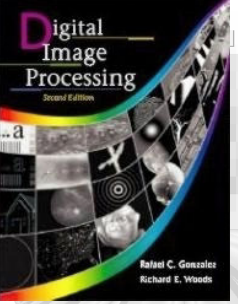
Image Compression

- Outline
  - – Fundamentals
    - Redundancies
  - – Image compression models
  - – Information theory impact
    - Entropy
    - Coding theorems
  - – Loss-less compression
    - Variable-length coding
    - Bit-place coding
    - Predictive coding
  - – Lossy compression
    - Predictive coding
    - Transform coding
  - – Compression standards

Fundamentals

- **Data compression**: process of reducing the amount of *data* required to represent a given quantity of *information*.
- Data and information
  - Data conveys information
- Data redundancy
  - Given two data sets with numbers of information-carrying units n_1 and n_2, correspondingly.
  - Compression ratio: $C_R = n_1/n_2$
  - Relative data redundancy: $R_D = 1 - 1/C_R$
- Three types of data redundancy
  - Coding redundancy
  - Interpixel redundandancy
  - Psychovisual redundancy



Fundamentals: Coding Redundancy

- Histogram analysis of grey-scale images
  - Given discrete random variable $r_k \in [0,1]$ representing L grey levels; k=0,1,2,…L – 1; each r_k occurs with probability $p_r(r_k)$, i.e. $p_r(r_k) = n_k / n$, where n_k is the number of times that k-th grey level appears and n is the total number of pixels.
  - Given $l(r_k)$ is the number of bits to represent each value of r_k. Average number of bits required to represent each pixel is

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

  - Total number of bits to code an $M \times N$ image is MNL_{avg}
  - Example: 8-level image coded by 3-bit binary code and a variable-length code

$$L^1_{avg} = \sum_{k=0}^{7} 3 p_r(r_k) = 3$$

$$L^2_{avg} = \sum_{k=0}^{7} l(r_k) p_r(r_k) = 2.7$$

$$C_R = 3/2.7 = 1.11$$

$$R_D = 1 - 1/1.11 = 0.099$$

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

TABLE 8.1 Example of variable-length coding.

Fundamentals: Coding Redundancy

- The fundamental basis of data compression through *variable-length coding*: functions $p_r(r_k)$ and $l(r_k)$ are inversely proportional.
- Code redundancy occurs when the code does not minimise the average length (not full advantage of probabilities is taken)

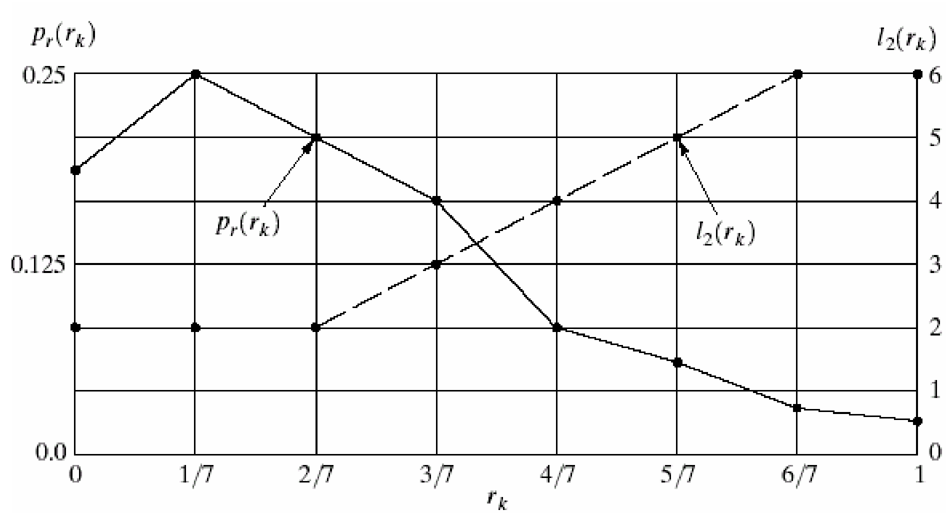


FIGURE 8.1 Graphic representation of the fundamental basis of data compression through variable-length coding.

Fundamentals: Interpixel Redundancy

- Related to the interpixel correlations within an image
    - spatial redundancy, geometric redundancy, interframe redundancy
- Measured through autocorrelation coefficients

$$\gamma(\Delta n) = A(\Delta n) / A(0)$$

where

$$A(\Delta n) = \frac{1}{N - \Delta n} \sum_{y=0}^{N-1-\Delta n} f(x, y) f(x, y + \Delta n)$$

- Transformations that remove interpixel redundancy are called *mappings*

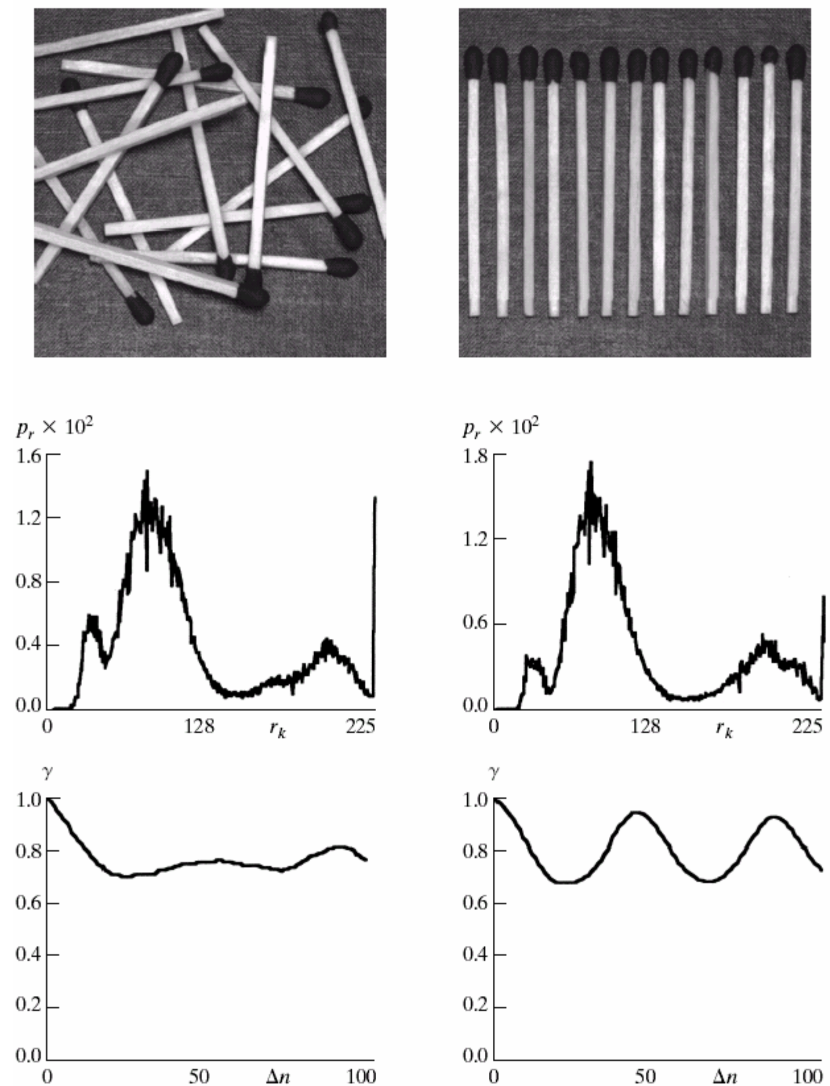


FIGURE 8.2 Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.

