LECTURE 07
IMAGE COMPRESSION

Punnarai Siricharoen, Ph.D.

OBJECTIVES

- To understand data, information, information theory and redundancy in an image
- To be able to describe image compression techniques including lossless and lossy techniques

CONTENT

- Fundamentals
- Image Compression Models
- Error-free compression 100% reconstruct
- Lossy compression ≠ 100% reconstruct
- Image compression standard – JPEG

FUNDAMENTALS

- *Image compression* involves reducing the size of image data files, while *retaining necessary information*
- Retaining necessary information depends upon the application
- Compression algorithm development starts with applications to two-dimensional (2-D) still images
- After the 2-D methods are developed, they are often extended to video (motion imaging). However, we will focus on image compression of a single frame of image data

FUNDAMENTALS

- The reduced file created by the compression process is called the *compressed image file* and is used to reconstruct the image, resulting in the *decompressed image file*
- The original image, before any compression is performed, is called the *uncompressed image file*
- The ratio of the original, uncompressed image file and the compressed file is referred to as the *compression ratio*

FUNDAMENTALS

To understand "*retaining necessary information*", we must differentiate between *data* and *information*

raw data

give same Info

8-bit

1. Data:

might redundancy → what we actually reduce

- For digital images, *data* refers to the pixel gray level values that correspond to the brightness of a pixel at a point in space
- Data are used to convey information, much like the way the alphabet is used to convey information via words

2. Information:

- Information is an interpretation of the data in a meaningful way; it can be application specific

COMPRESSION RATIO / BITS PER PIXEL

128 64 bit or something else

If n_1 and n_2 denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy R_D of the first data set (characterised by n_1) can be defined as

$$R_D = 1 - \frac{1}{C_R}$$

where C_R , commonly called the compression ratio, is

$$C_R = \frac{n_1}{n_2} = 2$$

when $n_1 = n_2$, $C_R = 1$ and $R_D = 0$ indicating that no redundant data relative to the first dataset.

when $n_2 << n_1$, $C_R \to \infty$ and $R_D \to 1$, implying significant compression.

COMPRESSION RATIO / BITS PER PIXEL

A discrete random variable r_k in the interval [0,1] represents the gray levels of an image and that each r_k occurs with probability $p_r(r_k)=\frac{n_k}{n}$, where k = 0,1,2,…,L-1. L is the number of gray levels, n_k is the number of times that kth gray level appears in the image, and n is the total number of pixels in the image. If the number of bits used to represent each value of r_k is $l(r_k)$, then the average number of bits required to represent each pixel is

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

The total number of bits required to code an $M \times N$ image is MNL_{avg}.

FUNDAMENTALS

There are two primary types of image compression methods:

- error-free
  Lossless compression methods: medical, remote sensor
  - Allows for the exact recreation of the original image data, and can compress complex images in range of 2 to 10 compression ratios
  - Preserves the data *exactly*
- *Lossy compression methods*
  - Compromising the accuracy of reconstructing image in exchange of increased compression
  - Data loss, original image cannot be re-created exactly
  - Can compress complex images 10:1 to 50:1 and retain high quality, and 100 to 200 times for lower quality, but acceptable images

FUNDAMENTALS

Compression algorithms are developed by taking advantage of the redundancy that is inherent in image data

Three primary types of redundancy that can be found in images are:

1. *Coding redundancy*
2. *Interpixel redundancy*
3. *Psychovisual redundancy*

FUNDAMENTALS

1. *Coding redundancy*

Occurs when the data used to represent the image is not utilized in an optimal manner

$L_{avg} = 3$

change a way

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

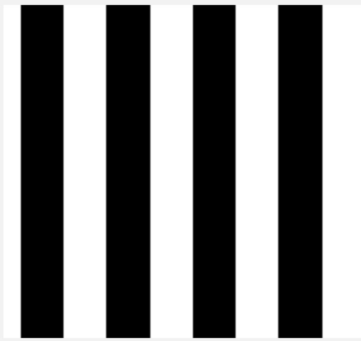
make Lavg reduce

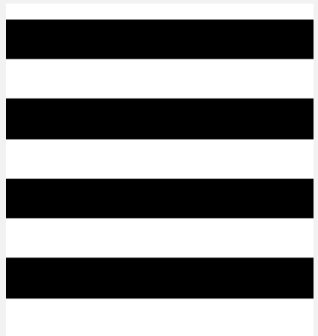
$$
\begin{aligned}
L_{avg} &= \sum_{k=0}^{7} l_2(r_k)p_r(r_k) \\
&= 2(0.19) + 2(0.25) + 2(0.21) + 3(0.16) + 4(0.08) \\
&\quad + 5(0.06) + 6(0.03) + 6(0.02) \\
&= 2.7 \text{ bits.}
\end{aligned}
$$

