Module 2

LOSSLESS IMAGE COMPRESSION SYSTEMS

Lesson 5

Other Coding Techniques

Instructional Objectives

At the end of this lesson, the students should be able to:

1. Convert a gray-scale image into bit-plane images.
2. Apply run-length coding on binary images.
3. State the basic principles of lossless predictive coding.
4. Obtain predicted image and error image from a given image and prediction scheme.
5. Reconstruct the original image from the error image and the predicted image.
6. Compare the entropies of the original image and the error image.

5.0 Introduction

So far, in lesson-3 and lesson-4, we have studied some of the popular lossless compression schemes, like Huffman Coding, Arithmetic Coding and Lempel-Ziv Coding. The first two fall under the category of entropy coding, whereas Lempel-Ziv coding relies on dictionary updating with new symbol groups. There are a few more techniques being used and some of these are used in conjunction with other lossless compression schemes.

In this lesson, we shall first discuss lossless coding schemes for binary images. This has got applications in facsimile images, which are essentially binary in nature. The binary image compression techniques are applicable for gray-scale images also, since any gray-scale image can be converted into bit-plane images. We shall discuss run-length coding, which is a popular and very effective binary image coding scheme, that is often used in association with other variable length coding schemes.

The rest of the lesson deals with *lossless predictive coding* scheme, in which the current pixel is predicted based on its neighboring pixels, received previously in time. The error in prediction is encoded, using any of the lossless coding techniques. Since, the entropy of the error image is much lower as compared to the entropy of the original image, significant bit reduction is possible using this technique.

5.1 Converting a gray-scale image into bit-plane images

We first note that any integer value s in the interval $[0,2^n-1]$ can be represented by a polynomial of base-2, as follows:

$$s = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots\cdots a_1 2^1 + a_0 2^0 \quad \text{.....(5.1)}$$

where, $a_{n-1}, a_{n-2}, \cdots a_0$ are the n binary coefficients associated with the corresponding powers of 2. This basically converts the integer s into the binary number representation $a_{n-1}, a_{n-2}, \cdots a_0$ with a_{n-1} as the most significant bit and a_0 as the least significant bits. Following this, all the pixel values having intensities in the range [0, 255] of an image array can be converted into its 8-bit binary representation. If we now consider an array containing only the i^{th} bit of every pixel $(i = 0,1,\cdots\cdots,7)$, we get the i^{th} bit plane image.

Let us consider a simple 4x4 array, whose elements are integers in the range 0-15 . The corresponding 4-bit binary representations are indicated within the parenthesis, s shown below:

13(1101)	12(1100)	13(1101)	9(1001)
15(1111)	13(1101)	11(1011)	10(1010)
14(1110)	11(1011)	12(1100)	9(1001)
11(1011)	12(1100)	14(1110)	7(0111)

The corresponding 4 bit planes are shown below:

1 1 1 1	1 1 1 0	0 0 0 0	1 0 1 1
1 1 1 1	1 1 0 0	1 0 1 1	1 1 1 0
1 1 1 1	1 0 1 0	1 1 0 0	0 1 0 1
1 1 1 0	0 1 1 1	1 0 1 1	1 0 0 1
Bit-plane-3(MSB)	Bit-plane-2	Bit-plane-1	Bit-plane-0(LSB)

5.2 Run-length Coding of bit-plane images

In one-dimensional run-length coding schemes for binary images, runs of continuous 1s or 0s in every row of an image are encoded together, resulting in substantial bit savings. Either, it is to be indicated whether the row begins with a run of 1s or 0s, or the encoding of a row compulsorily begins with 1s, where the count could be zero, if it begins with 0s.
Run-length encoding may be illustrated with the following example of a row of an image:

000110100011111001011111111111111101110000000010000111111100000001

Let us say, we adopt the second scheme, to begin with runs of 1s. Hence, the first run count in the given binary sequence is 0. Then we have a run of three 0s. Hence, the next count is 3. Proceeding in this manner, the reader may verify that the given binary sequence gets encoded to:

