尽管行程长度编码本身是一种压缩图像的有效方法(见 8.1.2 节中的例子),但其他的压缩方法通常可以通过对行程长度本身进行变长编码实现。实际上,黑色和白色的行程长度可以使用变长编码分别进行编码,这些变长编码是根据它们自身的统计数据特殊制定的。例如,符号 a_j 表示一个长度为 j 的黑色行程,可以估计出一个假想的黑色行程长度信源产生的符号 a_j 的概率,而符号 a_j 产生的概率是用整幅图像中长度为 j 的黑色行程的次数除黑色行程的总次数得到的。这种黑色行程长度信源熵的估计表示为 H_0,紧接着把这些概率代入式(8.3.3)。对于白色行程的熵有相似的结果,它表示为 H_1。图像的近似行程长度的熵为:

$$H_{RL} = \frac{H_0 + H_1}{L_0 + L_1} \tag{8.4.4}$$

这里,变量 L_0 和 L_1 分别表示黑色和白色行程长度的平均值。式(8.4.4)提供了对二值图像的行程长度使用变长编码时所要求的每像素平均比特数的估计。

二维行程长度编码

一维行程长度编码概念很容易扩展得到各种二维编码过程。其中较为熟知的一种方法是相对地址编码(RAC)。这种方法是以记录每次黑色和白色行程开始和结束的二值转换的原理为基础的。图 8.17(a)说明了这种方法的实现。注意,ec 是从当前转换 c 到当前线 e 的最后转换的距离,而 cc' 是在前面提到的线上从 c 到首次经过 e 的相似(距离相同)转换(表示为 c')的距离。如果 $ec \leqslant cc'$,则 RAC 编码的距离 d 等于 ec,并且用它表示在 c 处的当前转换。但是如果 $cc' < ec$,则令 d 等于 cc'。

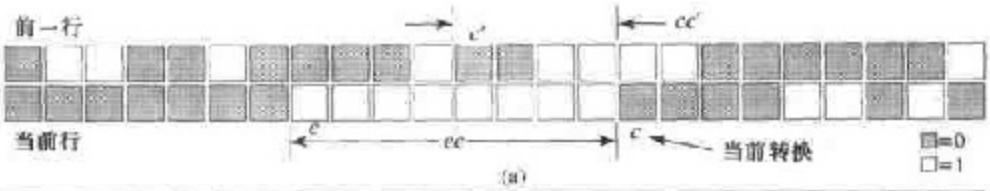
类似行程长度编码,相对地址编码要求按惯例判定行程值。另外,在每条线的起点和终点处假想的转换,以及一条假想的开始线(如,一条全白色的线)必须要考虑,以便能够适当处理图像的边界。最后,由于多数图像的 RAC 距离的概率分布实际上是不均匀的(见 8.1.1 节),因此,RAC 处理的最后步骤是通过使用适当的变长编码对选定的 RAC 距离(即最短距离)和距离 d 进行编码。如图 8.17(b)所示,可以使用一种近似于 B_1 编码的编码。最短距离分配给最短码字,而其他距离是这样编码的:使用一种前缀表示最短的 RAC 距离,使用第二前缀指定 d 在一个特定的距离范围内取值,用 d 的二进制表示[图 8.17(b)中表示为 $\times\times\times\cdots\times$]减去取值范围本身的基数距离。如果 ec 和 cc' 如图 8.17(a)所示为 +8 和 +4,正确的 RAC 码字应为 1100011。最后,如果 $d=0$,c 直接在 c' 下,而如果 $d=1$,则解码器可能必须判定最接近的转换点,因为编码 100 没有指明测量是相对于当前行还是相对于前一行进行的。

轮廓线追踪和编码

相对地址编码是一种表示二值图像中组成轮廓线的亮度转换的方法。另一种方法是通过一系列的边界点或通过一个单一的边界点和一个方向集合表示每条轮廓线。后一种方法有时称为直接轮廓跟踪法。在这一节中,将同时讲解另一种称为预测微分量化(PDQ)的方法。这种方法证明了上述两种方法的本质特点。它是一种面向扫描线的轮廓跟踪过程。

在预测微分量化中,对图像每个对象的前部和后部的轮廓线(图 8.18)同时进行跟踪以生成一个序列对(Δ',Δ'')。Δ'项是邻近线上前部轮廓的起始坐标之间的差。Δ''是从前到后的轮廓线长度的差。这些差同表示新轮廓线的开始(新的开始信息)和旧轮廓线的终结(聚合信息)