2. *Interpixel redundancy*

Occurs because adjacent pixels tend to be highly correlated, in most images the brightness levels do not change rapidly, but change gradually

ใช้ different between pixel แทน บอกตรงๆ





FUNDAMENTALS

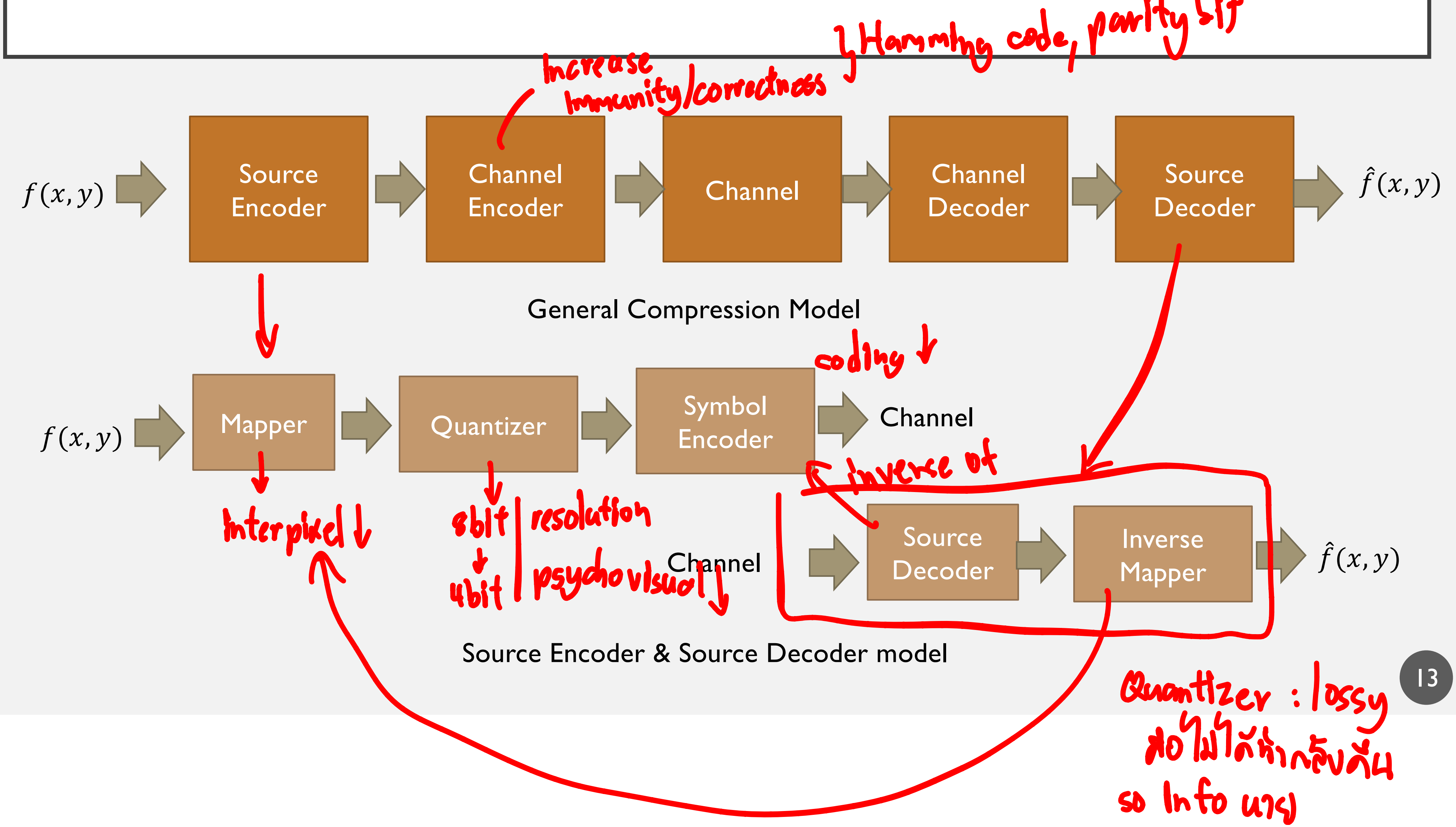
ลด Resolution

3. Psychovisual redundancy

Some information is more important to the human visual system than other types of information

Human perception qualitative analysis of every pixel value, human observe objects by distinguishing features, such as, edges, texture.

