

第4章

图像编码

第一讲

模拟图像信号在传输过程中极易受到各种噪声的干扰，而且模拟图像信号一旦受到污染则很难完全得到恢复。另外，在模拟领域中，要进行人与机器（计算机或智能机），机器与机器之间的信息交换以及对图像进行诸如压缩、增强、恢复、特征提取和识别等一系列处理都是比较困难的。

所以无论从完成图像通信与数据通信网的结合方面来看，还是从对图像信号进行各种处理的角度来看，图像信号数字化都是首当其冲的重要问题。本章将对图像编码进行较为详细的讨论。



4.1 图像编码分类

编码是把模拟制信号转换为数字制信号的一种技术。**图像编码**技术**不仅是**应用线性脉冲编码调制法（**线性PCM**），**更是**利用图像信号的统计特性及视觉对图像的生理学和心理学特性，对图像进行**信源编码**。在信息论中将数字通信过程概括为图4—1的形式。

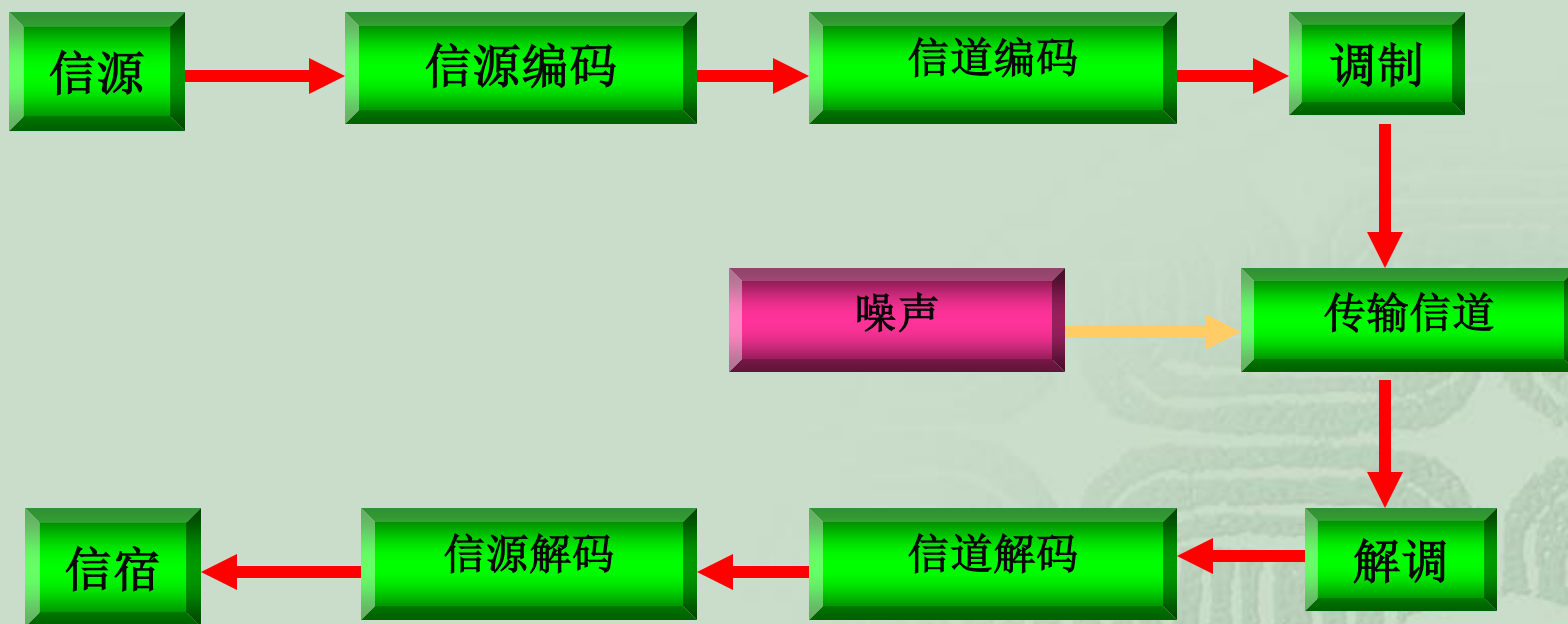


图4—1 数字通信系统模型



图4—1所示的模型是一个数字通信系统模型。在这样一个模型中有二次编、译码，

信源编码、信源译码

信道编码、信道译码。



信源编码的主要任务是解决有效性问题，也就是对信源实现压缩处理，使处理后的信号更适宜数字通信系统。解决有效性问题就是在编码过程中尽量提高编码效率，也就是力求用最少的数码传递最大的信息量。



信道编码的任务是解决可靠性问题。也就是尽量使处理过的信号在传输过程中不出错或少出错，即使出了错也要有能力尽量纠正错误。因此，在信道编码中往往引进用作误差控制的数码，以实现自动检错和纠错。

图像编码主要是要研究压缩数码率，即高效编码问题。



编码是信息处理科学中的经典研究课题，

就图像编码而言，已有近60多年的历史。

近年来，M. Kunt提出第一代、第二代编码的概念。



M. Kunt把1948年至1988年这四十年中研究的以去除冗余为基础的编码方法称为第一代编码，如：PCM、DPCM、 ΔM 、亚取样编码法，变换域的DFT、DCT、Walsh-Hadamard变换编码等方法以及以此为基础的混合编码法均属于经典的第一代编码法。



第二代编码方法多是80年代以后提出的

新的编码方法，如金字塔编码法、

Fractal编码、基于神经元网络的编码方

法、小波变换编码法、模型基编码法等。



现代编码法的特点是：

- ①充分考虑人的视觉特性；
- ②恰当地考虑对图像信号的分解与表述；
- ③采用图像的合成与识别方案压缩数据率。

这种分法尽管并没得到图像编码界全体同仁的广泛认可，但笔者认为对了解图像编码发展进程是有益的。

图像编码这一经典的研究课题，经**60**多年的研究
已有多种成熟的方法得到应用，特别是所谓的第一
代编码更是如此。

随着多媒体技术的发展已有若干编码标准由ITU-T
制定出来，如JPEG、H. 261、H. 263、MPEG1、MPEG2、
MPEG4、MPEG7、JBIG(二值图像压缩) (Joint Bo-
level Image Coding Expert Group) 等。

图像编码是属于信源编码的范畴，从信源编码的角度来分类，图像编码大致可分为

- 1) 匹配编码;
- 2) 变换编码;
- 3) 识别编码



另外，如果从目前已有的**实用方案的角度来分类**，可以分为三大类，即预测编码，变换编码及统计编码。而这些方法既适用于静止图像编码，也适用于电视信号编码。就具体编码方法而言可简略地概括在图4—2中。



