CHAPTER 4

DIGITAL IMAGE COMPRESSION

Contents

Introduction

Digital image coding and compression

Techniques and algorithms concerned with the minimization of the memory needed to represent and store digital images

Compression factors

- Transmission and storing of large images.
- Reduce of baud rate.
- Reduce of transmission time.

Introduction

Lossless compression techniques

These are used when raw image data are difficult to obtain or contain vital information that may be destroyed by compression, e.g. in medical diagnostic imaging.

Lossy compression techniques

These can be used when raw image data can be easily reproduced or when the information loss can be tolerated at the receiver site, e.g. in Digital Television, Teleconferencing.

Entropy Coding

- Pulse Coding Modulation (PCM) using *B* bits/pixel.
- The average number of bits per pixel can be reduced by assigning binary codes of different bit length to the various image intensities.
- The pdf (probability density function) *p(i)* can be estimated by calculating the digital image histogram.
- Assignment of short codewords to intensities having a hgh probability of occurrunce and larger codewords to less frequent image intensity levels.

Huffman coding

Entropy Coding

Average codeword length:

$$\bar{L} = \sum_{i=0}^{2^B-1} L(i)\,p(i)$$

The lengths $L(i)$ must be chosen in such a way that L is minimized.

Image entropy:

$$H(B) = -\sum_{i=0}^{2^B-1} p(i)\log_2 p(i)$$

Lower bound on average codeword length

$$\bar{L} \geq H(B)$$

The more higly predictable intensity levels are, the more efficient entropy coding is.

Huffman coding

- The image intensity levels are coded by using variable length codewords.
- No codeword can be the prefix of another codeword.

