还是给大家再强调一下，70 分的分析题，需要重点复习霍夫曼编码、算术编码、LZ77 编码、JPEG 编码、MPEG 编码、信息熵理论、LPC 线性预测。

一、霍夫曼编码

一： 什么是霍夫曼编码?

答：霍夫曼编码是一种常用的无损数据压缩技术，它根据输入数据中各个符号的频率分布，为每个符号分配不同长度的二进制编码。频率较高的符号被赋予较短的编码，而频率较低的符号则被赋予较长的编码，以实现较高的压缩率。

二： 霍夫曼编码的基本原理是什么?

答： 霍夫曼编码的基本原理是根据输入数据中各个符号的频率构建一颗霍夫曼树，然后通过从根节点到每个叶子节点的路径上的 0 和 1 来表示每个符号的编码。在霍夫曼树中，频率较高的符号位于树的较低层，而频率较低的符号位于树的较高层，从而使得编码长度较短的符号与频率较高的符号对应，进而实现压缩。

三： 霍夫曼编码的优点是什么?

答： 霍夫曼编码具有以下优点：

1. 较高的压缩率：霍夫曼编码根据符号的频率分布为每个符号分配最优的编码，使得高频率的符号具有较短的编码，从而实现较高的压缩率。
2. 无损压缩：霍夫曼编码是一种无损压缩技术，可以精确还原原始数据，没有任何信息损失。
3. 唯一解码性：通过构建霍夫曼树，每个符号的编码都是唯一的，可以避免编码冲突，确保解码的准确性。
4. 简单快速的编码与解码：使用霍夫曼编码进行数据的压缩与解压缩操作相对简单和高效。

二、算术编码

一： 什么是算术编码?

答： 算术编码是一种无损数据压缩技术，它将输入的数据流编码为一个介于 0 和 1 之间的小数。通过根据输入数据的频率分布来分配较长的编码给常见的符号，以及较短的编码给不常见的符号，算术编码可以实现较高的压缩率。

二： 算术编码的基本原理是什么?

答： 算术编码的基本原理是根据输入数据的频率分布为每个符号分配不同长度的编码。它使用一个初始的编码范围（通常是 0 到 1 之间的区间），然后根据每个符号的频率缩小编码范围。编码范围的缩小过程是根据每个符号的频率在当前编码范围内进行的。最终，编码范围会收敛到一个非常小的区间，其中包含了唯一表示输入数据的编码。

三： 算术编码的优点是什么?

答： 算术编码具有以下优点：

1. 较高的压缩率：算术编码可以根据输入数据的实际频率分布来分配编码，从而实现更高的压缩率。
2. 无损压缩：算术编码是一种无损压缩技术，它可以精确地还原原始数据，没有任何信息损失。
3. 适用于任意数据类型：算术编码可以应用于任何类型的数据，包括文本、图像、音频等。
4. 灵活性：算术编码可以根据输入数据的特性进行自适应编码，不需要事先了解数据

的统计特性。

三、LZ77 编码

一、 什么是 LZ77 编码?

答: LZ77 编码是一种常用的无损数据压缩算法，它通过利用数据中的重复信息来实现压缩。LZ77 编码将输入数据分割成窗口和查找缓冲区，并利用滑动窗口中已经出现的数据来表示当前位置的数据。

二、 LZ77 编码的基本原理是什么?

答: LZ77 编码的基本原理是通过寻找输入数据中的重复片段来实现压缩。它使用一个滑动窗口来存储最近出现的数据，并使用一个查找缓冲区来搜索滑动窗口中的重复片段。当找到重复片段时，LZ77 编码使用一个指针来指示滑动窗口中的起始位置和长度，从而表示重复的数据。通过这种方式，LZ77 编码可以用较少的位数来表示较长的重复片段，从而实现压缩。

三、 LZ77 编码的优点是什么?

答: LZ77 编码具有以下优点:

1. 重复数据压缩: LZ77 编码通过寻找输入数据中的重复片段来实现压缩，对于包含大量重复数据的输入，可以实现较高的压缩率。
2. 无损压缩: LZ77 编码是一种无损压缩技术，可以精确还原原始数据，没有任何信息损失。
3. 简单快速的编码与解码: LZ77 编码的编码与解码过程相对简单，可以快速进行压缩和解压缩操作。
4. 部分压缩: LZ77 编码可以实现部分压缩，即可以选择性地对输入数据中的某些片段进行压缩，而不是对整个数据流进行压缩。

四、对于输入串行 "ABABCABABDAB"，请使用 LZ77 编码进行压缩，并详细展示每一步的匹配过程和输出结果。

答:

1. 初始化: 输入串行为 "ABABCABABDAB"，输出串行为空。
2. 匹配过程:
    - 第一步: 在输入串行中找到最长的前缀与当前未匹配的部分相匹配。在这种情况下，最长匹配是 "AB"，其对应的距离为 0。
    - 输出: (0, 'A', 'B')
    - 更新: 将匹配的部分 "AB" 添加到输出串行中，并将输入串行中的匹配部分移除。
    - 输入串行更新为 "CABABDAB"。
    - 输出串行更新为 "(0, 'A', 'B')"
    - 第二步: 在输入串行中找到最长的前缀与当前未匹配的部分相匹配。在这种情况下，最长匹配是 "AB"，其对应的距离为 0。
    - 输出: (0, 'A', 'B')
    - 更新: 将匹配的部分 "AB" 添加到输出串行中，并将输入串行中的匹配部分移除。
    - 输入串行更新为 "DAB"。
    - 输出串行更新为 "(0, 'A', 'B') (0, 'A', 'B')"
    - 第三步: 在输入串行中找到最长的前缀与当前未匹配的部分相匹配。在这种