图像高效编码法

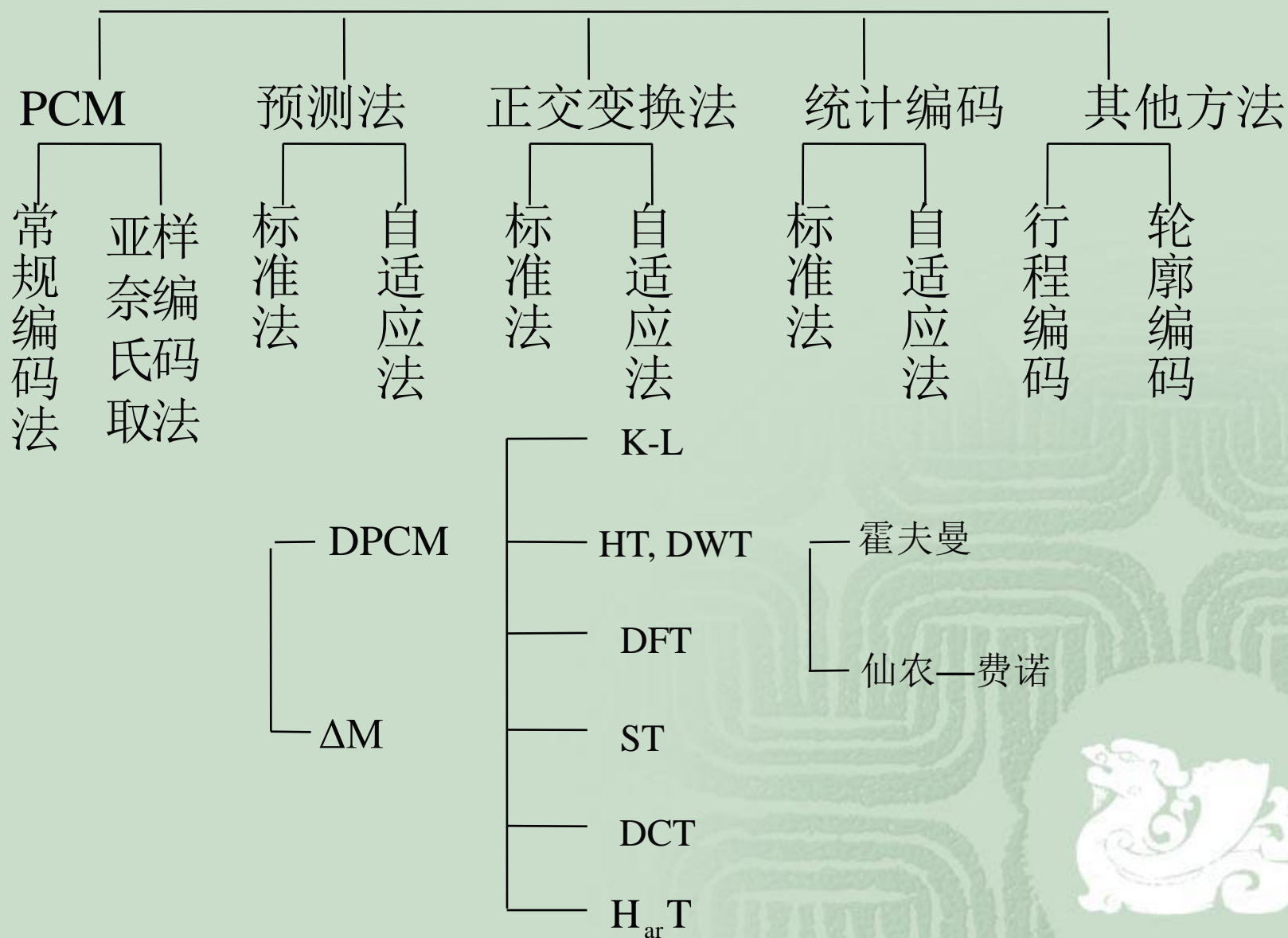


图4—2 图像高效编码分类

最后需要着重提及的是，上述各种具体方案并不是孤立的、单一地使用，往往是各种方法混合、交叉使用，以达到更高的编码效率，在ITU-T的建议标准中这一点尤为突出。



4.2 图像编码中的保真度准则

图像信号在编码和传输过程中会产生误差，尤其是在熵压缩编码中，产生的误差应在允许的范围之内。在这种情况下，**保真度准则**可以用来衡量编码方法或系统质量的优劣。通常，这种衡量的尺度可分为**客观**保真度准则和**主观**保真度准则。

4.2.1 客观保真度准则

通常使用的**客观保真度准则**有输入图像和输出图像的**均方根误差**；输入图像和输出图像的**均方根信噪比**两种。输入图像和输出图像的均方根误差是这样定义的。设输入图像是由 $N \times N$ 个像素组成，令其为 $f(x,y)$ ，其中 $x,y=0, 1, 2, 3, \dots, N-1$ 。



这样一幅图像经过压缩编码处理后，送至受信端，再经译码处理，重建原来图像。这里令重建图像为 $g(x,y)$ 。它同样包含 $N \times N$ 个像素，并且 $x,y=0, 1, 2 \dots N-1$ 。

在 $0, 1, 2, 3 \dots N-1$ 范围内 x,y 的任意值，输入像素和对应的输出像素之间的误差可用下式表示



$$e(x, y) = g(x, y) - f(x, y) \quad (4-1)$$

而包含 $N \times N$ 像素的图像的均方误差为

$$\begin{aligned} \bar{e}^2 &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^2(x, y) \\ &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [g(x, y) - f(x, y)]^2 \end{aligned} \quad (4-2)$$

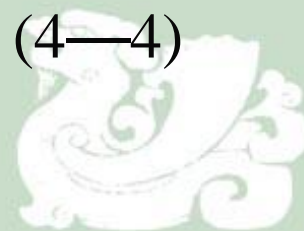


由式(4—2)可得到均方根误差为

$$e_{rms} = \left[\overline{e^2} \right]^{\frac{1}{2}} \quad (4—3)$$

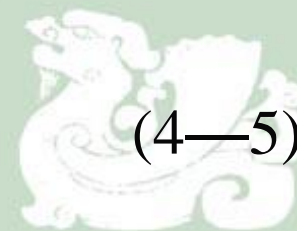
如果把输入/输出图像间的误差看作是噪声，那么，重建图像 $g(x, y)$ 可由下式表示

$$g(x, y) = f(x, y) + e(x, y) \quad (4—4)$$



在这种情况下，另一个客观保真度准则—重建图像的均方信噪比如下式表示

$$\begin{aligned}\left(\frac{S}{N}\right)_{ms} &= \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g^2(x, y)}{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} e^2(x, y)} \\ &= \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g^2(x, y)}{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [g(x, y) - f(x, y)]^2}\end{aligned}$$



均方根信噪比为

$$\left(\frac{S}{N}\right)_{rms} = \left\{ \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g^2(x, y)}{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} [g(x, y) - f(x, y)]^2} \right\}^{\frac{1}{2}} \quad (4-6)$$



4.2.2 主观保真度准则

图像处理的结果，绝大多数场合是给人观看，由研究人员来解释的，因此，图像质量的好坏与否，既与图像本身的客观质量有关，也与人的视觉系统的特性有关。

有时候客观保真度完全一样的两幅图像可能会有完全不同的视觉质量，所以又规定了主观保真度准则。这种方法是把图像显示给观察者，然后把评价结果加以平均，以此来评价一幅图像的主观质量。

一种方法是规定一种绝对尺度，例如：

- （1）优秀的：具有极高质量的图像；
- （2）好的：是可供观赏的高质量的图像，干扰并不令人讨厌；
- （3）可通过的：图像质量可以接受，干扰不讨厌；
- （4）边缘的：图像质量较低，希望能加以改善，干扰有些讨厌；
- （5）劣等的：图像质量很差，尚能观看，干扰显著地令人讨厌；
- （6）不能用：图像质量非常之差，无法观看。

另外常用的还有两种准则，即妨害准则和品质准则。

妨害准则分为5级：

- （1）没有妨害感觉；
- （2）有妨害，但不讨厌；
- （3）能感到妨害，但没有干扰；
- （4）妨害严重，并有明显干扰；
- （5）不能接收信息。



品质准则分为7级：

- （ 1 ） 非常好；
- （ 2 ） 好；
- （ 3 ） 稍好；
- （ 4 ） 普通；
- （ 5 ） 稍坏；
- （ 6 ） 恶劣；
- （ 7 ） 非常恶劣。



除此之外，还可以采用成对比较法，也就是同时出示两幅图像，让观察者表示更喜欢哪一幅图像，借此排出图像质量的等级。

也有采用随机抽取法来评定图像质量的。这种方法是把数量相等的原始图像和经编译码后的图像混杂在一起，然后让观察者挑出他认为质量差的图像。质量较差的图像可定义为处理过的图像，然后统计错挑的概率，显然错挑概率越大说明图像经处理后的劣化越小。总之，主观保真度评价方法的准则可不同，但其基本原理都一样，当然，对观察者的视觉条件应有一定的要求。

5.3 PCM编码

5.3.1 PCM 编码的基本原理

脉冲编码调制 (Pulse coding Modulation, PCM)

是将模拟图像信号变为数字信号的基本手段。图像信号的PCM 编码与语音信号PCM 编码相比并没有原则上的区别。但是，图像信号，特别是电视信号占的频带宽，要求响应速度快，因此，电路设计与实现上有较大的难度。