Average codeword length: $H(B) \leq \bar{L} \leq H(B) + 1$

Huffman coding

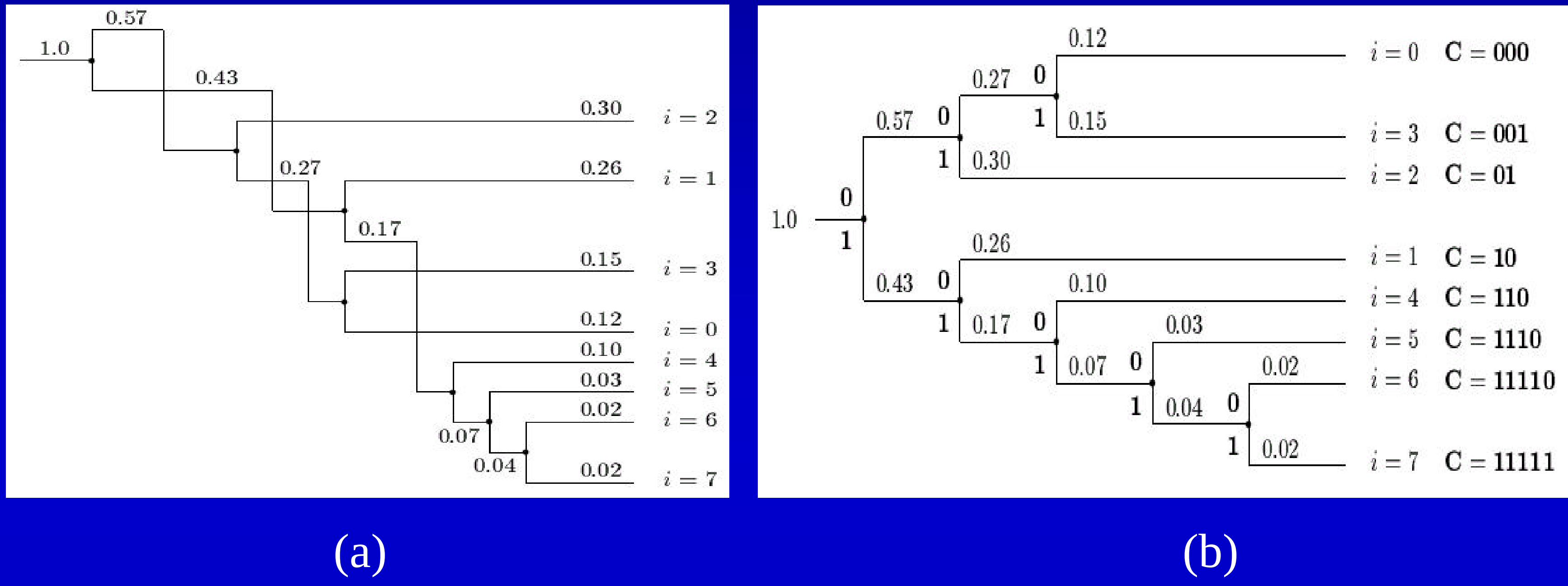

Figure1: (a) Construction of Huffman code tree, (b) tree rearrangement.

Huffman coding

Size of Huffman codebook: $L = 2^B$

The longest codeword may have up to L bits.

Practical modigication of the Huffman code

Truncated Huffman code:

- the Huffman code is truncated to $L_1 < L$
- the first L_1 intensity levels are Huffman coded
- the remaining intensity levels are coded by a prefix code followed by a fixed-length code

Huffman coding

Huffman coding combination with other coding schemes

- With predictive coding.
- With transform coding of greyscale images.
- With run-length coding of binary images.

Run-length coding

Each image line can be represented as follows:

$$(x_1,\ldots,x_M)\rightarrow(g_1,l_1),(g_2,l_2),\ldots,(g_k,l_k)$$

where:

$$g_1=x_1,\qquad g_k=x_M$$

$$\sum_{i=1}^{k} l_i = M$$

Each couple (g_i,l_i) is called gray-level run.

