



数字图象处理

中国科学技术大学
电子工程与信息科学系

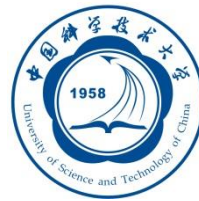
主讲教师：李厚强 (lihq@ustc.edu.cn)
周文罡 (zhwg@ustc.edu.cn)

助教：王宁 (wn6149@mail.ustc.edu.cn)



第8章 图像压缩

- 图像压缩是一种减少描绘一幅图像所需数据量的技术和科学，它是数字图像处理领域最有用、商业上最成功的的技术之一
- Web网页图像和高分辨率数字摄像机图片需要进行压缩，以节省存储空间和减少传输时间。同时，图像压缩在视频会议、遥感、文本和医学图像处理领域扮演重要角色
- 本章介绍数字图像压缩的理论和实践



第8章 图像压缩

□ 8.1 基础知识

- 冗余编码、空间冗余和时间冗余、不相关的信息量、图像信息的度量、保真度准则、图像压缩模型、图像格式容器和压缩标准

□ 8.2 一些基本的压缩方法

- | | |
|------------|-----------|
| ■ 霍夫曼编码 | ■ 基于符号的编码 |
| ■ Golomb编码 | ■ 比特平面编码 |
| ■ 算术编码 | ■ 块变换编码 |
| ■ LZW编码 | ■ 预测编码 |
| ■ 行程编码 | ■ 小波编码 |

□ 8.3 数字图像水印

8.1 基础知识

□ 相对数据冗余 R

$$R = 1 - \frac{1}{C}$$

← 压缩率

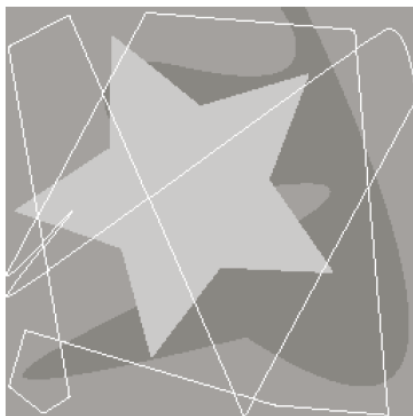
□ 三种主要类型的数据冗余：

- 编码冗余
- 空间和时间冗余
- 不相关的信息

编码冗余

空间冗余

无关信息



计算机产生的 $256 \times 256 \times 8$ 比特图像

编码冗余

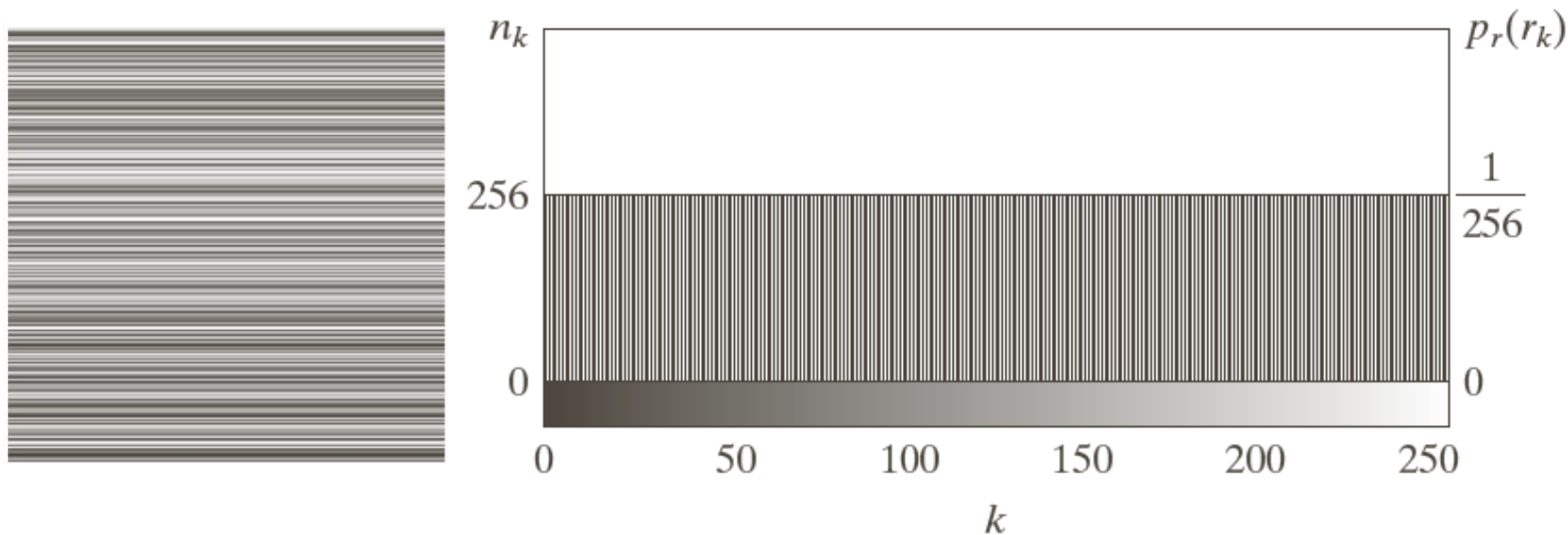
□ r_k 表示一副 $M \times N$ 凸显灰度, r_k 出现概率 $p_r(r_k)$

$$p_r(r_k) = \frac{n_k}{MN}, k = 0, 1, 2, \dots, L - 1$$

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

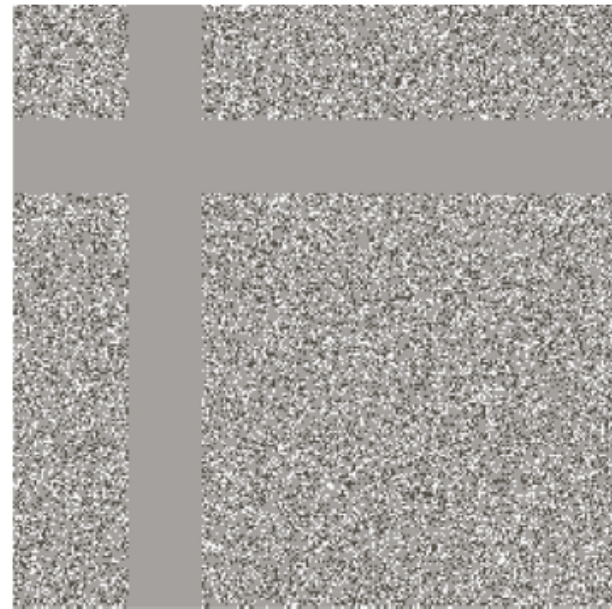
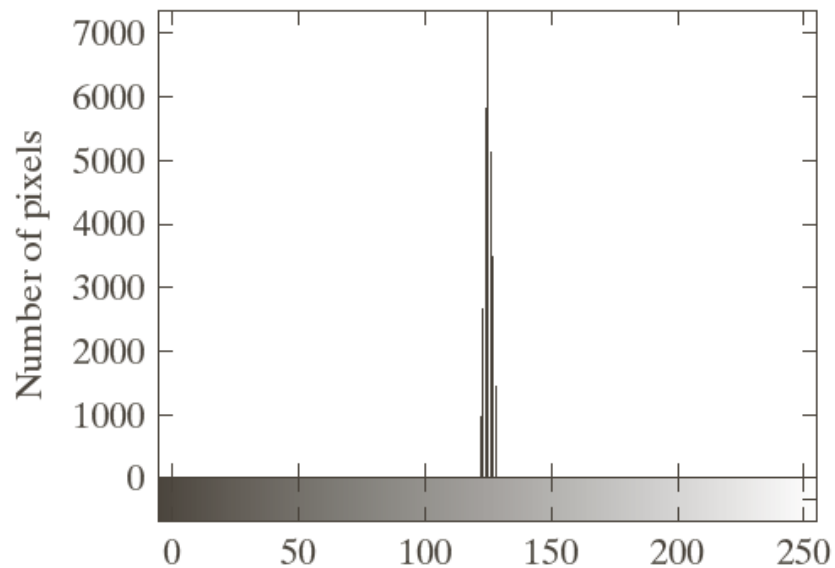
空间冗余和时间冗余



- 空间相关（x和y方向）：多数像素可根据相邻像素灰度进行合理预测
- 时间相关：当该图像是视频序列的一部分时

不相关信息

- 数字图像压缩中，被人类视觉系统忽略的信息或与图像与其的应用无关的信息需要被删除





图像信息的度量

- 概率为 $P(E)$ 的随机事件包含的信息

$$I(E) = -\log P(E)$$

- 信源的平均信息称为该信源的熵

$$H = -\sum_{j=1}^J P(a_j) \log P(a_j)$$

- 灰度信源的熵

$$\tilde{H} = \sum_{k=0}^{L-1} P_r(r_k) \log_2 p_r(r_k)$$

- 香农第一定理（无噪声编码定理）

$$\lim_{n \rightarrow \infty} \left[\frac{L_{avg,n}}{n} \right] = H$$



保真度准则

- $f(x, y)$ 和 $\hat{f}(x, y)$ 之间误差 $e(x, y)$

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

- 平均误差的平方根

$$e_{rms} = \left[\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]^{\frac{1}{2}}$$

- 均方信噪比

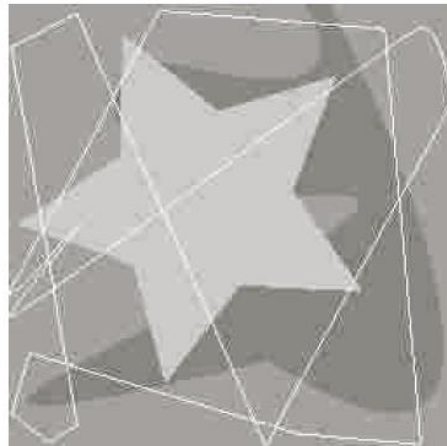
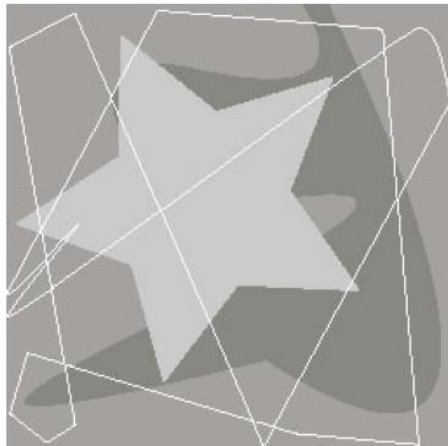
$$\text{SNR}_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2}$$

保真度准则

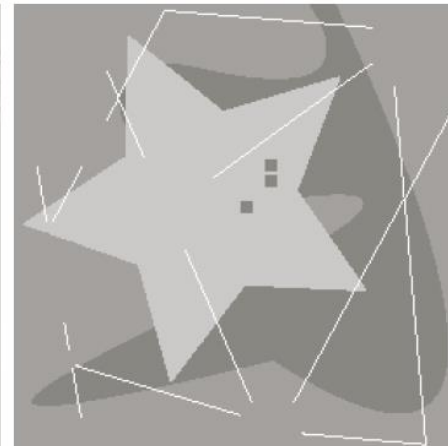
Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