图像信号PCM 编、译码原理方框图如图5—3所示。图像信号PCM

编码由前置低通滤波器、取样保持电路、量化器、编码器组成。

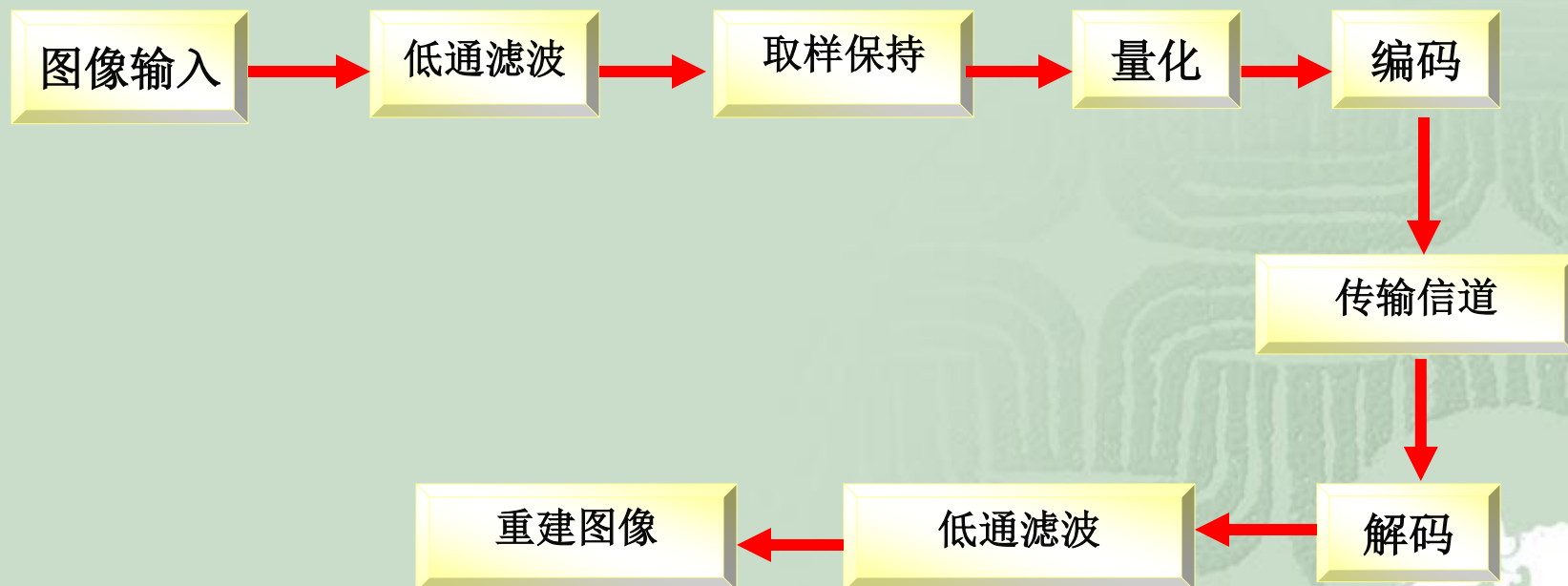


图5—3 图像信号PCM编、译码原理方框图

前置低通滤波器的作用：

- 1) 、为满足取样定理的带限要求，以减少折叠误差；
- 2) 、对杂散噪声也有一定的抑制作用。

取样保持电路：

将完成把时间上连续的模拟信号进行时间离散化的任务。取样周期由奈奎斯特（Nyquist）定理限定。



量化器:

把模拟信号的幅值离散化，经取样与量化处理后就可产生多值数字信号。

编码:

把多值的数字信号变成二进制的数字的多比特信号，以便传输或进行后续处理。



译码器的原理比较简单，它包括一个译码电路和一个低通滤波器。

译码器：

把数字信号恢复为模拟信号，这个模拟信号就是在接收端重建的图像信号。

滤波器：

内插和平滑作用。



5.3.2 PCM 编码的量化噪声

量化是对时间离散的模拟信号进行幅度离散化的过程，这个过程是去零取整的过程。量化后的样值与原信号相比大部分是近似关系。这样，把连续的数值限制在固定的台阶式的变化之下必然会带来畸变。这种畸变在接收端是无法克服的，只能使其尽量减小。



由量化带来的噪声可分为量化噪声和过载噪声。以正弦波输入为例，输入幅度较大和输入幅度较小时的量化噪声如图5—4所示。

(a) 是输入信号超过编码范围时的量化噪声和过载噪声的形成；

(b) 是信号未超过编码范围，只有量化噪声的情况。

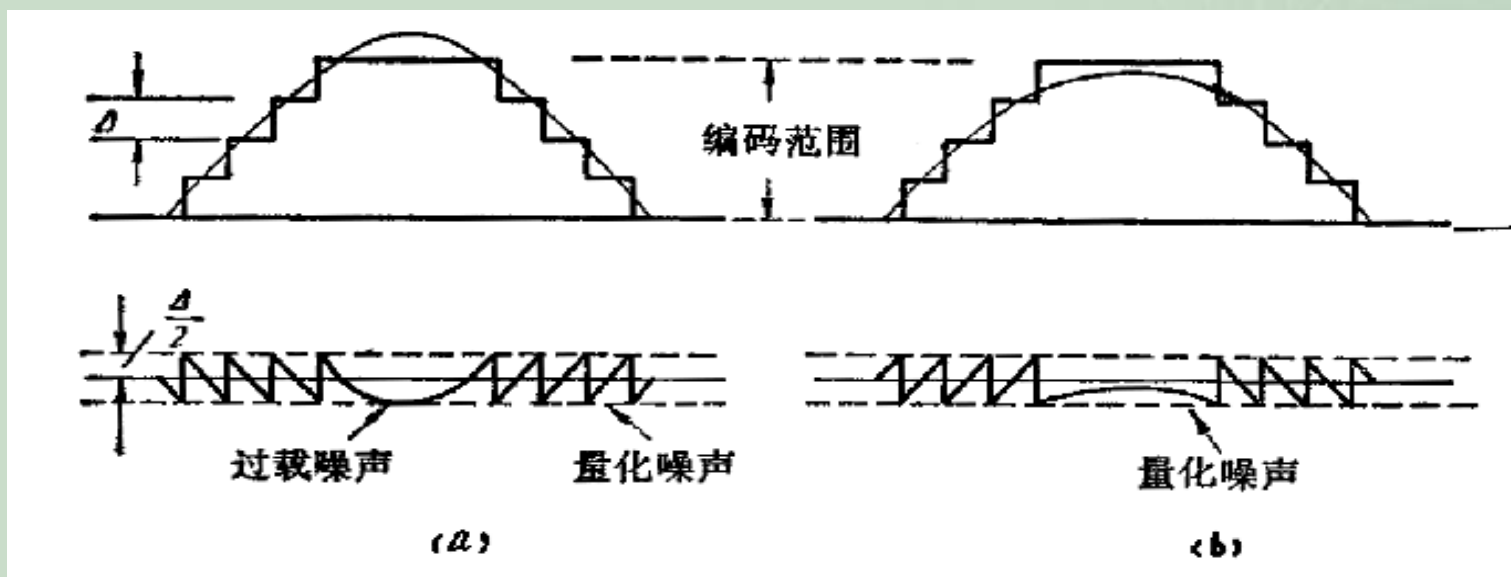


图5—4 量化噪声与过载噪声的形成

在PCM 编码中，量化噪声主要取决于码的位数，码位数越多（即量化阶数多）量化噪声的功率越小。一个量化阶的电压可由下式表示：

$$\Delta = \frac{V}{2^n} \quad (5-7)$$

式中： V 为输入信号电压； n 为样值用二进制数表示的比特数。

如果在整个输入幅度内量化阶是一个常数，就称这个量化为均匀量化，否则就是非均匀量化。线性PCM编码中均采用均匀量化法。在均匀量化中，设量化阶为 Δ ，量化噪声在 $-\frac{\Delta}{2} \sim +\frac{\Delta}{2}$ 内可看成是均匀分布的，因此，其功率可由下式表示：

$$P_Q = \int_{-\frac{\Delta}{2}}^{+\frac{\Delta}{2}} \frac{1}{\Delta} x^2 dx = \frac{\Delta^2}{12} \quad (5-8)$$

对于过载噪声，当量化特性输入过载点为 V 时，由下式表示

$$N_s = \int_{-\infty}^{-V} (x + V)^2 p(x) dx + \int_V^{\infty} (x - V)^2 p(x) dx \quad (5-9)$$

式中 N_s 为过载噪声， x 是输入信号值， $p(x)$ 为输入幅度的概率密度。如果用信噪比作为客观保真度准则的话，可推得PCM 编码在均匀量化下的量化信噪比如下：