情况下，最长匹配是 "D"，其对应的距离为 0。

- 输出：(0, 'D')
- 更新：将匹配的部分 "D" 添加到输出串行中，并将输入串行中的匹配部分移除。
- 输入串行更新为 "AB"。
- 输出串行更新为 "(0, 'A', 'B') (0, 'A', 'B') (0, 'D')"
- 第四步：在输入串行中找到最长的前缀与当前未匹配的部分相匹配。在这种情况下，最长匹配是 "A"，其对应的距离为 1。
- 输出：(1, 'A')
- 更新：将匹配的部分 "A" 添加到输出串行中，并将输入串行中的匹配部分移除。
- 输入串行更新为 "B"。
- 输出串行更新为 "(0, 'A', 'B') (0, 'A', 'B') (0, 'D') (1, 'A')"
- 第五步：在输入串行中找到最长的前缀与当前未匹配的部分相匹配。在这种情况下，最长匹配是 "B"，其对应的距离为 1。
- 输出：(1, 'B')
- 更新：将匹配的部分 "B" 添加到输出串行中，并将输入串行中的匹配部分移除。
- 输入串行更新为 ""。
- 输出串行更新为 "(0, 'A', 'B') (0, 'A', 'B') (0, 'D') (1, 'A') (1, 'B')"

3. 压缩结果：输出串行为 "(0, 'A', 'B') (0, 'A', 'B') (0, 'D') (1, 'A') (1, 'B')"

通过以上的步骤，我们得到了输入串行 "ABABCABABDAB" 的 LZ77 编码压缩结果为 "(0, 'A', 'B') (0, 'A', 'B') (0, 'D') (1, 'A') (1, 'B')"。

四、JPEG 编码

一、 JPEG 编码是用于什么类型的数据压缩?

答： JPEG 编码是一种常用的无损或有损的图像压缩算法，用于压缩数字图像数据。

二、 JPEG 编码的基本原理是什么?

答： JPEG 编码的基本原理是通过将图像数据转换到频域上的离散余弦变换（DCT）系数来实现压缩。JPEG 编码首先将图像分成小的图像块，并对每个图像块进行 DCT 变换以获取其频域表示。然后，对频域系数进行量化，并采用熵编码（通常是基于霍夫曼编码）来进一步压缩数据。

三、 JPEG 编码的优点是什么?

答： JPEG 编码具有以下优点：

1. 高压缩率：JPEG 编码可以实现较高的压缩率，尤其适用于彩色图像和复杂场景的图像。
2. 有损压缩：JPEG 编码可以通过调整压缩质量参数来实现不同程度的有损压缩，从而在一定程度上减小文档大小，而对视觉质量的影响相对较小。
3. 广泛支持：JPEG 是一种广泛支持的图像压缩格式，在许多设备和应用程序中得到良好的兼容性。
4. 高效的解码：JPEG 编码的解码速度相对较快，适用于实时图像传输和处理的场景。

五、MPEG 编码

一、 MPEG 编码是用于什么类型的数据压缩?

答: MPEG 编码是一种用于压缩数字音频和视频数据的标准，适用于多媒体数据的压缩与传输。

二、 MPEG 编码的基本原理是什么?

答: MPEG 编码的基本原理是通过移除视觉和听觉中的冗余信息，以及对不可感知的细节进行有损压缩来实现数据的压缩。MPEG 编码采用了一系列的压缩技术，包括运动估计、离散余弦变换、量化、熵编码等步骤，以降低数据的冗余性和数据量。

三、 MPEG 编码的优点是什么?

答: MPEG 编码具有以下优点:

1. 高压缩率: MPEG 编码可以实现较高的压缩率，减小音频和视频数据的文档大小，有利于存储和传输。
2. 视听质量平衡: MPEG 编码在进行有损压缩时，可以根据不同的压缩参数和编码配置来平衡视听质量和压缩率，以满足不同应用需求。
3. 广泛支持: MPEG 是一种广泛支持的音视频压缩标准，在各种设备和平台上都得到了良好的兼容性。
4. 可伸缩性: MPEG 编码支持可伸缩压缩，可以根据网络带宽和终端设备的能力进行适应性传输，提供不同质量级别的音视频流。

四、假设有一个视频串行，包含 3 帧，每帧的大小为 256x256 像素。第一帧为 I 帧，第二帧和第三帧为 P 帧。其中，第一帧的大小为 40KB，第二帧与第一帧的差别为 10KB，第三帧与第一帧的差别为 8KB。请计算使用 MPEG 编码对该视频串行进行压缩后的总体大小，并详细展示每一步的处理过程。

答:

1. 对第一帧进行压缩:
    - 第一帧为 I 帧，不依赖于其他帧，直接进行压缩。
    - 第一帧大小为 40KB。
2. 对第二帧进行压缩:
    - 第二帧为 P 帧，需要参考前一帧（I 帧）进行压缩。
    - 根据题目中的描述，第二帧与第一帧的差别为 10KB。
    - 使用运动补偿（Motion Compensation）将第二帧的差别表示为运动矢量和残差。
    - 运动矢量: 假设运动矢量大小为 4KB。
    - 残差: 差别为 10KB，减去运动矢量的大小（4KB），残差为 6KB。
    - 第二帧压缩后的大小为运动矢量大小（4KB）加上残差大小（6KB），总大小为 10KB。
3. 对第三帧进行压缩:
    - 第三帧为 P 帧，需要参考前一帧（I 帧）进行压缩。
    - 根据题目中的描述，第三帧与第一帧的差别为 8KB。
    - 使用运动补偿（Motion Compensation）将第三帧的差别表示为运动矢量和残差。
    - 运动矢量: 假设运动矢量大小为 2KB。
    - 残差: 差别为 8KB，减去运动矢量的大小（2KB），残差为 6KB。
    - 第三帧压缩后的大小为运动矢量大小（2KB）加上残差大小（6KB），总大小为 8KB。

4. 总体大小：
    - 第一帧大小为 40KB。
    - 第二帧压缩后的大小为 10KB。
    - 第三帧压缩后的大小为 8KB。
    - 总体大小为 40KB + 10KB + 8KB = 58KB。

通过以上的步骤，我们得到了使用 MPEG 编码对该视频串行进行压缩后的总体大小为 58KB。

六、信息熵理论

一、 什么是信息熵?