的特定信息一起表示每个对象。如果 Δ″用相邻线的后轮廓坐标之间的差(表示为 Δ‴)替代，则这种技术称为双增量编码(DDC)。



(a)

测量的距离	距离	编码	距离范围	编码 $h(d)$
cc'	0	0	1 - 4	0 ××
ec or ec' (left)	1	100	5 - 20	10 ××××
cc' (right)	1	101	21 - 84	110 ××××××
ee	$d(d>1)$	111 $h(d)$	85 - 340	1110 ××××××××
ee' (e' to left)	$d(d>1)$	1100 $h(d)$	341 - 364	11110 ××××××××××
ee' (e' to right)	$d(d>1)$	1101 $h(d)$	1365 - 5460	111110 ××××××××××××

(b)

图 8.17　一种相对地址编码(RAC)的说明

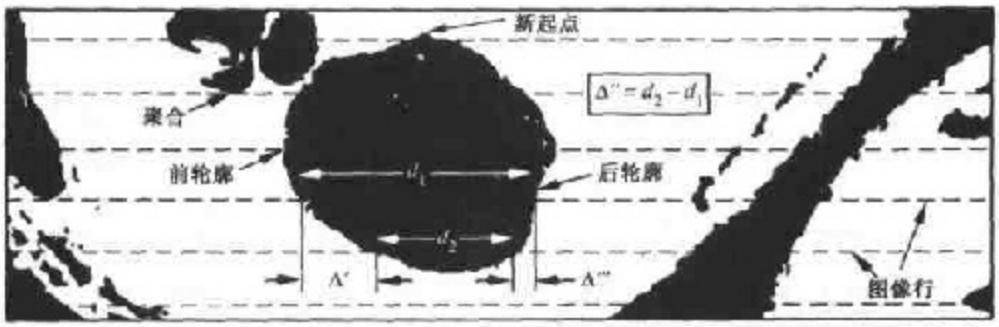


图 8.18　PDQ 算法的参数

新的开始和聚合信息允许在扫描线基础上产生(Δ′,Δ″)或(Δ′,Δ‴)，适当地连接到前一行和下一行中相对应的偶对。没有这些信息，解码器无法对另一个偶对参考一个差分偶对，或在图像中正确地放置轮廓线。为了避免对每个新的起点和聚合信息的行列坐标同时进行编码，通常使用惟一的编码识别不包含物体像素的扫描线。PDQ 和 DDC 编码过程的最后一步是用适当的变长编码表示 Δ′,Δ″或 Δ‴和新的起点与聚合的坐标。

例 8.14　二值压缩技术的比较

通过对前述二值压缩技术进行对比来对这一节进行总结。每种方法都用于压缩图 8.14 中的图像。得到的编码率和压缩率在表 8.8 和表 8.9 中列出。解释这些结果的时候，注意关于 RLC 行程长度的熵的一阶估计(见 8.3.4 节)及 PDQ 和 DDC 距离的计算和使用，是作为在用 8.4.1 节中的变长编码方法时所能达到的压缩性能的一种近似。

表 8.8 和表 8.9 中的结果说明所有的技术均能消除一部分像素间冗余。即，得到的编码率小于每幅图像的一阶熵的估计。行程编码被证明是最好的位平面编码图像的编码方法，而二维技术(比如 PDQ，DDC 和 RAC)在压缩二值图像时表现出更好的性能。再有，图 8.14(a)中图像的灰度编码相对简单的过程改进的编码性能在 1 比特/像素左右。最

后,注意这5种压缩方法都可以使用一个1到2之间的因子对单色图像进行压缩,而对图8.14(b)的二值图像的压缩使用2到5之间的一个因子。如表8.8所示,产生这种性能差异的原因是算法无法对位平面编码图像中的较低阶的位平面进行压缩。实际上,表中画横线的项表明算法引起了数据量膨胀。在这些情况下,用原始数据表示位平面,总共的编码率只增加了1比特/像素。