因为

$$\Delta = \frac{V}{2^n}$$

$$P_Q = \frac{\Delta^2}{12}$$

所以

$$P_Q = \frac{\frac{V^2}{(2^n)^2}}{12} \quad (5-10)$$



由信噪比的概念，则：

$$\begin{aligned}\left(\frac{S}{N}\right)_{dB} &= 10 \lg \frac{V^2}{P_Q} \\ &= 10 \lg \frac{V^2}{\frac{(2^n)^2}{12}} \\ &= 10 \lg [12 \times (2^n)^2] \\ &= 20 \lg \sqrt{12} + 20 \lg 2^n \\ &\approx (11 + 6n)\end{aligned}\tag{5—11}$$



由式(5—11)可见，每增加一位码可得到6dB的信噪比增益。

值得注意的是量化噪声不同于其他噪声，它的显著特点是仅在信号输入时才出现，所以它是数字化中特有的噪声。一般情况下，直接测量比较困难。



5.4.3 编码器、译码器

编码器的任务是把一个多值的数字量用多比特的二进制量来表示。如果量化器输出 M 个值，那么，对应于 M 个值中的任何一个值编码器将给定一个二进制码字。这个码字将由 m 个二进制数组成。通常情况下 $M = 2^m$ 。编码器的输入与输出关系是一一对应的，其过程是可逆的，因此，不会引入任何误差。



线性**PCM** 编码一般采用等长码，也就是说每一个码字都有相同的比特数。其中用得最为普遍的是自然二进制码，也有用格雷码的。以 **$M=8$** 为例的自然二进制码和格雷码列入表5—1中。



表5-1 M=8的自然二进制码和格雷码

输入	自然二进制	格雷码
m_1	0 0 0	0 0 0
m_2	0 0 1	0 0 1
m_3	0 1 0	0 1 1
m_4	0 1 1	0 1 0
m_5	1 0 0	1 1 0
m_6	1 0 1	1 1 1
m_7	1 1 0	1 0 1
m_8	1 1 1	1 0 0





(a) 原像



(b) 1bit编译码处理的图像





(c) 2bit编译码处理的图像



(d) 3bit编译码处理的图像





(e) 4bit编译码处理的图像 (f) 6bit编译码处理的图像

图 5—5 编码位数对画面质量的影响



5. 3. 4 非线性PCM 编码

在线性PCM编码中，量化阶是均匀的。这样，在小信号输入的情况下信噪比较低，在大信号输入的情况下信噪比较高。为了改善小信号在量化过程中的信噪比，采用一种瞬时压缩扩张技术。

这种技术实际上是降低大信号时的信噪比同时提高小信号时的信噪比，其结果是在不增加数码率的情况下，使信号在整个动态范围内有较均衡的信噪比。

瞬时压扩技术基本有两种方案。一种是如图5—8所示的方案。在这种方案中，首先将取样后的PAM信号进行非线性压缩，然后对压缩后的PAM信号进行线性编码。在接收端，首先进行线性译码，然后再送入瞬时扩张器进行非线性扩张，恢复原来的PAM信号。

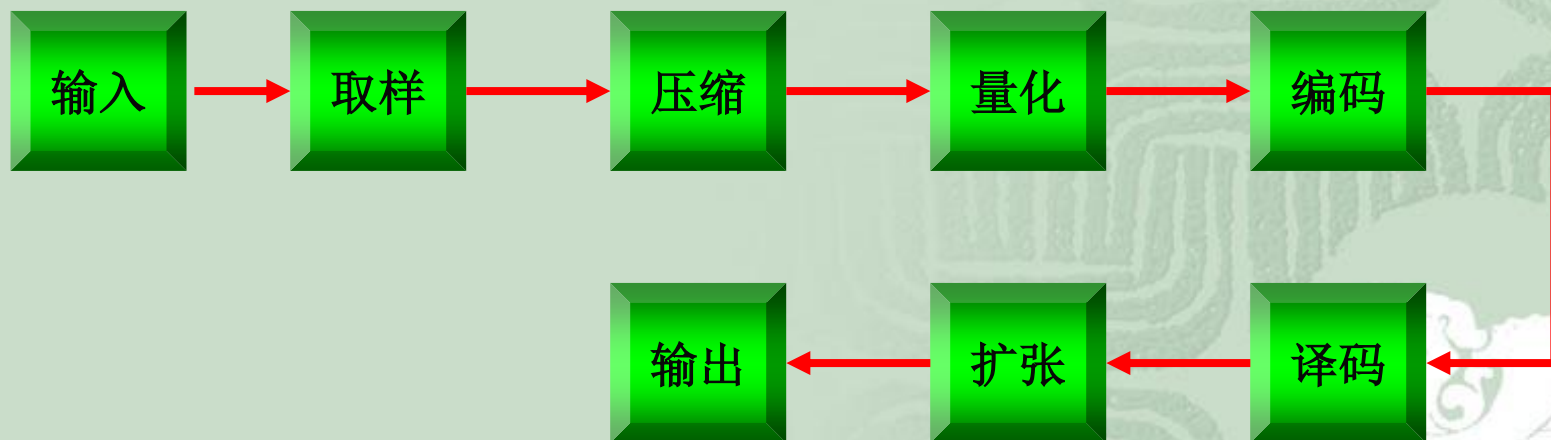


图5—8 一种非线性压扩方案

该方案一般采用二极管电压、电流的非线性特性来实现。通常采用图5—9所示的修正的对数特性。在图5—9中，对于压缩器来说， X 表示输入， Y 表示输出。对于扩张器来说， Y 表示输入， X 表示输出。参变量是表示压缩或扩张程度的量。显然，当 $\mu=0$ 时就是线性编码了。