答： 信息熵是信息理论中的一个概念，用于度量随机变量中的不确定性或信息量。在离散情况下，信息熵表示为对所有可能事件发生概率的期望值的负数加权和。它衡量了需要编码或传输一个随机变量的平均信息量。

二、 信息熵的计算公式是什么?

答： 对于离散随机变量，信息熵的计算公式为： $H(X) = -\Sigma\ p(x) \log2\ p(x)$ 其中，H(X)代表随机变量 X 的信息熵，p(x)表示随机变量 X 取值为 x 的概率，Σ表示对所有可能取值 x 求和，log2 表示以 2 为底的对数。

三、 信息熵的意义是什么?

答： 信息熵在信息理论中具有重要意义：

1. 衡量不确定性：信息熵可以度量随机变量中的不确定性或信息量。当一个随机变量的信息熵较高时，意味着它的可能取值较为均匀，具有较大的不确定性；当信息熵较低时，意味着它的可能取值更为集中，具有较小的不确定性。
2. 信息压缩：信息熵提供了对信息压缩的理论基础。通过对信息源进行编码，可以使得编码后的平均码长接近于信息熵，从而实现高效的信息压缩。
3. 通信效率：信息熵可以用于衡量通信系统中的传输效率。在通信中，通过减小信息源的信息熵，可以降低传输的带宽需求，提高通信效率。

七、LPC（Linear Predictive Coding）线性预测

一、 什么是 LPC 线性预测?

答： LPC 线性预测是一种数字信号处理技术，用于分析和压缩语音信号。它基于线性预测模型，通过线性组合当前样本和先前样本的加权和来预测未来样本的值。

二、 LPC 线性预测的基本原理是什么?

答： LPC 线性预测的基本原理是通过分析输入信号的自相关性，创建一个线性预测模型。该模型使用过去的信号样本来预测当前样本的值。具体步骤包括：

1. 计算输入信号的自相关系数。
2. 通过使用 Levinson-Durbin 算法或其他方法，估计线性预测模型的参数，如预测系数和误差增益。
3. 使用估计的模型参数，通过线性组合来预测未来的信号样本。

三、 LPC 线性预测的应用领域是什么?

答： LPC 线性预测在语音信号处理中有广泛的应用，包括以下领域：

1. 语音编码：LPC 线性预测可用于语音编码，通过创建预测模型和量化模型参数来实现语音信号的压缩，常用的编码算法包括 G.711 和 G.729 等。
2. 语音合成：LPC 线性预测可用于语音合成，根据语音信号的预测模型参数生成合成

语音。

3. 语音识别：LPC 线性预测可用于语音识别，通过对语音信号进行建模和分析，提取关键特征，用于语音识别任务中的特征匹配和模式识别。

课本：

4.2.3 常用的压缩编码

1. 熵编码

1）信息量和信息熵概念

一个消息的可能性越小，其信息越多；消息的可能性越大，则信息越少。在数学上，所传输的消息是其出现概率的单调下降函数。信息是指从 N 个相等可能事件中选出一个事件所需要的信息度量或含量，也就是在辨识 N 个事件中特定的一个事件的过程中所需要提问“是或否”的最少次数。例如，要从 256 个数中选定某一个数，可以先提问“是否大于 128?”，不论回答是或否都消去了半数的可能事件，这样继续问下去，只要提问 8 次这类问题，就能从 256 个数中选定某一个数，这是因为每提问一次都会得到 1 位的信息量。因此，在 256 个数中选定某一个数所需要的信息量是

$$\log_2 256 = 8 \text{ 位}$$

设从 N 个数中选定任一个数 x 的概率为 $P(x)$。假定选定任意一个数的概率都相等，即 $P(x)=1/N$，定义信息量为

$$I(x) = \log_2 N = -\log_2 \frac{1}{N} = -\log_2 P(x) = I[P(x)]$$

如果将信源所有可能事件的信息的量进行平均，就得到了信息的“熵”，其含义是信源 x 发出任意一个随机变量的平均信息量。信源 x 的符号集为 $X_i(i=1,2,\cdots,N)$，设 X_i 出现的概率为 $P(x_i)$，则信息源 x 的熵为：

$$H(x) = \sum_{i=1}^{n} P(x) I[P(x_i)] = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

2）哈夫曼(Huffman)编码

(1) 哈夫曼编码的方法

哈夫曼编码的方法是把信源符号按概率大小顺序排列，并设法按逆次序分配码字的长度。在分配码字长度时，首先将出现概率最小的两个符号的概率相加，合成一个概率，再把这个合成的概率看成是一个新组合符号的概率，重复上述做法，直到最后只剩下两个符号的概率为止。

(2) 哈夫曼编码的特点

① 用哈夫曼方法构造出来的码不是唯一，其原因有两个：一是在给两个分支赋值时，可以是左支(或上支)为 0，也可以是右支(或下支)为 0，造成编码的不唯一；二是当两个消息的概率相等时，谁前谁后也是随机的，构造出来的码字也不唯一。

② 哈夫曼编码码字字长参差不齐，因此硬件实现起来不大方便。

③ 哈夫曼编码对不同的信源的编码效率是不同的。当信源概率是 2 的负幂时，哈夫曼

码的编码效率达到100%；当信源概率相等时，其编码效率最低。因此，只有在概率分布很不均匀时，哈夫曼编码才会有显著的效果；在信源分布均匀的情况下，一般不使用哈夫曼编码。