0,3,2,1,1,3,5,2,1,1,15,1,3,9,1,4,7,7,1

Every run count values for 1s and 0s are associated with some probabilities, which may be determined through some statistical observations. It is therefore possible to estimate the entropies H_0 and H_1 for the runs of 0s and 1s respectively. If L0 and L_1 are the average values of run-lengths for 0s and 1s, the run-length entropy of the image is expressed as

$$H_R = \frac{H_0 + H_1}{L_0 + L_1} \quad \text{.....(5.2)}$$

We may apply some *variable length coding* to encode the run values. We shall find uses of *run-length* and *Huffman coding* combinations in multimedia standards, to be discussed in later lessons.

Although, run-length coding is essentially a compression technique for binary images, it can be applied to gray-scale images in an indirect way by first decomposing the gray-scale image to bit-plane images and then applying run-length coding on each of the bit-plane images.

What we have just described in this section is one-dimensional run-length coding. Two dimensional extension of run-length coding, using *relative address coding* (RAC) have also been reported in the literature [1].

5.3 Lossless Predictive Coding

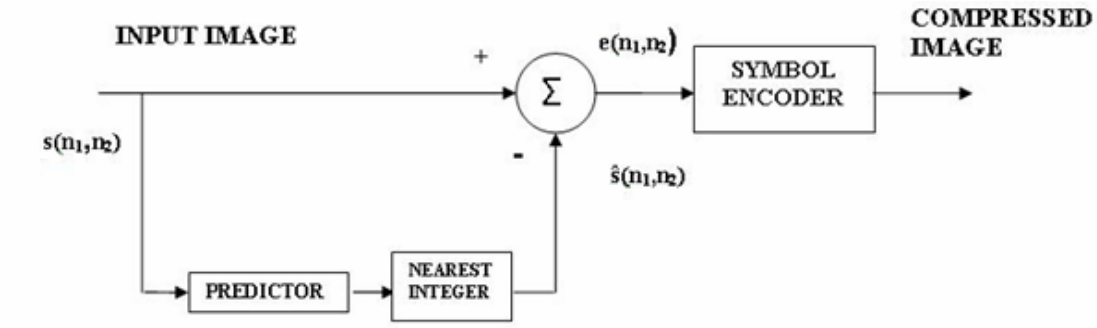

Fig 5.1 : *Lossless predictive encoder*

Fig.5.1 shows the block diagram of the *lossless predictive encoder*. Here, the current pixel $s(n_1, n_2)$ is predicted as $\hat{s}(n_1, n_2)$ using a linear combination of previously received pixels, which are spatial neighbors of the current pixel. The error in prediction, given by

$$e(n_1, n_2) = s(n_1, n_2) - \hat{s}(n_1, n_2) \quad \text{.....(5.3)}$$

is encoded using any of the lossless compression schemes discussed so far.
Since the predictor can only consider the pixels received so far, the prediction is based the set of pixels $\{s(n_1, n_2 - i) \mid i = 1,2,\cdots\cdots, n_2\}$ from the current row and the set of pixels $\{s(n_1 - i, j) \mid i = 1,2,\cdots\cdots, n_1;\ j = 0,1,\cdots\cdots, N_2 - 1\}$ from the previous row, where N_2 is the number of columns. In practice, we normally consider the closest past neighbors of $s(n_1, n_2)$ and obtain the predicted value $\hat{s}(n_1, n_2)$ as a linear combination of these pixels, illustrated in Fig.5.2 and mathematically expressed as:

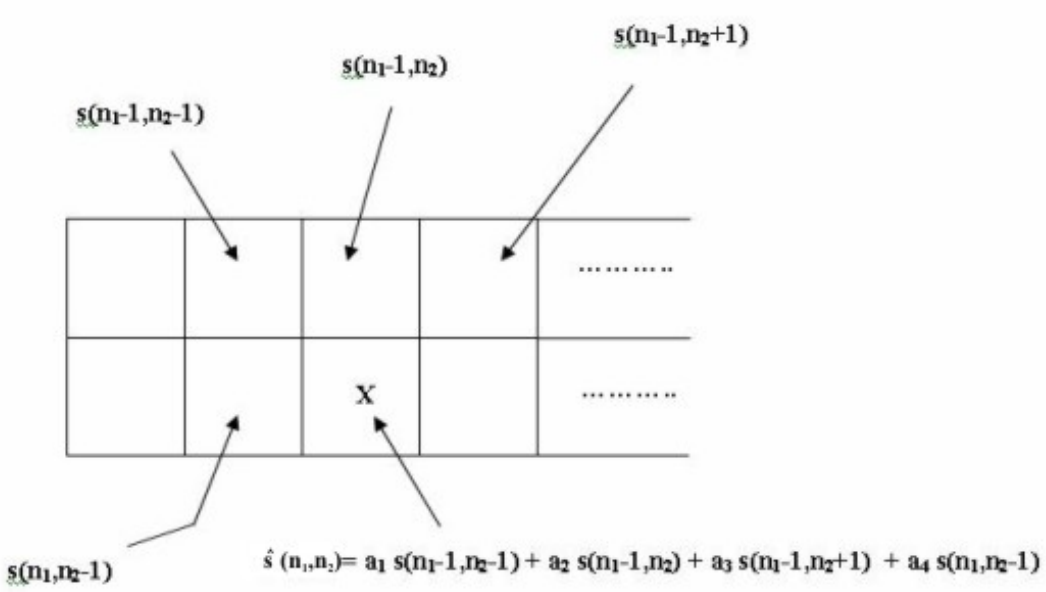

Fig 5.2 : *Predicted pixel as a linear combination of past neighbours*

$$\hat{s}(n_1, n_2) = a_1 s(n_1 - 1, n_2 - 1) + a_2 s(n_1 - 1, n_2) + a_3 s(n_1 - 1, n_2 + 1) + a_4 s(n_1, n_2 - 1) \quad \text{.....(5.4)}$$

where, a_1, a_2, a_3, a_4 are the coefficients associated with the respective neighboring pixels, such that $a_1 + a_2 + a_3 + a_4 = 1$. In the simplest case of $a_1 = a_2 = a_3 = 0$, we have $a_4 = 1$ and $\hat{s}(n_1, n_2) = s(n_1, n_2 - 1)$, which realizes a previous pixel predictor.

Equation (5.4) describes a *linear predictive coding*. The first pixel $s(0,0)$ of an image does not have any previously received neighbor and will be encoded directly without any prediction (in other words, $\hat{s}(n_1, n_2) = 0$). In absence of any quantization, the error encoding is lossless. Although, the pixel intensities are always expressed in integers, the predicted intensities computed using equation (5.4) and consequently the error intensity computed through equation (5.3) may have fractional components and mathematical rounding operation will be necessary. However, the rounding operation does not create any loss, since fractional intensity change can never be perceived.

The corresponding *lossless predictive decoder* is shown in Fig.5.3. In the decoder, the same prediction mechanism is replicated. It receives the encoded error signal $e(n_1, n_2)$ and reconstructs the current pixel $s(n_1, n_2)$ using