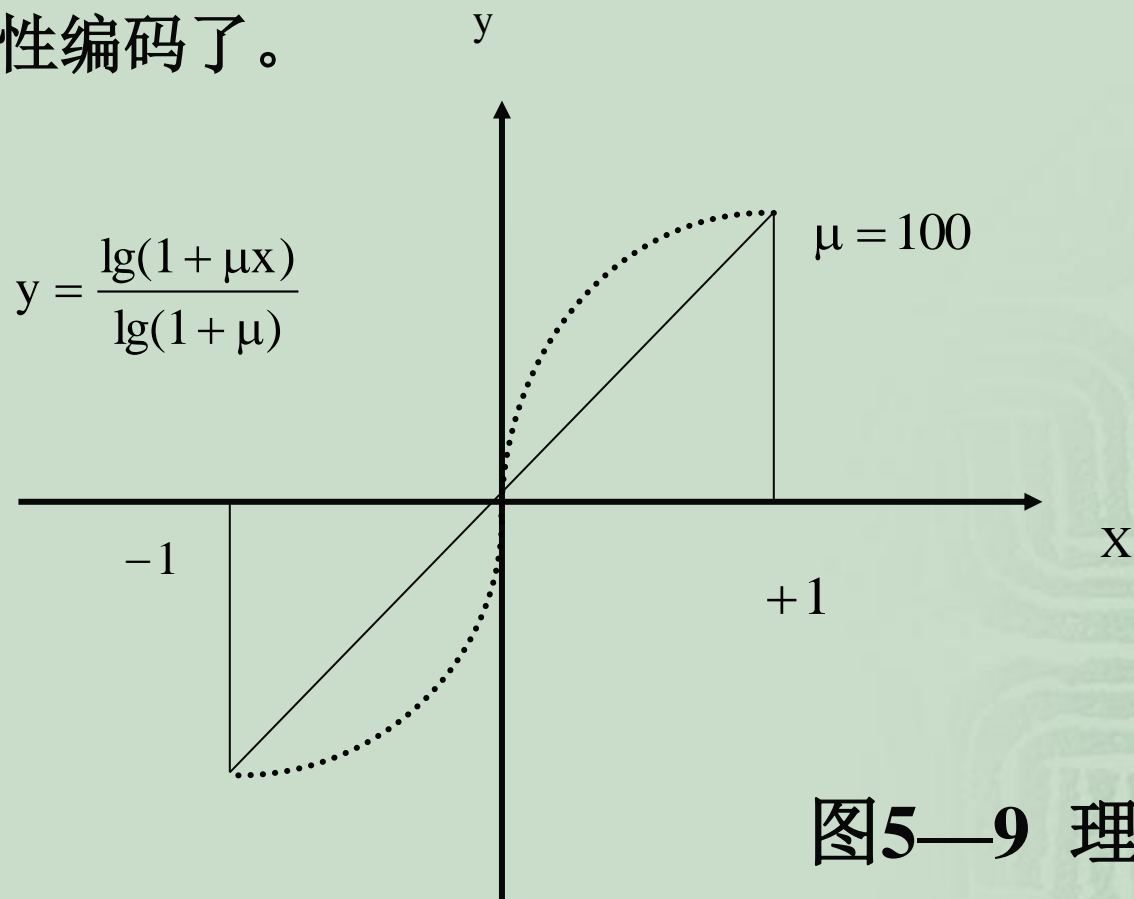


图5—9 理论压扩曲线

另外一种方案是数字化非线性压扩技术，其原理框图如图5-10所示。

这种方案是把编码与压缩，译码与扩张都分别在编码和译码中一次完成。

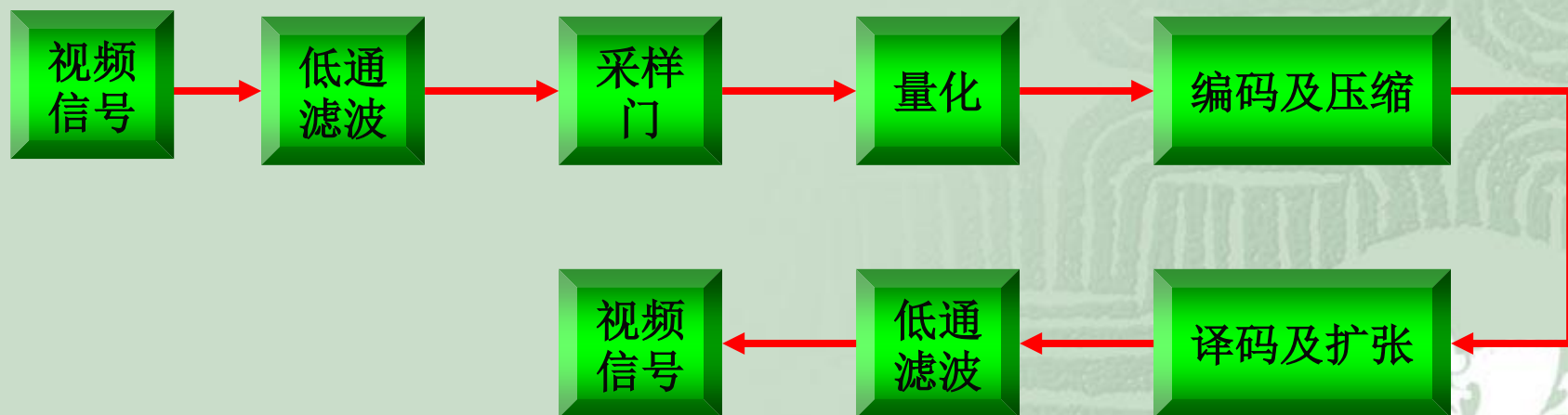


图5—10 数字化非线性压扩技术框图

数字式非线性编码的压扩特性有 μ 特性、 A 特性等等。根据 CCITT1970 年的建议，通常采用 13 折线 ($A=87.6$) 的压扩特性。

13 折线压扩特性如图 5—11 所示。各折线的斜率列于表 5—2 中。由图中可见，各段折线的斜率是不一样的；4 至 8 段的小信号区的信噪比都得到了改善。图中的 $u_{\text{入}}$ 表示压缩器的输入， $u_{\text{出}}$ 表示压缩器的输出， V 为过载点电压。



$$y = \frac{u_{\text{出}}}{V}$$

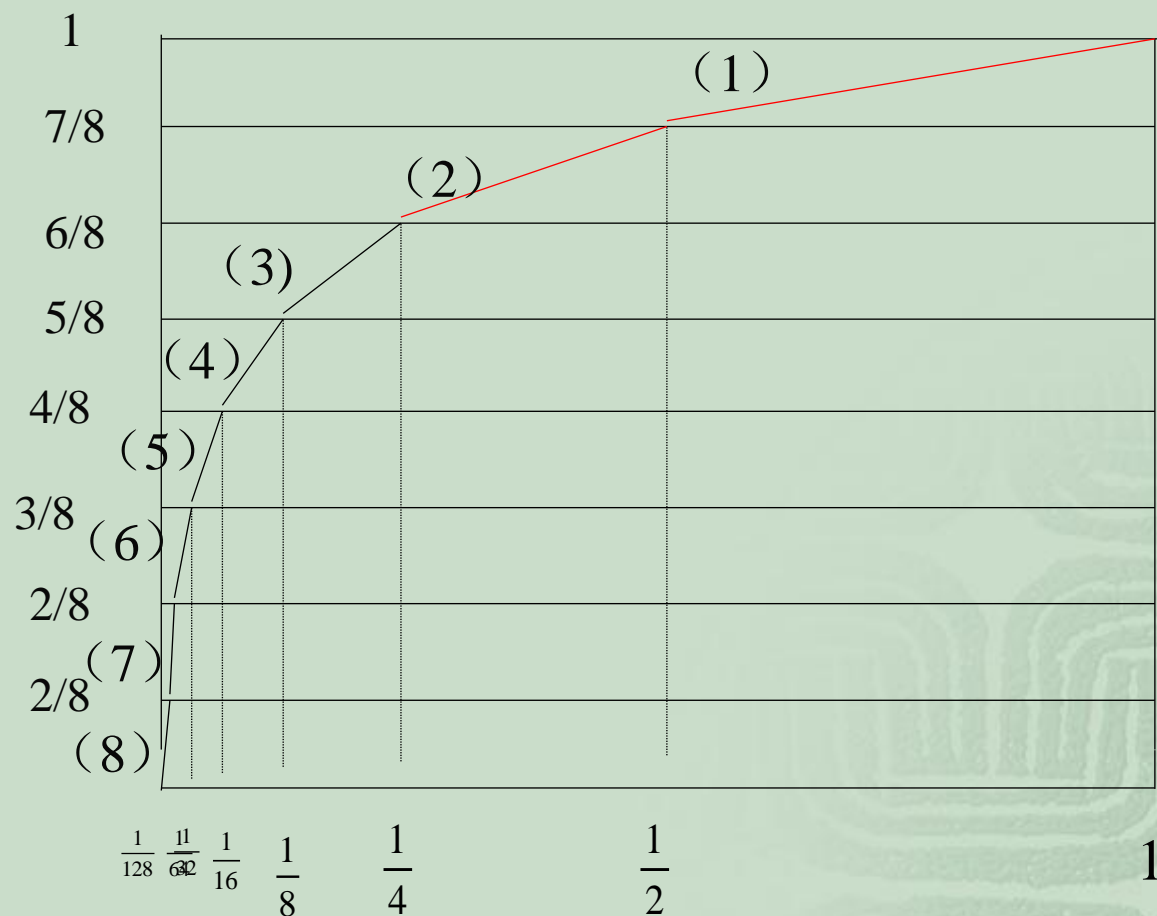


图5—11 13折线压扩特性(信号为正时的八段)

表 5—2

各种线段斜率表

折线段	1	2	3	4	5	6	7	8
斜 率	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4	8	16	16

图5—10只画出了信号在正半周时的情况，负半周时也一样。由于正半周的7、8两段和负半周的7、8两段斜率都一样，所以在整个特性中这四段连成一条直线。因此，总共有13条直线段，简称13折线。