④ 对信源进行哈夫曼编码后，形成了一个哈夫曼表。解码时，必须参照哈夫曼编码表才能正确译码。在信源的存储与传输过程中，必须首先存储或传输这一哈夫曼编码表。在实际计算压缩效果时，必须考虑哈夫曼编码表占有的比特数。在某些应用场合，信源概率服从于某一分布或存在一定规律(这主要由大量的统计得到)，这样就可以在发送端和接收端先固定哈夫曼编码表，在传输数据时就省去了传输哈夫曼编码表，这种方法称为哈夫曼编码表的默认使用。虽然这种方法对某一个特定应用来说不一定最好，但从总体上说，只要哈夫曼编码表基于大量概率统计，其编码效果是足够好的。

使用默认的哈夫曼编码表有两方面的好处：一是降低了编码的时间，改变了编码和解码的时间不对称性；二是使用硬件实现，编码和解码电路相对简单。这种方法适用于实时性要求较强的场合。

主教材图4.7和图4.8给出了一个具体例子，说明了哈夫曼编码过程、步骤以及码字的构成等。

3) 算术编码

(1) 算术编码原理

算术编码的方法是将被编码的信源消息表示成实数轴0～1之间的一个间隔(也称子区间)，消息越长，编码表示它的间隔就越小，表示这一间隔所需的二进制位数就越多。信源中连续符号根据某一模式生成概率的大小来缩小间隔，可能出现的符号要比不太可能出现的符号缩小范围少，只增加了较少的比特。

下面举一个简单的例子来说明算术编码的原理过程。

例：已知信源 $\boldsymbol{X}=\begin{Bmatrix} 0 & 1 \\ 1/4 & 3/4 \end{Bmatrix}$，对1011进行算术编码。

① 二进制信源符号只有两个，即0和1，设置小概率 $Q_e=1/4$，大概率 $P_e=1-Q_e=3/4$。

② 设 C 为子区间的左端起始位置，L 为子区间的长度(等效于符号概率)，根据(a)知，符号0的子区间为[0,1/4)；0的子区间左端 $B=0$，子区间长 $L=1/4$；符号1区间为[1/4,1)；1的子区间左端 $B=1/4$，子区间长 $L=3/4$。

③ 在编码运算过程中，随着消息符号出现，子区间按下列规则缩小。

规则A：新子区间左端＝前子区间左端＋当前子区间左端×前子区间长度

规则B：新子区间长度＝前子区间长度×当前子区间的长度

④ 初始子区间为[0,1)，即 $0\leqslant X<1$。

编码算法过程如下：

步序	符号	C	L
1	1	1/4	3/4
2	0	1/4+0×3/4=1/4	3/4×1/4=3/16
3	1	1/4+1/4×3/16=19/64	3/16×3/4=9/64
4	1	9/64+1/4×9/64=85/256	9/64×3/4=27/256

最后的子区间左端(起始位置) $C=(85/256)d=(0.01010101)$b。最后的子区间长度

$L=(27/256)d=(0.0111)b$ 编码结果为子区间头尾之间取值，其值为 0.011，可编码为 011，原来 4 个符号 1011 被压缩为 3 个符号 011。

解码过程是逆过程，首先将区间[1,0)，按 Q_e 靠近 0 侧，P_e 靠近 1 侧分割成两个子区间，判断被解码字落在哪个子区间，而赋予对应符号。有关解码算法举例参考主教材的内容。

(2) 算术编码的特点

- 算术编码的模式选择直接影响编码效率，有固定模式，也有自适应模式。
- 算术编码的自适应模式无须先定义概率模型，对无法进行概率统计的信源合适，在这点上优越于哈夫曼编码。
- 在信源符号概率接近时，算术编码比哈夫曼编码效率高。
- 算术编码的硬件实现比哈夫曼编码要复杂些。
- 算术编码在 JPEG 的扩展系统中被推荐代替哈夫曼编码。

JPEG 标准采用了混合编码方法，定义了两种基本压缩算法：一种是基于空间线性预测技术(即差分脉冲调制)的无失真压缩算法；另一种是基于离散余弦变换，并应用行程编码和熵编码的有失真压缩算法。JPEG 标准的有损压缩率为 10∶1～100∶1，如果压缩率小于 40，所获图像与原图主观效果几乎一样。JPEG 标准的无损压缩率大约为 4∶1。

JPEG 算法主要存储颜色变化，尤其是亮度变化，因为人眼对亮度变化要比对颜色变化更为敏感。只要压缩后重建的图像与原来图像在亮度变化、颜色变化上相似，在人眼看来就是同样的图像。其原理是不重建原始画面，而生成与原始画面类似的图像，丢掉那些未被注意到的颜色。

1) JPEG 压缩的适用范围

JPEG 只有帧内编码，每帧可随机存取。JPEG 连续色调图像压缩方法满足以下要求：

① 可大范围调节图像压缩率及其相应的图像保真度，解码器可参数化，用户应用可以选择期望的压缩/质量比。

② 可应用于任何连续色调数字图像，不限制图像的景象内容。

③ 只需有一定能力的 CPU 就可实现，而不要求很高档的机器，复杂的软件本身要易于操作。

④ 可运行 4 种模式：无失真压缩、基于 DCT 的顺序工作方式、基于 DCT 的累进工作方式和基于 DCT 的分层工作方式。

2) 无失真预测编码

JPEG 选择了简单的线性预测编码方法，具有硬件实现容易、重建质量好的优点。无失真预测编码器框图，如图 4.3 所示。

无失真预测采用三邻域采样值法，由 a、b、c 预测 x，如图 4.4 所示。以 x' 表示 x 的预测值(x 是该点的实际值)，从 x 中减去 x' 得到一差值，再对差值进行无失真的熵编码(可采用哈夫曼算术编码)。预测值 x' 可用表 4.1 所列的选择值，表 4.1 中的 1、2、3 是一维预测，4、5、6、7 是二维预测。无失真预测编码的压缩比一般为 2∶1。