Run-length coding

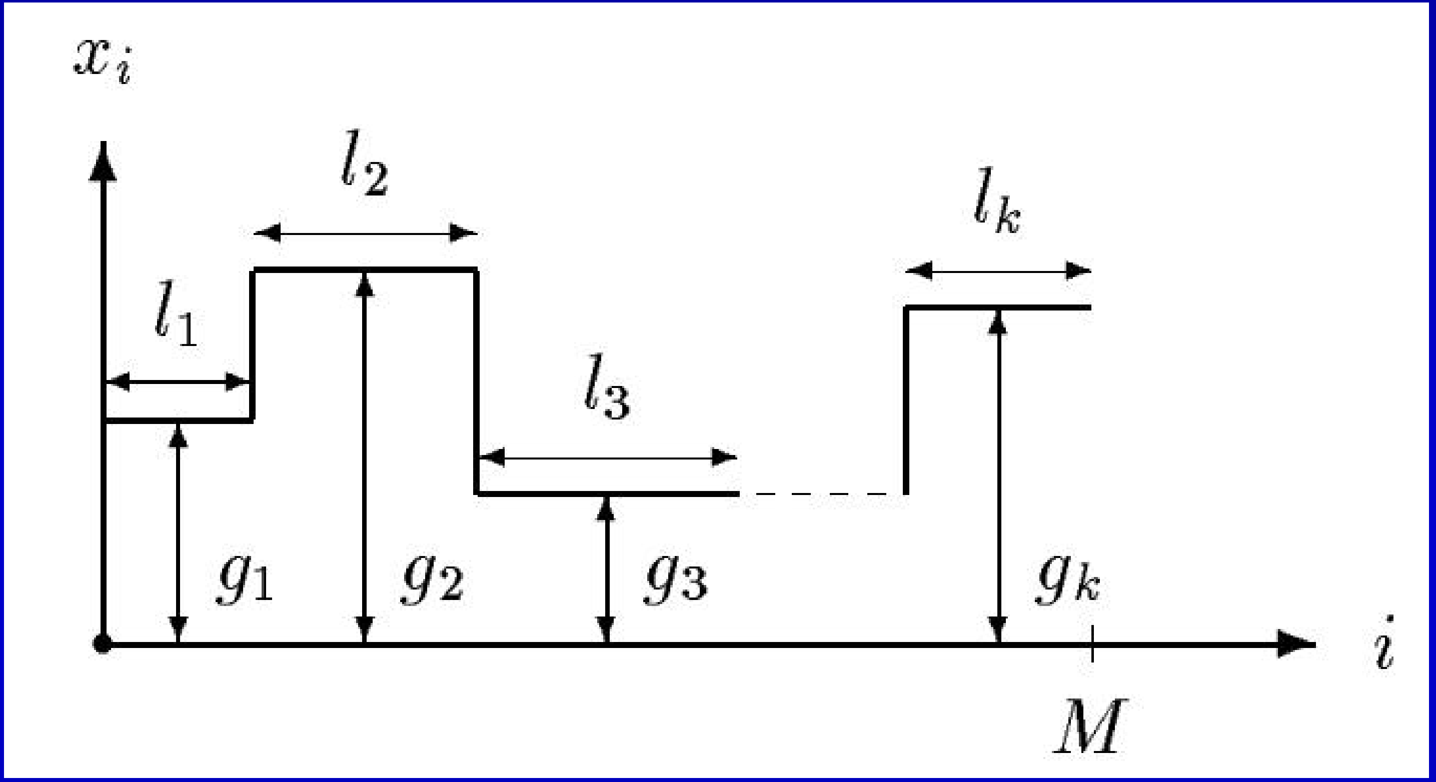

Figure 2: Graphical representation of an image line.

Run-length coding

- The resulting compression is considerable if the grey-level runs are relatively large.
- The savings are even larger when the image is binary.
- An end of line (EOL) codeword indicates the start of an image and the end of a line.

Run-length coding

EOL	DATA	EOL	DATA	FILL	EOL	· · ·	EOL	DATA	EOL	EOL	EOL	EOL	EOL	EOL

Figure 3: Format of an image coded by a run-length code.

- The run-length coding has been standardized by CCITT.

- It has been included in Group 3 coding schemes (FAX transmission).

Run-length coding

Terminating Codewords		
Run length	White run	Black run
0	00110101	0000110111
1	000111	010
2	0111	11
3	1000	10
4	1011	011
5	1100	0011
6	1110	0010
7	1111	00011
8	10011	000101
9	10100	000100
10	00111	0000100
11	01000	0000101
12	001000	0000111
13	000011	00000100
14	110100	00000111
15	110101	000011000

Figure 4: Part of modified Huffman codebook for run-length coding (CCITT).

Modified READ coding

Run-length coding

It is a one-dimensional scheme that cannot take into account vertical correlations among run transitions in consecutive image lines.

Modified READ coding (Relative Element Address Designate)

It is a two-dimensional coding scheme that codes a binary image line with reference to the previous line.

