 由 B 预测,下面部分 MVP 由 A 预测。

(3) 若当前块尺寸为 8×16,则左面部分 MVP 由 A 预测,右面部分 MVP 由 C 预测。

(4) 若为跳跃宏块(skipped MB),则用第一种方法生成 16×16 块的 MVP。若有一个或者几个已传输块不存在时,MVP 的选择方法需要做相应的调整。

4) 多参考帧

H.264 引入了多参考帧的预测,不仅可以使用前后相邻帧,而且可以参考前向与后向多个帧提高预测的精确性,如图 3.14 所示。因此,采用多参考帧会对视频图像产生更好的主观质量,对当前帧编码更加有效。实验表明,与只采用一个参考帧预测相比,使用 5 个参考帧时比特率可以降低 5%~10%。然而从实现的角度看,多参考帧将产生额外的处理延时和更大的内存空间要求。

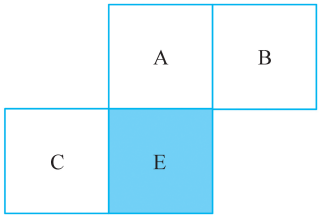


图 3.13 邻近块的选择方法

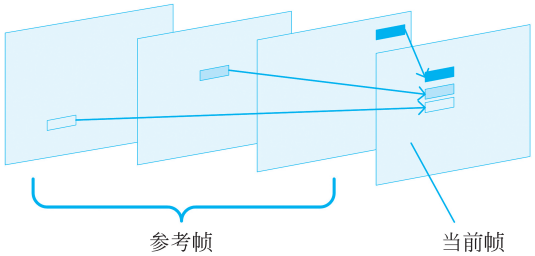


图 3.14 多参考帧预测

3. 整数 DCT 与量化

H.264 引入了 4×4 整数 DCT,降低了算法的复杂度,将变换运算中的比例因数合并到量化过程中,整个变换过程无乘法运算,只需要加法和移位运算,同时避免了以往标准中使用的通用 8×8DCT 的逆变换经常出现的失配问题。量化过程根据图像的动态范围大小确定量化参数,既可以保留图像必要的细节,又可以减少码流。

整数 DCT 的处理过程如图 3.15 所示，对 16×16 亮度残差数据进行整数 DCT 时，如果是帧内 16×16 预测模式的亮度块，则进一步将其中 4×4 块的直流分量进行哈达马变换及量化；对 8×8 色度残差数据进行整数 DCT 时，对 Cr 或 Cb 块中的 2×2 直流分量系数矩阵也进行哈达马变换及量化。

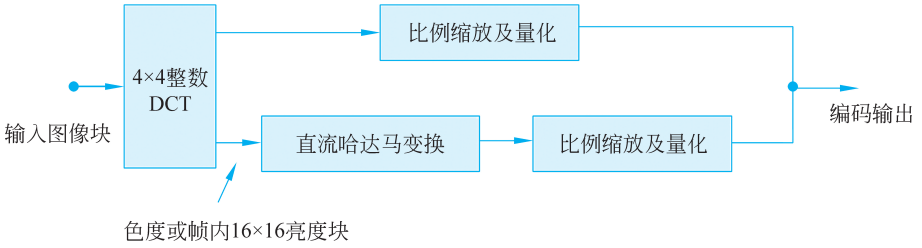


图 3.15　整数 DCT 的处理过程

4. 去块滤波器

在进行基于分块的视频编码时，对块的预测、补偿、变换以及量化在码率较低时会遇到块效应。为了降低图像的块效应失真，H.264 中引入了去块滤波器对解码宏块进行滤波，平滑块边缘，滤波后的帧用于后续帧的运动补偿预测，从而避免了假边界累积误差导致的图像质量下降，提高图像的主观视觉效果。

去块滤波器在处理时以 4×4 块为单位，如图 3.16 所示。对每个亮度宏块，先对宏块最左的边界 a 进行滤波，然后依次从左到右处理宏块内 3 个垂直边界 b、c 和 d。对水平边界，从上到下依次处理 e、f、g 和 h。色度滤波次序类似，依次处理 i、j、k、l。

去块滤波器的处理可以在 3 个层面上进行。在分片层面，OffsetA 和 OffsetB 为在编码器中选择的偏移量，该偏移量用于调整阈值 α 与 β，从而调整全局滤波强度；在块边界层面，滤波强度依赖于边界两边图像块的帧间/帧内预测、运动矢量差及编码残差等；在图像像素层面，滤波强度取决于像素值在边界的梯度及量化参数。