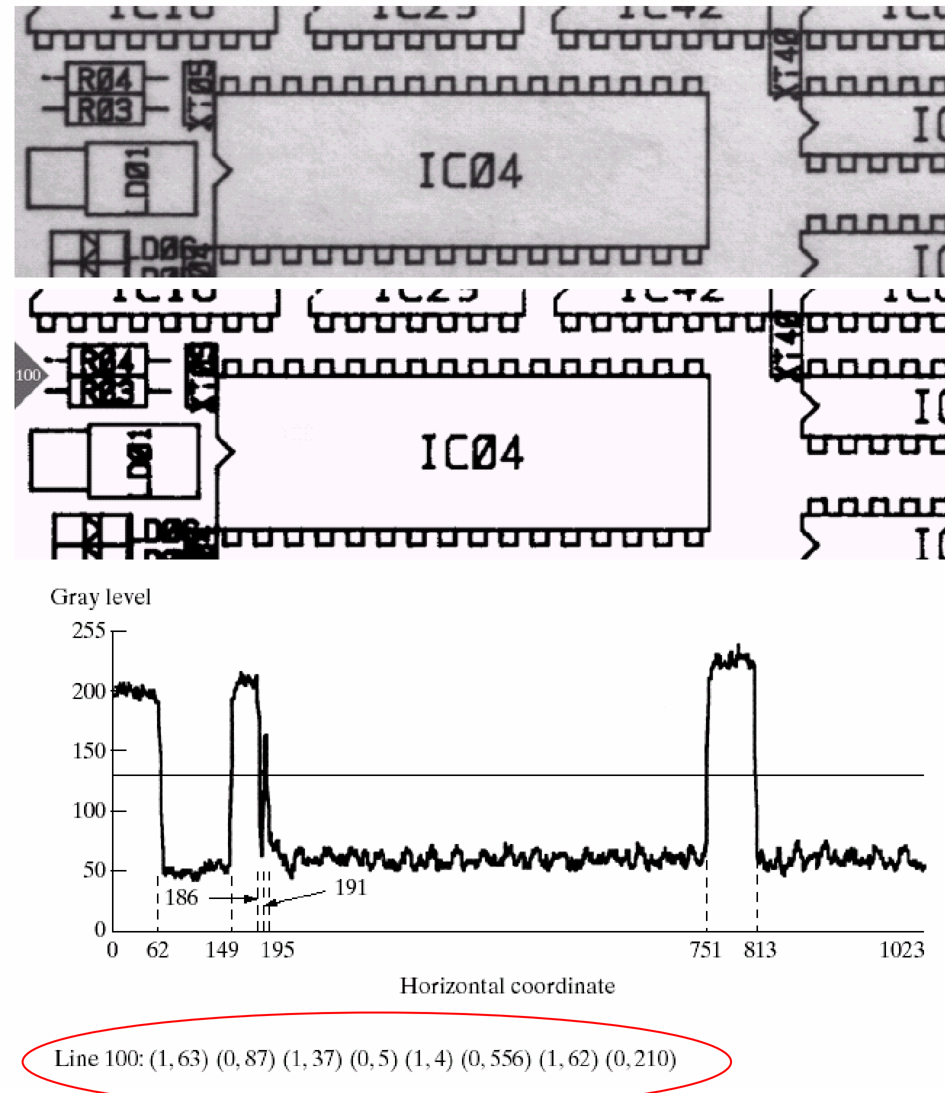
Fundamentals: Interpixel Redundancy

Example of a non-visual mapping:

- 1024x343 grey-scale image transformed to a binary image
- Each line is represented by a sequence of pairs (g_i, w_i), where g_i is the i-th level and w_i is the number of successive sample of this level (run length).
- For line 100 only 88 bits are needed to represent the 1024 bits of binary data
- For the whole image 12166 runs are sufficient
- 11 bits represent each run-length pair

$$C_R = \frac{1024.343.1}{12166.11} = 2.63$$

$$R_D = 1 - 1/2.63 = 0.62$$



a b c d

FIGURE 8.3 Illustration of run-length coding: (a) original image. (b) Binary image with line 100 marked. (c) Line profile and binarization threshold. (d) Run-length code.

Fundamentals: Psycho-visual Redundancy

- Eye does not respond with equal sensitivity to all visual information. Certain information simply has less relative importance. This information is said to be *psycho-visually redundant*. It can be eliminated without significantly imparing the quality of image perception.
- Some knowledge about how the brain recognizes pictures is needed
- Psycho-visual redundancy differs from the previous two. It is associated with real visual information. Its elimination results in a loss of quantitative information and it is referred to as *quantization*. It is an irreversible operation and leads to *lossy* data compression.
- Example: *improved grey-scale (IGS) quantization*.

Pixel	Gray Level	Sum	IGS Code
$i-1$	N/A	0000 0000	N/A
i	0110 1100	0110 1100	0110
$i+1$	1000 1011	1001 0111	1001
$i+2$	1000 0111	1000 1110	1000
$i+3$	1111 0100	1111 0100	1111

TABLE 8.2 IGS quantization procedure.

Fundamentals: Psycho-visual Redundancy

a b c

FIGURE 8.4 (a) Original image. (b) Uniform quantization to 16 levels. (c) IGS quantization to 16 levels.

Fundamentals: Fidelity Criteria

- Objective and subjective fidelity criteria to assess the information loss
- Root-mean-square error or Mean-square signal-to-noise ratio
- $$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - \hat{f}(x,y)]^2 \right]^{1/2}$$ $$SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y)]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x,y) - \hat{f}(x,y)]^2}$$

- Rating scales (see Table 8.3) or Side-by-side comparisons

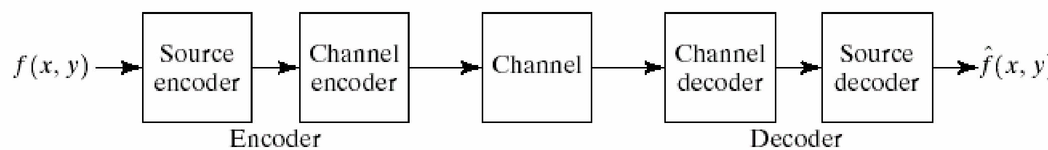
{-3, -2, -1, 0, 1, 2, 3}≡{*much worse, worse, slightly worse, the same, slightly better, better, much better*}

TABLE 8.3 Rating scale of the Television Allocations Study Organization. (Frendendall and Behrend.)

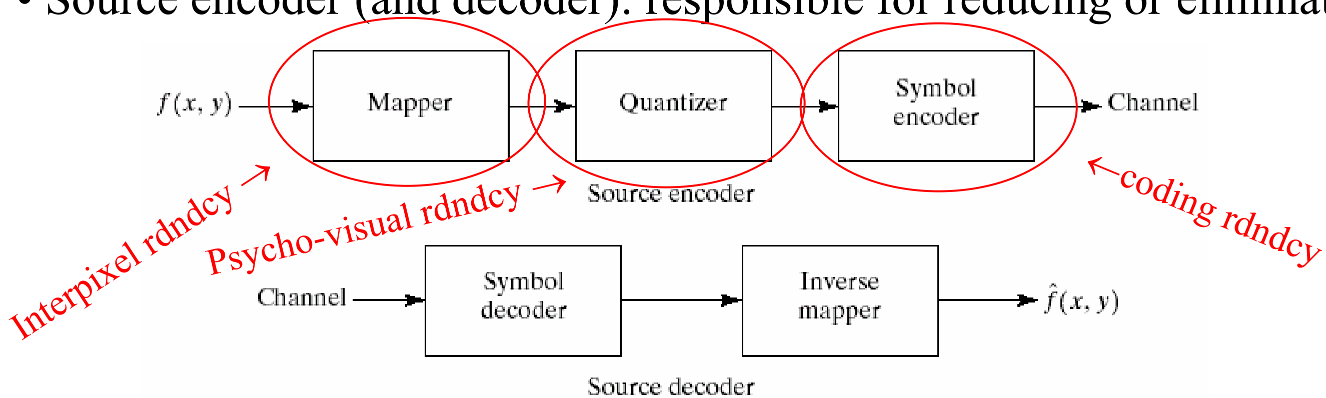
Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor 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.

Image Compression Models

- A general compression system consists of an encoder and a decoder.



- Source encoder (and decoder): responsible for reducing or eliminating redundancies.



- Channel encoder (and decoder): reduces the impact of channel noise by 'controlled redundancy'. Example: *Hamming* (7,4) *code*

given a 4-bit number $b_3b_2b_1b_0$

$h_1 = b_3 \oplus b_2 \oplus b_0$ $\quad h_3 = b_3$

$h_2 = b_3 \oplus b_1 \oplus b_0$ $\quad h_5 = b_2$

$h_4 = b_2 \oplus b_1 \oplus b_0$ $\quad h_6 = b_1$

parity check by a nonzero word $c_4c_2c_1$

$c_1 = h_1 \oplus h_3 \oplus h_5 \oplus h_7$

$c_2 = h_3 \oplus h_3 \oplus h_6 \oplus h_7$

$c_1 = h_4 \oplus h_5 \oplus h_6 \oplus h_7$

the corrected code word $h_3h_5h_6h_7$

Elements of Information Theory

Measuring information

The generation of information is modeled as a probabilistic process. Random event E occurs with probability $P(E)$

$$I(E) = \log \frac{1}{P(E)} = -\log P(E)$$

The base of the logarithm determines the units used to measure the information. If the base 2 is selected the resulting information unit is called bit. If $P(E)$=0.5 (two possible equally likely events) the information is one bit.