电视配置研究组
组织的等级尺度

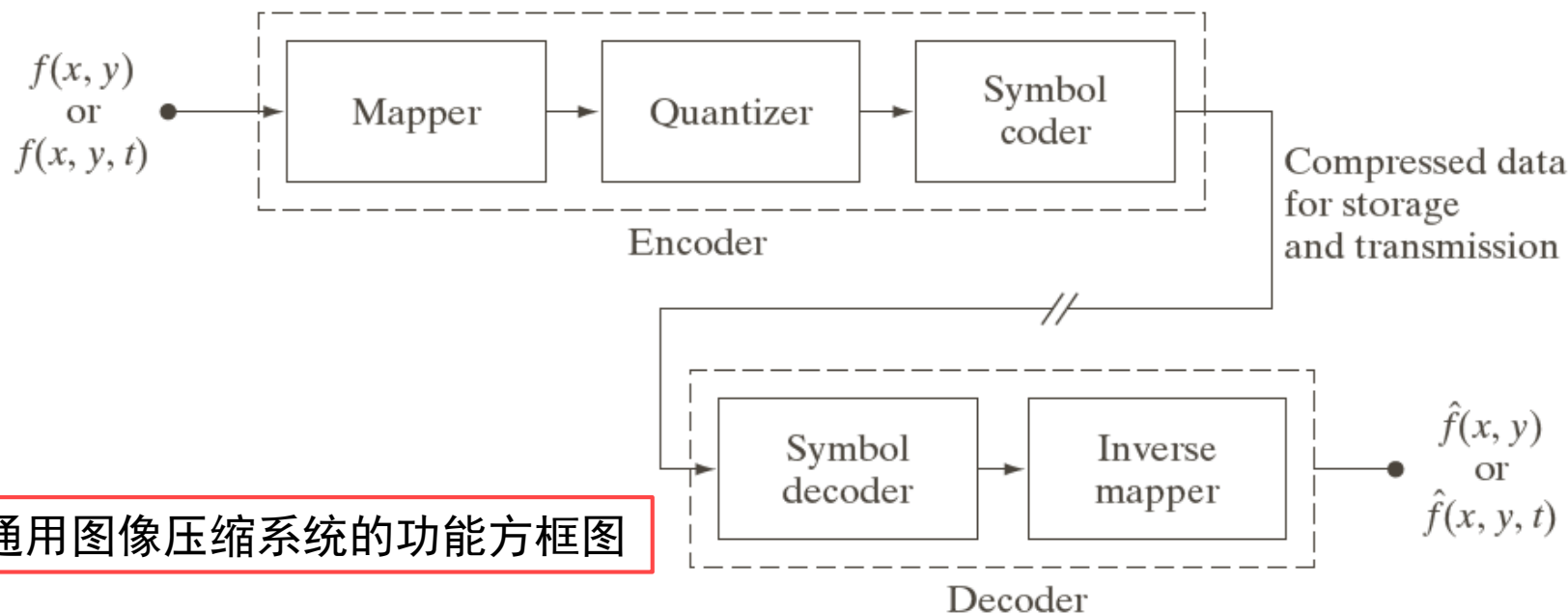
压缩重建的典型图像



人工生成的图像



图像压缩模型



一个通用图像压缩系统的功能方框图

□ 编码或压缩过程

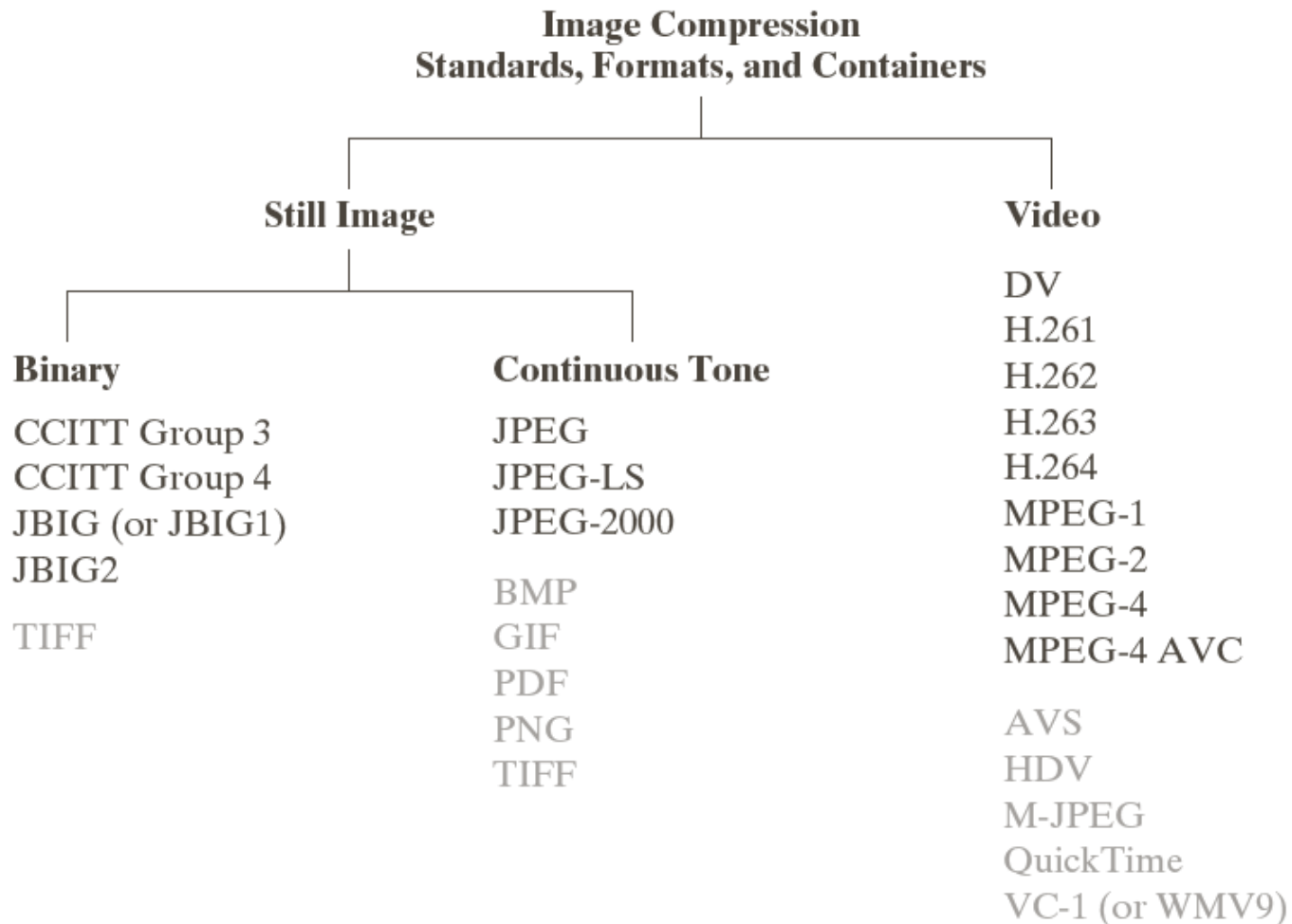
- 映射器：降低空间和时间冗余
- 量化器：根据预设保真度准则，排除压缩表示的无关信息
- 符号编码器：生成定长编码或变长编码，最小化编码冗余

□ 解码或解压缩过程

- 符号解码器和反映射器



图像格式、容器和压缩标准





8.2 一些基本的压缩方法

- ☐ 霍夫曼编码
- ☐ Golomb编码
- ☐ 算数编码
- ☐ LZW编码
- ☐ 行程编码
- ☐ 基于符号的编码
- ☐ 比特平面编码
- ☐ 块变换编码
- ☐ 预测编码
- ☐ 小波编码

霍夫曼编码

霍夫曼编码步骤：

(1) 概率排序，缩减和简化信源符号数量

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6 0.4
a_6	0.3	0.3	0.3	0.3	
a_1	0.1	0.1	0.2 0.1	0.3	
a_4	0.1	0.1			
a_3	0.06	0.1			
a_5	0.04				

平均长度

$$L_{avg} = 0.4 \times 1 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 4 + 0.06 \times 5 + 0.04 \times 5 = 2.2$$



霍夫曼编码

霍夫曼编码步骤：

(2) 对每个简化后的信源进行编码，从最小信源开始，直到遍历原始信源

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Golomb编码

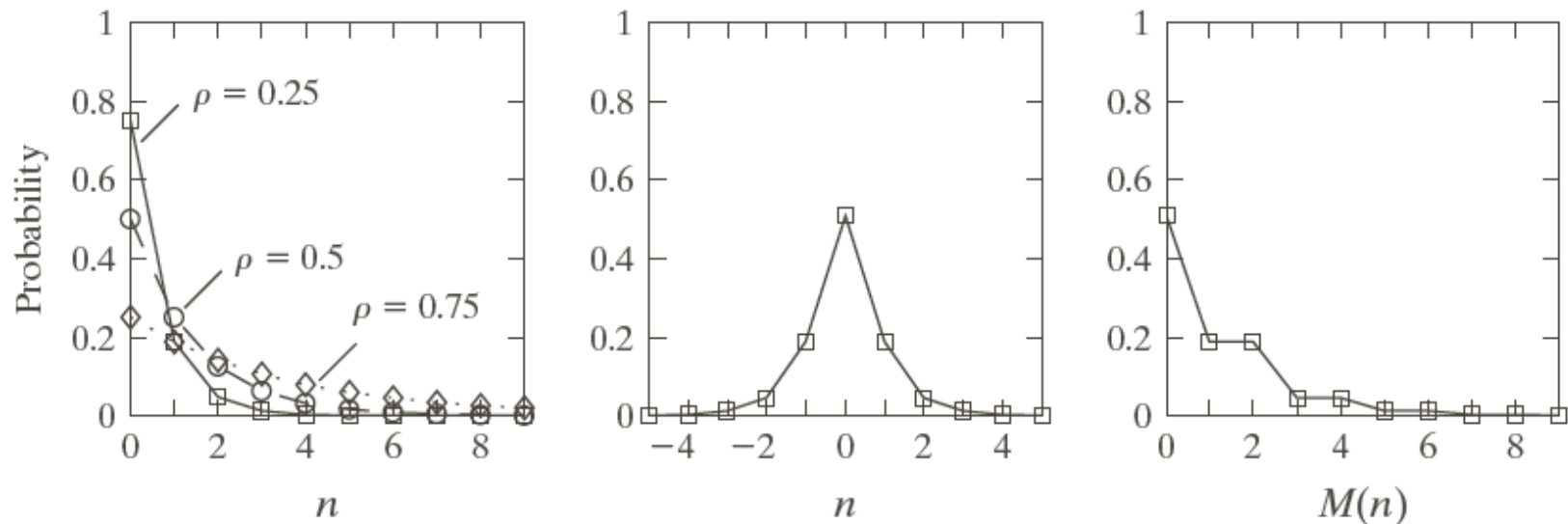
- 具有指数衰减概率分布输入的非负整数编码
- 给定一个非负整数 n 和一个正整数除数 $m > 0$ 后，表示为 $G_m(n)$ 的 n 关于 m 的Golomb编码是商 $\lfloor n/m \rfloor$ 的一元编码和 $n \bmod m$ 的二进制表示的并集
- $G_m(n)$ 的构建步骤：
 - 步骤1：生成商 $\lfloor n/m \rfloor$ 的一元编码
 - 步骤2：令 $k = \lfloor \log_2 m \rfloor$, $c = 2^k - m$, $r = n \bmod m$ ，并计算截短的余数 r' ，使其满足
$$r' = \begin{cases} r \text{截短至 } k-1 \text{ 比特,} & 0 \leq r < c \\ r + c \text{截短至 } k \text{ 比特,} & \text{其他} \end{cases}$$
 - 步骤3：连接步骤1和步骤2的结果

Golomb编码

□ 整数0~9的几种Golomb编码

n	$G_1(n)$	$G_2(n)$	$G_4(n)$	$G_{\text{exp}}^0(n)$
0	0	00	000	0
1	10	01	001	100
2	110	100	010	101
3	1110	101	011	11000
4	11110	1100	1000	11001
5	111110	1101	1001	11010
6	1111110	11100	1010	11011
7	11111110	11101	1011	1110000
8	111111110	111100	11000	1110001
9	1111111110	111101	11001	1110010

Golomb编码



- (a) 来自(1)式的三个单边的聚合分布；
 (b) 一个双边的指数衰减分布
 (c) 使用式(2)重新对图(b)重排序的结果

$$P(n) = (1 - \rho)\rho^n \quad (1)$$

$$M(n) = \begin{cases} 2n, & n \geq 0 \\ 2|n| - 1, & n < 0 \end{cases} \quad (2)$$

Golomb编码

□ k 阶指数Golomb编码 $G_{\text{exp}}^k(n)$ 计算步骤

- 步骤1: 寻找一个整数 $i, i \geq 0$, 使其满足

$$\sum_{j=0}^{i-1} 2^{j+k} \leq n < \sum_{j=0}^i 2^{j+k}$$

并形成 i 的一元码。若 $k = 0$, 则 $i = \lfloor \log_2(n + 1) \rfloor$

- 步骤2: 把二进制表示

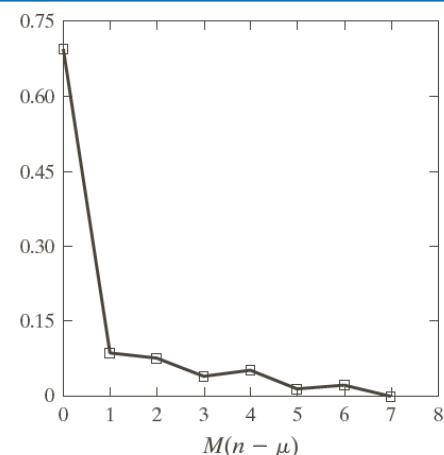
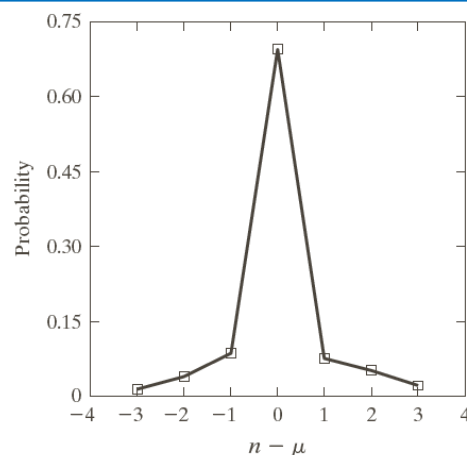
$$n - \sum_{j=0}^{i-1} 2^{j+k}$$