IMAGE COMPRESSION MODELS



General Compression Model

Source Encoder & Source Decoder model

ELEMENTS OF INFORMATION THEORY

for example
Intensity of particular value

- Measuring Information:
  - E – a random event, P(E)-probability of event E occurs
  - Unit of information: $I(E) = \log\frac{1}{P(E)} = -\log(\mathrm{P(E)})$ scale down the prob
    เพราะ $0 \leq P(E) \leq 1$ log แล้วติดลบเสมอ
  - If base m log is used, the measurement is in m-ary units; if base 2 log is selected, the resulting unit of information is called a bit.

ถ้า $P(E) = 1$
log แล้วได้ 0
แปลว่า no info

- Average Information per source output H(z)
  - A set of source value (intensity value) = $\{a_1, a_2, .., a_J\}$
  - $\sum_{j=1}^{J} P(a_j) = 1$
  - H(z) = $-\sum_{j=1}^{J} P(a_j) log_2 P(a_j)$
  - H(z) is also called uncertainty or entropy of the source

เราสามารถใช้เพื่อหาจำนวน bit
ที่เหมาะสมสำหรับ represent info นี้ได้

randomness

เช่น Intensity
2 สี อย่างละ $\frac{1}{2}, \frac{1}{2}$

$-1 \sum \frac{1}{2} \times -1 + \frac{1}{2} \times -1$

$-1 \times -1 = 1$

ตอบใช้ 1 bit/pixel
พอ

ELEMENTS OF INFORMATION THEORY

entropy = $-\left(2\left(\frac{3}{8}\log\left(\frac{3}{8}\right)\right)+2\left(\frac{1}{8}\log\left(\frac{1}{8}\right)\right)\right)$

= 1.81 bits / pixel

- Using information theory
  - $H(z) = -\sum_{j=1}^{J} P(a_j) log_2 P(a_j)$

Estimating the information content (or entropy) of the simple 8-bit image

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

interpixel redun เพราะค่าใกล้ๆ คล้ายคลึง

Gray Level	Count	Probability
21	12	3/8
95	4	1/8
169	4	1/8
243	12	3/8

The entropy of the image is approximately 1.81 bits/pixel or 58 total bits. 1.81 × 8 × 4

ELEMENTS OF INFORMATION THEORY

- Using information theory
  - $H(z) = -\sum_{j=1}^{J} P(a_j) log_2 P(a_j)$

$$\text{Entropy} = -\left(2\left(\frac{1}{4}\log_2\frac{1}{4}\right) + 4\left(\frac{1}{8}\log_2\frac{1}{8}\right)\right)$$

$= 2.5$ แต่เป็น pair

จริงๆ เลย ≈ 1.25 bit/pixel

Estimating the information content (or entropy) of the simple 8-bit image จับคู่ ลด inter pixel

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

Gray-level Pair	Count	Probability
(21, 21)	8	1/4
(21, 95)	4	1/8
(95, 169)	4	1/8
(169, 243)	4	1/8
(243, 243)	8	1/4
(243, 21)	4	1/8

The resulting entropy estimate is 2.5/2 or 1.25 bits / pixel, where division by 2 is a consequence of considering two pixels at a time.

ELEMENTS OF INFORMATION THEORY

- Using information theory
  - $H(z) = -\sum_{j=1}^{J} P(a_j) log_2 P(a_j)$

หา different จาก pixel ก่อนหน้าใน row

Estimating the information content (or entropy) of the simple 8-bit image

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

mapping to reduce entropy

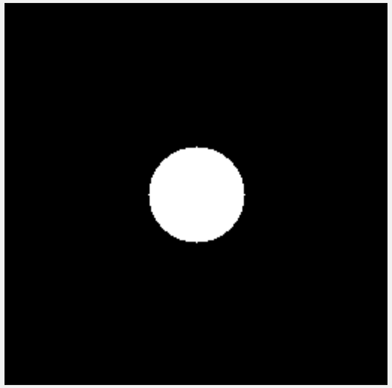
21	0	0	74	74	74	0	0
21	0	0	74	74	74	0	0
21	0	0	74	74	74	0	0
21	0	0	74	74	74	0	0

Gray Level or Difference	Count	Probability
0	12	1/2
21	4	1/8
74	12	3/8

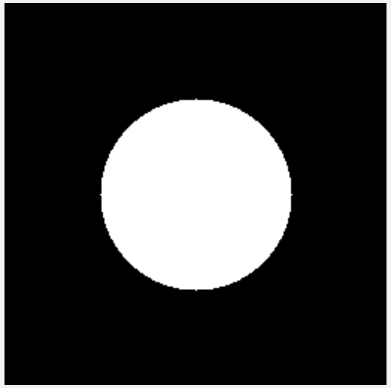
By the variable-length coding, the mapped difference image, the original image can be represented with only 1.41 bits / pixel