Elements of Information Theory

- The information channel: the physical medium that links the source to the user (tel. line, wire, electromagnetic waves, etc.)
- The source
  - Modeled as a discrete random variable
  - Source alphabet $A=\{a_j\}$
  - Symbols (letters) a_j with probabilities $P(a_j)$
  - The average self-information obtained from k outputs is

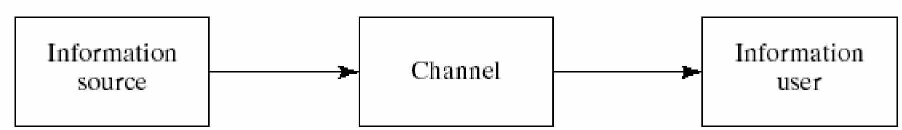
$$-kP(a_1)\log P(a_1) - kP(a_2)\log P(a_2) - \ldots - kP(a_J)\log P(a_J) =$$

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

  - The average information per source output is

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

  - (*uncertainty* or *entropy*)



Ensemble $(A, \mathbf{z})$
$A = \{a_j\}$
$\mathbf{z} = [P(a_1), P(a_2), \ldots, P(a_J)]^T$

$Q = [q_{kj}]$

Ensemble $(B, \mathbf{v})$
$B = \{b_k\}$
$\mathbf{v} = [P(b_1), P(b_2), \ldots, P(b_K)]^T$

Elements of Information Theory

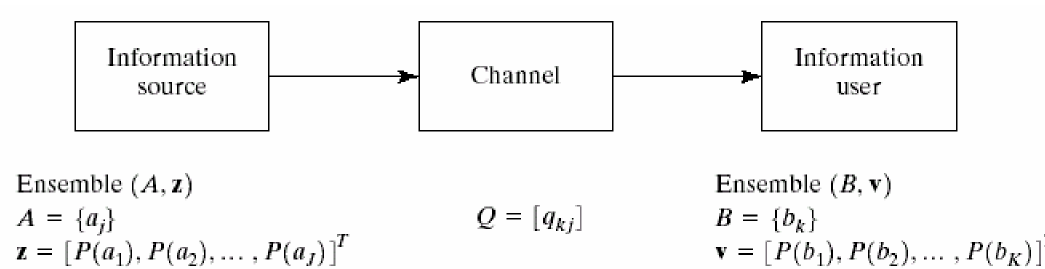
- The channel
  - The information in the output of the channel is a discrete random variable
  - Channel alphabet $B=\{b_k\}$ with letters b_k with probabilities $P(b_k)$
  - The probabilities are related to the source **z** by conditional probabilities

$$P(b_k) = \sum_{j=1}^{J} P(b_k \mid a_j) \log P(a_j)$$

  - Channel (transition) matrix

$$\mathbf{Q} = \begin{bmatrix} P(b_1 \mid a_1) & P(b_1 \mid a_2) & \dots & P(b_1 \mid a_J) \\ P(b_2 \mid a_1) & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ P(b_K \mid a_1) & P(b_K \mid a_2) & \dots & P(b_K \mid a_J) \end{bmatrix}$$

$$\mathbf{Q} = [q_{kj}], \text{ where } q_{kj} = P(b_k \mid a_j)$$



Elements of Information Theory

- The capacity of the channel
  - The conditional entropy function for each b_k

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

  - The averaged value over all b_k (*equivocation*)

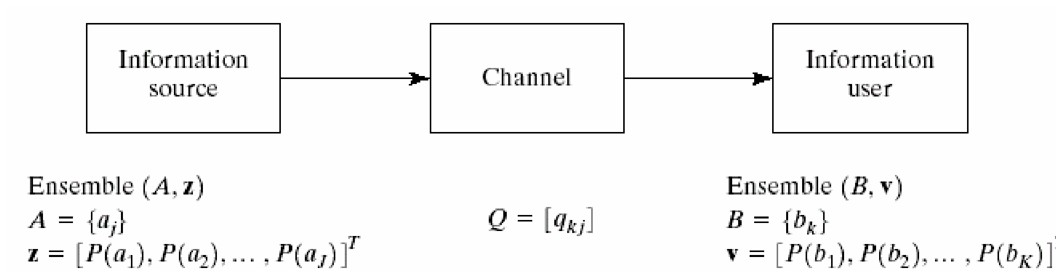
$$H(\mathbf{z} \mid \mathbf{v}) = \sum_{k=1}^{K} H(\mathbf{z} \mid b_k) P(b_k) = -\sum_{j=1}^{J} \sum_{k=1}^{K} P(a_j, b_k) \log P(a_j \mid b_k)$$

  - The mutual information of **z** and **v**

$$I(\mathbf{z}, \mathbf{v}) = H(\mathbf{z}) - H(\mathbf{z} \mid \mathbf{v}) = \sum_{j=1}^{J} \sum_{k=1}^{K} P(a_j, b_k) \log \frac{P(a_j, b_k)}{P(a_j) P(b_k)} = \sum_{j=1}^{J} \sum_{k=1}^{K} P(a_j) q_{kj} \log \frac{q_{kj}}{\sum_{j=1}^{J} P(a_j) q_{kj}}$$

  - The capacity

$$C = \max_{\mathbf{z}} [I(\mathbf{z}, \mathbf{v})]$$



Fundamental Coding Theorems

- The noiseless coding theorem: how to compact information as much as possible
  - Consider a source of information with finite ensemble $(A, \mathbf{z})$ and statistically independent source symbols: so-called *zero-memory source*
  - Non-extended source produces one symbol and the n-th extension produces block random variable of n symbols. Its entropy is n times the entropy of the single symbol source
  - The source output is an n-tuple of symbols
  - The final result is $H(\mathbf{z}) \leq \frac{L'_{avg}}{n} < H(\mathbf{z}) + \frac{1}{n}$ where L'_{avg} is the average word length of the code corresponding to the nth extension
  - The efficiency of any encoding strategy is

$$\eta = n \frac{H(\mathbf{z})}{L'_{avg}}$$

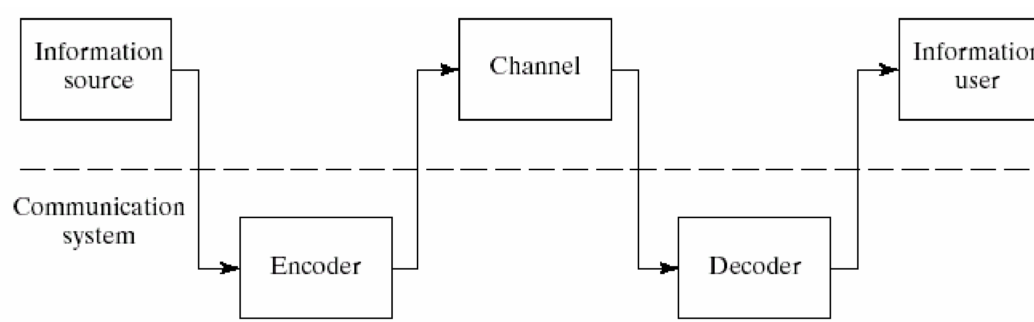
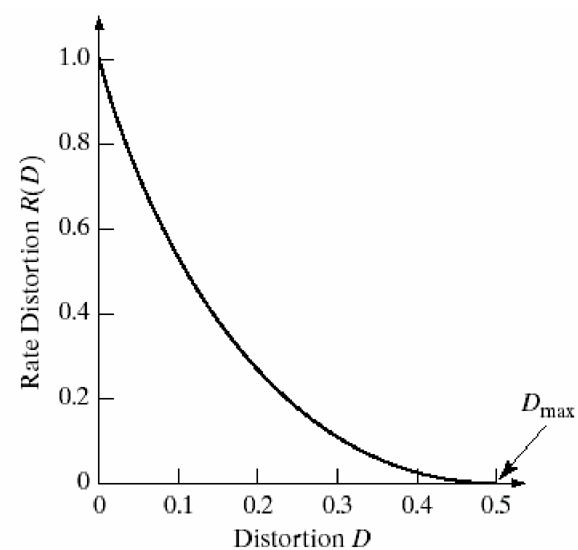


FIGURE 8.9 A communication system model.

Fundamental Coding Theorems

- The noisy coding theorem: how to ensure a reliable communication
  - The simplest way: by a repetitive messaging
  - Code of size φ and block length r has rate $R = \log(\varphi/r)$
  - For any rate $R <$ C, the capacity of the channel, there exist an integer r and code of block length r and rate R such that the probability of block decoding error is less than (any) $\varepsilon > 0$.
- The source coding theorem: the channel is error-free but the encoding process is lossy, hence a rate-distortion compromise



Error-free Compression

- Variable-length coding reduces only coding redundancy by assigning the shortest possible code words to the most probable (grey) levels.
- Huffman coding (codes one symbol at a time)
  - Order the probabilities of the symbols and combine the lowest probability symbols into a single symbol

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

  - Code each reduced source working back to the original source.

Original source			Source reduction							
Sym.	Prob.	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								

$L_{avg} = 2.2$ bits/symbol

$E = 2.14$ bits/symbol

code eff. $= 0.973$

Error-free Compression

- For a large number of symbols to be coded Huffman coding is computationally complex. Other codes provide a trade-off between coding efficiency and simplicity.
- Truncated Huffman coding: codes with Huffman code only the most probable symbols. All others are coded by prefix and fixed-length code
- B-code: made by continuation bits and information bits
- Shift codes:
    - Arranging the symbols by their (monoliticaly decreasing) probabilities
    - dividing the total number of symbols into equally sized blocks
    - coding the individual elements within all blocks identically
    - adding shift-up and/or shift-down symbols to identify each block.

Error-free Compression

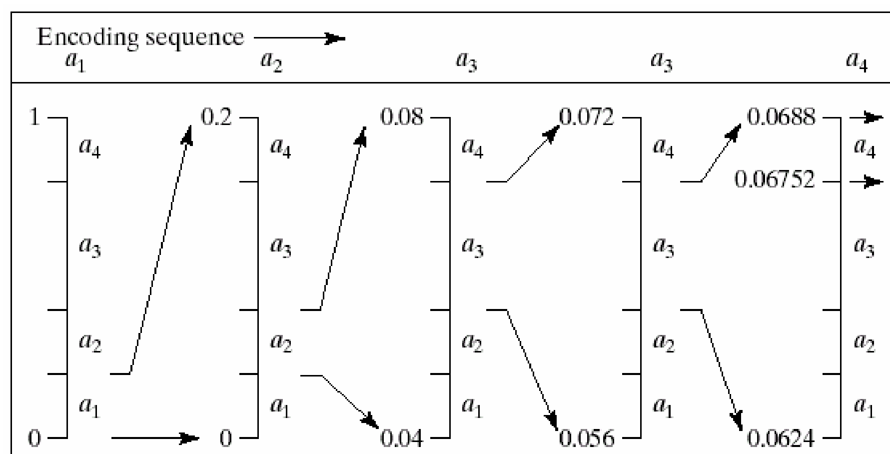
TABLE 8.5 Variable-length codes.