表8.8 图8.14(a)无误差位平面编码结果:$H \approx 6.82$比特/像素

方法	位平面编码率(比特/像素)								编码率	压缩率
	7	6	5	4	3	2	1	0		
二值位平面编码										
CBC(4×4)	0.14	0.24	0.60	0.79	0.99	—	—	—	5.75	1.4:1
RLC	0.09	0.19	0.51	0.68	0.87	1.00	1.00	1.00	5.33	1.5:1
PDQ	0.07	0.18	0.79	—	—	—	—	—	6.04	1.3:1
DDC	0.07	0.18	0.79	—	—	—	—	—	6.03	1.3:1
RAC	0.06	0.15	0.62	0.91	—	—	—	—	5.17	1.4:1
灰度位平面编码										
CBC(4×4)	0.14	0.18	0.48	0.40	0.61	0.98	—	—	4.80	1.7:1
RLC	0.09	0.13	0.40	0.33	0.51	0.85	1.00	1.00	4.29	1.9:1
PDQ	0.07	0.12	0.61	0.40	0.82	—	—	—	5.02	1.6:1
DDC	0.07	0.11	0.61	0.40	0.81	—	—	—	5.00	1.6:1
RAC	0.06	0.10	0.49	0.31	0.62	—	—	—	4.05	1.8:1

表8.9 图8.14(b)的无误差二值图像压缩结果:$H \approx 0.55$比特/像素

	WBS(1×8)	WBS(4×4)	RLC	PDQ	DDC	RAC
编码率(比特/像素)	0.48	0.39	0.32	0.23	0.22	0.23
压缩率	2.1:1	2.6:1	3.1:1	4.4:1	4.7:1	4.4:1

8.4.4 无损预测编码

现在转向无误差压缩方法,这种方法不需要将图像分解为一个位平面的集合。这种方法通常称为无损预测编码,它是基于通过对每个像素新增的信息进行提取和编码,来消除在空间上较为接近像素之间的冗余信息。一个像素的新增信息被定义为此像素实际值和预测值之间的差异。

图8.19显示了一个无损预测编码系统的基本组成部分。这个系统由一个编码器和一个解码器组成,每部分都包含一个相同的预测器。由于输入图像的每个表示为f_n的连续像素都要送入编码器,预测器根据以往的一些输入生成输入像素的预期值。预测器的输出被四舍五入为最接近的整数$\hat{f}_n$,并将这个整数用于计算差异或预测误差:

$$e_n = f_n - \hat{f}_n \tag{8.4.5}$$

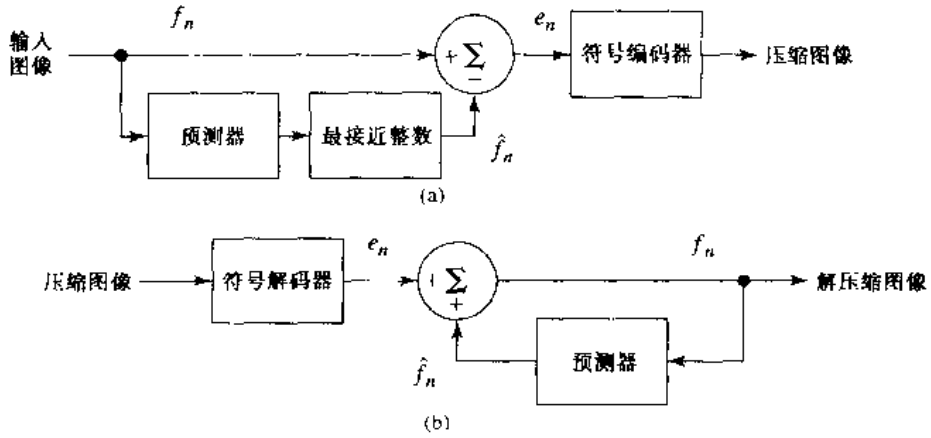


图 8.19　一个无损预测编码模型。(a)编码器,(b)解码器

这个误差使用变长代码(通过符号编码器)进行编码以生成压缩数据流的下一个元素。图 8.19(b)中的解码器根据接收到的变长码字对 e_n 进行重构并实现反运算:

$$f_n = e_n + \hat{f}_n \tag{8.4.6}$$

有多种局部的、全局的和自适应(见 8.5.1 节)的方法可以用于生成 $\hat{f}_n$。然而,在大多数情况下,预测是根据前 m 个像素的线性组合生成的。即,

$$\hat{f}_n = \text{round}\left[\sum_{i=1}^{m} \alpha_i f_{n-i}\right] \tag{8.4.7}$$

这里,m 是线性预测器的阶,round 是表示四舍五入或取最接近的整数的运算的函数,α_i($i = 1, 2, \cdots, m$)是预测系数。在光栅扫描应用中,下标 n 根据它们的发生时间指示预测器输出。即,式(8.4.5)到式(8.4.7)中的和可以用更明确的符号和替代,这里 t 表示时间。在其他情况下,n 用做图像的空间坐标和/或帧数(某一时间内的图像序列)的下标。例如,在一维线性预测编码中,式(8.4.7)可以写为:

$$\hat{f}_n(x,y) = \text{round}\left[\sum_{i=1}^{m} \alpha f(x, y-i)\right] \tag{8.4.8}$$