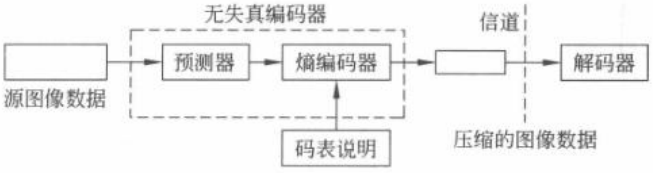


图 4.3　无失真预测编码器框图

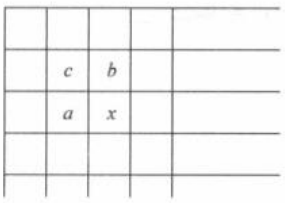


图 4.4　三邻域预测方法

表 4.1　无失真预测编码预测值可选择值

选　择　值	预　　测	选　择　值	预　　测
0	非预测	4	$A+B-C$
1	a	5	$A+[(B-C)/2]$
2	b	6	$B+[(A-C)/2]$
3	c	7	$(A+B)/2$

3）基于 DCT 的有失真压缩编码

基于离散余弦变换(DCT)的压缩编码算法有两种不同层次的系统：基本系统和增强系统，并且定义了两种工作方式：顺序方式和累进方式。基本系统采用顺序工作方式，编码过程中只采用哈夫曼编码，解码只能存储两套哈夫曼表。增强系统是基本系统的扩充或增强，它采用累进工作方式，编码过程可采用自适应能力的算术编码。

(1) 离散余弦变换

JPEG 采用子块为 8×8 的二维离散余弦变换。在编码器的输入端，把原始图像顺序地分割成一系列 8×8 的子块。设原始图像的采样精度为 P 位，是无符号整数，输入时把$[0, 2^{P-1}]$范围内的无符号整数变成$[-2^{P-1}, 2^{P-1}-1]$范围内的有符号整数，以此作为离散余弦正变换(Forward DCT)的输入。在解码器的输出端经离散余弦逆变换(Inverse DCT)后，得到一系列 8×8 的图像数据块，需将其数值范围由$[-2^{P-1}, 2^{P-1}-1]$再变回到$[0, 2^P-1]$范围内的无符号整数来获得重构图像。

下面是 8×8 FDCT 和 8×8 IDCT 的数学变换公式：

$$F(u,v)=\frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right]$$

逆变换如下：

$$f(x,y)=\frac{1}{4}\left[\sum_{x=0}^{7}\sum_{y=0}^{7}C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right]$$

其中 $\begin{cases}C(u),C(v)=1/\sqrt{2}, & \text{当 } u,v=0\\ C(u),C(v)=1, & \text{其他}\end{cases}$

二维快速余弦变换算法是把 8×8 子块不断地对半分成无交叠子块，重新排列数据块直到 1×1 子块，并直接对数据进行运算操作。

离散余弦变换逆变换由 64 个离散余弦变换系数经逆变换重建 64 点的图像，由于计算过程中不可避免的精度损失以及系数的量化，这个 64 点的图像不能完全恢复到原始图像。一个单分量，如图像的灰度信号符，基于离散余弦变换的压缩编码过程如图 4.5 所示。其解码过程如图 4.6 所示。对于彩色图像则可近似作为多分量进行压缩和解压缩。

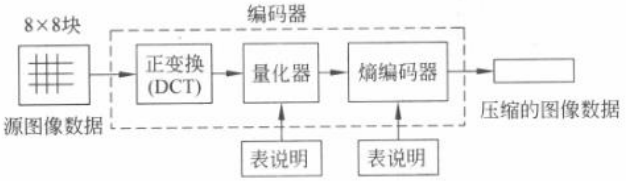


图 4.5　基于 DCT 压缩编码过程

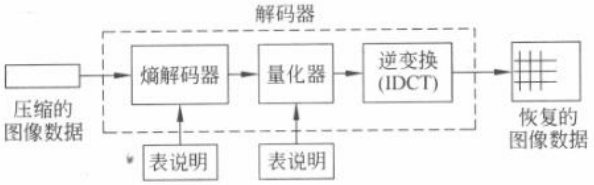


图 4.6　基于 DCT 的压缩解码过程

(2) 离散余弦变换系数的量化处理

为达到压缩数据的目的，对离散余弦变换系数 $F(u,v)$需作量化处理。不同频率的余弦函数的视觉效果不同。量化步长是量化表的元素，可按不同频率的视觉值来选择量化表中元素的大小。实际设计中可通过心理视觉实验确定对应于不同频率的视觉阈值，以确定不同频率的量化器步长。

量化处理是多到一的映射，是造成离散余弦变换编码解码信息损失的根源。在 JPEG 中采用线性均匀量化器。量化定义为 64 个离弦系数除以量化步长后，四舍五入取整，表达式如下：

$$F^Q(u,v) = \text{IntegerRound}[F(u,v)/Q(u,v)]$$

其中，$Q(u,v)$为量化器步长，是量化表的元素(量化表的元素随离散余弦变换系数的位置和彩色分量的不同有不同的值)。量化表的尺寸为 8×8，与 64 个变换系数一一对应。量化表可由用户规定，但 JPEG 给出了参考值，并作为编码器的一个输入。量化表的多个元素值为 1～255 之间的任意整数，共值规定了它到对应的离散余弦变换系数的量化器步长。

反量化表达式如下：

$$F^Q(u,v) = F^Q(u,v)Q(u,v)$$

量化结果一般是频率低的分量系数大，频率高的分量系数小且大多为零。

(3) 直流(DC)系数的编码和交流(AC)系数的行程编码

64 个变换系数中，直流系数处于矩阵的左上角，是 64 个图像采样的平均值。相邻 8×8 块之间的直流系数有很强的相关性。JPEG 对于量化后的直流系数采用差分编码(DPCM)，即对相邻块之间的直流系数的差值 $DIFF = DC_j - DC_{j-1}$编码。其余 63 个交流系数采用行程编码。从左上方 AC_{01}开始沿对角线方向，以 Z 字形进行行程扫描，直至 AC_{77}扫描结束。量化后的交流系数中通常会有许多零值，以 Z 字形路经进行行程编辑，可增加行程中连续的零的个数，如图 4.7 所示。