Source symbol	Probability	Binary Code	Huffman	Truncated Huffman	B_2-Code	Binary Shift	Huffman Shift
Block 1							
a_1	0.2	00000	10	11	C00	000	10
a_2	0.1	00001	110	011	C01	001	11
a_3	0.1	00010	111	0000	C10	010	110
a_4	0.06	00011	0101	0101	C11	011	100
a_5	0.05	00100	00000	00010	C00C00	100	101
a_6	0.05	00101	00001	00011	C00C01	101	1110
a_7	0.05	00110	00010	00100	C00C10	110	1111
Block 2							
a_8	0.04	00111	00011	00101	C00C11	111 000	00 10
a_9	0.04	01000	00110	00110	C01C00	111 001	00 11
a_{10}	0.04	01001	00111	00111	C01C01	111 010	00 110
a_{11}	0.04	01010	00100	01000	C01C10	111 011	00 100
a_{12}	0.03	01011	01001	01001	C01C11	111 100	00 101
a_{13}	0.03	01100	01110	100000	C10C00	111 101	00 1110
a_{14}	0.03	01101	01111	100001	C10C01	111 110	00 1111
Block 3							
a_{15}	0.03	01110	01100	100010	C10C10	111 111 000	00 00 10
a_{16}	0.02	01111	010000	100011	C10C11	111 111 001	00 00 11
a_{17}	0.02	10000	010001	100100	C11C00	111 111 010	00 00 110
a_{18}	0.02	10001	001010	100101	C11C01	111 111 011	00 00 100
a_{19}	0.02	10010	001011	100110	C11C10	111 111 100	00 00 101
a_{20}	0.02	10011	011010	100111	C11C11	111 111 101	00 00 1110
a_{21}	0.01	10100	011011	101000	C00C00C00	111 111 110	00 00 1111
Entropy	4.0						
Average length		5.0	4.05	4.24	4.65	4.59	4.13

Error-free Compression

- Arithmetic coding
  - An entire sequence of source symbols (a message) is assigned a single arithmetic code word
  - The code word defines an interval of real numbers between 0 and 1
  - Each symbol reduces the size of the interval in accordance with its probability.
  - Example: five-symbol message from a four-symbol source is being coded

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)



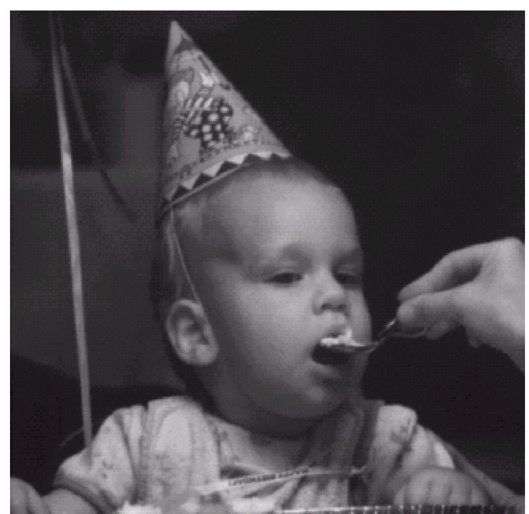
Error-free Compression: Bit-plane Coding

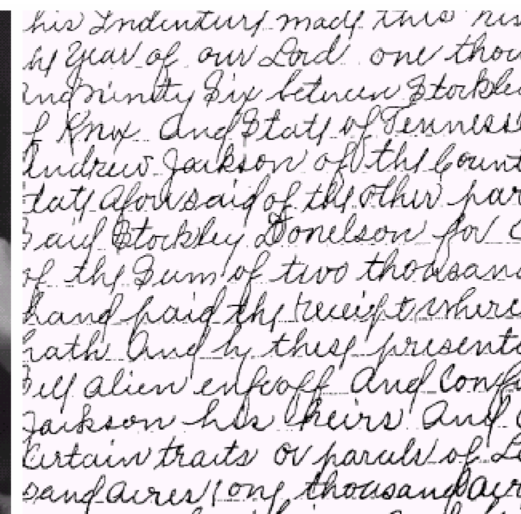
- Bit-plane decomposition: $a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \cdots + a_1 2^1 + a_0 2^0$
  - Direct separation into m 1-bit planes
  - Gray code (successive code words differ in only one bit position)

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

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

Example: two 1024×1024 images: one monochrome a one binary





Error-free Compression: Bit-plane Coding

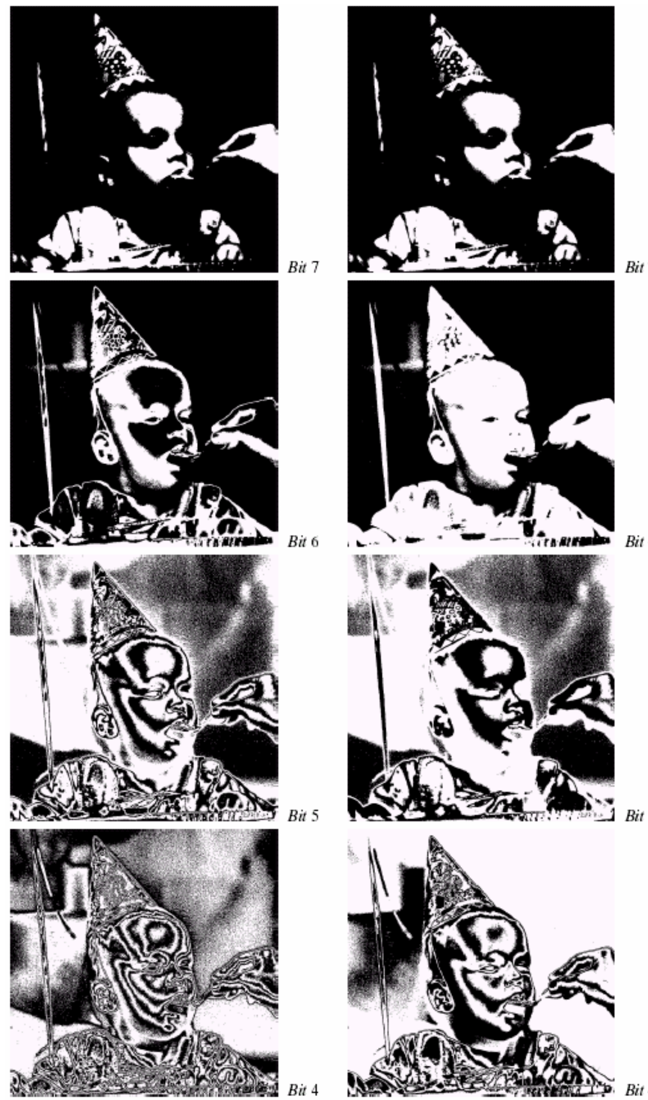


FIGURE 8.15 The four most significant binary (left column) and Gray-coded (right column) bit planes of the image in Fig. 8.14(a).

Error-free Compression: Bit-plane Coding

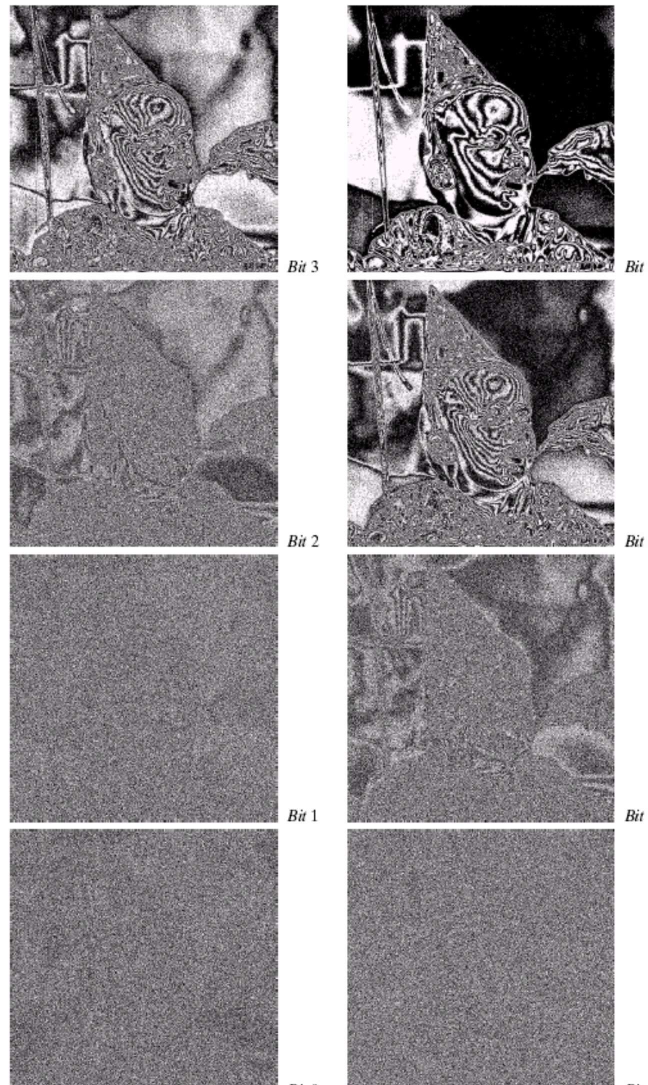
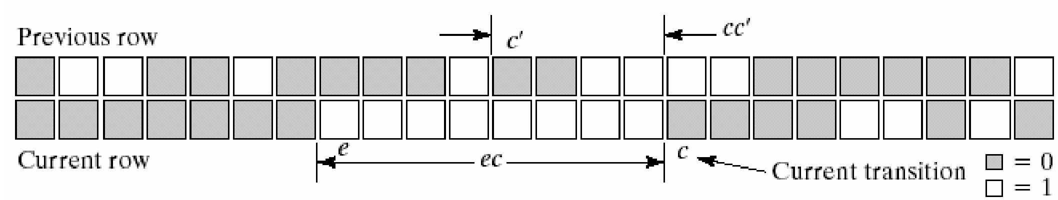