这里每个下标变量都被直接表示为一个空间坐标 x 和 y 的函数。注意,式(8.4.8)表明一维线性预测 $\hat{f}(x,y)$仅是一个当前行上前几个像素的函数。在二维预测编码中,预测是一个在从左到右、从上到下扫描图像过程中前几个像素的函数。在三维情况下,预测是以这些像素和前面几帧的像素为基础的。式(8.4.8)不能对每一行的前 m 个像素进行求解,因此这些像素必须用其他的方法进行编码(比如,霍夫曼编码)并被视为预测编码处理的额外开销。相似的方法可用于更高维的情况。

例 8.15　预测编码

考虑使用简单的一阶线性预测器对图 8.14(a)的单色图像进行编码:

$$\hat{f}_n(x,y) = \text{round}[\alpha f(x, y-1)] \tag{8.4.9}$$

这种一般形式的预测器称为前像素预测器,相应的预测编码过程称为差分编码或前像素编码。图 8.20(a)显示了根据式(8.4.9)令 $\alpha = 1$ 时得到的预测误差图像。在这幅图像中,灰度级 128 表示一个零的预测误差,而所有非零的正的和负的预测误差(在估算范围内的

和超出估算范围的)都与 8 相乘并分别被显示成更亮的和更暗的灰度区。预测图像的均值为 128.02,这个值对应的平均预测误差仅为 0.02 比特。

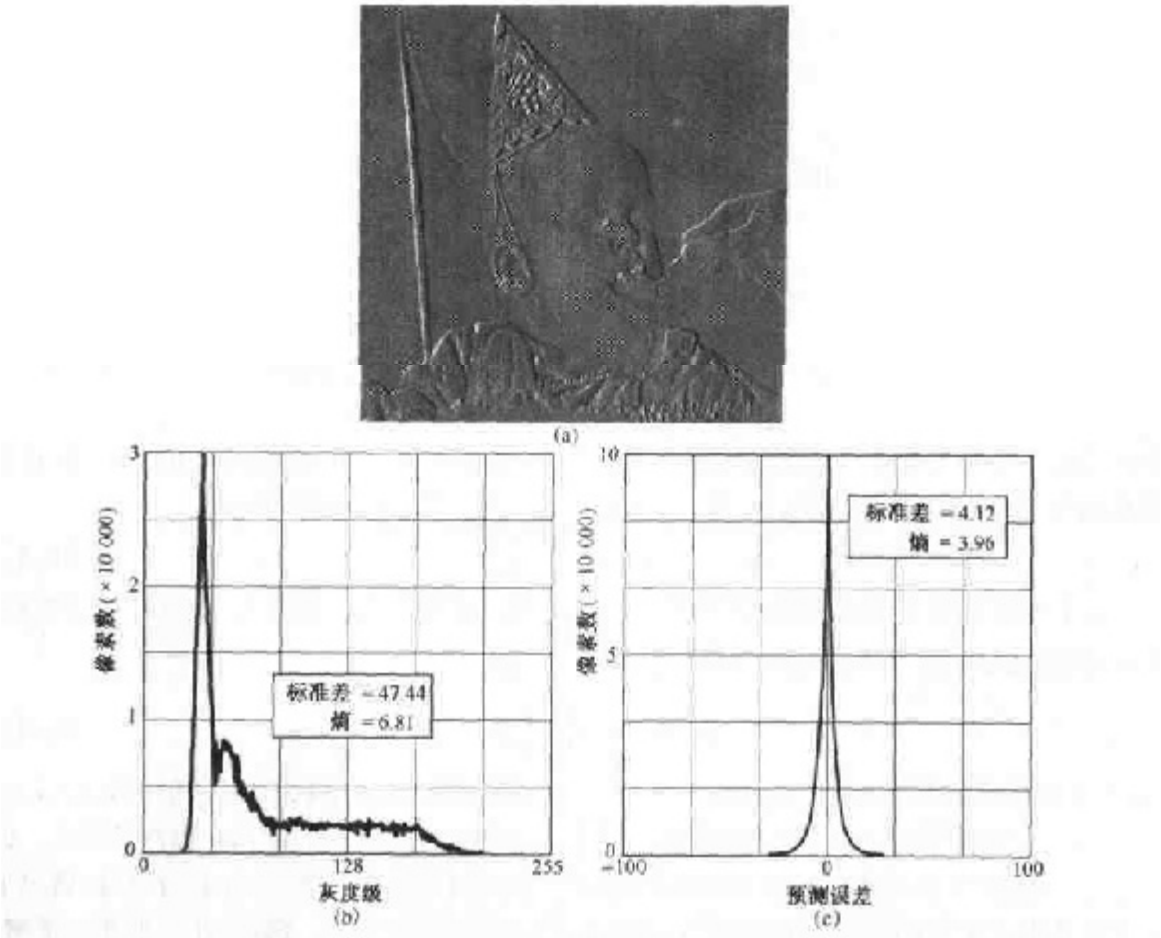


图 8.20 (a)根据式(8.4.9)得到的预测误差图像,(b)原图像的灰度级直方图,(c)预测误差的直方图

图 8.20(b)和(c)显示了图 8.14(a)中图像的灰度直方图和根据式(8.4.9)得到的预测误差图像的直方图。注意,图 8.20(c)中的预测误差的变化比原图像中灰度级的变化小得多。另外,预测误差的熵的一阶估计明显小于对应的原图像的一阶估计(3.96 比特/像素对 6.81比特/像素)。这个熵的减少反映了通过预测编码处理消除了大量的冗余,尽管事实上对 m 比特的图像需要$(m+1)$比特数据准确表示根据式(8.4.5)得到的误差序列。虽然 8.4.1 节的任何变长编码过程都可以用于对这个误差序列进行编码,但得到的压缩结果被限制在大约 8/3.96 或约 2:1 的水平。通常,任何无损预测编码方法的最大压缩估计,可以通过将用于表示原始图像中每一个像素的平均比特数除以预测误差数据的熵的一阶估计而得到。