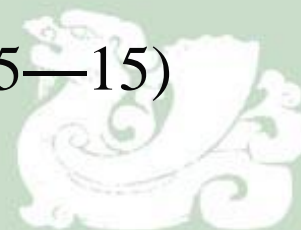


如果令 $x = \frac{u_{\lambda}}{V}$, $y = \frac{u_{\text{出}}}{V}$, 上述13折线可用

下式近似表示

$$y = \frac{Ax}{1 + \ln A} \quad 0 < x \leq \frac{1}{A} \quad (5-14)$$

$$y = \frac{1 + \ln Ax}{1 + \ln A} \quad \frac{1}{A} < x \leq 1 \quad (5-15)$$



式中 A 是一常数，不同的 A 值可决定一条不同的曲线。在 origin 处的斜率由下式表示

$$\frac{dy}{dx} = \frac{A}{1 + \ln A} \quad (5-16)$$



当原点处的斜率为16时，则

$$\frac{dy}{dx} = \frac{A}{1 + \ln A} = 16$$

可求得 $A=87.6$ 。图5—11所示的折线就是
 $A=87.6$ 的13折线压缩特性。



总的来说，如果采用第一种方案，当 $\mu = 100$ 时，在输入信号最小情况下信噪比可提高26dB，

当采用第二种方案时，在 $A=87.6$ 的情况，最小信号输入时的信噪比改善大约24dB。



5. 4 统计编码

高效编码的主要方法是尽可能去除信源中的冗余成份，从而以最少的数码率传递最大的信息量。**冗余度存在于**像素间的相关性及像素值出现概率的不均等性之中。

对于**有记忆性信源**来说首先要去除像素间的相关性，从而达到压缩数码率的目的。

对于**无记忆性信源**来说，像素间没有相关性，可以利用像素灰度值出现概率的不均等性，采用某种编码方法，也可以达到压缩数码率的目的。**这种根据像素灰度值出现概率的分布特性而进行的压缩编码叫统计编码。**

5.4.1 编码效率与冗余度

为了确定一个衡量编码方法优劣的准则，首先讨论一下编码效率与冗余度的问题。设某个无记忆信源共有 M 个消息，记作 $\{u_1, u_2, u_3, \dots, u_M\}$ 。其中消息各自出现的概率分别为 $\{p_1, p_2, p_3, \dots, p_M\}$ 。可把这个信源用下式表示

$$X = \left\{ \begin{array}{l} u_1, u_2, u_3, \dots, u_M \\ p_1, p_2, p_3, \dots, p_M \end{array} \right\} \quad (5-21)$$

根据该信源的消息集合，在字母集中选取符号进行编码 $A = \{a_1, a_2, a_3, \dots, a_n\}$ 。一般情况下取二元字母集 $A\{1, 0\}$ 。通常，这一离散信源中的各个消息出现的概率并不相等。根据信息论中熵的定义，可计算出该信源的熵(平均信息量)如下式

$$H(X) = -\sum_{i=1}^M p_i \log_a p_i \quad (5-22)$$

式中 $H(X)$ 代表熵， p_i 代表第*i*个消息出现的概率。

例如，设一离散信源如下

$$X = \left\{ \begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{array} \right\}$$

由式(5—22)可算出该信源的熵

$$\begin{aligned} H(X) &= -\sum_{i=1}^4 p_i \log_2 p_i \\ &= -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{8} \log_2 \frac{1}{8} - \frac{1}{8} \log_2 \frac{1}{8} \end{aligned}$$

$$= \frac{7}{4} \quad \text{比特/消息}$$



设对应于每个消息的码字由 N_i 个符号组成。也就是说每个消息所对应的码字长度各为 N_i 。那么，每个消息的平均码长可用下式表示

$$\bar{N} = \sum_{i=1}^M p_i N_i \quad (5-24)$$

式中 \bar{N} 代表平均码长， M 为信源中包含的消息的个数， P_i 为第 i 个消息出现的概率， N_i 为第 i 个消息对应的码长。

就平均而言，每个符号所含有的熵为

$$S = \frac{H(X)}{\bar{N}} \quad (5-25)$$



编码符号是在字母集 A 中选取的。如果编码后形成一个新的等概率的无记忆信源，字母数为 n ，那么，它的最大熵应为 $\log_a n$ 比特/符号，因此，这是极限值。

如果 $\frac{H(X)}{\bar{N}} = \log_a n$ ，则可认为编码效率已达到 100%，若 $\frac{H(X)}{\bar{N}} < \log_a n$ ，则可认为编码效率较低。



由上述概念，编码效率如下式表示：

$$\eta = \frac{H(X)}{\bar{N} \log_a n} \quad (5-26)$$

式中 η 代表编码效率， $H(X)$ 为信源的熵， \bar{N} 为平均码长， n 为字母集合中的字母数。



如果以比特 (bit) 作单位, \log_a 的底为2, 根据上述定义, 则 $\frac{H(X)}{\bar{N}} = \log_2 n$ $\eta = 100\%$

$$\frac{H(X)}{\bar{N}} < \log_2 n \quad \eta < 100\%$$

显然, 如果 $\eta \neq 100\%$, 就说明还有冗余度。

因此, 冗余度如下式表示

$$R_d = 1 - \eta = \frac{\bar{N} \log_2 n - H(X)}{\bar{N} \log_2 n} \quad (5-27)$$

统计编码要研究的问题就在于设法减小 \bar{N} ，使 η 尽量趋近于1， R_d 趋近于0。显然 \bar{N} 值有一个理论最低限，当 $\eta = 1$ 时， \bar{N} 的最低限就是 $H(X) / \log_2 n$ 。可以根据这一准则来衡量编码方法的优劣。下面举例加以说明。



例：一个信源 X 和一个字母集合 A 如下

$$X = \left\{ \begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} \end{array} \right\}$$

$$A = \{ 0, 1, 2, 3 \}$$



可求得信源 X 的熵

$$H(X) = \frac{7}{4} \text{ bit/消息}$$

平均码长

$$\bar{N} = 1 \times \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} \right) = 1$$



所以

$$\eta = \frac{\frac{7}{4}}{1 \times \log_2 4} = \frac{7}{8}$$

$$R_d = 1 - \eta = 1 - \frac{7}{8} = \frac{1}{8} \text{ (bit)}$$

显然，编码后还有 $\frac{1}{8}$ bit的冗余度， \overline{N} 没有达到
的最低限。



如果取

$A = \{0, 1\}, n = 2$ 那么可以编成如下等长码

$$u_1 = 00$$

$$u_3 = 10$$

$$u_2 = 01$$

$$u_4 = 11$$

此时

$$\bar{N} = 2 \times \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} \right) = 2$$



$$\eta = \frac{\frac{7}{4}}{2 \times \log_2 2} = \frac{7}{8}$$

$$R_d = 1 - \frac{7}{8} = \frac{1}{8}$$

同样有 $\frac{1}{8}$ 的冗余度。



上例中的两种编码方法，其特点是码字长度均相等，这种码叫等长码。显然此例中的两种等长码均没有达到最低限。怎样才能使信源编码达到最低限呢？



再看下例的编码方法选 $A = \{0,1\}$, $n=2$, 作为编码字符集。在这种编码中, 不用等长码, 而是采用下面的原则来编码, 即 P_i 大的消息编短码, P_i 小的消息编长码。



例:

$u_1 : 0$

$u_2 : 10$

$u_3 : 110$

$u_4 : 111$

可计算出平均码长

$$\bar{N} = 1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{8} + 3 \times \frac{1}{8} = \frac{7}{4}$$



其效率

$$\eta = \frac{\frac{7}{4}}{\frac{7}{4} \times \log_2 2} = 1$$

冗余度

$$R_d = 0$$



由此可见，这种编码法的码字平均长度达到了最低限。这说明用变长编码法可达到较高的效率。采用这种编码方法，信源中的消息与码字是一一对应的，因而译码时也是准确无误的。在编、译码过程中并不损失任何信息。它是一种信息保持编码法。



5.4.2 三种常用的统计编码法

变长编码是统计编码中最为主要的一种方法。变长编码的目标就是使平均码长达到低限，也就是使 \bar{N} 最优，但是，这种最优必须在一定的限制下进行。编码的基本限制就是码字要有单义性和非续长性。



单义代码是指任意一个有限长的码字序列只能被分割成一个一个的码字，而任何其他分割方法都会产生一些不属于码字集中的码字。符合这个条件的代码就叫单义代码。

非续长代码是指任意一个码字都不是其他码字的续长。换句话说，就是码字集合中的任意一个码字都不是由其中一个码字在后面添上一些码元构成的。

很容易看出**非续长代码一定是单义的**，但是，**单义代码却不一定是非续长的**。

表5—3

四种代码表

信 源	概 率	码 I	码 II	码 III	码 IV
u_1	$\frac{1}{2}$	0	0	0	0
u_2	$\frac{1}{4}$	0	1	1 0	0 1
u_3	$\frac{1}{8}$	1	0 0	1 1 0	0 1 1
u_4	$\frac{1}{8}$	1 0	1 1	1 1 1	0 1 1 1