Modified READ coding

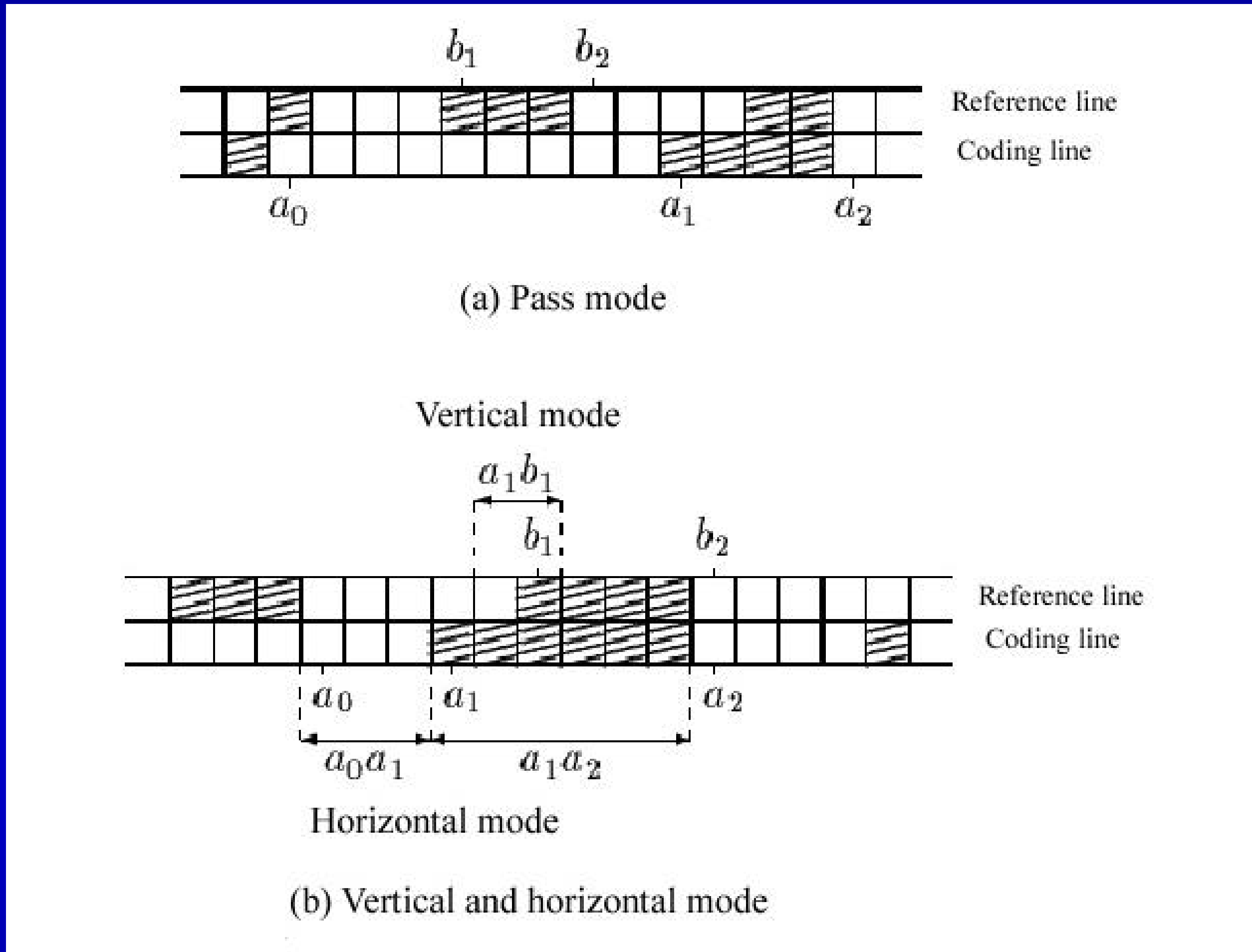

Figure 5: Transition elements in modified READ coding.

Modified READ coding

- Pass mode
- Vertical mode
- Horizontal mode

Mode	Changing elements to be coded	Notation	Codeword
Pass	b_1, b_2	P	0001
Horizontal	a_0, a_1, a_2, a_3	H	001+M(a_0a_1)+M(a_0a_1)
Vertical	a_1 just under b_1, a_1 b_1=0 a_1 to the right of b_1, a_1 b_1=1 a_1 to the right of b_1, a_1 b_1=2 a_1 to the right of b_1, a_1 b_1=3 a_1 to the left of b_1, a_1 b_1=1 a_1 to the left of b_1, a_1 b_1=2 a_1 to the left of b_1, a_1 b_1=3	$V(0)$ $V_R(1)$ $V_R(2)$ $V_R(3)$ $V_L(1)$ $V_L(2)$ $V_L(3)$	1 011 000011 0000011 010 000010 0000010
	EOL 1-d coding of next line 2-d coding of next line		000000000001 EOL+’1’ EOL+’0’

Figure 6: Code table of modified READ code.

LZW compression

- General-purpose compression scheme proposed by Lempel-Ziv and Welch.