前面的例子重点强调了,用无损预测编码实现的压缩量与将输入图像映射到预测误差序列得到的熵的减少直接相关。

因为通过预测和差分处理,消除了大量的像素间冗余,所以,预测误差的概率密度函数通常在零处有一个很高的峰,并表现出变化相对较小(同输入灰度级分布相比较而言)的特征。

实际上，预测误差的密度函数通常使用零均值不相关的拉普拉斯概率密度函数进行模型化：

$$p_e(e) = \frac{1}{\sqrt{2}\sigma_e} e^{\frac{-\sqrt{2}|e|}{\sigma_e}} \tag{8.4.10}$$

这里 σ_e 是 e 的标准差。

8.5　有损压缩

与前一节讲述的无误差方法不同，有损编码是以在图像重构的准确度上做出让步而换取压缩能力增加的概念为基础的。如果产生的失真(可能是明显可见的也可能不很明显)是可以容忍的，则压缩能力上的增加就是有效的。实际上，许多种有损编码技术有能力根据压缩比率超过 100:1 的数据重构实际上不可区分的单色图像，并且生成的图像与对原图进行 10:1 到 50:1压缩的图像之间没有本质上的区别。单色图像的无差错编码很少能在数据压缩上得到大于3:1的结果。正如 8.2 节中指明的，这两种方法之间主要的差别在于，是否存在图 8.6 中的量化器模块。

8.5.1　有损预测编码

这一节，在 8.4.4 节介绍的模型中增加一个数字量化器，并分析在重构准确度和压缩性能之间得到的折中。如图 8.21 所显示的，量化器充当无误差编码器的取整函数，它被插入在符号编码器和形成预测误差的那一点之间。

这个量化器将预测误差映射成有限范围内的输出，表示为 $\dot{e}_n$，这个输出确定了与有损预测编码相联系的压缩和失真的量。

为了适应量化步骤的加入，必须更改图 8.19(a)中的无误差编码器，以便由编码器和解码器生成的预测相等。如图 8.21(a)所示，这是通过在反馈环中设置一个有损编码器的预测器来实现的，这里，表示为 f_n 的反馈环的输入是过去预测函数和对应的量化误差产生的。即，

$$\dot{f}_n = \dot{e}_n + \hat{f}_n \tag{8.5.1}$$

这里，$\hat{f}_n$ 同 8.4.4 节中所定义的一样。这个闭合的循环结构可以防止在解码器的输出部位形成误差。注意，从图 8.21(b)可以看出，解码器的输出也是由式(8.5.1)给出的。