截短到 $k + 1$ 个最低阶比特

- 连接步骤1和步骤2的结果



- (a) 从图像每个像素中减去平均灰度后该图像的概率分布
- (b) 对图(a)使用上页(2)式得到的映射



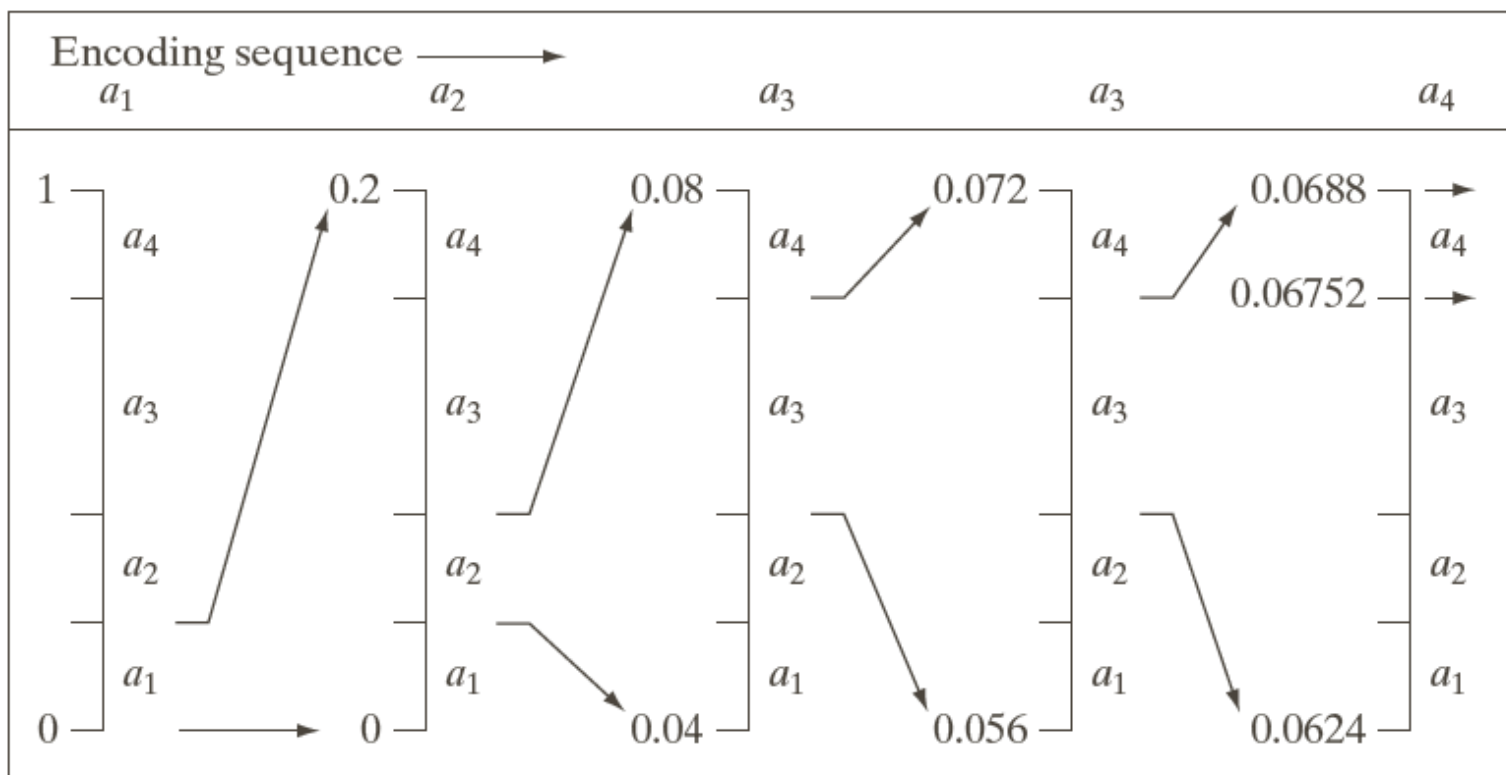
算术编码

示例：对来自一个4符号信源的5符号序列进行编码

算术编码过程

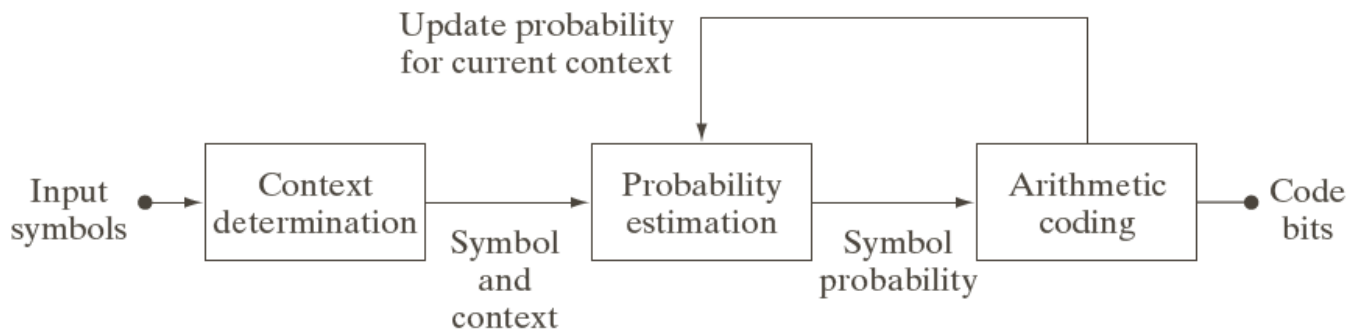
Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

信源



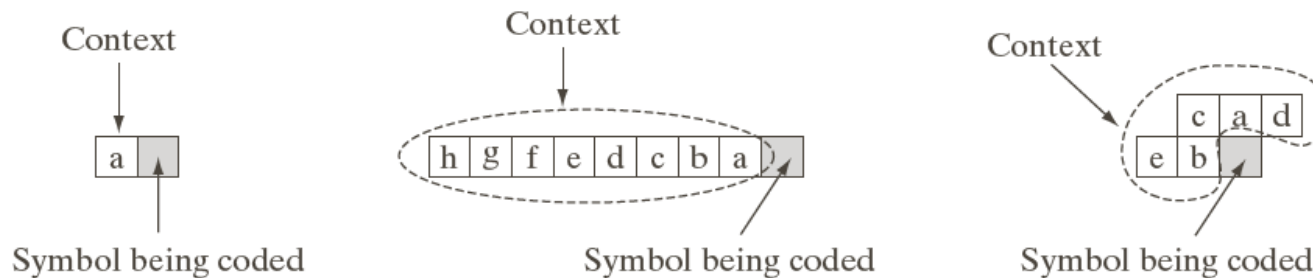
算术编码

□ 自适应上下文相关的概率估计



a
b c d

FIGURE 8.13
 (a) An adaptive, context-based arithmetic coding approach (often used for binary source symbols).
 (b)–(d) Three possible context models.



(a) 自适应上下文相关的算术编码方法（通常用于二值信源符号）
 (b)~(d) 三种可能的上下文模型



LZW编码

- 无误差压缩图像中空间冗余的方法
- 将定长码字分配各变长信源符号序列

39 39 126 126
39 39 126 126
39 39 126 126
39 39 126 126

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126		126		

LZW编码示例



行程编码

- 处理行（或列）重复灰度的图像
- 行程对指定一个新灰度的开始和具有该灰度的连续像素的数量
- 主要适用于压缩二值图像
- BMP文件格式的RLE是行程编码的一种形式
- 两种古老且应用最广的二值图像压缩标准CCITT Group3和4采用了行程编码

行程编码

□ BMP文件格式的RLE

- 图像数据使用两种不同的模式来表示：编码和绝对值
- 编码模式用两个字节RLE表示，第一个字节规定连续像素的数目，它有包含在第二个字节中的彩色索引
- 绝对值模式中，第一个字节是0，第二个字节是4中可能的条件之一。
 - ✓ 当第二个字节是0或1时，就到达一行或一幅图像末尾
 - ✓ 当第二个字节是2，则下两个字节包含相对图像中一个新空间位置的无符号水平偏移或垂直偏移
 - ✓ 当第二个字节在3和255之间，则它指定了非压缩像素的数量

Second Byte Value	Condition
0	End of line
1	End of image
2	Move to a new position
3-255	Specify pixels individually

TABLE 8.8

BMP absolute coding mode options. In this mode, the first byte of the BMP pair is 0.



行程编码

□ CCITT垂直模式编码示例

- CCITT Group3和4标准采用的二维压缩方法是逐行方法
- 对每个黑到白或白到黑的行程转换的位置，根据位于当前编码行的参考元素a0的位置来编码
- 已编码的前一行称为参考行，每幅新图像的第一行的参考行为一个虚构的白色行
- 所用的二维编码技术称为相对元素地址指派（READ）编码

行程编码

CCITT单一扫描行的二维编码过程

TABLE 8.9
CCITT two-dimensional code table.

Mode	Code Word
Pass	0001
Horizontal	$001 + M(a_0a_1) + M(a_1a_2)$
Vertical	
a_1 below b_1	1
a_1 one to the right of b_1	011
a_1 two to the right of b_1	000011
a_1 three to the right of b_1	0000011
a_1 one to the left of b_1	010
a_1 two to the left of b_1	000010
a_1 three to the left of b_1	0000010
Extension	0000001xxx

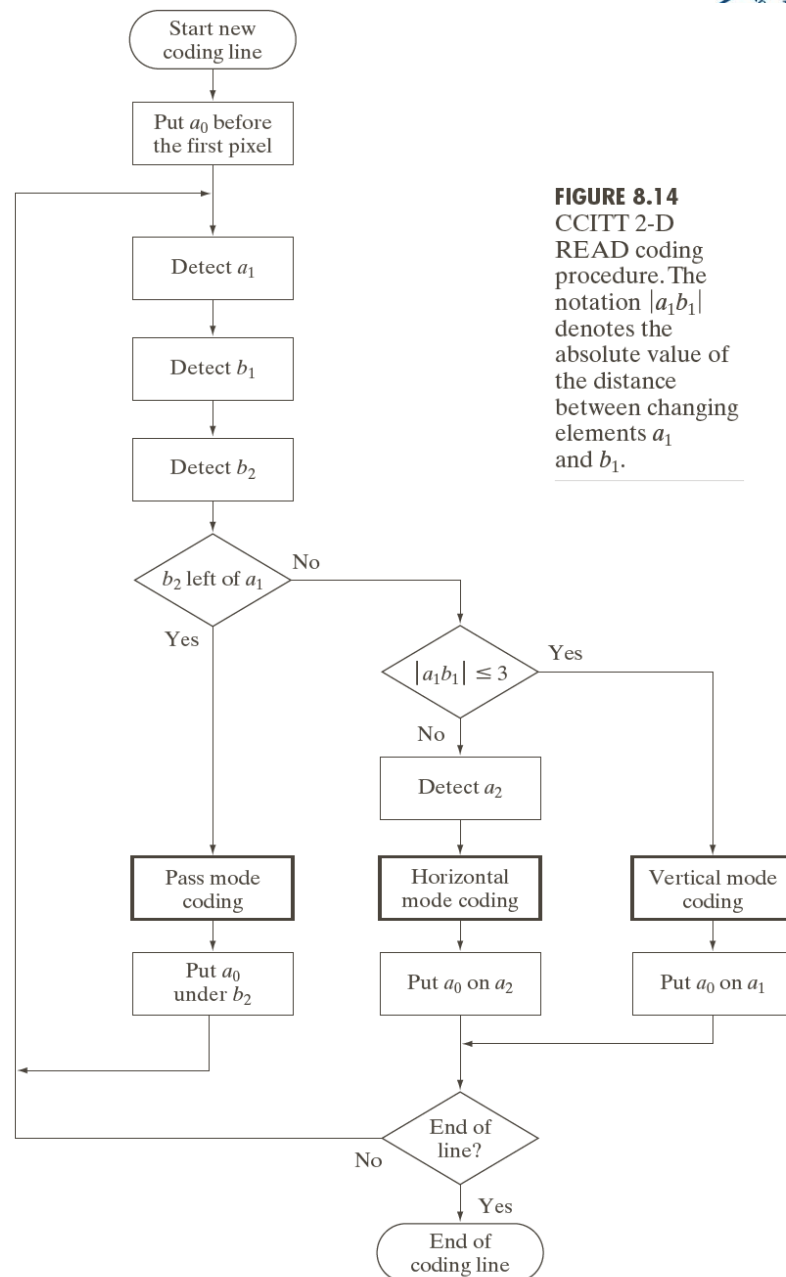
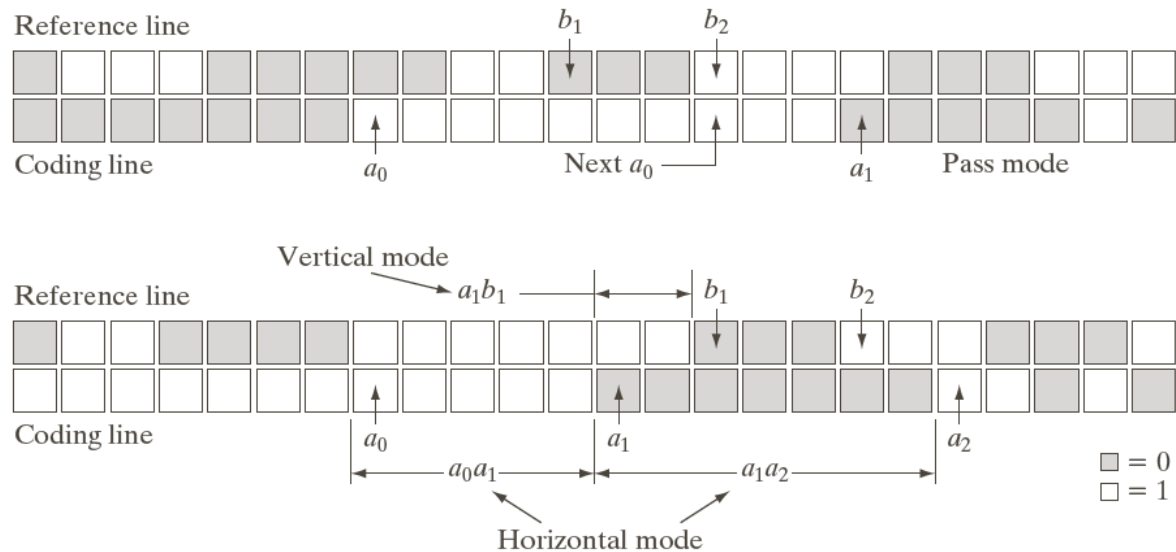


FIGURE 8.14
CCITT 2-D
READ coding
procedure. The
notation $|a_1b_1|$
denotes the
absolute value of
the distance
between changing
elements a_1
and b_1 .

行程编码

CCITT垂直模式编码实例



行程编码

□ CCITT压缩实例

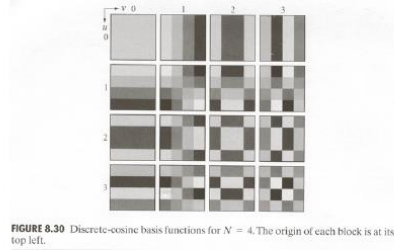


FIGURE 8.30 Discrete-cosine basis functions for $N = 4$. The origin of each block is at its top left.

where

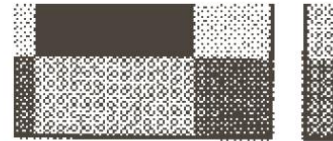
$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1, 2, \dots, N-1 \end{cases} \quad (8.5-33)$$

and similarly for $\alpha(v)$. Figure 8.30 shows $g(x, y, u, v)$ for the case $N = 4$. The computation follows the same format as explained for Fig. 8.29, with the difference that the values of g are not integers. In Fig. 8.30, the lighter gray levels correspond to larger values of g .

Figures 8.31(a), (c), and (e) show three approximations of the 512×512 monochrome image in Fig. 8.23. These pictures were obtained by dividing the original image into subimages of size 8×8 , representing each subimage using one of the transforms just described (i.e., the DFT, WHT, or DCT transform), truncating 50% of the resulting coefficients, and taking the inverse transform of the truncated coefficient arrays.