FIGURE 8.16 The four least significant binary (left column) and Gray-coded (right column) bit planes of the image in Fig. 8.14(a).

Error-free Compression: Bit-plane Coding

- Constant Area Coding (CAS)
  - The image is divided into blocks of size p×q, classified as white, black or mixed; the most probable is assigned 0, the other two categories are assigned 10 and 11.
  - White block skipping (WBS): white blocks are coded by 0 and all others by a 1 followed by the bit pattern of the block
- One-dimensional run-length coding
  - Represent (row-wise) sucsessive runs of black or white pixels
  - The run-lengths can be coded additionally by a variable-length code
- Two-dimensional run-length coding



Distance measured	Distance	Code	Distance range	Code $h(d)$
cc'	0	0	1 – 4	0 xx
ec or cc' (left)	1	100	5 – 20	10 xxxx
cc' (right)	1	101	21 – 84	110 xxxxxx
ec	$d(d>1)$	111 $h(d)$	85 – 340	1110 xxxxxxxx
cc' (c' to left)	$d(d>1)$	1100 $h(d)$	341 – 364	11110 xxxxxxxxxx
cc' (c' to right)	$d(d>1)$	1101 $h(d)$	1365 – 5460	111110 xxxxxxxxxxxx

Lossless Predictive Coding

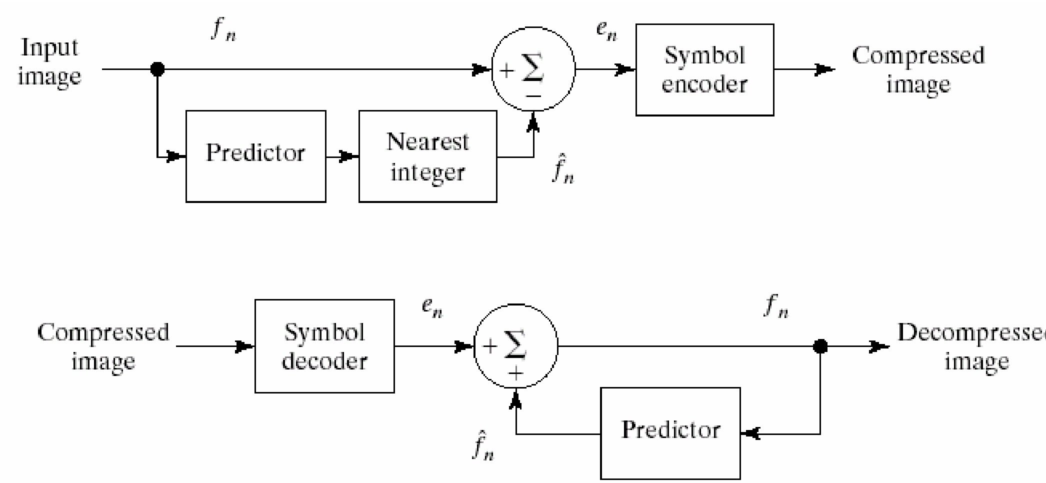
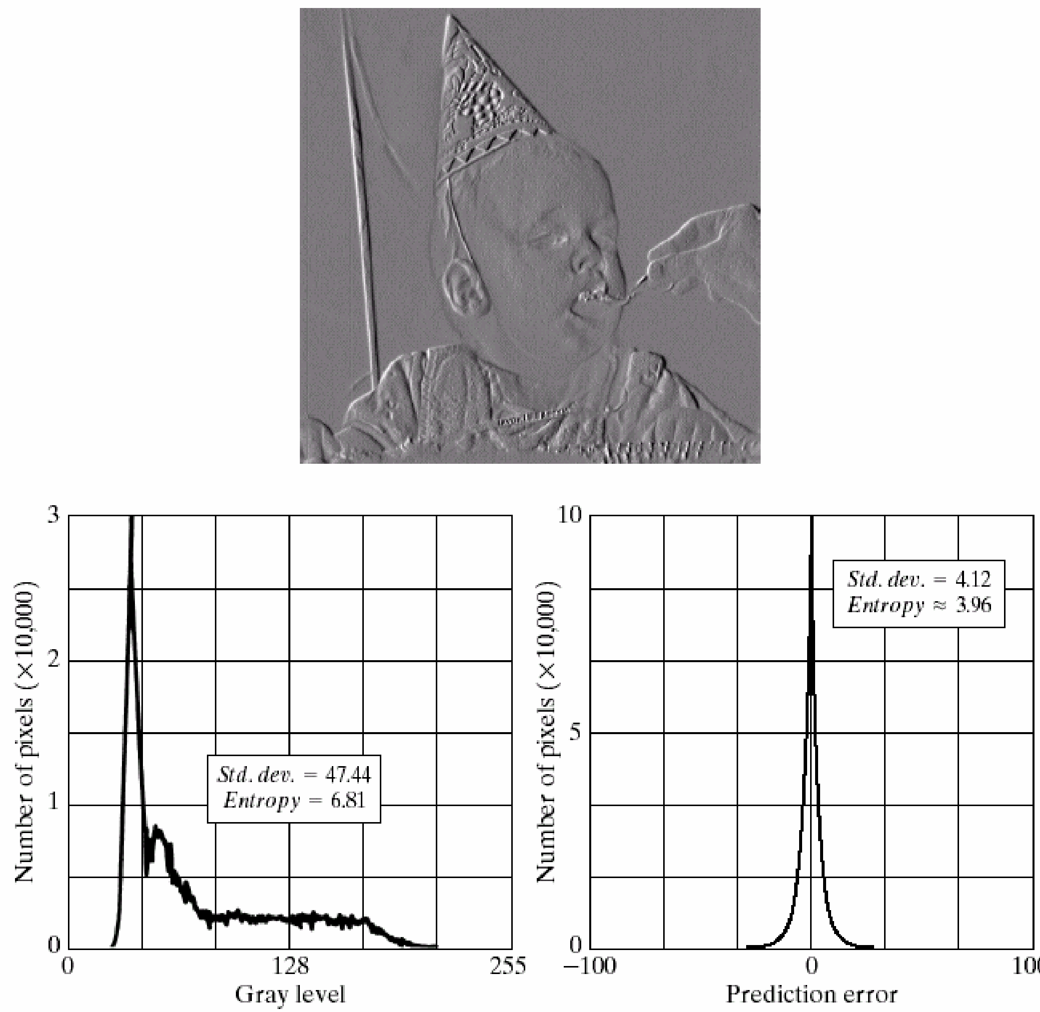


FIGURE 8.19 A lossless predictive coding model: (a) encoder; (b) decoder.

Lossless Predictive Coding



a
b c

FIGURE 8.20 (a) The prediction error image resulting from Eq. (8.4-9). (b) Gray-level histogram of the original image. (c) Histogram of the prediction error.