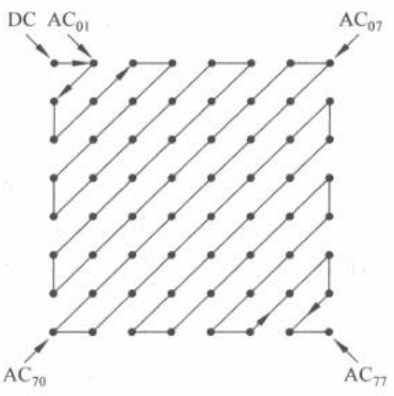


图 4.7　Z 字形排列

2. 运动图像压缩编码标准(MPEG)

MPEG 标准是 ISO/IEC 委员会的第 11172 号标准，是针对全活动视频的压缩标准。该标准包括 MPEG 视频、MPEG 音频和 MPEG 系统 3 大部分。MPEG 视频是面向位速率约 1.5Mb/s 全屏幕运动图像的数据压缩；MPEG 音频是面向每通道数率为 64Kb/s、128Kb/s 和 192Kb/s 的数字音频信号的压缩。

MPEG 输入图像亮度信号的分辨率为 360×240，色度信号的分辨率为 180×120，每秒 29.97 帧，采用双向运动补偿。MPEG 把输入的视频信号分成组，用 3 种图像格式标出：帧内图像、预测图像和差补图像。每组中的第 1 帧用帧内图像格式编码，第 1M、2M、3M 帧(M 一般选为 3)用预测图像格式编码，其他各帧使用差补图像格式编码。差补图像不仅利用过去的帧内图像或预测图像，也利用未来的帧内图像或预测图像进行运动补偿，因此可以达到更高的图像压缩率。

1) MPEG-1 标准

MPEG-1 标准是运动图像专家小组 1981 年制定的数字存储运动图像及伴音编码标准。该标准分为视频、音频和系统 3 部分。它是一个通用标准，既考虑了应用要求，又独立于具体应用之上。视频部分为 1.5Mb/s 活动图像压缩编码算法，对于带宽为 1.5Mb/s 的位流，能够获得可接受的图像质量。该算法帧内编码采用二维余弦变换、自适应量化、行程编码、变字长编码和 DPCM 技术，帧间编码采用运动补偿预测和运动补偿内插技术。MPEG-1 对于较低的传输速率、窄带宽的应用(如单速 CD-ROM)是相当完善的，并通过插值可处理大于 352×240 的画面。

MPEG 应用的数字存储媒体包括光盘(CD-ROM)、数字录音带(DAT)、磁盘和可写光盘，通信网络包括集成服务数字网络(ISDN)和局域网(LAN)等。视频压缩算法必须有与存储相适应的性质，即能够随机访问、快进和快退、检索、倒放、音像同步、容错，延时控制(在 150ms 之内)，可编辑性，以及灵活的视频窗口格式。实现这些特性对各种应用十分重要。

设计 MPEG 视频压缩算法的一个矛盾是：仅靠帧内编码无法达到在保证画面质量前提下的高压缩比，而满足随机访问条件的最好算法是帧内编码。为满足高压缩和随机访问两方面要求，MPEG 采取了预测和插值两种帧间编码技术。

MPEG 视频压缩算法有两个技术基础：块基(block-based)运动补偿(缩减时间冗余)和域基(domain-based)变换(缩减空间冗余)。运动补偿技术采用因果和非因果两种编码方法(即纯预测编码和插值预测编码)。剩余信号(预测误差)在缩减空间冗余时被进一步压缩，与运动有关的信息包含在 16×16 的剩块中，与空间信息一起进行变换。为获得最高效率，用可变长代码压缩运动信息。

(1) 时域冗余量的减少

MPEG 中考虑了 3 种画面：内帧(I)、预测帧(P)和内插帧(B)(双向预测)。这样做的原因有两个：一是考虑随机访问视频存储的重要性；二是运动补偿可显著降低位速率。内帧经过中度压缩可作为随机访问点；预测帧以参照帧(I 或 P)基础进行编码，它又是后面预测帧的参照帧；内插帧压缩比最高，它需要前后两个参照帧，但它本身不可作为参照帧使用。3 种画面的相互关系如图 4.8 所示。

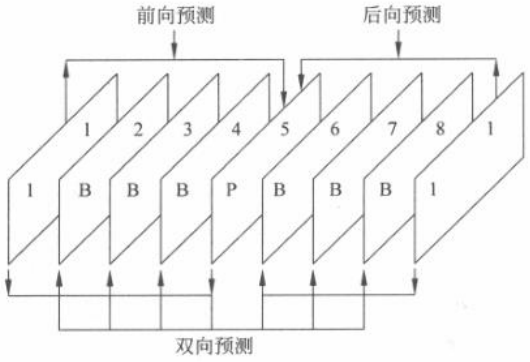


图 4.8　内帧、预测帧和内插帧三种画面的相互关系

在预测编码中，运动补偿方法可大大提高编码效率。运动补偿技术假设每帧当前画面

都可以以前面某一帧为原型经过变换而得到，这一变换是局部的，即画面上各点位移的方向和大小不必相同。运动信息由一个前向预测、一个后向预测和两个双向预测微块向量构成。每个16×16块的运动信息都与邻块有些不同，通过对比可确定运动向量的变化范围，并使之与时间分辨率、空间分辨率及画面内容相匹配。标准所定的范围很大，足够适用于一些特殊情况。