In each case, the 32 retained coefficients were selected on the basis of maximum magnitude. When we disregard any quantization or coding issues, this process amounts to compressing the original image by a factor of 2. Note that in all cases, the 32 discarded coefficients had little visual impact on reconstructed image quality. Their elimination, however, was accompanied by some mean-square error, which can be seen in the scaled error images of Figs. 8.31(b), (d), and (f). The actual rms errors were 1.28, 0.86, and 0.68 gray levels, respectively.

EXAMPLE 8.19:
Transform coding
with the DFT,
WHT, and DCT.



a b

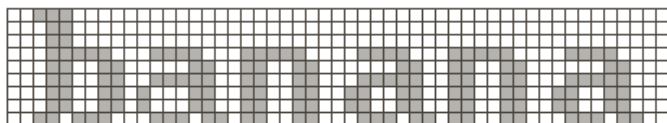
FIGURE 8.16 A binary scan of a book page: (a) scaled to show the general page content; (b) scaled to show the binary pixels used in dithering.

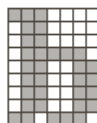


$r N = 4. \text{TI}$

- (a) 图的文档不压缩需要1952*2697比特，658068字节
- 该文档的非PDF文件需要663445字节
- CCITT Group3 压缩可使该文件减少到123497字节，压缩率C=5.37
- CCITT Group4压缩将该文件减少到110456字节，压缩率约增加6

基于符号的编码

- 图像被表示为多幅频繁发生的子图像的集合，称为符号
- 符号字典：符号的集合
- 图像以一个三元组 $\{(x_1, y_1, t_1), \{(x_2, y_2, t_2)\}, \dots\}$ 的集合来编码， (x_i, y_i) 规定图像中某个符号的位置， t_i 表示该子图在字典中的地址

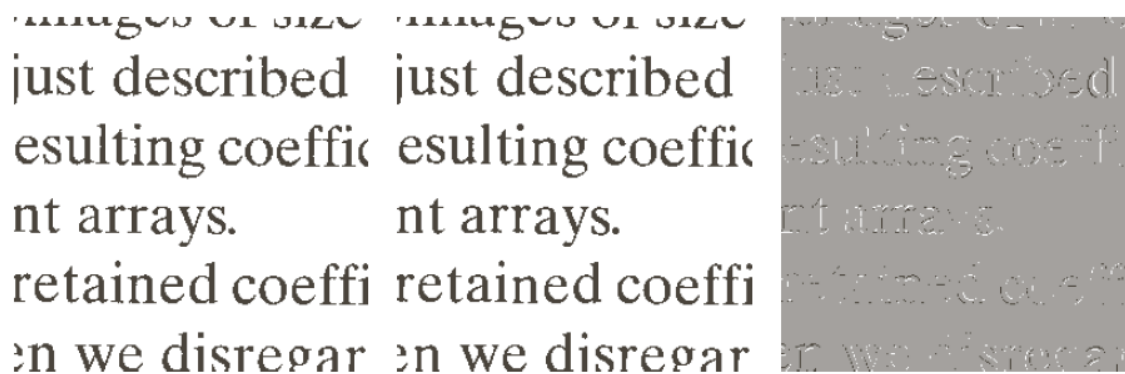


Token	Symbol	Triplet
0		(0, 2, 0) (3, 10, 1) (3, 18, 2)
1		(3, 26, 1) (3, 34, 2) (3, 42, 1)
2		

(a) 两灰度级文档 (b) 符号字典 (c) 文档中用于定位的三元组

基于符号的编码

- JBIG2是针对二级图像压缩的国际标准
- JBIG2包括无损压缩和感知有损压缩
- 感知无损指忽略字典位图和图像中相应字符特例之间的差别，但是不影响可读性



(a)无损压缩和重建 (b)感知无损压缩 (c)两种压缩程度的差值

比特平面编码

- 将一幅多级图像分解为一系列二值图像，然后使用二值压缩方法进行编码
- 两种分解方法

- 基2多项式

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \dots + a_12^1 + a_02^0$$

- m比特格雷码表示图像

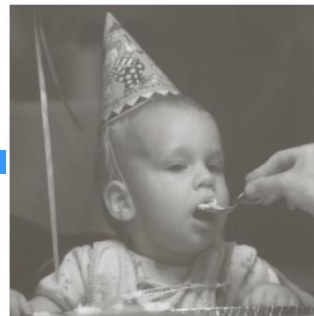
- ✓ 将m比特分解为格雷码 $g_{m-1} \dots g_2 g_1 g_0$

$$g_i = a_i \oplus a_{i+1}, 0 \leq i \leq m-2$$

$$g_{m-1} = a_{m-1}$$

比特平面编码

- (a) 一幅256比特单色图像
(b)-(h) 图(a)中图像的4个最高有效二进制编码和格雷编码比特平面



All bits



a_7



a_6



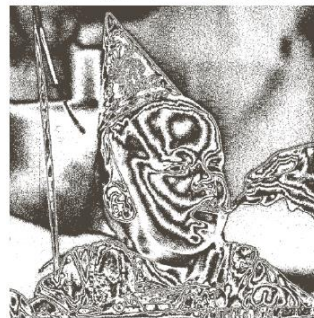
g_6



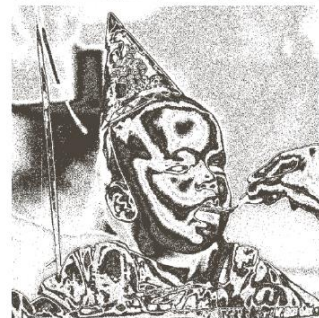
a_5



g_5



a_4



g_4

比特平面编码

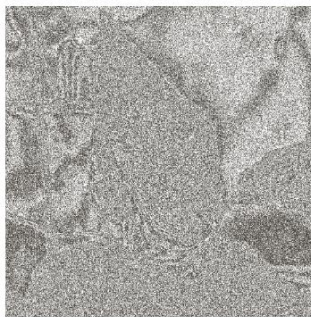
(a)-(h) 图(a)中图像的4个最低有效二进制编码和格雷编码比特平面



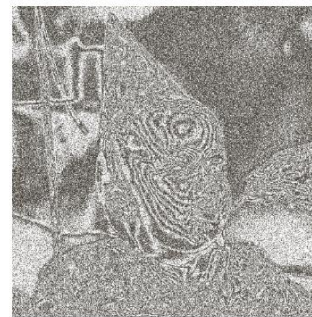
a_3



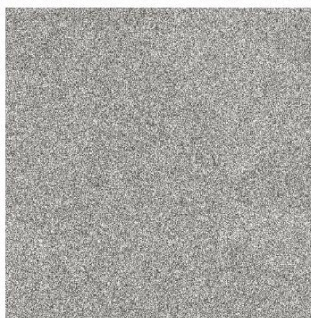
g_3



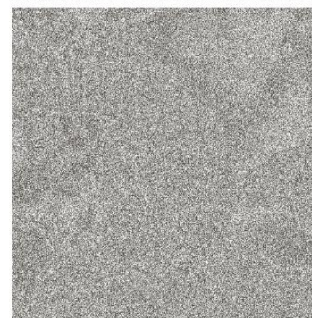
a_2



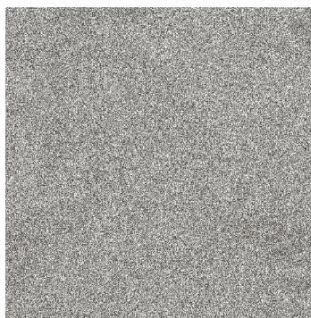
g_2



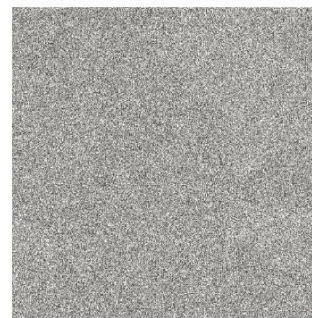
a_1



g_1



a_0



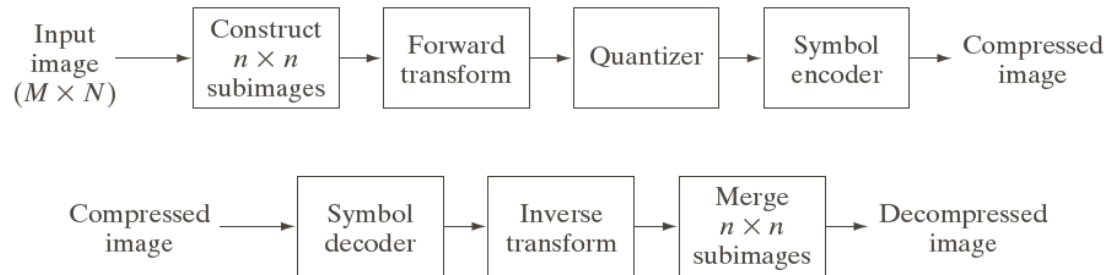
g_0

比特平面编码

Coefficient <i>m</i>	Binary Code (PDF bits)	Gray Code (PDF bits)	Compression Ratio
7	6,999	6,999	1.00
6	12,791	11,024	1.16
5	40,104	36,914	1.09
4	55,911	47,415	1.18
3	78,915	67,787	1.16
2	101,535	92,630	1.10
1	107,909	105,286	1.03
0	99,753	107,909	0.92

图像的二进制编码和格雷编码比特平面的JBIG2无损编码结果

块变换编码



块变换编码系统

正变换
$$T(u, v) = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} g(x, y) r(x, y, u, v)$$

反变换
$$g(x, y) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u, v) s(x, y, u, v)$$

块变换编码

□ 傅里叶变换核

$$r(x, y, u, v) = e^{-j2\pi(ux+vy)/n}$$

$$s(x, y, u, v) = \frac{1}{n^2} e^{j2\pi(ux+vy)/n}$$

□ Walsh-Hadamard变换 (WHT)

$$r(x, y, u, v) = s(x, y, u, v) = \frac{1}{n} (-1)^{\sum_{i=0}^{m-1} [b_i(x)p_i(u) + b_i(y)p_i(v)]}$$

$$n = 2^m$$

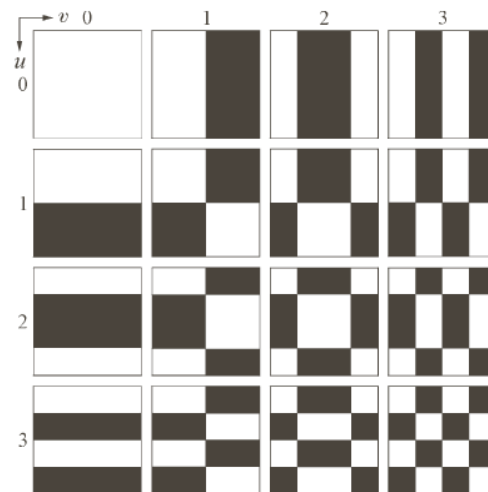
$b_k(z)$ 是 z 的二进制表示中的第 k 位

$$p_0(u) = b_{m-1}(u)$$

$$p_1(u) = b_{m-1}(u) + b_{m-2}(u)$$

...

$$p_{m-1}(u) = b_1(u) + b_0(u)$$



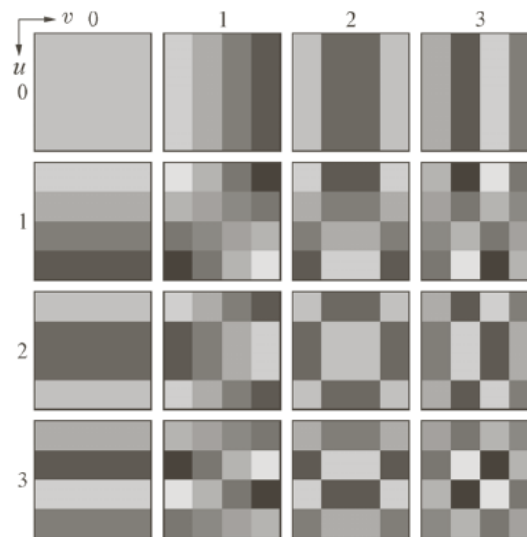
n=4时的WHT的基函数

块变换编码

□ 离散余弦变换 (DCT)

$$r(x, y, u, v) = s(x, y, u, v) = \alpha(u)\alpha(v) \cos\left[\frac{(2x+1)u\pi}{2n}\right] \cos\left[\frac{(2y+1)v\pi}{2n}\right]$$

$$\alpha(u) = \begin{cases} \sqrt{1/n}, u = 0 \\ \sqrt{2/n}, u = 1, 2, \dots, n-1 \end{cases}$$



n=4时的DCT的基函数

块变换编码



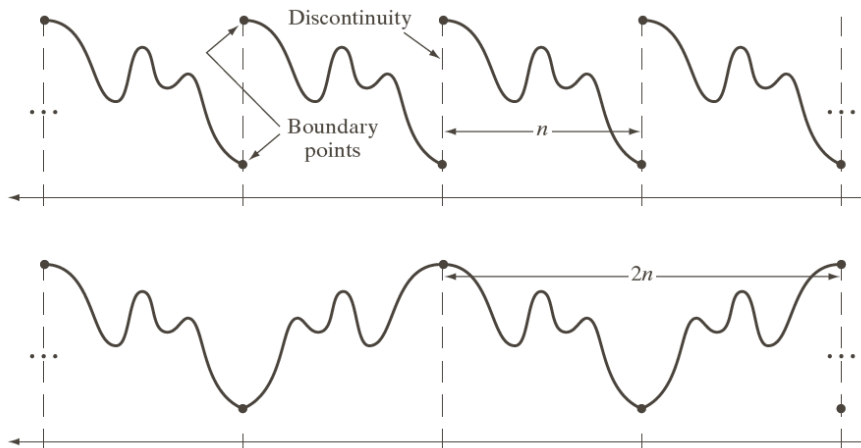
a	b	c
d	e	f

FIGURE 8.24 Approximations of Fig. 8.9(a) using the (a) Fourier, (b) Walsh-Hadamard, and (c) cosine transforms, together with the corresponding scaled error images in (d)–(f).