- It can be used for the compression of any binary data file.

- It is incorporated in several de facto image storage standards (e.g. TIFF, GIF).

LZW compression

- It is a lossless, fast and effective algorithm and can operate on images of any bit depth.

- LZW compression is based on the construction of a code table that maps frequently encountered bit strings to output codewords.

- The digital image as well as the coded one is treated as a one-dimensional bit string.

LZW compression

- The speed of both LZW compression and decompression depends on the implementation of the code table and on the efficient searching in it.

- The decompression is usually faster because no search is needed.

- The compression ratio ranges from 1:1.5 to 1:3.

- Substantial compression can be obtained for binary or bitmap images (moderate compression for raw greyscale images).

Predictive coding

- One way to describe information redundancy in digital images is to use predictability in local image neighbourhoods.

- The pixel intensity *f(n,m)* can be predicted from the intensities of its neighbourhooding pixels *Á*:

$$\hat{f}(n,m) = L\left[f(n-i, m-j), \quad (i,j) \in A, \quad (i,j) \neq (0,0) \right]$$

Predictive coding

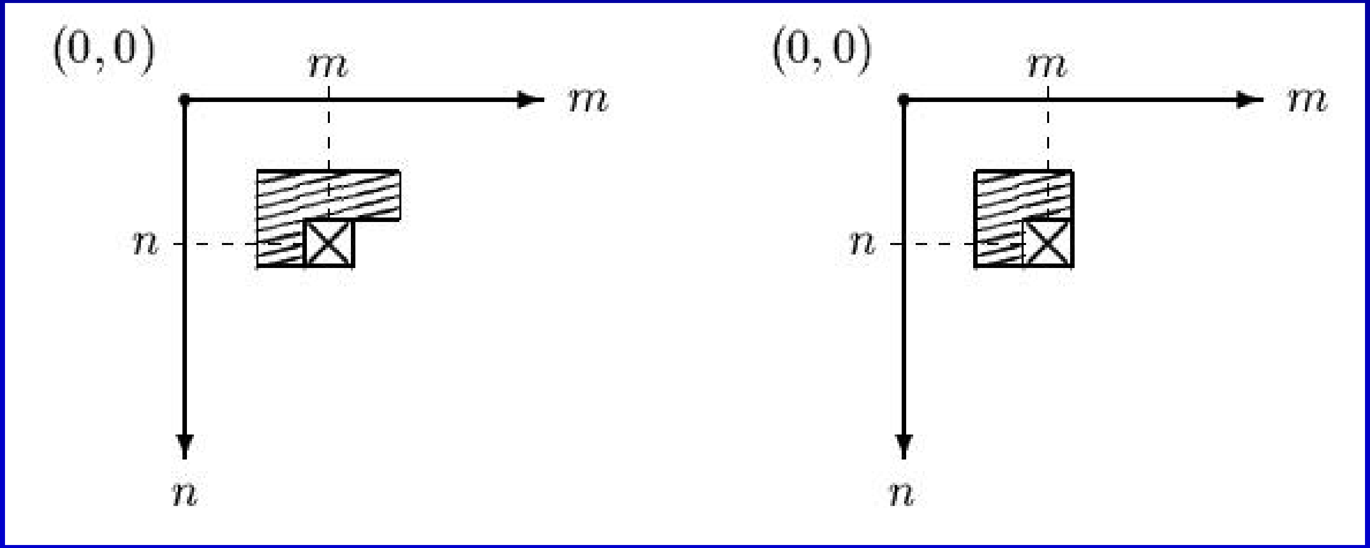

Figure 7: Causal windows used in image prediction.