(2) 空域冗余量的减少

MPEG视频信息的帧内图和预测图都有很高的空域冗余度，用于减少这方面冗余的技术很多，但由于运动补偿处理基于块的特性，基于块的技术用于此处更合适些。在基于块的空间冗余压缩技术中，变换编码技术和矢量量化技术是非常有用的。在正交变换中，DCT具有许多明显的优点，且相对来说较易实现，所以帧内压缩也采用基于DCT的方法。这和静态图像的压缩标准(JPEG)相同，实现的步骤也一样。只是在JPEG压缩算法中，针对静止图像，对DCT系数采用等宽量化。在MPEG中的视频信号包含有静止画面(帧内图)和运动信息(帧间预测图)等不同的内容，故量化器的设计需做特殊的考虑。一方面量化器结合行程编码能使大部分数据得以压缩；另一方面要求通过量化器，编码器使之输出一个与信道传输速率匹配的比特流。因此，如何设计一个能够满足上述要求，且有满意的视觉质量的量化器，在MPEG中是十分重要的。

2) MPEG-2标准

MPEG-2是MPEG-1的扩充、丰富和完善。MPEG-2标准的视频数据速率为4～5Mb/s，能提供720×480(NTSC)或720×576(PAL)分辨率的广播级质量的视像，适用于包括宽屏幕和高清晰度电视(HDTV)在内的高质量电视和广播。

MPEG-2的特点如下：

① 解码器通常支持MPEG-1和MPEG-2两种标准。

② 基本分辨率为720×480，传输速率为每秒30帧，并具有CD的音质。

③ 允许在一定范围内改变压缩比，以便在画面质量、存储容量和带宽之间做出权衡。可以30∶1或更低的压缩比提供广播级的质量。

④ 压缩比可高达200∶1，能够以每秒30帧的速度播放全屏影像。实际的压缩比依赖于节目的内容和所需的重放质量。运动和背景的变化越多，所得到的压缩比就越低。

⑤ 能够对分辨率可变的视频信号进行压缩编码，预计的传输速率将为10Mb/s。

通常MPEG-2要用几百秒的时间去压缩用于播放1秒钟的电视画面，实时的MPEG-2压缩编码一直是很困难的，实现起来十分昂贵。与P×64相比实时性好，而MPEG压缩率高。

DVD格式的视频部分将采用MPEG-2压缩标准，音频部分压缩标准将随电视制式而异。MPEG-2压缩标准已被以欧洲为主的国家采纳并用于PAL制的音频中，美国和日本在NTSC制中采用的是AC3音频压缩标准。

有关MPEG-4和MPEG-7和MPEG-21标准在主教材中均作了详细的介绍。

8. 简述MPEG和JPEG的主要差别。

8. 答：MPEG 视频压缩技术是针对运动图像的数据压缩技术。为了提高压缩比，帧内图像数据和帧间图像数据压缩技术必须同时使用。

MPEG 通过帧运动补偿有效地压缩了数据的比特数，而 MPEG 采用了 3 种图像，帧内图、预测图和双向预测图，有效地减少了冗余信息。对于 MPEG 来说，帧间数据压缩、运动补偿和双向预测，这是和 JPEG 主要不同的地方。而 JPEG 和 MPEG 相同的地方均采用了 DCT 帧内图像数据压缩编码。

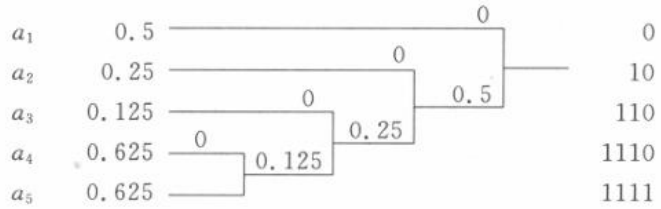
在 JPEG 压缩算法中，针对静态图像对 DCT 系数采用等宽量化，而 MPEG 中视频信号包含有静止画面（帧内图）和运动信息（帧间预测图）等不同的内容，量化器的设计不能采用等宽量化，需要作特殊考虑。从两方面设计，一是量化器综合行程编码能使大部分数据得到压缩；另一方面是通过量化器、编码器使之输出一个与信道传输速率匹配的比特流。

9. 信源符号及其概率如下：

a	a_1	a_2	a_3	a_4	a_5
$p(a)$	0.5	0.25	0.125	0.0625	0.0625

求其 Huffman 编码、信息熵及平均码长。

9. 解：



则：$a_1=0$　$a_2=10$　$a_3=110$　$a_4=1110$　$a_5=1111$

信息熵：

$$H=-\sum_{i=1}^{n=5}P_i\log_2(p_i)=-\left(\frac{1}{2}\log_2\frac{1}{2}+\frac{1}{4}\log_2\frac{1}{4}+\frac{1}{8}\log_2\frac{1}{8}+\frac{1}{16}\log_2\frac{1}{16}+\frac{1}{16}\log_2\frac{1}{16}\right)$$

$$=\frac{1}{2}+\frac{1}{2}+\frac{3}{8}+\frac{1}{4}+\frac{1}{4}=1+\frac{7}{8}=1.875\text{ 位 / 字符}$$

$a_1\sim a_5$ 码长分别为 1,2,3,4,4。

则平均码长 $\bar{N}=\sum_{i=1}^{n=5}P_iL_j=\frac{1}{2}\times1+\frac{1}{4}\times2+\frac{1}{8}\times3+\frac{1}{16}\times4+\frac{1}{16}\times4$

$=1.875$ 位 / 字符

10. 说明 JPEG 静态图像压缩编码原理及其实现技术。

10. 答：JPEG 是由国际电报电话咨询委员会（CCITT）和国际标准化协会（OSI）联合组成的一个图像专家小组开发研制出的连续色调、多级灰度、静止图像的数字图像压缩编码方法。JPEG 适于静止图像的压缩，此外，电视图像序列的帧内图像的压缩编码也常采用 JPEG 压缩标准。JPEG 数字图像压缩文件，作为一种数据类型，如同文本和图形文件一样地储存和传输。基于离散余弦变换（DCT）的编码方法是 JPEG 算法的核心内容。算法的编