块变换编码

□ DCT的优点

- 用单片集成电路即可实现
- 可将最多信息装入最少系数中
- 可使块效应最小化
- 隐含周期性

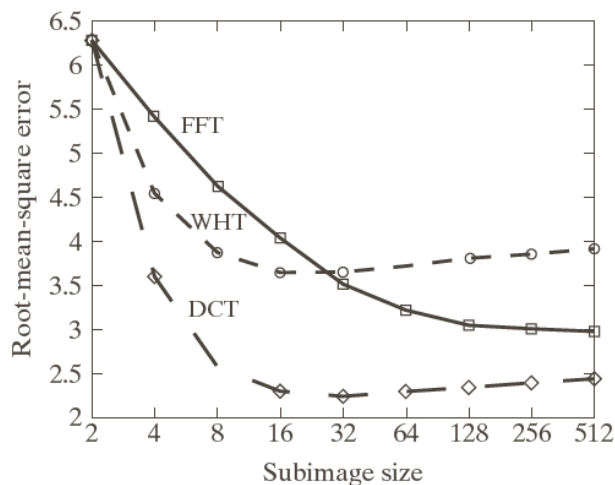


(a)一维DFT中隐含的周期性 (b)一维DCT中隐含的周期性

块变换编码

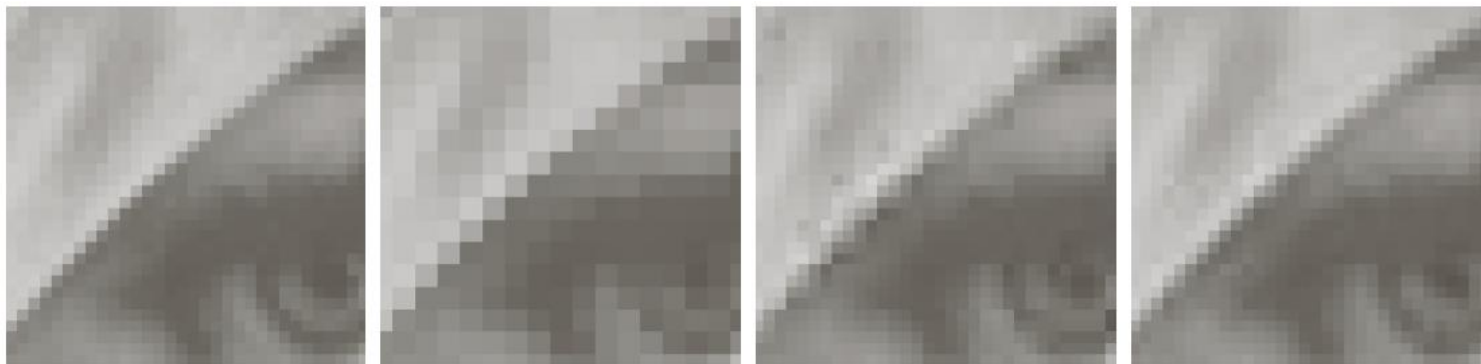
□ 子图像尺寸的选择

- 子图像的尺寸是影响编码误差和计算复杂性的重要因素
- 图像划分越细，编码效率越高，同时复杂度越高
- 最常用的子图像尺寸为 8×8 和 16×16



重建误差与子图像尺寸的关系

块变换编码



- (a)使用25%DCT系数对原图的近似
- (b)使用 2×2 大小的子图的近似
- (c)使用 4×4 大小的子图的近似
- (d) 使用 8×8 大小的子图的近似

块变换编码

□ 区域编码

- 保留的系数是根据最大方差进行选择

□ 阈值编码

- 保留的系数是根据最大幅度进行选择

□ 比特分配

- 对变换后的子图像的系数进行截取、量化和编码的过程



a b
c d

FIGURE 8.28
Approximations
of Fig. 8.9(a) using
12.5% of the
 8×8 DCT
coefficients:
(a)—(b) threshold
coding results;
(c)—(d) zonal
coding results. The
difference images
are scaled by 4.

块变换编码

□ 区域编码的实现

- 具有最大方差的变换系数被保留下来
- 区域取样可看做变换矩阵 $T(u,v)$ 乘以区域模板的对应元素

1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

8	7	6	4	3	2	1	0
7	6	5	4	3	2	1	0
6	5	4	3	3	1	1	0
4	4	3	3	2	1	0	0
3	3	3	2	1	1	0	0
2	2	1	1	1	0	0	0
1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0

1	1	0	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

a b
c d

FIGURE 8.29

A typical
(a) zonal mask,
(b) zonal bit
allocation,
(c) threshold
mask, and
(d) thresholded
coefficient
ordering
sequence. Shading
highlights the
coefficients that
are retained.

块变换编码

□ 阈值编码的实现

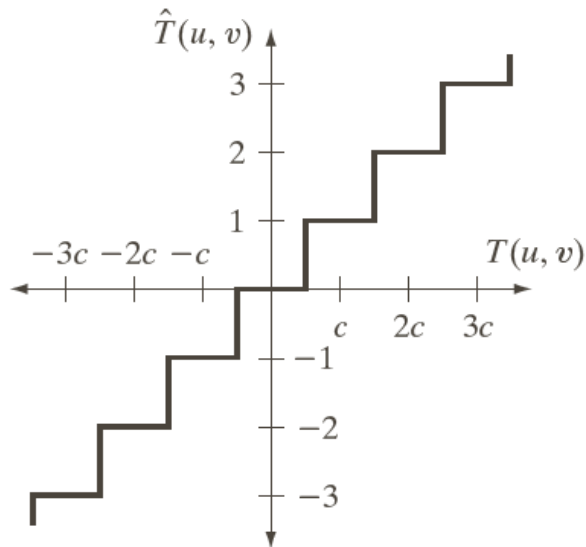
- 区域编码对所有子图像采用单个固定模板来实现
- 阈值编码在不同子图像保留系数的位置上具有自适应性
- 阈值模板函数的生成方法有三种
 - ✓ 对所有子图像采用单一的全局阈值
 - 对不同图像的压缩水平不同，取决于超过全局阈值的系数数量
 - ✓ 对每幅子图像采用不同的阈值
 - 对每幅子图像丢掉相同数量的系数，代码率恒定
 - ✓ 阈值随子图像内每个系数位置的变化而变化

$$\hat{T}(u,v) = \text{round}\left(\frac{T(u,v)}{Z(u,v)}\right) \quad \dot{T} = \hat{T}(u,v) Z(u,v)$$

$$Z = \begin{bmatrix} Z(0,0) & Z(0,1) & \dots & Z(0,n-1) \\ Z(1,0) & M & \dots & M \\ M & & & \\ Z(n-1,0) & Z(n-1,1) & \dots & Z(n-1,n-1) \end{bmatrix} \quad \text{变换标准化阵列}$$

块变换编码

□ 阈值编码的实现



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	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

a b

FIGURE 8.30
(a) A threshold coding quantization curve [see Eq. (8.2-29)]. (b) A typical normalization matrix.

块变换编码

□ 阈值编码的实现



FIGURE 8.31 Approximations of Fig. 8.9(a) using the DCT and normalization array of Fig. 8.30(b): (a) \mathbf{Z} , (b) $2\mathbf{Z}$, (c) $4\mathbf{Z}$, (d) $8\mathbf{Z}$, (e) $16\mathbf{Z}$, and (f) $32\mathbf{Z}$.

相应的均方误差：3.83,4.93,6.62,9.35,13.94和22.46



块变换编码

□ JPEG

- 广泛使用的连续色调静止帧压缩标准
- JPEG定义了三种编码系统
 - ✓ 有损的基本编码系统，以DCT为基础
 - ✓ 扩展的编码系统，用于更大压缩、更高精度或渐进式重建
 - ✓ 面向可逆压缩的无损独立编码系统

块变换编码

□ JPEG基本编码与解码

■ 8×8 子图像

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

■ 移位-128个灰度级

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34

块变换编码

□ JPEG基本编码与解码

■ 前向DCT

-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

■ 量化、缩放和截断

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

块变换编码

□ JPEG基本编码与解码

■ Z型扫描系数重排

`[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]`

■ 完整编码阵列

`1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001
001 100101 11100110 110110 0110 11110100 000 1010`

■ 重新生成的量化系数阵列

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

块变换编码

□ JPEG基本编码与解码

■ 去规格化

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

■ 反DCT变换

-70	-64	-61	-64	-69	-66	-58	-50
-72	-73	-61	-39	-30	-40	-54	-59
-68	-78	-58	-9	13	-12	-48	-64
-59	-77	-57	0	22	-13	-51	-60
-54	-75	-64	-23	-13	-44	-63	-56
-52	-71	-72	-54	-54	-71	-71	-54
-45	-59	-70	-68	-67	-67	-61	-50
-35	-47	-61	-66	-60	-48	-44	-44

块变换编码

□ JPEG基本编码与解码

■ 移位128个灰度级

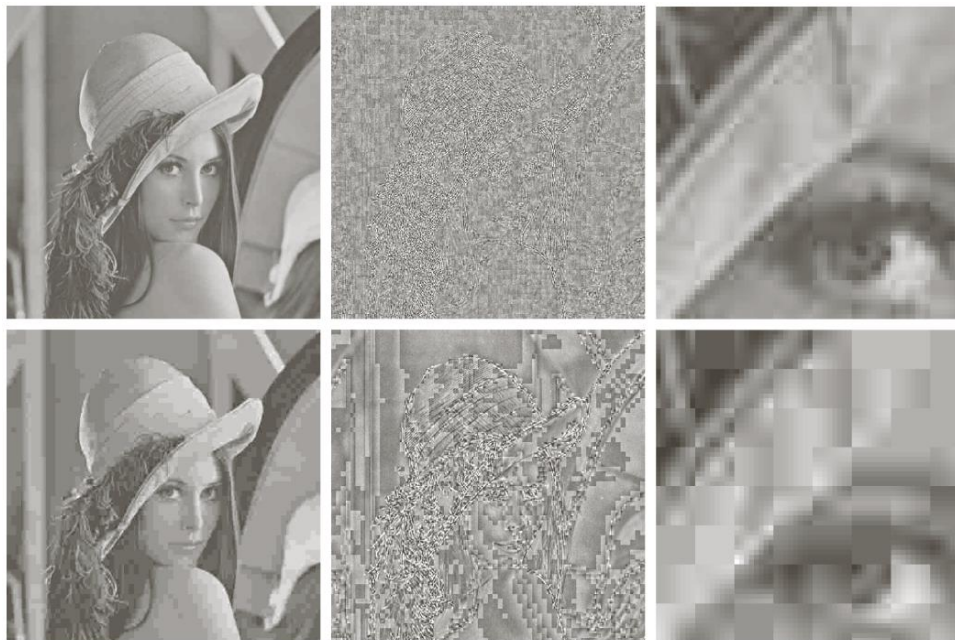
58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

■ 原图与重建的差别

-6	-9	-6	2	11	-1	-6	-5
7	4	-1	1	11	-3	-5	3
2	9	-2	-6	-3	-12	-14	9
-6	7	0	-4	-5	-9	-7	1
-7	8	4	-1	6	4	3	-2
3	8	4	-4	2	6	1	1
2	2	5	-1	-6	0	-2	5
-6	-2	2	6	-4	-4	-6	10

块变换编码

□ JPEG基本编码与解码



a b c
d e f

FIGURE 8.32 Two JPEG approximations of Fig. 8.9(a). Each row contains a result after compression and reconstruction, the scaled difference between the result and the original image, and a zoomed portion of the reconstructed image.

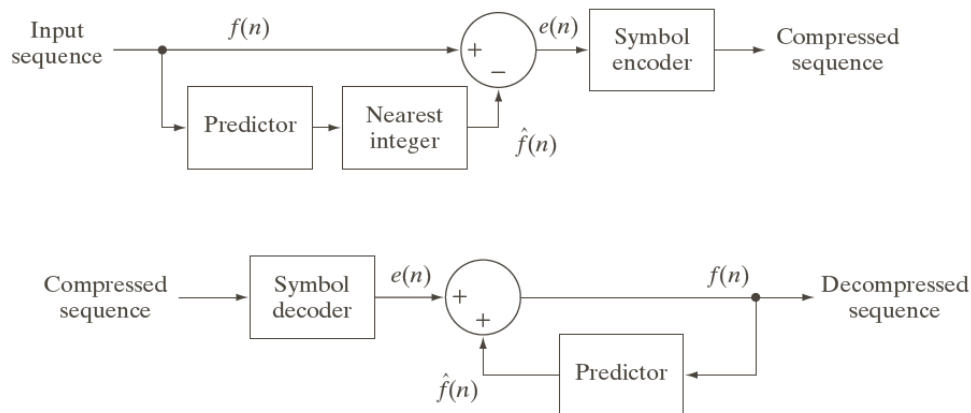
第一个结果压缩率25:1，第二个结果压缩率52:1

预测编码

概述

- 预测编码通过消除紧邻像素在空间和时间上的冗余来实现
- 一个像素的新信息定义为实际值和预测值的差值

无损预测编码



a
b

FIGURE 8.33
A lossless
predictive coding
model:
(a) encoder;
(b) decoder.