Lossy Compression

- Lossy encoding is based on the concept of compromising the accuracy of the reconstructed image in exchange for increased compression
- Quantization block is an essential part of the encoder
- Lossy predictive coding
  - The quantizer maps the prediction error into a limited range of outputs
  - The predictor is placed in the feedback loop, thus achieving the same prediction block in the decoder

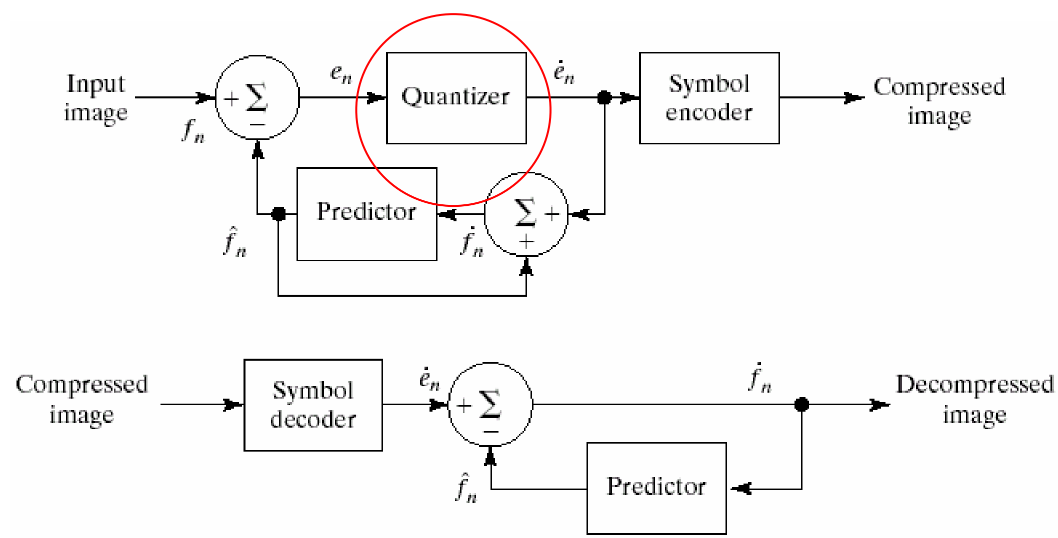
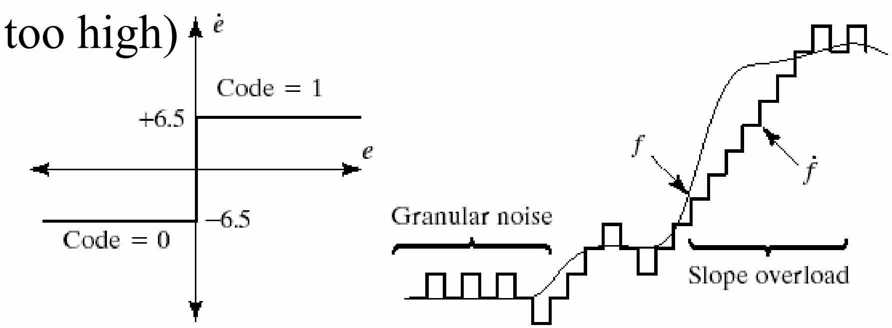


FIGURE 8.21 A lossy predictive coding model: (a) encoder and (b) decoder.

Lossy Compression

- Delta modulation: an one-bit per pixel representation $\hat{f}_n = \alpha \dot{f}_{n-1}$ $\quad \dot{e}_n = \begin{cases} +\zeta & \text{for } e_n > 0 \\ -\zeta & \text{otherwise} \end{cases}$
- Example for $\alpha = 1, \quad \zeta = 6.5$
  - Effects such as *granular noise* (α too high) or *slope overload* (α too small)



Input		Encoder				Decoder		Error
n	f	$\hat{f}$	e	$\dot{e}$	$\dot{f}$	$\hat{f}$	$\dot{f}$	$[f - \dot{f}]$
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
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Lossy Compression

- *Differential pulse code modulation*: an optimal in mean-square sense predictor.

- Constraints:
  - quantization error assumed to be negligible
  - the prediction is a linear combination of m previous pixels.

- Computation of the autocorrelation matrix is difficult in practice. Instead, a model (e.g. 2-D Markov source) is used

- The sum of the prediction coefficients should be smaller than 1 to ensure the same output grey levels

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

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

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

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

$$\text{where } [\mathbf{R}]_{m\times m} = \begin{bmatrix} E\{f_{n-1}f_{n-1}\} & E\{f_{n-1}f_{n-2}\} & \dots & E\{f_{n-1}f_{n-m}\} \\ E\{f_{n-2}f_{n-1}\} & . & \dots & . \\ . & . & \dots & . \\ E\{f_{n-m}f_{n-1}\} & E\{f_{n-m}f_{n-2}\} & & E\{f_{n-m}f_{n-m}\} \end{bmatrix}$$

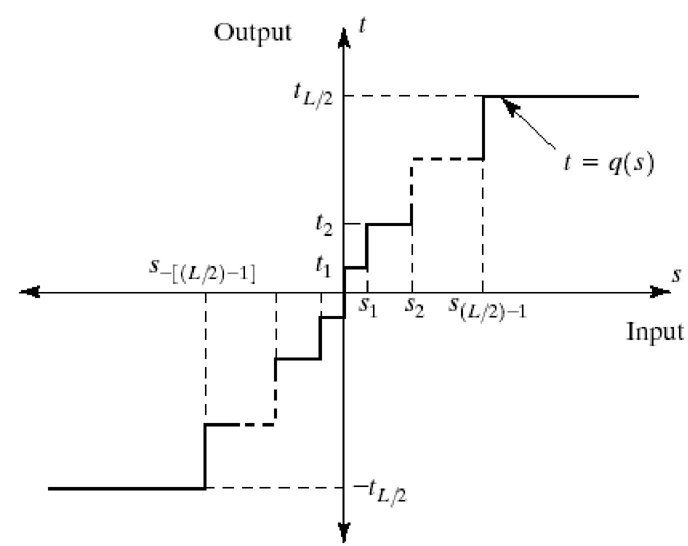
$$\mathbf{r} = \begin{bmatrix} E\{f_n f_{n-1}\} \\ E\{f_n f_{n-2}\} \\ \vdots \\ E\{f_n f_{n-m}\} \end{bmatrix}; \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}$$

Lossy Compression

- *Optimal quantization*: given a staircase function $t = q(s)$, the design problem is to select the best s_i and t_i for an input density function $p(s)$
  - The reconstruction levels t_i are the centroids of the areas under p(s)
  - The decision levels s_i are halfway between the reconstruction levels
  - The obtained quantizer is called (L-level) Lloyd-Max quantizer
-

$$\int_{s_{i-1}}^{s_i} (s - t_i)\, p(s)\, ds = 0 \qquad i = 1,2,\ldots,L/2$$

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



Transform Coding

- General scheme
  - The transform has to decorrelate the pixels or to compact as much information as possible into the smallest number of transform coefficients
  - The quantization selectively eliminates or more coarsely quantizes the less informative coefficients
  - Variable-length coding eliminates the remained coding redundancy

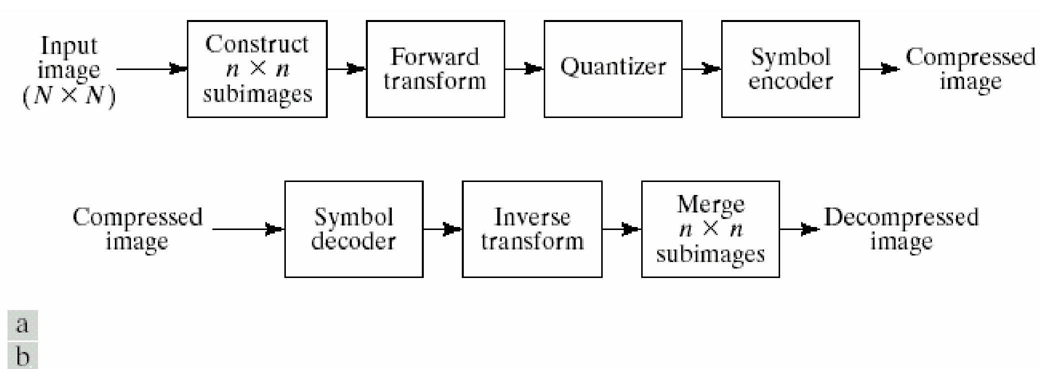


FIGURE 8.28 A transform coding system: (a) encoder; (b) decoder.

Transform Coding

- Transform selection
  - Defined by forward and inverse transformation kernels (basis functions, basis images)

    $$T(u,v)=\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)g(x,y,u,v) \text{ for } u,v=0,1,2,\ldots N-1$$

    $$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}T(u,v)h(x,y,u,v) \text{ for } x,y=0,1,2,\ldots N-1$$

  - Separable kernels $g(x,y,u,v)=g_1(x,u)g_2(y,v)$
  - Symmetric kernels $g(x,y,u,v)=g_1(x,u)g_1(y,v)$

- Fourier transform

  $$g(x,y,u,v)=\frac{1}{N^2}e^{-j2\pi(ux+yv)/N}$$

  $$h(x,y,u,v)=e^{j2\pi(ux+yv)/N}$$

- Walsh-Hadamard transform

  $$g(x,y,u,v)=h(x,y,u,v)=\frac{1}{N}(-1)^{\sum_{i=0}^{m}\left[b_i(x)p_i(u)+b_i(y)p_i(v)\right]} \text{ where } N=2^m$$

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

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

  $$p_2(u)=b_{m-2}(u)+b_{m-3}(u)$$

  $$\vdots$$

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

Transform Coding

- Walsh-Hadamard transform kernel consists of alternating plus and minus 1's arranged in a checkerboard pattern

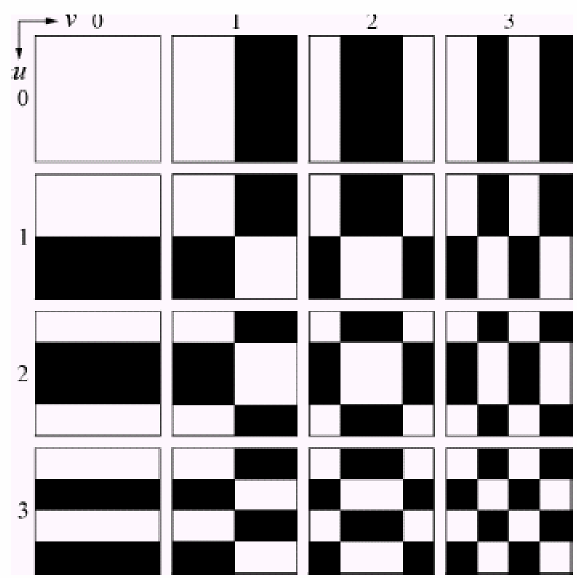


FIGURE 8.29 Walsh-Hadamard basis functions for $N = 4$. The origin of each block is at its top left.