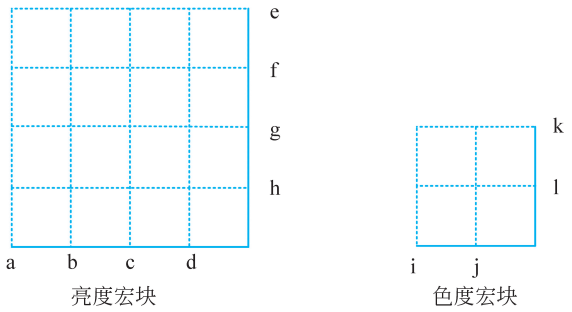


图 3.16　边界滤波顺序

由于视频图像本身还存在物体的真实边界，在进行去块滤波时，应尽可能判断边界的真实性，保留图像的细节，不能盲目通过平滑图像达到消除块效应的目的。一般而言，真实边界的两侧像素梯度比因量化造成的虚假边界两侧的像素梯度大。在 H.264 中，给定两个阈值 α 与 β 用于判断是否对边界进行滤波，当高于阈值时，则认为该边界为真实边界。

5. 熵编码

H.264 标准规定的熵编码有两种：一种是可变长编码方案，包括统一的变长编码

(Universal Variable Length Coding,UVLC)和基于上下文的自适应变长编码(Context-based Adaptive Variable Length Coding,CAVLC);另一种是基于上下文的自适应二进制算术编码(Context-based Adaptive Binary Arithmetic Coding,CABAC)。这两种方案都利用了上下文信息,使编码最大限度地利用了视频流的统计信息,有效降低了编码冗余。当熵编码模式设置为 0 时,残差数据使用 CAVLC 编码,其他参数,如宏块类型、量化步长参数、参考帧索引、运动矢量等,采用 UVLC 编码。UVLC 由传统的 VLC 改进而来,它利用统一的指数哥伦布码表进行编码。当熵编码模式设置为 1 时,采用 CABAC 编码对语法元素进行编码。

1) 指数哥伦布编码

指数哥伦布编码使用一张码表对不同对象进行编码,故编码方法简单,且解码器容易识别码字前缀,从而在发生比特错误时能快速重新获得同步。指数哥伦布编码是具有规则结构的变长码,每个码字的长度为 $2M+1$ 比特,其中包括最前面 M 比特的 0、中间 1 比特的 1 和后面 M 比特的 INFO 字段。在对各种参数(如宏块类型、运动矢量等)进行编码时,把参数先映射为 code_num,再对 code_num 进行编码。

2) CAVLC

CAVLC 是一种基于上下文的自适应游程编码。当熵编码模式设置为 0 时,使用 CAVLC 对以 Z 形扫描得到的 4×4 残差块变换系数进行编码。由于经过预测、变换和量化后的 4×4 残差系数是稀疏矩阵,多数系数为 0,故用游程编码可以取得更好的压缩效果。由于相邻块的非零系数个数具有相关性,CAVLC 依据这种相关性自适应选择相应的码表,体现了基于邻近块的上下文原理。同时,残差系数中低频系数较大,高频系数较小,CAVLC 利用这一特点,并根据邻近已编码系数的大小自适应选择相关码表。

3) CABAC

CABAC 使用算术编码方法,根据元素的上下文为其选择可能的概率模型,并根据局部统计特性自适应地进行概率估计,从而提高压缩性能。和 CAVLC 相比,CABAC 平均效率可以提高 10%~15%。其缺点在于编码速度较低。

3.3.3 H.264 的传输与存储

1. H.264 分层编码结构

H.264 视频编码标准引入了分层结构,将图像压缩系统分成视频编码层(Video Coding Layer,VCL)和网络抽象层(Network Abstraction Layer,NAL),使压缩编码与网络传输分离,使编码层能够移植到不同的网络结构中,图 3.17 为 H.264 的分层结构。视频编码层进行视频编码、解码操作,而网络抽象层专门为视频编码信息提供文件头信息,安排格式以方便网络传输和介质存储,使网络对于视频编码层是透明的,具有较强的网络友好性和错误隐藏能力。

VCL 是 H.264 的核心部分,其编码输出的是 VCL 数据(表示编码视频数据的比特序列),在传输和存储之前被映射到网络抽象层单元(NAL Unit,NALU)。NAL 在外围,它根据视频信号传输介质把 VCL 的内容封装起来。NAL 数据的基本单位是 NAL 单元,而 VCL 自上而下包括序列、图像组、图像、条带组、条带、宏块组、宏块和块,如图 3.18 所示。划分条带主要是为了适应不同传输网络的最大传输单元(Maximum Transfer Unit,MTU)