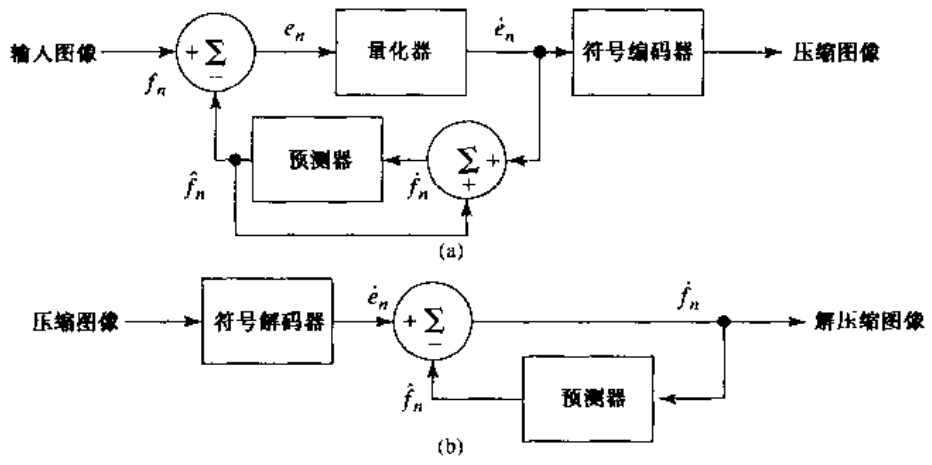


图 8.21　有损预测编码模型。(a)编码器和(b)解码器

例 8.16 德尔塔调制

德尔塔调制(DM)是一种简单但众所周知的有损预测编码形式,其中,预测器和量化器定义为:

$$\hat{f}_n = \alpha \dot{f}_{n-1} \tag{8.5.2}$$

和

$$\dot{e}_n = \begin{cases} +\zeta & e_n > 0 \\ -\zeta & \text{其他} \end{cases} \tag{8.5.3}$$

这里,α 是一个预测系数(通常小于 1),而 ζ 是一个正的常量。量化器的输出 $\dot{e}_n$ 可以用 1 比特表示[图 8.22(a)],因此,图 8.21(a)的符号编码器可以使用 1 比特的固定长度编码。得到的 DM 码率是 1 比特/像素。

图 8.22(c)说明了调制处理的机理,这里的计算要求压缩和重构被列入表格的输入序列{14,15,14,15,13,15,15,14,20,26,27,28,27,27,29,37,47,62,75,77,78,79,80,81,81,82,82},$\alpha = 1$,$\zeta = 6.5$。这个处理过程从第一个输入像素无误差地传递到解码器开始。由于在编码器和解码器都设置了初始条件 $\dot{f}_0 = f_0 = 14$,余下的输出可以通过反复计算式(8.5.2)、式(8.4.5)、式(8.5.3)和式(8.5.1)得到。这样,例如,当 $n = 1$ 时,$\hat{f} = (1)(14) = 14$,$e_1 = 15 - 14 = 1$,$\dot{e}_1 = +6.5$(因为 $e_1 > 0$),$\dot{f}_1 = 6.5 + 14 = 20.5$,得到的重构误差是(15 - 20.5)或 -5.5灰度级。

图 8.22(b)显示了图 8.22(c)中显示的图表化的表格数据。其中显示了输入和完整的解码输出(f_n 和 $\dot{f}_n$)。注意,在从 $n = 14$ 到 19 这一段快速变化的区域中,ζ 太小不足以表示输入的最大变化,从而产生了被称为斜率过载的失真。另外,当 ζ 太大无法表示输入的最小变化的时候,如在从 $n = 0$ 到 $n = 7$ 的相对平滑的区域中,会出现颗粒噪声。在大多数图像中,这两种现象会导致图像的边缘变得模糊,呈现粒状或有噪声的表面(即,失真的平滑区域)。

在前例中提到的失真对所有有损预测编码方法都是很普遍的。这些失真的严重性取决于所使用的量化和预测方法之间的一个相互作用的复杂集合。不考虑它们相互间的作用,预测器通常是在无量化误差的假设之下设计的,且量化器被设计成其自身的误差最小。就是说,预测器和量化器是彼此独立进行设计的。