Transform Coding

- Discrete Cosine Transform $g(x,y,u,v) = h(x,y,u,v) = \alpha(u)\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right]$

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

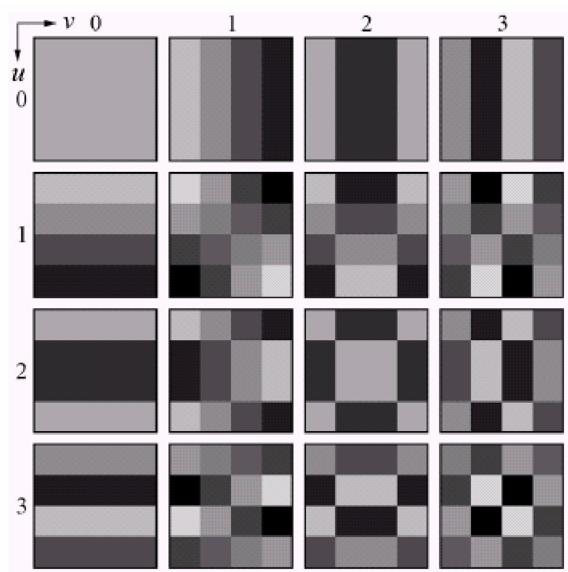
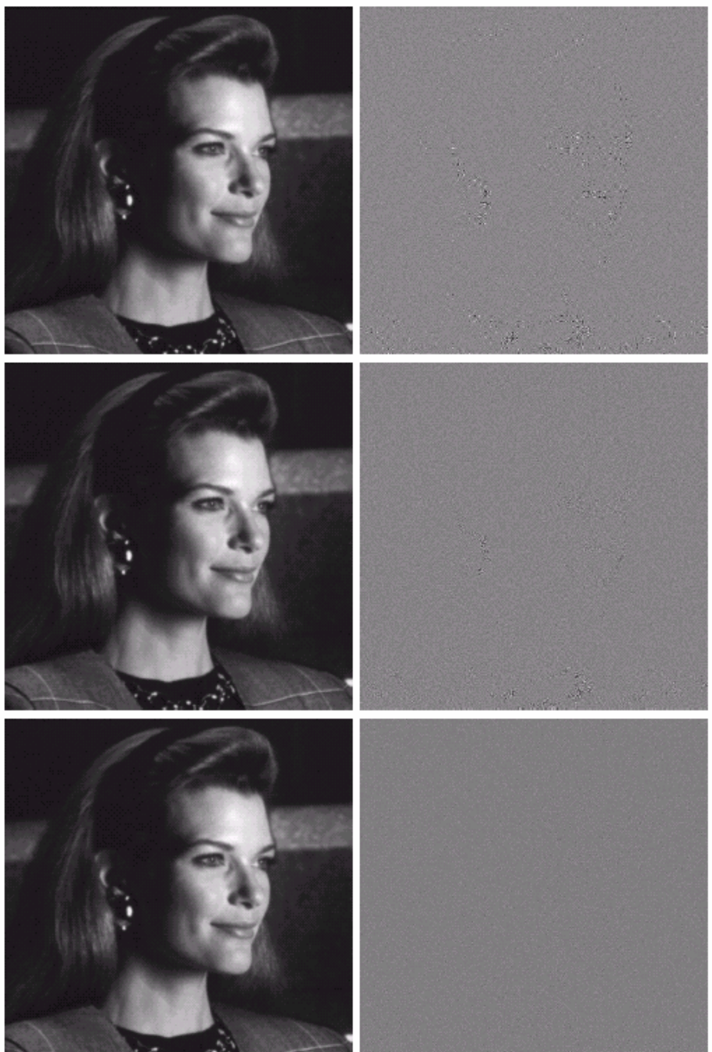


FIGURE 8.30 Discrete-cosine basis functions for $N = 4$. The origin of each block is at its top left.

Transform Coding



50% of transform coefficients disregarded

Residual images are not zero but some mean-square error

$rms_{Fourier}=1.28$

$rms_{Walsh}=0.86$

$rms_{DCT}=0.68$

a	b
c	d
e	f

FIGURE 8.31 Approximations of Fig. 8.23 using the (a) Fourier, (c) Hadamard, and (e) cosine transforms, together with the corresponding scaled error images.

Transform Coding

- Basis images and energy compaction
  - The matrix **F** is defined as a linear combination of n^2 matrices of size $n \times n$ (basis matrices)
  - masking function to get truncated expansion
  - The mean-square approximation error is the sum of the variances of the discarded coefficients
- The Optimal Karhunen-Loeve Transform and its approximations

$$f(x,y) = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1} T(u,v)h(x,y,u,v)$$

$$\mathbf{F} = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1} T(u,v)\mathbf{H}_{uv}$$

$$\mathbf{H}_{uv} = \begin{bmatrix} h(0,0,u,v) & h(0,1,u,v) & \ldots & h(0,n-1,u,v) \\ h(1,0,u,v) & \cdot & \ldots & \cdot \\ \cdot & \cdot & \ldots & \cdot \\ h(n-1,0,u,v) & h(n-1,1,u,v) & \ldots & h(n-1,n-1,u,v) \end{bmatrix}$$

$$\gamma(u,v) = \begin{cases} 0 & \text{if } T(u,v) \text{ is non - significant} \\ 1 & \text{if } T(u,v) \text{ is significant} \end{cases}$$

$$\hat{\mathbf{F}} = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1} \gamma(u,v)T(u,v)\mathbf{H}_{uv}$$

$$e_{ms} = E\left\{\left\|\mathbf{F} - \hat{\mathbf{F}}\right\|^2\right\} = \sum_{u=0}^{n-1}\sum_{v=0}^{n-1} \sigma^2_{T(u,v)}\left[1 - \gamma(u,v)\right]$$

Transform Coding

- Boundary (Gibbs-like) effects: more visible for the Fourier transform and less visible for the DCT

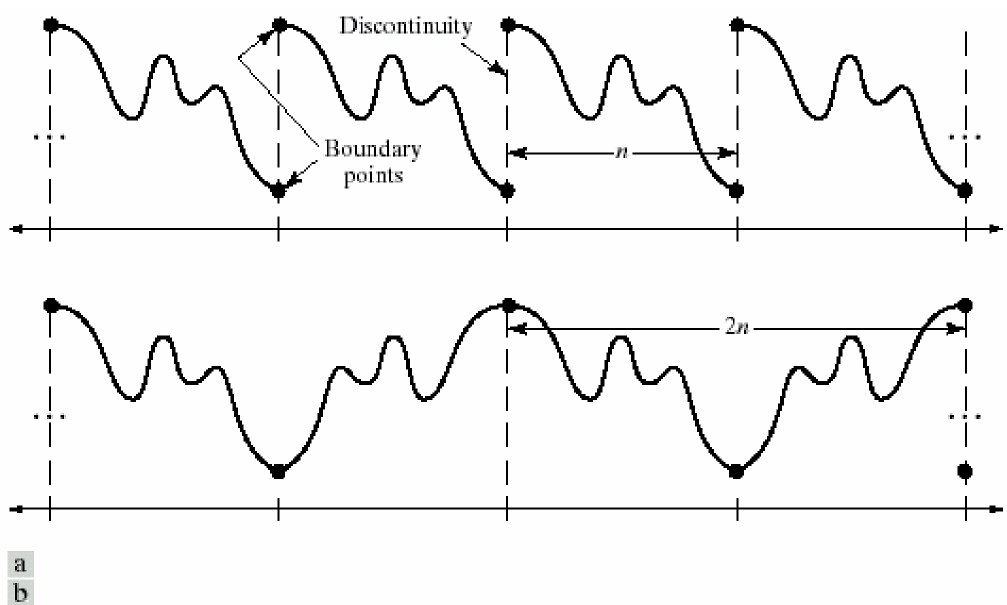


FIGURE 8.32 The periodicity implicit in the 1-D (a) DFT and (b) DCT.