$$s(n_1, n_2) = \hat{s}(n_1, n_2) + e(n_1, n_2) \quad \text{.....(5.4)}$$

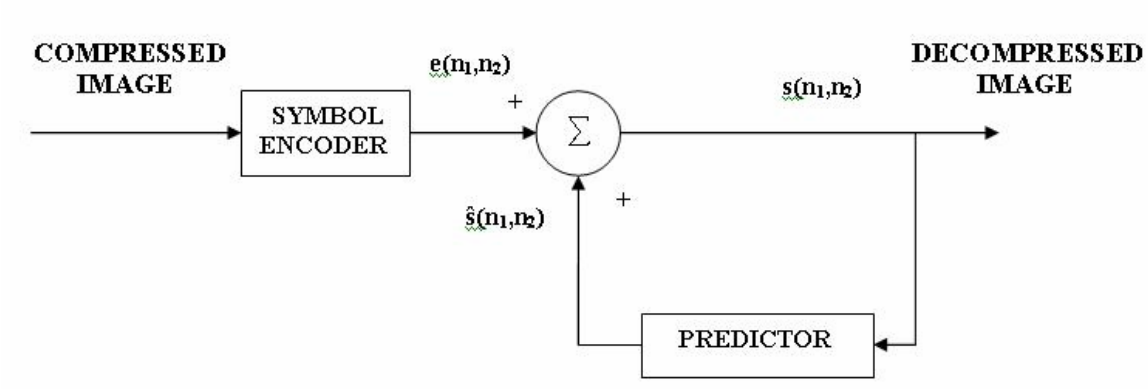

Fig 5.3 : *Lossless predictive decoder*

5.3.1 Example of Predictive Coding:

As an illustrative example, we are now going to apply linear predictive coding on an image, shown in Fig.5.4.

Fig5_4.pgm

Fig.5.4 An example image (Lena)

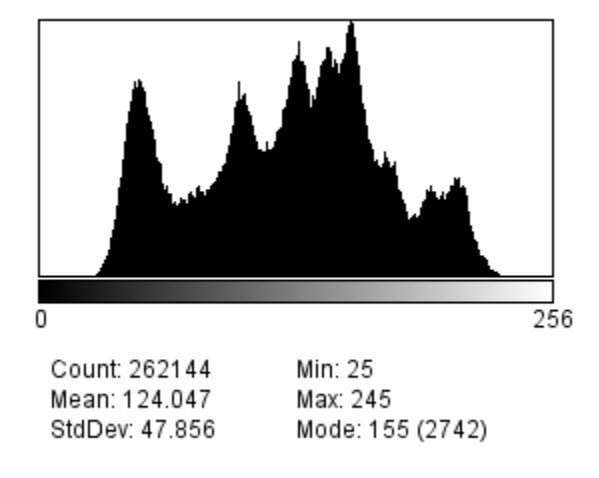

Fig 5.5 Histogram of original image

The corresponding histogram of image intensities is shown in Fig.5.5. For a large number of samples, the histogram approximates the probability density function (pdf) of the intensity and the probability Pi of the intensity level i ($i = 0,1,\cdots\cdots,255$) is given by $P_i = \frac{n_i}{N}$, where n_i is the number of pixels having intensity value i and N is the total pixel count. We can now apply equation (3.2) to measure the entropy of the original image and it is found to be (calculate the value). Now, we apply the closest past neighbor prediction, given in equation (5.4) to obtain a predicted image $\hat{s}(n_1, n_2)$. Subtracting the predicted image from the original, we obtain the error image, shown in Fig.5.6.

Fig5_6.pgm

Fig.5.6 *Error Image*

Since the error values can be positive or negative, we have added an offset of intensity value 127 on all the pixels to display the negative intensity values as well. It is clearly seen that the error image has very little intensity variations and its histogram is shown in Fig.5.7.