解码过程如图 A-2 所示。编码处理过程包括原图像数据输入、正向 DCT 变换器、量化器、熵编码器和压缩图像数据的输出，除此之外还附有量化表和熵编码表(即哈夫曼表)；接收端由信道收到压缩图像数据流后，经过熵解码器、逆量化器、逆变换(IDCT)，恢复并重构出数字图像，量化表和熵编码表同发送端完全一致。编码原图像输入，可以是单色图像的灰度值，也可以是彩色图像的亮度分量或色差分量信号。DCT 的变换压缩是对一系列 8×8 采样数据作块变换压缩处理，可以对一幅像，从左到右、从上到下、一块一块(8×8/块)地变换压缩，或者对多幅图轮流取 8×8 采样数据块压缩。解码输出数据，需按照编码时的分块顺序作重构处理，得到恢复数字图像。

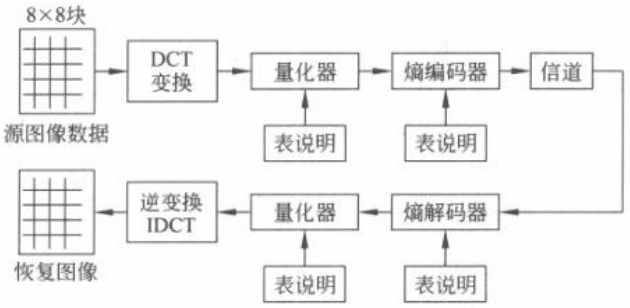


图 A-2　基于 DCT 编码解码过程

具体的实现技术如下：

① 首先把一幅图像分 8×8 的子块，按图 A-2 进行离散余弦正变换(FDCT)和离散余弦逆变换(IDCT)。

在编码器的输入端，原始图像被分成一系列 8×8 的块，作为离散余弦正变换(FDCT)的输入。在解码器的输出端，离散余弦逆变换(IDCT)输出许多 8×8 的数据块，用以重构图像。8×8 FDCT 和 8×8 IDCT 数学定义表达式如下：

FDCT：

$$F(u,v)=\frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7}f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right]$$

IDCT：

$$F(x,y)=\frac{1}{4}\left[\sum_{u=0}^{7}\sum_{v=0}^{7}C(u)C(v)f(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}\right]$$

两式中，$C(u),C(v)=1/\sqrt{2}$，　当 $u=v=0$

$C(u),C(v)=1$，　　其他

离散余弦正变换(FDCT)可看作一个谐波分析仪，同时可把离散余弦逆变换(IDCT)看做一个谐波合成器。每个 8×8 二维原图像采样数据块，实际上是 64 点离散信号，该信号是空间二维参数 x 和 y 的函数。FDCT 把这些信号作为输入，然后把它分解成 64 个正交基信号，每个正交基信号对应于 64 个二维(2D)空间频率中的一个，这些空间频率是由输入信号的频谱组成的。FDCT 的输出是 64 个基信号的幅值(即 DCT 系数)，每个系数值由 64 点输入信号唯一地确定，即离散余弦变换的变换系数。在频域平面上变换系数是二维频域变量 u 和 v 的函数。对应于 $u=0,v=0$ 的系数，称做直流分量(DC 系数)，其余 63 个系数称做交

流分量(AC 系数)。因为在一幅图像中像素之间的灰度或色差信号变化缓慢,在 8×8 子块中像素之间相关性很强,所以通过离散余弦正变换处理后,在空间频率低频范围内集中了数值大的系数,这样为数据压缩提供了可能。远离直流系数的高频交流系数大多为零或趋于零。如果 FDCT 和 IDCT 变换计算中计算精度足够高,并且 DCT 系数没有被量化,那么原始的 64 点信号就能精确地恢复。

② 量化过程如下。

为了达到压缩数据的目的,对 DCT 系数 $F(u,v)$需作量化处理。量化处理是一个多到一的映射,它是造成 DCT 编解码信息损失的根源。在 JPEG 标准中采用线性均匀量化器。量化定义为,对 64 个 DCT 变换系数 $F(u,v)$除以量化步长 $Q(u,v)$后四舍五入取整。即量化器步长是量化表的元素,量化表元素随 DCT 变换系数的位置而改变,同一像素的亮度量化表和色差量化表不同值,量化表的尺寸也是 64,与 64 个变换系数一一对应。量化表中的每一个元素值为 1～255 之间的任意整数,其值规定了对应位置变换系数的量化器步长。在接收端要进行逆量化,逆量化的计算公式为:

$$F^Q(u,v) = F^Q(u,v) \cdot Q(u,v)$$

不同频率的余弦函数对视觉的影响不同,量化处理是在一定的主观保真度图像质量的前提下,根据不同频率的视觉阈值来选择量化表中的元素值的大小的。根据心理视觉加权函数得到亮度化表和色度量化表。DCT 变换系数 $F(u,v)$除以量化表中对应位置的量化步长,其幅值下降,动态范围变窄,高频系数的零值数目增加。

③ 熵编码过程如下。

为进一步达到压缩数据的目的,需对量化后的 DC 系数和行程编码后的 AC 系数进行基于统计特性的熵编码。64 个变换系数经量化后,坐标 $u=v=0$ 的值是直流分量(即 DC 系数)。DC 系数是 64 个图像采样平均值。因为相邻的 8×8 块之间有强的相关性,所以相邻块的 DC 系数值很接近,对量化后前后两块之间的 DC 系数差值进行编码,可以用较少的比特数。DC 系数包含了整个图像能量的主要部分。经量化后的 63 个 AC 系数编码是从左上方 AC($u=7,v=7$)开始,沿箭头方向,以 Z 字形行程扫描,直到 AC($u=7,v=7$)扫描结束。量化后特性编码的 AC 系数通常有许多零值,沿 Z 字形路径行进,可使零 AC 系数集中,便于使用行程编码方法。63 个 AC 系数行程编码和码字,可用两个字节表示。JPEG 建议使用两种熵编码方法,即 Huffman 编码和自适应二进制算术编码。熵编码可分成两步进行,首先把 DC 和 AC 系数转换成一个中间格式的符号序列,第二步是给这些符号赋以变长码字。