Transform Coding

- Subimage size selection: computational complexity versus coding efficiency

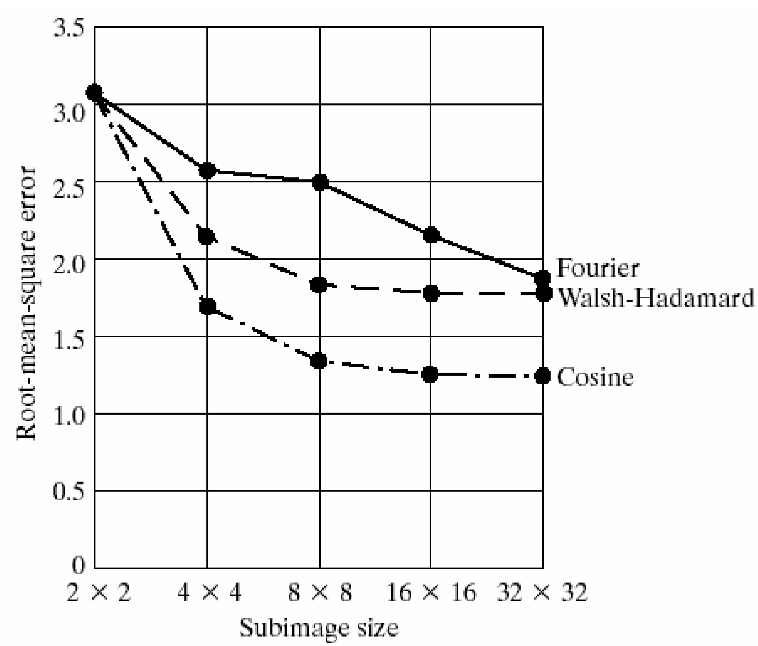


FIGURE 8.33 Reconstruction error versus subimage size.

Transform coding

a b
c d
e f

FIGURE 8.34 Approximations of Fig. 8.23 using 25% of the DCT coefficients: (a) and (b) 8×8 subimage results; (c) zoomed original; (d) 2×2 result; (e) 4×4 result; and (f) 8×8 result.

Transform Coding

- Bit allocation: zonal coding (based on maximum variance) or threshold coding (based on maximum amplitude)

- Zonal coding implementation: information is viewed as uncertainty. Transform coefficients with maximum variance carry te most of image information and have to be retained.

(a)

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

(b)

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

(c)

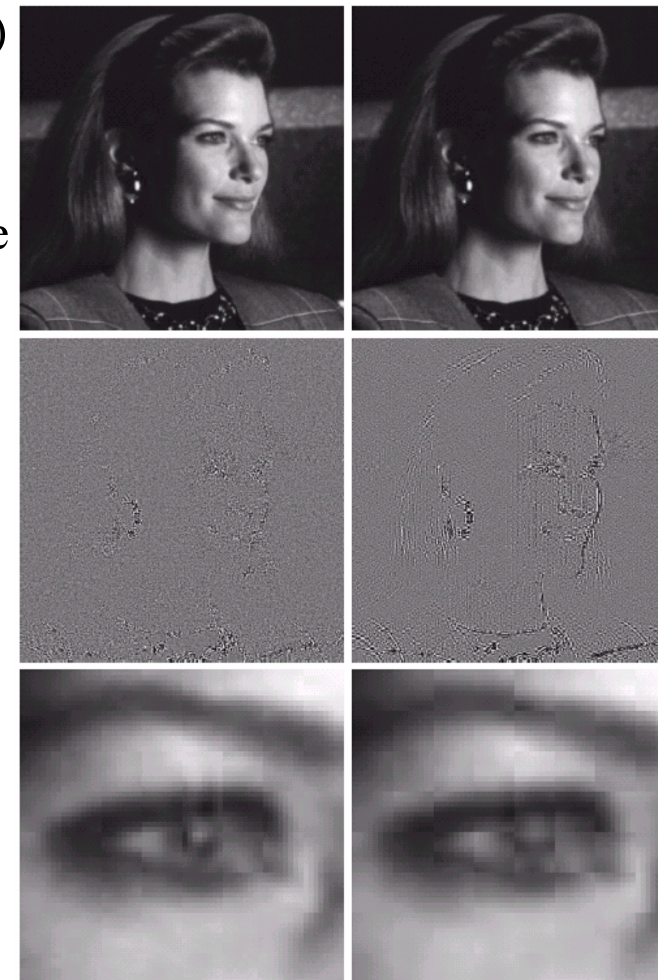
1	1	0	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(d)

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 8.36 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.



a b
c d
e f

FIGURE 8.35 Approximations of Fig. 8.23 using 12.5% of the 8 × 8 DCT coefficients: (a), (c), and (e) threshold coding results; (b), (d), and (f) zonal coding results.

Transform Coding

- Zonal coding implementation: information is viewed as uncertainty. Transform coefficients with maximum variance carry te most of image information and have to be retained.

- Zonal masks are showing the number of bits used to code each coefficient

- Fixed number of bits for each coeff.

- Bit budget distributed unequally
  - Lloyd-Max quantizer
  - Rayleigh pdf model for the dc
  - Laplacian for the remaining coeff.

(a)

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

(b)

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

(c)

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

(d)

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 8.36 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.

Transform Coding

- Threshold coding implementation: Transform coefficients of largest magnitude make to most significant contribution to reconstructed subimage quality.
- The retained coefficients are reordered in 1-D manner (zig-zag scanning) and then run-length coded
- Thresholding
  - Single global threshold to all coeff.
    - level of compression varies
  - Different for different subimages
    - constant coding rate
  - Varying (with the location) threshold
    - combines thresholding with quantization

(a)

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

(b)

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

(c)

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

(d)

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

FIGURE 8.36 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.

Transform Coding

Z is a transform normalization array

Before restoration, the normalized (thresholded and quantized) image must be denormalized.

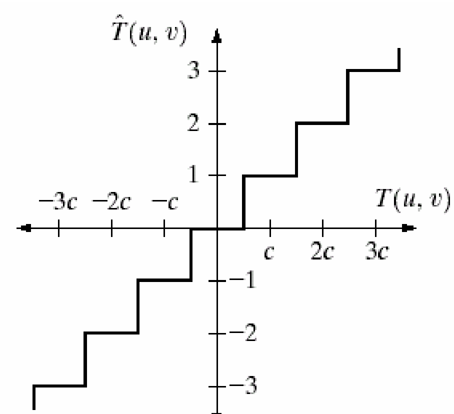
$$\mathbf{Z} = \begin{bmatrix} Z(0,0) & Z(0,1) & \dots & Z(0,n-1) \\ Z(1,0) & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ Z(n-1,0) & Z(n-1,1) & \dots & Z(n-1,n-1) \end{bmatrix}$$

$$\hat{T}(u,v) = round\left[\frac{T(u,v)}{Z(u,v)}\right]$$

$$\dot{T}(u,v) = \hat{T}(u,v)Z(u,v)$$

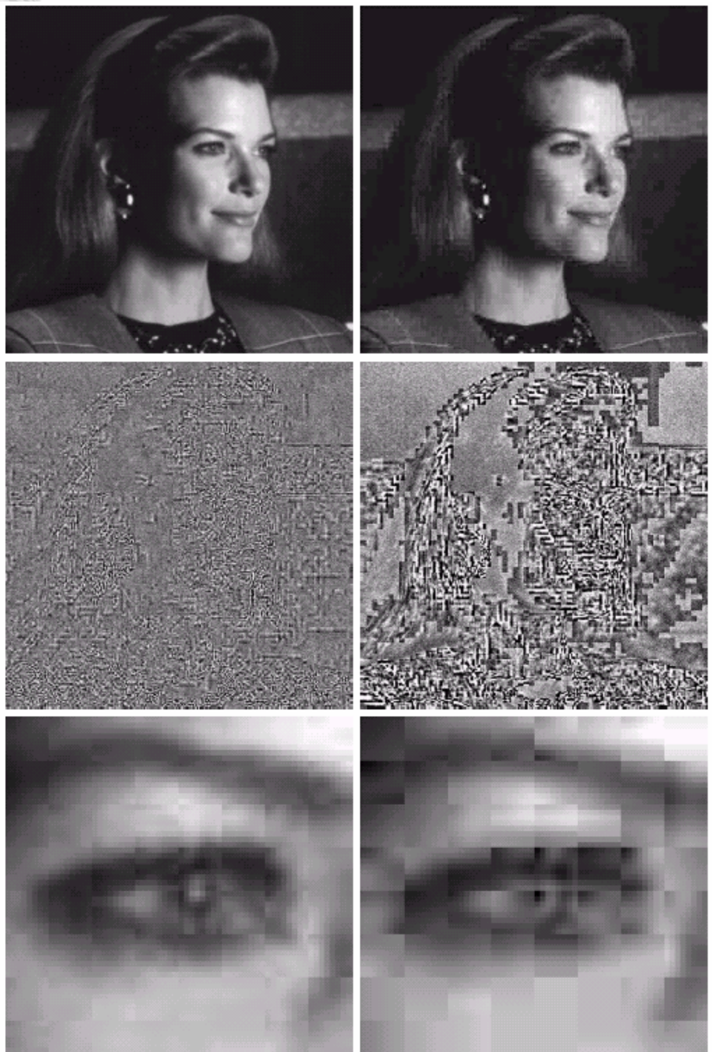
a b

FIGURE 8.37 (a) A threshold coding quantization curve [see Eq. (8.5-40)]. (b) A typical normalization matrix.



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

Transform Coding



Cr1=34

Cr2=67

a b
c d
e f

FIGURE 8.38 Left column: Approximations of Fig. 8.23 using the DCT and normalization array of Fig. 8.37(b). Right column: Similar results for 4**Z**.

Image Compression

End of Part 1