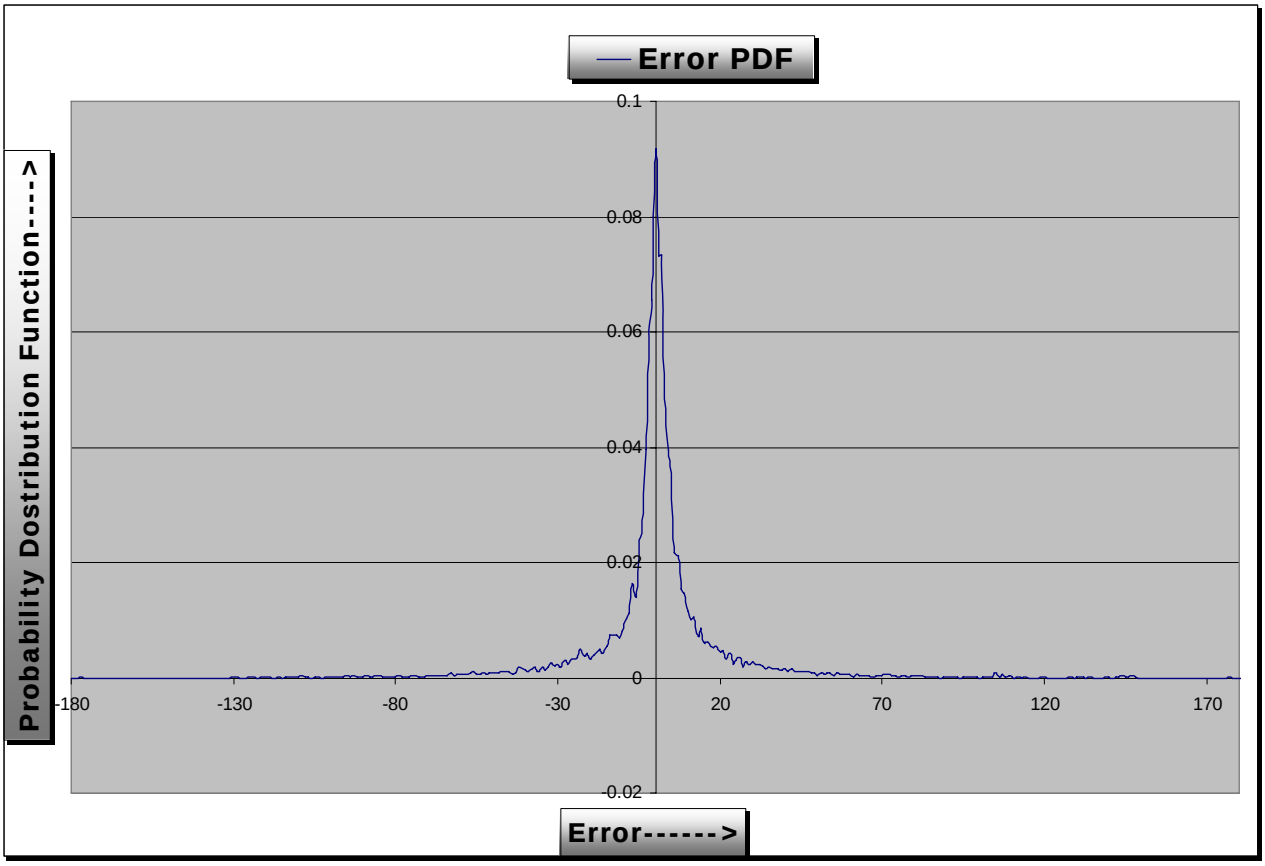

Fig 5.7 Histogram of error image

The histogram is peaked about 127 and by calculating the entropy on the error image, we obtain (calculate the value). The error image entropy being much lower than that of the original image, Shannon's theorem on noiseless coding suggests that it is possible to achieve considerable bit reduction, if we apply one of the entropy coding schemes, like *Arithmetic Coding* and *Huffman Coding* to encode the error image.

Questions

NOTE: The students are advised to thoroughly read this lesson first and then answer the following questions. Only after attempting all the questions, they should click to the solution button and verify their answers.

PART-A

A.1. How is it possible to convert a gray-scale image into bit-planes. Explain briefly.

A.2. State the basic principles of run-length coding of binary images.

A.3. Is it possible to apply run-length coding on gray-scale images? Explain.

A.4. Express the run-length entropy in terms of entropies and average run-lengths for the runs of 0s and 1s.

A.5. State the basic principles of lossless predictive coding.