预测误差

$$e(n) = f(n) - \hat{f}(n)$$

重建

$$f(n) = e(n) + \hat{f}(n)$$

预测编码

□ 无损预测编码

- 多数情况下，预测可由m个样值进行线性组合

$$\hat{f}(n) = \text{round} \left[\sum_{i=1}^m \alpha_i f(n-i) \right]$$

- 对一幅图像，预测值可以是来自当前扫描行（一维线性预测编码）、当前行和前几个扫描行
- 一维线性预测图像编码

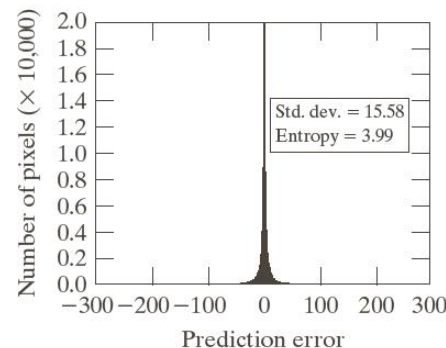
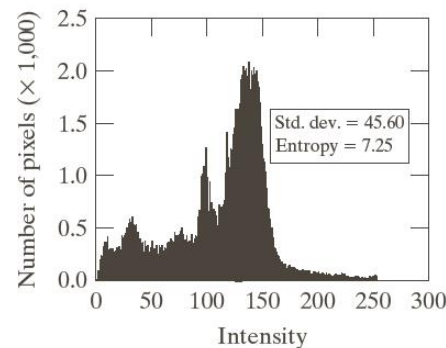
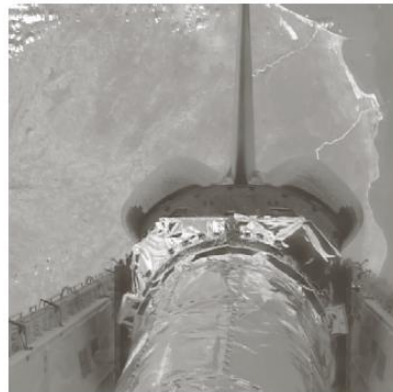
$$\hat{f}(x, y) = \text{round} \left[\sum_{i=1}^m \alpha_i f(x, y-i) \right]$$

预测编码

□ 无损预测编码

■ 一维线性预测图像编码示例1

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



a b
c d

FIGURE 8.34

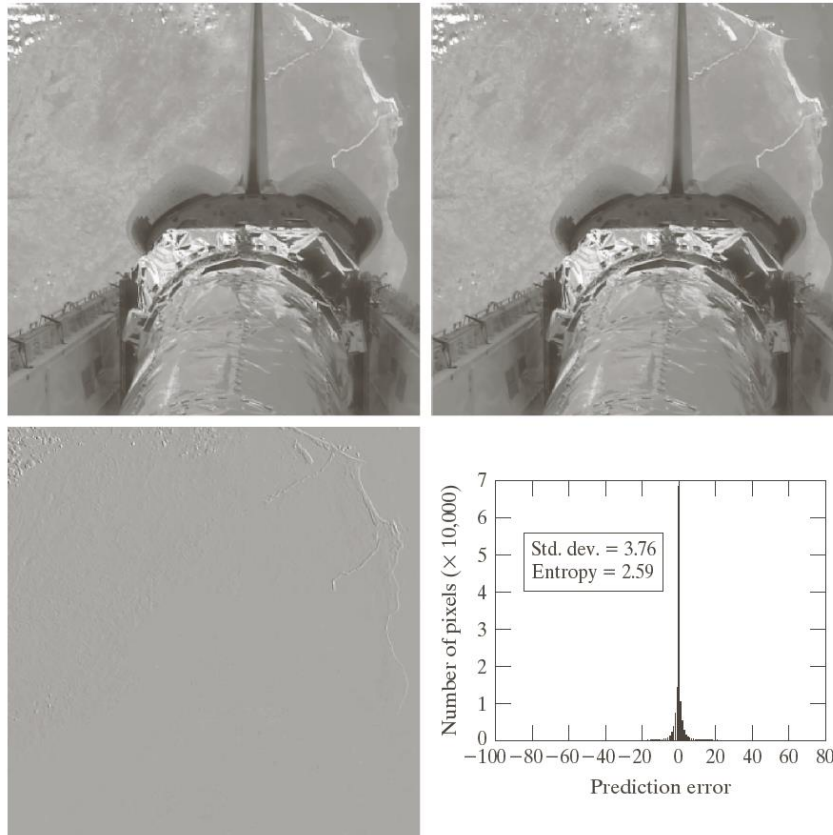
(a) A view of the Earth from an orbiting space shuttle. (b) The intensity histogram of (a). (c) The prediction error image resulting from Eq. (8.2-34). (d) A histogram of the prediction error. (Original image courtesy of NASA.)

预测编码

□ 无损预测编码

■ 一维线性预测图像编码示例2

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



a b
c d

FIGURE 8.35

(a) and (b) Two views of Earth from an orbiting space shuttle video. (c) The prediction error image resulting from Eq. (8.2-36). (d) A histogram of the prediction error.

(Original images courtesy of NASA.)

预测编码

□ 运动补偿预测残差

- 视频序列相邻帧相关性较强
- 使用运动补偿预测比空间预测更高效
- 运动补偿广泛用于视频编码中

□ 运动补偿基本原理

- 将视频帧分为互不重叠的字块，称为宏块
- 每个宏块在前一帧（称为参考帧）中找到最合适的块
- 使用运动矢量表示当前宏块在参考帧的位置
- 为提高运动搜索的精度，往往采用亚像素精度

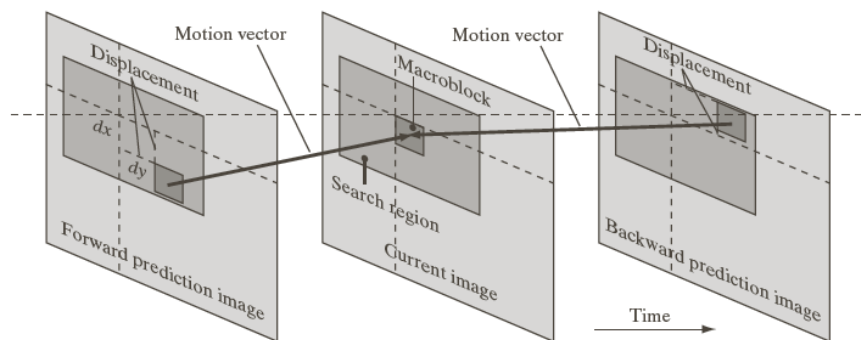
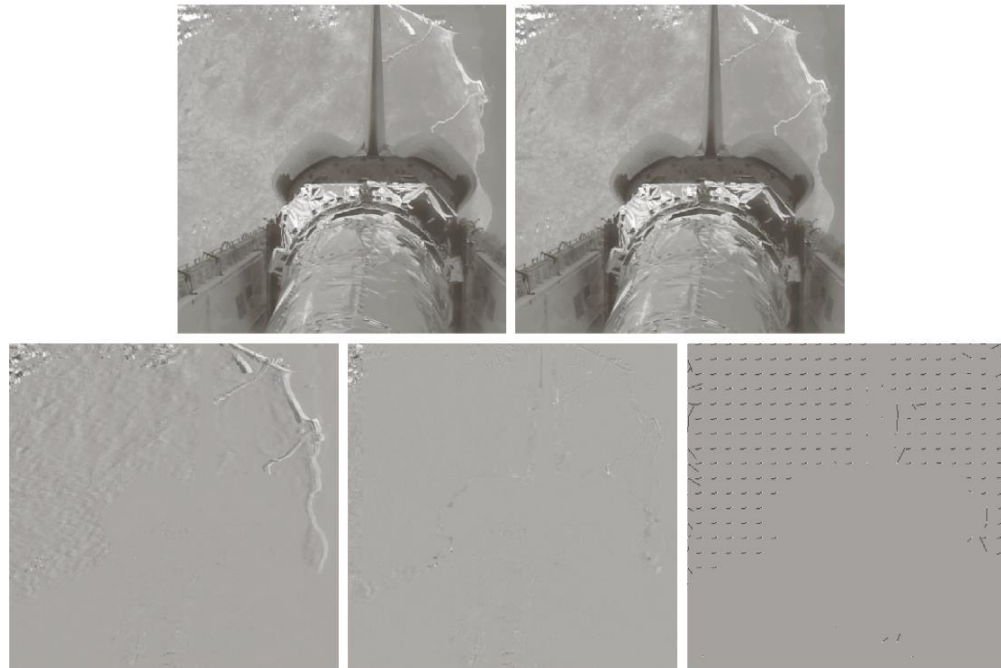


FIGURE 8.36
Macroblock
motion
specification.

预测编码

□ 运动补偿预测示例1

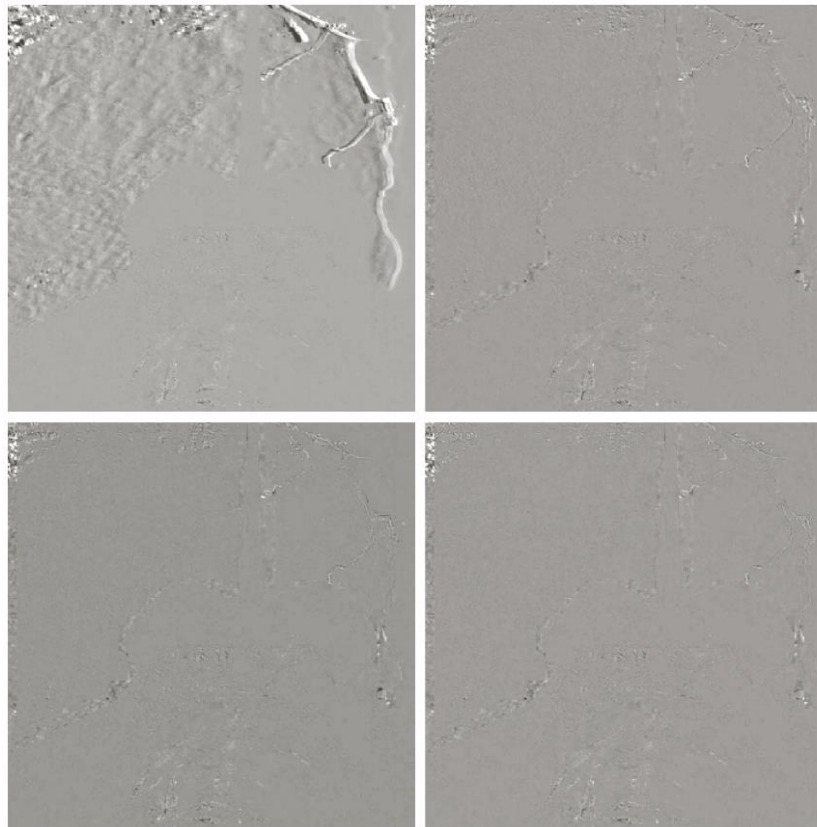


a b
c d e

FIGURE 8.37 (a) and (b) Two views of Earth that are thirteen frames apart in an orbiting space shuttle video. (c) A prediction error image without motion compensation. (d) The prediction residual with motion compensation. (e) The motion vectors associated with (d). The white dots in (d) represent the arrow heads of the motion vectors that are depicted. (Original images courtesy of NASA.)

预测编码

□ 运动补偿预测示例2



a b
c d

FIGURE 8.38

Sub-pixel motion compensated prediction residuals: (a) without motion compensation; (b) single pixel precision; (c) $\frac{1}{2}$ pixel precision; and (d) $\frac{1}{4}$ pixel precision. (All prediction errors have been scaled to the full intensity range and then multiplied by 2 to increase their visibility.)

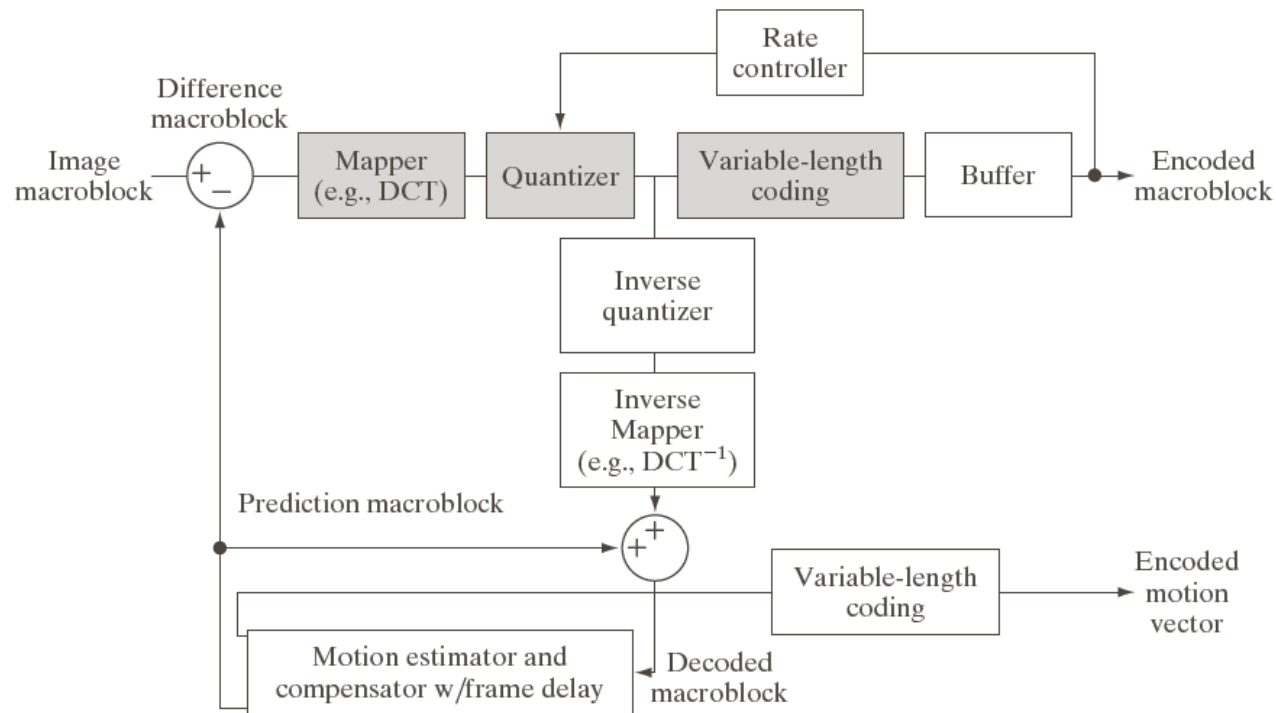
预测编码

- 运动估计的过程要求的计算量较大
- 只有编码端需要运动估计
- 视频压缩标准未包含运动估计，只在宏块维数、运动矢量精度等设置约束

Parameter	H.261	MPEG-1	H.262 MPEG-2	H.263	MPEG-4	VC-1 WMV-9	H.264 MPEG-4 AVC
Motion vector precision	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
Macroblock sizes	16×16	16×16	16×16 16×8	16×16 8×8	16×16 8×8	16×16 8×8	16×16 16×8 8×16 8×8 8×4 4×8 4×4
Transform	8×8 DCT	8×8 DCT	8×8 DCT	8×8 DCT	8×8 DCT	8×8 8×4 4×8 4×4 Integer DCT	4×4 8×8 Integer
Interframe predictions	P	P, B	P, B	P, B	P, B	P, B	P, B
I-frame intra-predictions	No	No	No	No	No	No	Yes