例如，在表5—3中，列出四种代码，

码 I：显然码 I 缺乏单义可译性。

码 II：也缺乏单义可译性，又是可续长的。

码 III：既具备单义可译性又是非续长的码，它是
可用的。

码 IV 也具有单义可译性，但是却缺乏非续长性。

从上面的例子可知，使 \bar{N} 最短的码只是在单义可译性和非续长性的约束下才有意义。至于变长码的存在定理以及 \bar{N} 的最低限是否存在等问题，在信息论中都有详细的定理加以证明及讨论，在此不加赘述了。



最为常用的变长编码方法是霍夫曼
(*Huffman*) 码和仙农-费诺 (*Shannon-
Fano*) 码。下面详细地讨论一下这两种码的
构成方法



5.4.3 霍夫曼码

霍夫曼码变长编码法能得到一组最优的变长码。设原始信源有 M 个消息，即

$$X = \left\{ \begin{array}{cccccc} u_1 & u_2 & u_3 & \cdots & u_M \\ p_1 & p_2 & p_3 & \cdots & p_M \end{array} \right\} \quad (5-28)$$



可用下述步骤编出霍夫曼码：

第一步，把信源 X 中的消息按出现的概率从大到小的顺序排列，

即

$$p_1 \geq p_2 \geq p_3 \geq \cdots \geq p_M$$



第二步，把最后两个出现概率最小的消息合并成一个消息，从而使信源的消息数减少一个，并同时再次将信源中的消息的概率从大到小排列一次，得

$$X_1 = \left\{ \begin{array}{ccccccc} u_1' & u_2' & u_3' & \cdots & u_{M-1}' \\ p_1 & p_2 & p_3 & \cdots & p_{M-1} \end{array} \right\} \quad (5-29)$$



第三步，重复上述步骤，直到信源最后为 X^0 形式为止。这里 X^0 有如下形式

$$X^0 = \begin{Bmatrix} u_1^0 & u_2^0 \\ p_1^0 & p_2^0 \end{Bmatrix} \quad (5-30)$$



第四步，将被合并的消息分别赋以1和0或0和1。对最后 X^0 也对 u_1^0 和 u_2^0 对应地赋以1和0或0和1。

通过上述步骤就可以构成最优变长码（霍夫曼码）。下面举例说明具体构成方法。



例：求下述信源的霍夫曼码

$$X = \left\{ \begin{array}{cccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ 0.25 & 0.25 & 0.20 & 0.15 & 0.10 & 0.05 \end{array} \right\}$$

由上述步骤，合并最小的两项做一个新的信源

$$X' = \left\{ \begin{array}{ccccc} u'_1 & u'_2 & u'_3 & u'_4 & u'_5 \\ 0.25 & 0.25 & 0.20 & 0.15 & 0.15 \end{array} \right\}$$

这样可给 u_5 赋0， u_6 赋1，其中 $u'_5 = (u_5 + u_6)$ 。 X' 中消息的概率大小顺序正好符合从大到小的规律，故不必重排。



再做新的信源 $X'' = \begin{Bmatrix} u_1'' & u_2'' & u_3'' & u_4'' \\ 0.25 & 0.25 & 0.20 & 0.30 \end{Bmatrix}$

重排得 $X'' = \begin{Bmatrix} u_4'' & u_1'' & u_2'' & u_3'' \\ 0.30 & 0.25 & 0.25 & 0.20 \end{Bmatrix}$

将 u_4' 赋0, u_5' 赋1。将 u_2'' 和 u_3'' 合并构成新的

信源 $X''' = \begin{Bmatrix} u_1''' & u_2''' & u_3''' \\ 0.30 & 0.25 & 0.45 \end{Bmatrix}$

重排得 $X''' = \begin{Bmatrix} u_3''' & u_1''' & u_2''' \\ 0.45 & 0.30 & 0.25 \end{Bmatrix}$



将 u_2'' 赋0, u_3'' 赋1。最后得

$$X^0 = \begin{Bmatrix} u_1^0 & u_2^0 \\ 0.45 & 0.55 \end{Bmatrix}$$

重排得 $X^0 = \begin{Bmatrix} u_2^0 & u_1^0 \\ 0.55 & 0.45 \end{Bmatrix}$

u_1''' 赋0, u_2''' 赋1。最后 u_2^0 赋0, u_1^0 赋1。

编码结果可总结于表5—4中。编码图如图5—18所示。

如对合并的消息赋以1, 0值, 则会得到如表5—5所示的另外一组码。



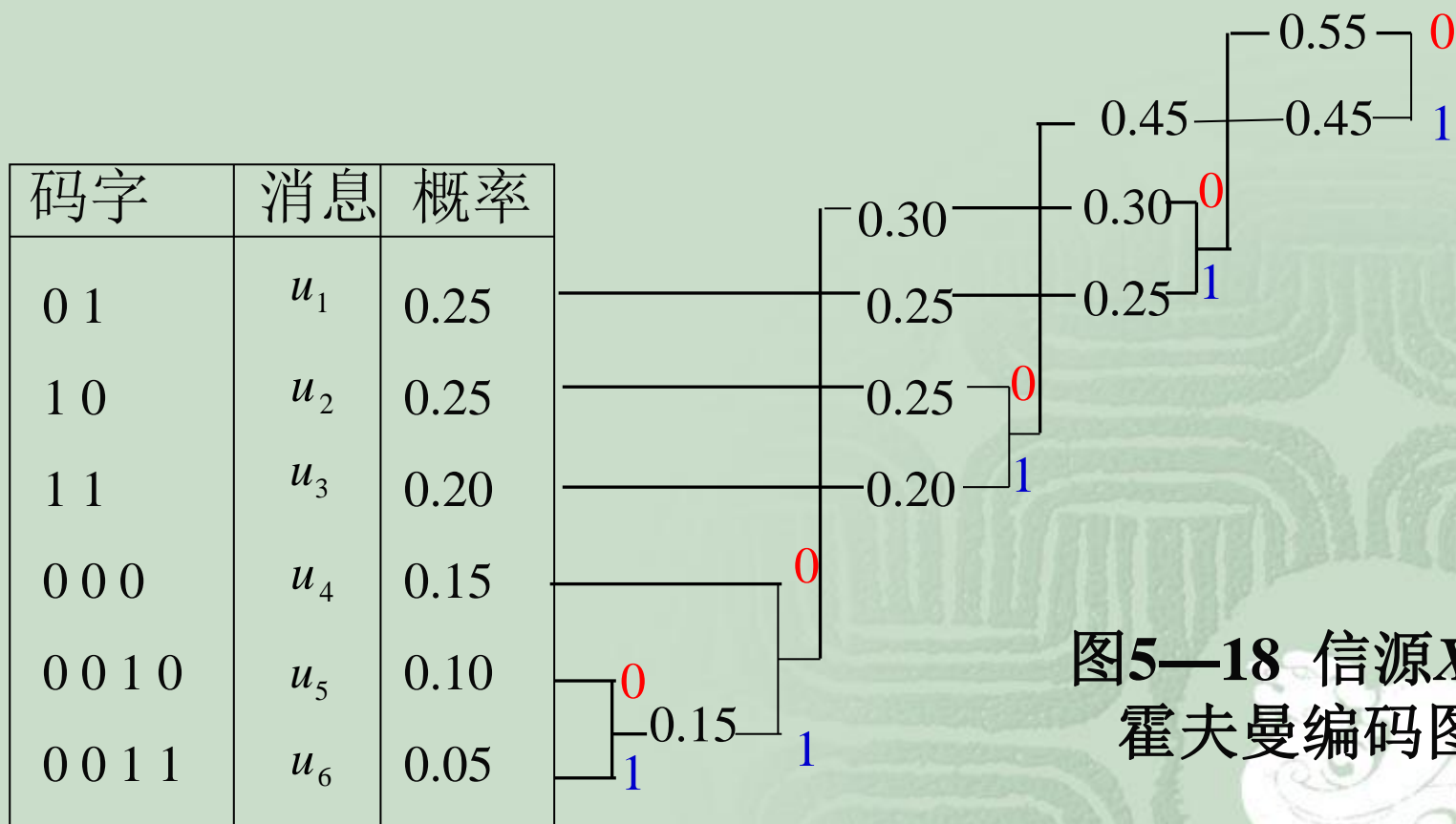


表5—4信源X的霍夫曼编码表

码 字	消 息	概 率
0 1	u_1	0.25
1 0	u_2	0.25
1 1	u_3	0.20
0 0 0	u_4	0.15
0 0 1 0	u_5	0.10
0 0 1 1	u_6	0.05