最佳预测器

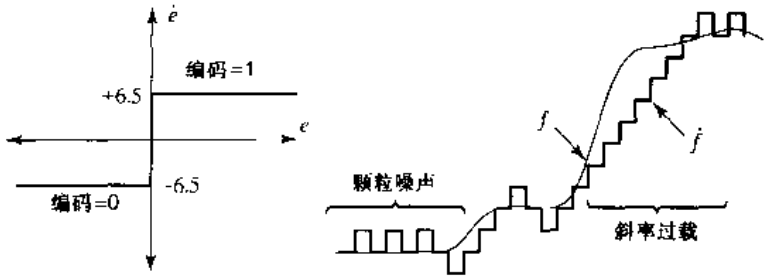
在大多数预测编码应用中的最佳预测器可将编码器的均方预测误差降至最小[①]:

$$E\{e_n^2\} = E\{[f_n - \hat{f}_n]^2\} \tag{8.5.4}$$

限制条件为:

$$\dot{f}_n = \dot{e}_n + \hat{f}_n \approx e_n + \hat{f}_n = f_n \tag{8.5.5}$$

① 符号 $E\{\cdot\}$ 代表期望算子。



(a)　　(b)

输入		编码器				解码器		误差
n	f	$\hat{f}$	e	$\dot{e}$	$\dot{f}$	$\hat{f}$	$\dot{f}$	$[f-\hat{f}]$
0	14	—	—	—	14.0	—	14.0	0.0
1	15	14.0	1.0	6.5	20.5	14.0	20.5	-5.5
2	14	20.5	-6.5	-6.5	14.0	20.5	14.0	0.0
3	15	14.0	1.0	6.5	20.5	14.0	20.5	-5.5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
14	29	20.5	8.5	6.5	27.0	20.5	27.0	2.0
15	37	27.0	10.0	6.5	33.5	27.0	33.5	3.5
16	47	33.5	13.5	6.5	40.0	33.5	40.0	7.0
17	62	40.0	22.0	6.5	46.5	40.0	46.5	15.5
18	75	46.5	28.5	6.5	53.0	46.5	53.0	22.0
19	77	53.0	24.0	6.5	59.6	53.0	59.6	17.5
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

(c)

图 8.22　德尔塔调制的一个例子

和

$$\hat{f}_n = \sum_{i=1}^{m} \alpha_i f_{n-i} \tag{8.5.6}$$

即，选择最小化均方预测误差为最佳的准则。量化误差假设是可以忽略的（$\dot{e}_n \approx e_n$），且预测被限定为前 m 个像素的线性组合[①]。这些限制并不是基本的，但它们在相当大程度上简化了分析，并且同时减小了预测器计算的复杂性。如此得到的预测编码方法被称为差分脉冲编码调制（DPCM）。

在这些条件下，最佳预测器的设计问题变成相对简单的选择 m 个预测系数的应用，这些系数使下式最小：

$$E\{e_n^2\} = E\left\{\left[f_n - \sum_{i=1}^{m} \alpha_i f_{n-i}\right]^2\right\} \tag{8.5.7}$$

相对于每个系数对式(8.5.7)求导，令导数为零，在假设 f_n 具有零均值和方差为 σ^2 的条件下解出联立方程的解集，得到：

$$\boldsymbol{\alpha} = \mathbf{R}^{-1}\mathbf{r} \tag{8.5.8}$$

这里 $\mathbf{R}^{-1}$ 是下列 $m \times m$ 自相关矩阵的逆矩阵：

① 一般来讲，一个非高斯场的图像的最佳预测器是用于估计像素的一个非线性函数。

$$\mathbf{R} = \begin{bmatrix} E\{f_{n-1}f_{n-1}\} & E\{f_{n-1}f_{n-2}\} & \cdots & E\{f_{n-1}f_{n-m}\} \\ E\{f_{n-2}f_{n-1}\} & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & & & \\ E\{f_{n-m}f_{n-1}\} & E\{f_{n-m}f_{n-2}\} & \cdots & E\{f_{n-m}f_{n-m}\} \end{bmatrix} \tag{8.5.9}$$

$\mathbf{r}$ 和 $\boldsymbol{\alpha}$ 是 m 元向量:

$$\mathbf{r} = \begin{bmatrix} E\{f_n f_{n-1}\} \\ E\{f_n f_{n-2}\} \\ \vdots \\ E\{f_n f_{n-m}\} \end{bmatrix}, \quad \boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} \tag{8.5.10}$$

因此,对任何输入图像,使式(8.5.7)取得最小的系数可以通过一系列基本的矩阵操作确定。另外,这些系数只取决于原始图像中像素的自相关。使用这些最佳系数得到的预测误差的方差为:

$$\sigma_e^2 = \sigma^2 - \boldsymbol{\alpha}^T\mathbf{r} = \sigma^2 - \sum_{i=1}^{m} E\{f_n f_{n-i}\}\alpha_i \tag{8.5.11}$$