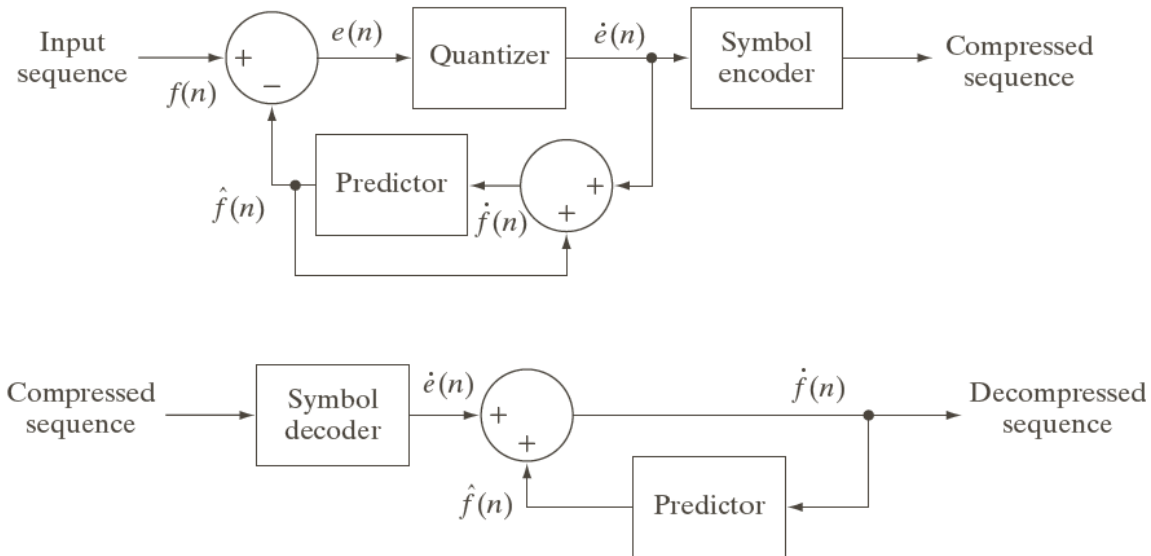
预测编码

□ 典型的基于运动补偿的视频编码器



预测编码

□ 有损预测编码



$$\dot{f}(n) = \dot{e}(n) + \hat{f}(n)$$



预测编码

□ 有损预测编码

■ △调制

✓ △调制是有损预测编码形式

✓ 预测器：

$$\hat{f}(n) = \alpha \dot{f}(n-1)$$

✓ 量化器

$$\hat{e}(n) = \begin{cases} +\xi, & e(n) > 0 \\ -\xi, & \text{others} \end{cases}$$



预测编码

□ 有损预测编码

■ △调制

✓ △调制是有损预测编码形式

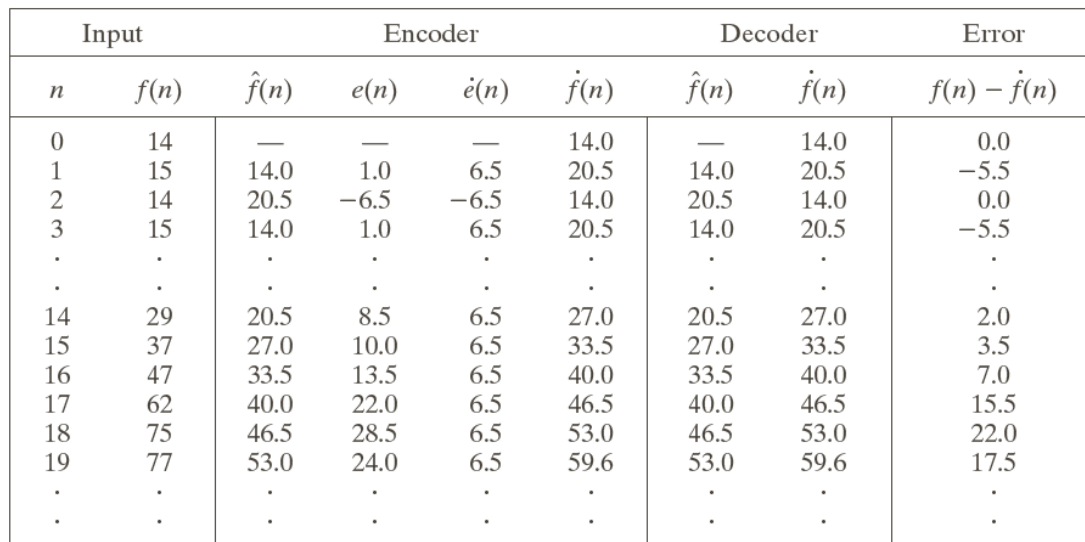
✓ 预测器：

$$\hat{f}(n) = \alpha \dot{f}(n-1)$$

✓ 量化器

$$\hat{e}(n) = \begin{cases} +\xi, & e(n) > 0 \\ -\xi, & \text{others} \end{cases}$$

■ Δ 调制



预测编码

□ 最佳预测器

- 均方误差最小的预测器
- 量化误差可以忽略
- 预测被约束为前m个样本的线性组合
- 称为差分脉冲编码调制 (DPCM)

$$E\{e^2(n)\} = E\left\{\left[f(n) - \hat{f}(n)\right]^2\right\}$$

约束条件:

$$\dot{\hat{f}}(n) = \dot{e}(n) + \hat{\dot{f}}(n) \approx e(n) + \dot{f}(n)$$
$$\hat{f}(n) = \sum_{i=1}^m \alpha_i f(n-i)$$

预测编码

- 最佳预测器设计问题转化为选择m个预测系数下式最小

$$E\{e^2(n)\} = E\left\{\left[f(n) - \sum_{i=1}^m \alpha_i f(n-i)\right]^2\right\}$$

- 在f(n)具有零均值和方差 σ^2 的假设下，求解：

$$\alpha = R^{-1}r$$

$$R = \begin{bmatrix} E\{f(n-1)f(n-1)\} & E\{f(n-1)f(n-2)\} & \text{L} & E\{f(n-1)f(n-m)\} \\ E\{f(n-2)f(n-1)\} & \text{M} & \text{L} & \text{M} \\ \text{M} & \text{M} & & \text{M} \\ \text{M} & \text{M} & & \text{M} \\ \text{M} & & & \\ E\{f(n-m)f(n-1)\} & E\{f(n-m)f(n-2)\} & \text{L} & E\{f(n-m)f(n-m)\} \end{bmatrix} \quad r = \begin{bmatrix} E\{f(n)f(n-1)\} \\ E\{f(n)f(n-2)\} \\ \text{M} \\ \text{M} \\ \text{M} \\ E\{f(n)f(n-m)\} \end{bmatrix} \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \text{M} \\ \alpha_m \end{bmatrix}$$

- 预测误差的方差：

$$\sigma_e^2 = \sigma^2 - \alpha^T r = \sigma^2 - \sum_{i=1}^m E\{f(n)f(n-i)\}\alpha_i$$

预测编码

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

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

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

d) $\hat{f}(x, y) = \begin{cases} 0.97 f(x, y - 1), & \Delta h \leq \Delta v \\ 0.97 f(x - 1, y), & \text{other} \end{cases}$

$$\Delta h = |f(x - 1, y) - f(x - 1, y - 1)|$$

$$\Delta v = |f(x, y - 1) - f(x - 1, y - 1)|$$



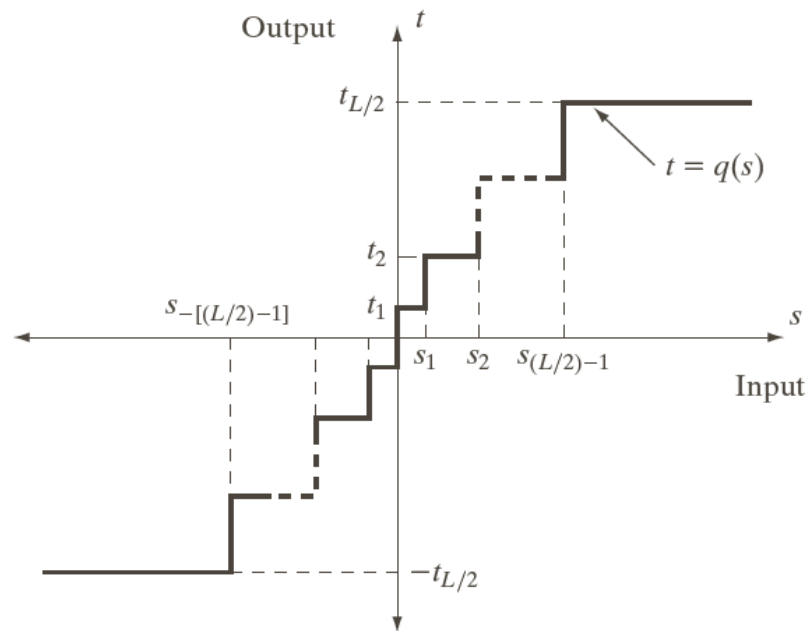
a	b
c	d

FIGURE 8.43
A comparison of
four linear
prediction
techniques.

预测编码

□ 最佳量化

■ 典型量化函数



预测编码

□ 最佳量化

■ 最小均方量化误差准则下的量化

- ✓ 输入概率密度函数 $p(s)$
- ✓ 目标 $\min E\{(s_i - t_i)^2\}$
- ✓ 最小误差的条件:

$$\int_{s_{i-1}}^{s_i} (s - t_i) p(s) ds = 0, \quad i = 1, 2, \dots, \frac{L}{2}$$

$$s_i = \begin{cases} 0, & i = 0 \\ \frac{t_i + t_{i+1}}{2}, & i = 1, 2, \dots, L/2 - 1 \\ \infty, & i = L/2 \end{cases}$$

$$s_{-i} = -s_i, \quad t_{-i} = -t_i$$

预测编码

□ 最佳量化

■ 最小均方量化误差准则下的量化

Levels	2		4		8	
i	s_i	t_i	s_i	t_i	s_i	t_i
1	∞	0.707	1.102	0.395	0.504	0.222
2			∞	1.810	1.181	0.785
3					2.285	1.576
4					∞	2.994
θ	1.414		1.087		0.731	

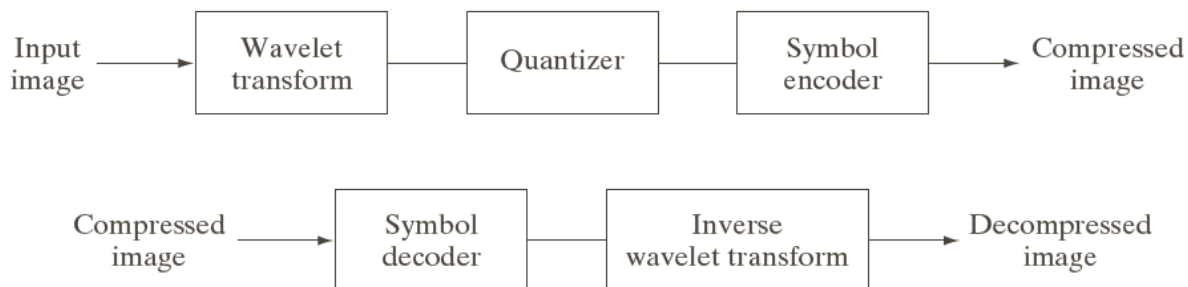
TABLE 8.12

Lloyd-Max quantizers for a Laplacian probability density function of unit variance.

小波编码

□ 概念

- 对图像像素解除相关的变换系数进行编码，比对原图像像素本身进行编码效率更高



小波编码

□ 小波基

■ 图像编码中广泛使用的小波有：

- ✓ Harr小波
- ✓ Daubechies小波
- ✓ 对称小波
- ✓ 双正交小波

TABLE 8.13

Wavelet transform filter taps and zeroed coefficients when truncating the transforms in Fig. 8.46 below 1.5.

a b
c d

FIGURE 8.46
Three-scale wavelet transforms of Fig. 8.9(a) with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies Feauveau biorthogonal wavelets.



Wavelet	Filter Taps (Scaling + Wavelet)	Zeroed Coefficients
Haar (see Ex. 7.10)	2 + 2	33.8%
Daubechies (see Fig. 7.8)	8 + 8	40.9%
Symlet (see Fig. 7.26)	8 + 8	41.2%
Biorthogonal (see Fig. 7.39)	17 + 11	42.1%

小波编码

□ 分解级别选择

- 变换分解级别的数量是影响计算复杂性和重建误差的重要因素
- 正反变换计算操作次数随分解级数的增加而增加

Decomposition Level (Scales or Filter Bank Iterations)	Approximation Coefficient Image	Truncated Coefficients (%)	Reconstruction Error (rms)
1	256 × 256	74.7%	3.27
2	128 × 128	91.7%	4.23
3	64 × 64	95.1%	4.54
4	32 × 32	95.6%	4.61
5	16 × 16	95.5%	4.63

TABLE 8.14
Decomposition
level impact on
wavelet coding
the 512 × 512
image of
Fig. 8.9(a).