表5—5 信源X的另一组霍夫曼编码表

码 字	消 息	概 率
1 0	u_1	0.25
0 1	u_2	0.25
0 0	u_3	0.20
1 1 1	u_4	0.15
1 1 0 1	u_5	0.10
1 1 0 0	u_6	0.05

下面计算一下信源的熵，平均码长，效率及冗余度。

$$\begin{aligned} H(X) &= -\sum_{i=1}^6 p_i \log_2 p_i \\ &= -0.25 \log_2 0.25 - 0.25 \log_2 0.25 - 0.20 \log_2 0.20 \\ &\quad - 0.15 \log_2 0.15 - 0.10 \log_2 0.10 - 0.05 \log_2 0.05 = 2.42 \end{aligned}$$

$$\bar{N} = 2 \times 0.25 + 2 \times 0.25 + 2 \times 0.20 + 3 \times 0.15 + 4 \times 0.10 + 4 \times 0.05 = 2.45$$

$$\eta = \frac{H(X)}{\bar{N} \log_2 n} = \frac{2.42}{2.45 \log_2 2} = 0.98 = 98\%$$

$$R_d = 1 - \eta = 1 - 98\% = 2\%$$

所以，对于信源 X 的霍夫曼码的编码效率为98%，
尚有2%的冗余度。

5.4.4 仙农—费诺码

另外一种常用的变长编码是仙农—费诺码。这种码有时也可以得到最优编码性能。它的编码准则要符合非续长条件，在码字中1和0是独立的，而且是（或差不多是）等概率的。

这样的准则一方面可保证无需间隔来区分码字，同时又保证每传输 1 位码就有 1 bit 的信息量。



仙农—费诺码的编码程序可由下述几个步骤来完成：

第一步：设信源 X 有非递增的概率分布

$$X = \begin{Bmatrix} u_1 & u_2 & u_3 & \cdots & u_M \\ p_1 & p_2 & p_3 & \cdots & p_M \end{Bmatrix} \quad (5-31)$$

其中 $p_1 \geq p_2 \geq p_3 \geq \cdots \geq p_M$ 。把 X 分成两个子集合，得

$$X_1 = \begin{Bmatrix} u_1 & u_2 & u_3 & \cdots & u_k \\ p_1 & p_2 & p_3 & \cdots & p_k \end{Bmatrix} \quad (5-32)$$

$$X_2 = \begin{Bmatrix} u_{k+1} & u_{k+2} & u_{k+3} & \cdots & u_M \\ p_{k+1} & p_{k+2} & p_{k+3} & \cdots & p_M \end{Bmatrix} \quad (5-33)$$

并且保证

$$\sum_{i=1}^k p_i \approx \sum_{i=k+1}^M p_i \quad (5-34)$$

成立或差不多成立。

第二步：给两个子集中的消息赋值， X_1 赋1，

X_2 赋 0，或给 X_1 赋 0， X_2 赋 1。



第三步：重复第一步骤，将两个子集 X_1 , X_2 再细分为2个子集，并且也同样使两个小子集里消息的概率之和相等或近似相等。

然后，重复第二步骤赋值。以这样的步骤重复下去，直到每个子集内只包含一个消息为止。对每个消息所赋过的值依次排列出来就可以构成仙农—费诺码



下面举例说明仙农—费诺码的具体构成方法

例：设有信源

$$X = \left\{ \begin{array}{cccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{8} & \frac{1}{8} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \end{array} \right\}$$

其编码流程图如图5—19所示。编码表如表5—6所示。

如果对各子集赋以另外一种值，即1，0，那么，同样会得到另一种编码结果，其编码表如表5—7所示。

码字	消息	概率				
00	u_1	$\frac{1}{4}$	0	0		
01	u_2	$\frac{1}{4}$		1		
100	u_3	$\frac{1}{8}$	1	0	0	
101	u_4	$\frac{1}{8}$			1	
1100	u_5	$\frac{1}{16}$		1	0	0
1101	u_6	$\frac{1}{16}$				1
1110	u_7	$\frac{1}{16}$			1	0
1111	u_8	$\frac{1}{16}$				1

图5—19 仙农-费诺码编码流程图



表5—7 仙农-费诺码编码表 表5—8 另一种仙农-费诺码编码表

消 息	概 率	码 字		消 息	概 率	码 字
u_1	$\frac{1}{4}$	0 0		u_1	$\frac{1}{4}$	1 1
u_2	$\frac{1}{4}$	0 1		u_2	$\frac{1}{4}$	1 0
u_3	$\frac{1}{8}$	1 0 0		u_3	$\frac{1}{8}$	0 1 1
u_4	$\frac{1}{8}$	1 0 1		u_4	$\frac{1}{8}$	0 1 0
u_5	$\frac{1}{16}$	1 1 0 0		u_5	$\frac{1}{16}$	0 0 1 1
u_6	$\frac{1}{16}$	1 1 0 1		u_6	$\frac{1}{16}$	0 0 1 0
u_7	$\frac{1}{16}$	1 1 1 0		u_7	$\frac{1}{16}$	0 0 0 1
u_8	$\frac{1}{16}$	1 1 1 1		u_8	$\frac{1}{16}$	0 0 0 0

下面计算一下仙农—费诺码的平均码长，效率及冗余度。信

源的熵为：

$$\begin{aligned} H(X) &= -\frac{1}{4}\log_2 \frac{1}{4} - \frac{1}{4}\log_2 \frac{1}{4} - \frac{1}{8}\log_2 \frac{1}{8} - \frac{1}{8}\log_2 \frac{1}{8} \\ &\quad - \frac{1}{16}\log_2 \frac{1}{16} - \frac{1}{16}\log_2 \frac{1}{16} - \frac{1}{16}\log_2 \frac{1}{16} - \frac{1}{16}\log_2 \frac{1}{16} \\ &= 2\frac{3}{4} \quad \text{比特/消息} \end{aligned}$$

平均码长

$$\begin{aligned} \bar{N} &= \sum_{i=1}^8 p_i N_i \\ &= \frac{1}{4} \times 2 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 + \frac{1}{16} \times 4 \\ &\quad + \frac{1}{16} \times 4 + \frac{1}{16} \times 4 + \frac{1}{16} \times 4 = 2\frac{3}{4} \end{aligned}$$

显然， $\eta=1, R_d=0$ 效率已达到100%。这一结果

并不奇怪，对于仙农—费诺码来说，如果满足下式

$$p(u_i) = 2^{-N_i} \quad (5—35)$$

且

$$\sum_{i=1}^M 2^{-N_i} = 1 \quad (5—36)$$

就会使编码效率达到100%。式中的 $p(u_i)$ 为消息 u_i

出现的概率， N_i 是码字的长度。如果不满足上述条件

就不会有100%的效率，如下例中图5—20所示。

例：设有一信源

$$X = \left\{ \begin{array}{ccccccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 & u_9 \\ 0.49 & 0.14 & 0.14 & 0.07 & 0.07 & 0.04 & 0.02 & 0.02 & 0.01 \end{array} \right\}$$

编码流程及形成的码字如图5—20所示。可以计算

出信源的熵为 $H(X) = 2.313$ 平均码长 $\bar{N} = 2.33$

效率 $\eta = 0.993$ 冗余度 $R_d = 0.007$

由此例可见，由于信源不满足式(5—35)和(5—36)的条件，编码效率不能达到100%。然而从结果上看，它仍然是一种相当好的编码。

编码	消息	概率								
0	u_1	0.49	0							
100	u_2	0.14	1	0	0					
101	u_3	0.14			1					
1100	u_4	0.07		1	0	0				
1101	u_5	0.07				1				
1110	u_6	0.04			1	0				
11110	u_7	0.02				1	0			
111110	u_8	0.02				1	1	0		
111111	u_9	0.01					1			

图5—20 编码流程图

特点:

- 1) 、 Huffman码和Shannon - Fano码不是唯一的;
- 2) 、 Huffman码和Shannon - Fano码缺乏构造性, 即, 不能用数学方法建立一一对应关系, 只能通过查表的方法构成对应关系。如果消息数目很大, 所需的存储器就大, 设备就复杂。
- 3) 、 非等长码在传输、译码、存储都不方便。