- When causal prediction is used, then this is based on already reconstructed pixel values in the past:

$$\hat{f}(n,m) = L\left[f_r(n-i, m-j), \quad (i,j) \in A \right]$$

Predictive coding

- Let us suppose that it is sufficient to code the error:

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

- If $e_q(n,m)$ is the quantized and coded value of the error e(n,m), the pixel value can be reconstructed as follows:

$$f_r(n,m) = L\left[f_r(n-i, m-j), \quad (i,j) \in A\right] + e_q(n,m)$$

- If the prediction is good, the error term has a small dynamic range and a substantial compression can be achieved.
- For the reconstruction, the transmission of the prediction coefficients and of the coded error is needed.

Predictive coding

Predictive Differential Pulse Code Modulation (DPCM)

- Extensively used in telecommunications.
- It is a lossy coding scheme.
- The quantization of the error signal always creates an irrecoverable amount of distortion.

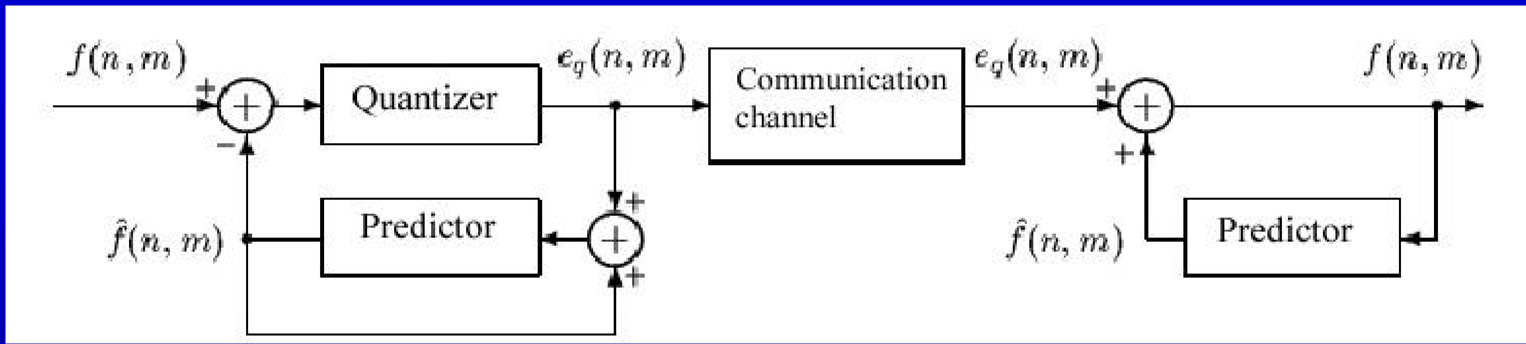

Figure 8: Predictive differential pulse code modulation (DPCM).

Predictive Differential Pulse Code Modulation (DPCM) with entropy coding

- It is a lossless coding scheme.
- The performance of the DPCM depends greatly on the predictor used and on the choice of its coefficients.

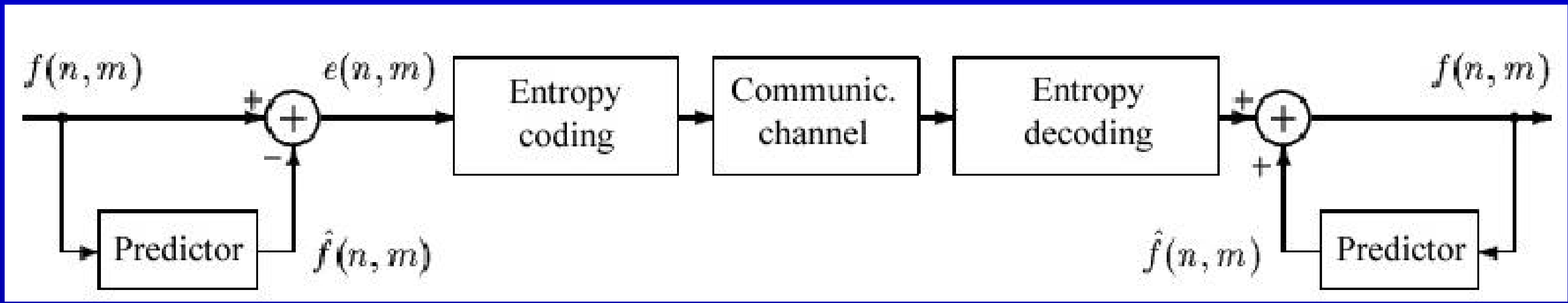

Figure 9: DPCM with entropy coding

Predictive coding

- Let us suppose that the image line *f*(*m*) can be modeled as a stationary AR process:

$$f(m) = \sum_{k=1}^{p} a(k) f(m-k) + e(m), \qquad E\left[e^2(m)\right] = s^2$$

where *å*(*m*) is a white additive Gaussian noise uncorrelated to *f*(*m*).

- A natural choice for the prediction scheme is:

$$\hat{f}(m) = \sum_{k=1}^{p} a(k) f_r(m-k)$$

Predictive coding

- The quantized error signal $e_q(m)$ is transmitted to the receiver:

$$e_q(m) = Q[e(m)] = Q[f(m) - \hat{f}(m)]$$

- The image row is reconstructed by using:

$$f_r(m) = \sum_{k=1}^{p} a(k) f_r(m-k) + e_q(m)$$

- The prediction coefficients can be chosen by solving the set of normal equations (where $R(k)$ is the autocorrelation function):

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(p-1) \\ R(1) & R(0) & \cdots & R(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ R(p-1) & R(p-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ \vdots \\ a(p) \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}$$

Predictive coding

- One-dimensional prediction models can be extended to two-dimensional ones of the form:

$$\hat{f}(n,m)=\sum_{(i,j)\in A}\sum a(i,j)f(n-i,m-j)$$

- The 2-d AR prediction model coefficients can be optimally chosen by minimizing the mean square error:

$$E\left[\left|f(n,m)-\sum_{(i,j)\in A}\sum a(i,j)f(n-i,m-j)\right|^2\right]$$

- This minimization leads to the solution of a set of normal equations of the form:

$$R(k,l)=\sum_{(i,j)\in A}\sum a(i,j)R(k-i,l-j)$$

Predictive coding

- The error image can be obtained and quantized, once the coefficients $a(i,j)$ are known:

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

- The digital image can be reconstructed at the receiver as follows:

$$f_r(n,m) = \sum_{(i,j)\in A}\sum a(i,j) f_r(n-i,m-j) + e_q(n,m)$$

- The autocorrelation coefficients can be estimated by using:

$$R(i,j) = \frac{1}{(2N+1)(2M+1)} \sum_{i=-N}^{N}\sum_{j=-M}^{M} f(k,l) f(k+i,l+j)$$

Predictive coding

- Predictive DPCM is a simple digital compression technique.

- It can be easily implemented in both software and hardware.

- It can be used to achieve moderate compression ratios.

- Its is sensitive to channel noise.

- Noise bursts propagate in the entire image row, or even in the entire decode image.

Transform image coding

We try to use image transforms in an effort to concentrate the image energy in a few transform coefficients.