□ 量化器设计

- 系数量化是影响小波编码压缩和重建误差的最重要因素
- 进一步改进
 - 引入以零为中心的较大量化间隔，称为死区
 - 从一个尺度到另一个尺度自适应调制量化间隔的大小

小波编码

□ 量化器设计

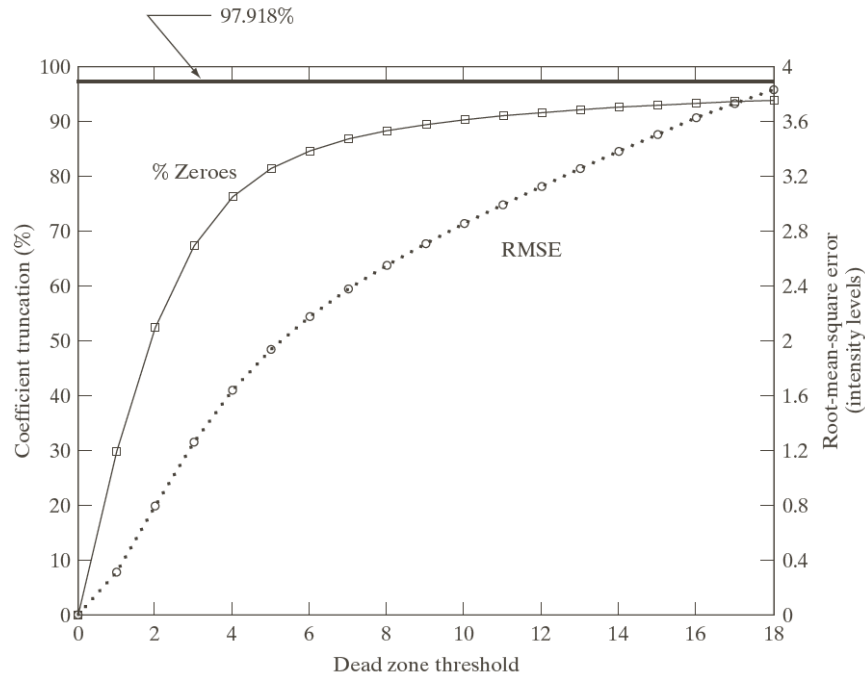


FIGURE 8.47 The impact of dead zone interval selection on wavelet coding.

死区尺寸增大时，被截断系数的数量同样增大

小波编码

□ JPEG2000

- JPEG2000扩充了JPEG标准，在连续色调静止图像压缩和压缩数据的访问提供了更多灵活性
- JPEG2000编码步骤
 - 对编码的图像样本进行直流电平平移和去相关性
 - 将各个分量分成多个像块，即被单独处理的像素的矩阵阵列
 - 计算每个像块分量的行和列的一维离散小波变换
 - 无损压缩以双正交、5-3系数尺度小波向量为基础
 - 有损压缩中，采用9-7系数尺度-小波向量

$$Y(2n+1) = X(2n+1) + \alpha[X(2n) + X(2n+2)], i_0 - 3 \leq 2n+1 < i_1 + 3$$

$$Y(2n) = X(2n) + \beta[Y(2n-1) + Y(2n+1)], i_0 - 2 \leq 2n < i_1 + 2$$

$$Y(2n+1) = Y(2n+1) + \gamma[Y(2n) + Y(2n+2)], i_0 - 1 \leq 2n+1 < i_1 + 1$$

$$Y(2n) = Y(2n) + \delta[Y(2n-1) + Y(2n+1)], i_0 \leq 2n < i_1$$

$$Y(2n+1) = -K g Y(2n+1), i_0 \leq 2n+1 < i_1$$

$$Y(2n) = Y(2n) / K, i_0 \leq 2n < i_1$$

提升和尺度操作



小波编码

□ JPEG2000

□ JPEG2000编码步骤

- 计算每个像块分量的行和列的一维离散小波变换

Filter Tap	Highpass Wavelet Coefficient	Lowpass Scaling Coefficient
0	-1.115087052456994	0.6029490182363579
±1	0.5912717631142470	0.2668641184428723
±2	0.05754352622849957	-0.07822326652898785
±3	-0.09127176311424948	-0.01686411844287495
±4	0	0.02674875741080976

TABLE 8.15

Impulse responses of the low- and highpass analysis filters for an irreversible 9-7 wavelet transform.

小波编码

□ JPEG2000

□ JPEG2000编码步骤

- 变换生成四个子带，分别是像块分量的低分辨率近似及分量的水平、垂直和对角频率特征。将后续迭代限制为前面的分解的近似系数，把该变换重复N次，可得到N尺度小波变换

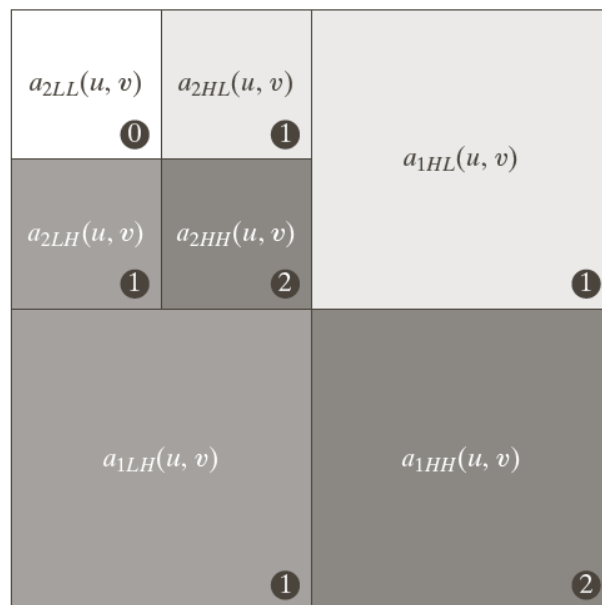


FIGURE 8.48
JPEG 2000
two-scale wavelet
transform
tile-component
coefficient
notation and
analysis gain.

小波编码

□ JPEG2000

□ JPEG2000编码步骤

- 量化, 子带b的系数 $a_b(u, v)$ 量化为 $q_b(u, v)$

$$q_b(u, v) = \text{sign}[a_b(u, v)] \text{floor} \left[\frac{|a_b(u, v)|}{\Delta_b} \right]$$

$$\Delta_b = 2^{R_b - g_b} \left(1 + \frac{\mu_b}{2^{11}} \right)$$

- 每个像块分量的子带系数被排列为称为码块的矩形块, 码块被单独编码, 一次一个比特平面

小波编码

□ JPEG2000



FIGURE 8.49 Four JPEG-2000 approximations of Fig. 8.9(a). Each row contains a result after compression and reconstruction, the scaled difference between the result and the original image, and a zoomed portion of the reconstructed image. (Compare the results in rows 1 and 2 with the JPEG results in Fig. 8.32.)

数字图像水印

- 数字图像水印处理是把数据插入到一幅图像中的过程，从而形成对图像的保护。
- 简单的可见水印

$$f_w = (1 - \alpha)f + \alpha w$$

其中f为原图，w为水印



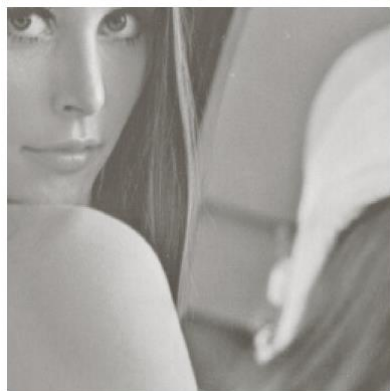
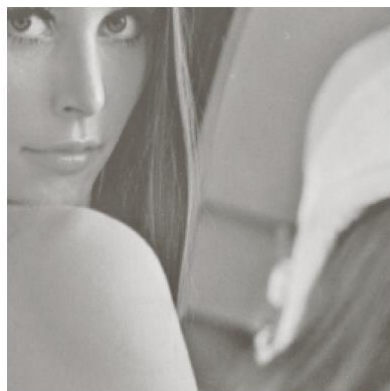
a
b c

FIGURE 8.50
A simple visible watermark:
(a) watermark;
(b) the watermarked image; and (c) the difference between the watermarked image and the original (non-watermarked) image.

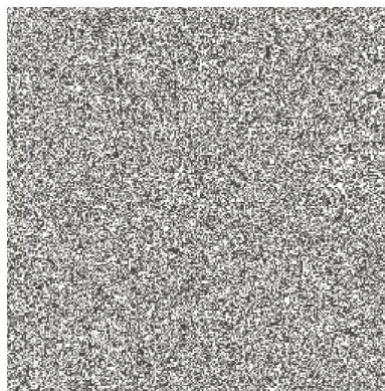
数字图像水印

□ 不可见水印

- 不可见水印裸眼是看不见的
- 使用合适的解码算法可以恢复原图
- 不可见水印是通过作为可见的冗余信息插入来保证的



Digital Image
Processing

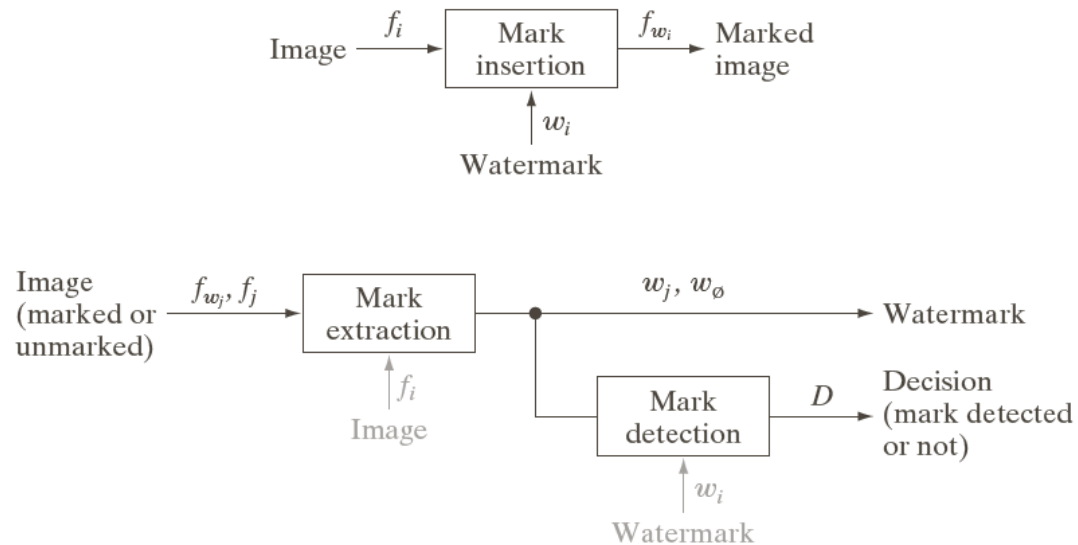


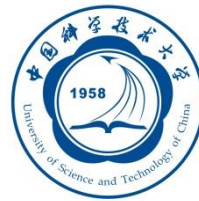
a b
c d

FIGURE 8.51 A simple invisible watermark: (a) watermarked image; (b) the extracted watermark; (c) the watermarked image after high quality JPEG compression and decompression; and (d) the extracted watermark from (c).

数字图像水印

□ 典型的图像水印处理系统





数字图像水印

□ 基于DCT的不可见的鲁棒水印

- 计算被加入水印图像的二维DCT
- 按幅值定位它的K个最大系数 c_1, c_2, \dots, c_k .
- 通过生成一个K元素伪随机数序列 w_1, w_2, \dots, w_k 创建一个水印, 这些伪随机数取自 $\mu = 0, \sigma^2 = 1$ 的高斯分布
- 使用下式将步骤3得到的水印嵌入步骤2的K个最大DCT系数中

$$c'_i = c_i g(1 + \alpha w_i), 1 \leq i \leq K$$

- 计算步骤4结果的反DCT

数字图像水印

□ 基于DCT的不可见的鲁棒水印



a b
c d

FIGURE 8.53 (a) and (c) Two watermarked versions of Fig. 8.9(a); (b) and (d) the differences (scaled in intensity) between the watermarked versions and the unmarked image. These two images show the intensity contribution (although scaled dramatically) of the pseudo-random watermarks on the original image.

数字图像水印

- 基于DCT的不可见的鲁棒水印的判断
- 确定一幅特点图像是否是一幅事先加水印图像的副本，该图像具有水印 w_1, w_2, \dots, w_k 和DCT系数 c_1, c_2, \dots, c_k .
 - 计算问题中图像的二维DCT
 - 提取K个DCT系数，记为 $\hat{c}_1, \hat{c}_2, \dots, \hat{c}_K$ 。若问题图像是事先加过水印的图像（未被修改），则对 $1 \leq i \leq K$ 有 $\hat{c}_i = c'_i$ 。否则若它是加过水印图像的修改副本，则 $\hat{c}_i \approx c'_i$
 - 使用下式计算 $\hat{\omega}_1, \hat{\omega}_2, \dots, \hat{\omega}_K$

$$\hat{\omega}_i = \frac{\hat{c}_i - c_i}{\alpha c_i}, 1 \leq i \leq K$$

- 使用相关系数度量 $\hat{\omega}_1, \hat{\omega}_2, \dots, \hat{\omega}_K$ 和 $\omega_1, \omega_2, \dots, \omega_K$ 的相似度

$$\gamma = \frac{\sum_{i=1}^K (\bar{\hat{\omega}} - \bar{\hat{\omega}})(\bar{\omega}_i - \bar{\omega})}{\sqrt{\sum_{i=1}^K (\bar{\hat{\omega}} - \bar{\hat{\omega}})^2} \sqrt{\sum_{i=1}^K (\bar{\omega}_i - \bar{\omega})^2}}, 1 \leq i \leq K$$

数字图像水印

- 确定一幅特点图像是否是一幅事先加水印图像的副本，该图像具有水印 $\omega_1, \omega_2, \dots, \omega_K$ 和DCT系数 c_1, c_2, \dots, c_k .
- 将度量相似度 γ 和一个预定义的阈值 T 进行比较，进行二值检测判决。D=1代表水印 $\omega_1, \omega_2, \dots, \omega_K$ 存在， D=0代表水印 $\omega_1, \omega_2, \dots, \omega_K$ 不存在

$$D = \begin{cases} 1, & \gamma \geq T \\ 0, & \text{other} \end{cases}$$

数字图像水印



FIGURE 8.54 Attacks on the watermarked image in Fig. 8.53(a): (a) lossy JPEG compression and decompression with an rms error of 7 intensity levels; (b) lossy JPEG compression and decompression with an rms error of 10 intensity levels (note the blocking artifact); (c) smoothing by spatial filtering; (d) the addition of Gaussian noise; (e) histogram equalization; and (f) rotation. Each image is a modified version of the watermarked image in Fig. 8.53(a). After modification, they retain their watermarks to varying degrees, as indicated by the correlation coefficients below each image.



第8章 图像压缩

Thanks