A.6. Which has more entropy in lossless predictive coding of natural images – the original or the error image? Justify your answer.

PART-B: Multiple Choice

In the following questions, click the best out of the four choices.

B.1 Four consecutive pixels in a line have intensity values 221, 229, 143 and 186 in the range [0,255]. The bit-plane number-0 corresponds to the least significant bit. The bit-plane number-4 stores the following bit pattern to store the four pixels as above.

(A) 1011 (B) 0111

(C) 1010 (D) 1001

B.2 We have three binary images to encode using run-length coding. The first one is a checker-board image whose alternate pixels are black and white. The second one contains horizontal stripes of black and white in alternate lines. The third one contains vertical stripes of black and white – each having a width of 4 pixels. Rank according to the compressions achievable (highest to the lowest)

(A) First, second, third. (B) Second, third, first.

(C) Third, first, second. (D) Third, second, first.

B.3 Encode the bit stream
000000011111111111000000110101110001111111000000
using run-length coding, which always starts with runs of 1s. The corresponding code is

(A) 8,11,6,,2,1,1,1,3,3,9,6. (B) 0, 8,11,6,,2,1,1,1,3,3,9,6.

(C) 0, 8,11,6,,2,1,1,3,3,9,6 (D) 0, 8,11,6,,2,1,1,1,3,3,9,6,0.

B.4 In one line of an image, three consecutive pixel values are 37, 43 and 29. The next pixel is to be predicted by a linear prediction that is based on the last two pixels with a coefficient of 0.8 for the last and 0.2 for the second last. The predicted pixel with integer approximation is

(A) 32 (B) 38

(C) 39 (D) 40

B.5 A line of image containing consecutive pixel values 34, 33, 33, 37, 32, 35, 38, 35 is to be predicted by the previous pixel only. The corresponding error image has pixel values

(A) 0, -1, 0, 4, -5, 3, 3, -3 (B) -34, 1, 0, -4, 5, -3, -3, 3

(C) 34, -1, 0, 4, -5, 3, 3, -3 (D) , 1, 0, -4, 5, -3, -35.

B.6. Linear predictive coding will not lead to significant compression if

(A) the image has uniform intensity values.

(B) the image is a natural one.

(C) the image is very large in size.

(D) the image has only random noise with no spatial correlation.

PART-C: Computer Assignments

C-1. Pick up any monochrome image from the archive.
(a) Obtain the bit-plane mapped images for all the eight bit planes.
(b) Represent the image by a 8-bit gray code $g_7 g_6 \cdots\cdots g_1 g_0$ defined as follows:

$$g_7 = b_7$$
$$g_i = b_i \oplus b_{i+1}, \qquad 0 \le i \le 6$$

where, $b_7 b_6 \cdots\cdots b_1 b_0$ represents the binary values. Obtain the gray-coded bit plane images for all the eight planes.
(c) Compare the two sets of bit-plane images in (a) and (b). In what sense should gray-coded bit-planes be better? Justify your answer.

C-2. (a) On the above bit-plane images, perform 1-D run-length coding, as described in Section-5.2. If each run is to be represented by a 6-bit value, calculate the compression ratio (compressed bits: uncompressed bits) for (i) binary-mapped bit-planes and (ii) gray-coded bit-planes.

(b) From the statistics of gray-coded bit-planes, obtain the probabilities of the run-lengths. Assign Huffman codes to the run-lengths and calculate the compression ratio for the resulting encoded bit stream. Compare this result with that of (a)-(ii).

C-3. (a) Using the same monochrome image, obtain the predicted image and the error image, using $a_1 = a_2 = a_3 = a_4 = 0.25$ in equation (5.4).
(b) Compute the histograms and the entropies of the original image and the error image.

SOLUTIONS

A.1 A.2

A.3 A.4

A.5 A.6

B.1 (D) B.2 (B) B.3 (B)

B.4 (A) B.5 (C)B.6 (D)

C.1
C.2

REFERENCES

1.