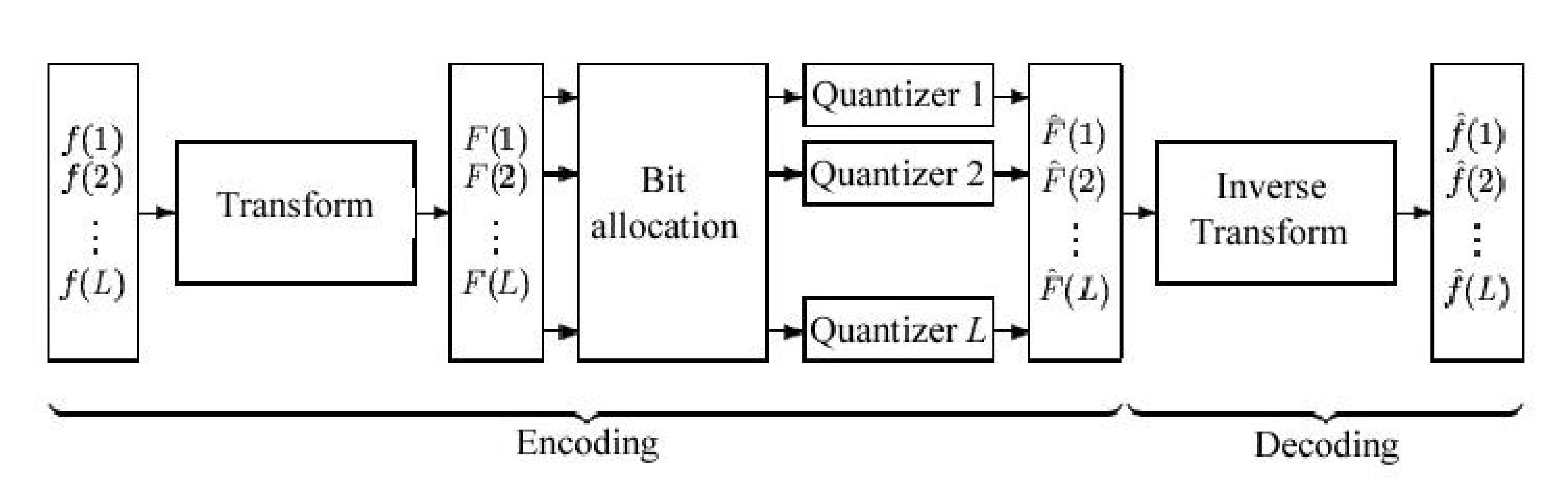

Figure 10: Transform encoding/decoding

Transform image coding

Let **f** be the vector representing an image of size $L = N \times M$. The transform vector **F** is given by:

$$\mathbf{F} = \mathbf{A}\mathbf{f}$$

where **Á** is the transform matrix. The inverse transform is defined as follows:

$$\mathbf{f} = \mathbf{A}^{-1}\mathbf{F}$$

<u>Unitary</u> transform definition:

$$\mathbf{A}\mathbf{A}^{*T} = \mathbf{A}^{T}\mathbf{A}^{*} = \mathbf{I}$$

Transform image coding

A unitary transform satisfies energy conservation:

$$\|\mathbf{f}\|^2 = \sum_{k=1}^{L} |f(k)|^2 = \sum_{k=1}^{L} |F(k)|^2 = \|\mathbf{F}\|^2$$

- The DC coefficient and some other “low-frequency” coefficients tend to concentrate most of the signal energy.
- A varying number of bits can be allocated to the remaining coefficients.
- The transform coefficients $F(k)$, $1 \le k \le K$ are quantized using K quantizers.
- The decoding is done by applying the inverse transform to the encoded coefficient vector.

Transform image coding

Problems to be solved for a good transform coding algorithm

- The choice of the transform to be used (DFT, WHT, DCT, DST etc.).
- The choice of image block size.
- The determination of bit allocation. If the average number of bits per pixel is *Â* the following relation must be satisfied:

$$\frac{1}{L}\sum_{k=1}^{L} n_k = B$$

- The average distortion due to coefficient quantization is given by:

$$E = \frac{1}{L}\sum_{k=1}^{L} E\left[\left|F(k) - Q[F(k)]\right|^2\right] = \frac{1}{L}\sum_{k=1}^{L} \sigma_k^2 q(n_k)$$

where $Q[.]$ denotes quantization and σ_k^2 is the variance of the transform coefficient $F(k)$.

- A reasonable choice of the bit numbers is given by:

$$n_k = B + \frac{1}{2}\left[\log_2 \sigma_k^2 - \frac{1}{L}\log_2\left(\prod_{k=1}^{L} \sigma_k^2\right)\right]$$

Transform image coding

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

Figure 12: Binary allocation of a 16×16 block cosine transform.

Transform image coding

(a)

(b)

Figure 13: (a) Original image. (b) JPEG compressed image.