尽管计算式(8.5.8)的机理非常简单,但构造 $\mathbf{R}$ 和 $\mathbf{r}$ 需要的自相关的计算实现起来是非常困难的。因此,在实践中,几乎从不使用局部预测(这些预测的预测系数是逐图像进行计算的)。

在大多数情况下,通过假设一个简单的图像模型并将相应的自相关代入式(8.5.9)和式(8.5.10)去计算一系列全局系数。比如,当假设一个带有可分离的自相关函数的二维马尔可夫信源(见 8.3.3 节):

$$E\{f(x,y)f(x-i,y-i)\} = \sigma^2\rho_v^i\rho_h^i \tag{8.5.12}$$

和广义的四阶线性预测器:

$$\begin{aligned} \hat{f}(x,y) = {} & \alpha_1 f(x,y-1) + \alpha_2 f(x-1,y-1) \\ & + \alpha_3 f(x-1,y) + \alpha_4 f(x-1,y+1) \end{aligned} \tag{8.5.13}$$

的时候,得到的最佳系数是:

$$\alpha_1 = \rho_h \quad \alpha_2 = -\rho_v\rho_h \quad \alpha_3 = \rho_v \quad \alpha_4 = 0 \tag{8.5.14}$$

这里,ρ_h 和 ρ_v 分别是所研究的图像的水平和垂直相关系数。

最后,式(8.5.6)中预测系数的和通常要小于或等于 1。即,

$$\sum_{i=1}^{m} \alpha_i \leqslant 1 \tag{8.5.15}$$

这种限制是为了确保预测器的输出能落到灰度级的允许范围内,并减少传输噪声的影响,传输噪声的影响通常在重构图像中表现为水平的条纹。减小 DPCM 解码器对输入噪声的敏感性是很重要的,因为单个差错(在适当的环境下)会传播到所有以后的输出。即,解码器的输出会变得不稳定。通过进一步限制式(8.5.15)必须严格小于 1,输入差错的影响被限制在少量的输出中。

例 8.17 预测技术的对比

考虑由对图 8.23 中的单色图像进行 DPCM 编码产生的预测误差,假设量化误差为零且用

下列四个预测器的一个：

$$\hat{f}(x,y) = 0.97f(x,y-1) \tag{8.5.16}$$

$$\hat{f}(x,y) = 0.5f(x,y-1) + 0.5f(x-1,y) \tag{8.5.17}$$

$$\hat{f}(x,y) = 0.75f(x,y-1) + 0.75f(x-1,y) - 0.5f(x-1,y-1) \tag{8.5.18}$$

$$\hat{f}(x,y) = \begin{cases} 0.97f(x,y-1) & \Delta h \leqslant \Delta v \\ 0.97f(x-1,y) & \text{其他} \end{cases} \tag{8.5.19}$$

这里，$\Delta h = |f(x-1,y) - f(x-1,y-1)|$ 和 $\Delta v = |f(x,y-1) - f(x-1,y-1)|$ 表示在点(x,y)的水平和垂直梯度。式(8.5.16)到式(8.5.18)定义了一个 α_i 的相对鲁棒性的集合。这个鲁棒性的集合在图像很宽的范围上提供了令人满意的性能。设计式(8.5.19)的自适应预测器，通过计算图像的方向特性的局部度量(Δh 和 Δv)，并选择适合度量性能的预测器来改善边缘的重现质量。

图 8.23　一幅 512 × 512 大小的 8 比特单色图像

图 8.24(a)到(d)显示了预测误差图像，这些图像是使用式(8.5.16)到式(8.5.19)的预测器得到的。注意随着预测器阶数的增加明显可见的误差减少了[①]。预测误差分布的标准差遵从相似的模式。它们分别是 4.9,3.7,3.3 和 4.1 个灰度级。

最佳量化

图 8.25 中所示的阶梯量化函数 $t = q(s)$是关于 s 的奇函数[即，$q(-s) = -q(s)$]，这个函数完全可以用在图形的第一象限所显示的 s_i 和 t_i 的 $L/2$ 值描述。这些断点定义了函数的不连续性，并被称为量化器的判决和重构级。为了方便起见，如果 s 在半开区间$(s_i, s_{i+1}]$内，则可以将 s 看做到 t_i 的映射。

① 用多于前 3 个或前 4 个像素的预测因子，在预测因子的复杂性增加时无法获得更大的压缩增益(Habibi[1971])。