Original image,
entropy = 7.032 *bpp*

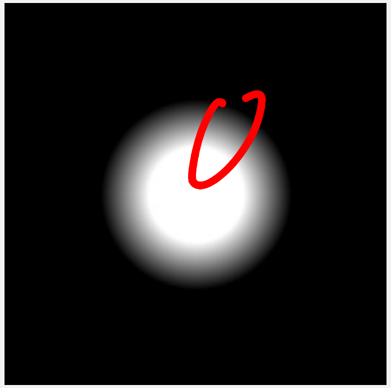
Image after binary threshold,
entropy = 0.976 *bpp*



Circle with a radius of 32,
entropy = 0.283 *bpp*



Circle with a radius of 64,
entropy = 0.716 *bpp*



Circle with a radius of 32,
and a linear blur radius of 64,
entropy = 2.030 *bpp*

Reference: Scott E Umbaugh, SIUE 2005

ERROR-FREE COMPRESSION

- Only acceptable means of data reduction
- For example, medical & business documents, lossy compression is prohibited for legal reasons
- Satellite image -> high cost of collecting data -> loss is undesirable
- Digital Radiography -> loss of info compromising diagnostic accuracy
- Compression ratio: 2 - 10

ERROR-FREE COMPRESSION

- 1.Variable-Length Coding บางสี ใช้bit เยอะ, บางใช้ bit น้อย
  ไม่ได้ constant # of bit (fixed bit)
  - Reducing only coding redundancy
  - To construct a variable-length code that assigns the shortest possible codeword to the most probable gray scale

ERROR-FREE COMPRESSION

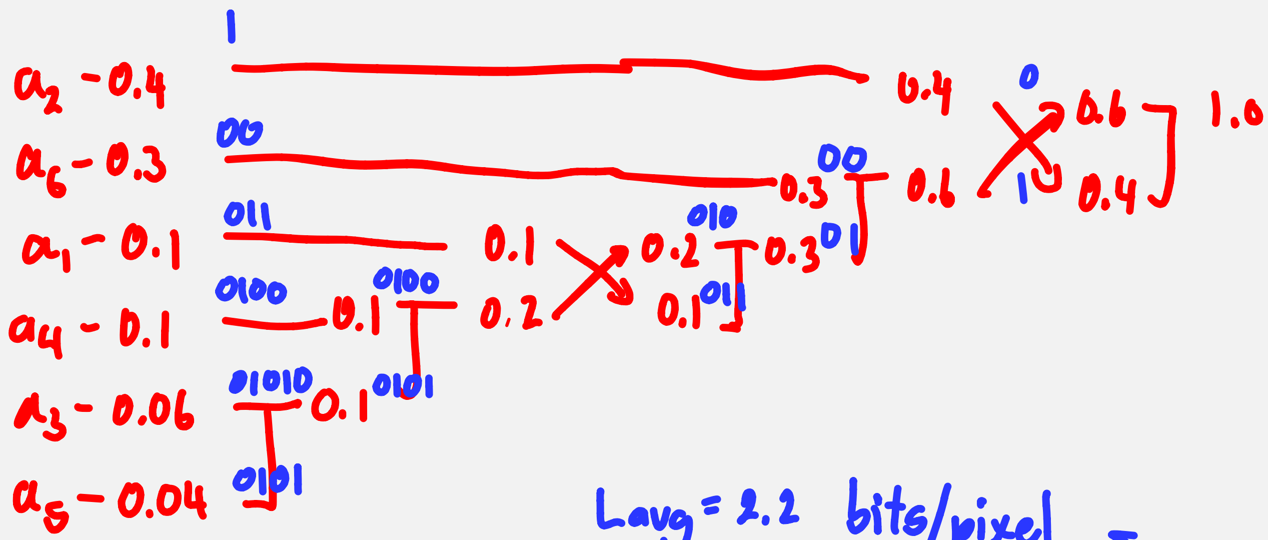
- 1.1) Huffman coding
  - The most popular technique for removing coding redundancy
  - Yield smallest possible number of code symbols
  - **Step**

1. Find the gray level probabilities for the image by finding the histogram
2. Order the input probabilities (histogram magnitudes) from largest to smallest
3. Combine the smallest two by addition
4. GOTO step 2, until only two probabilities remain
5. By working backward along the tree, generate code by alternating assignment of 0 and 1

ERROR-FREE COMPRESSION

- Example: Huffman coding

Symbol	Prob
a_1	0.1
a_2	0.4
a_3	0.06
a_4	0.1
a_5	0.04
a_6	0.3



L_{avg} = 2.2 bits/pixel

entropy = 2.14 bits/pixel } very close to ideal

ERROR-FREE COMPRESSION

- 1.2) Arithmetic coding
  - Arithmetic coding transforms input data into a single floating point number between 0 and 1 or [0,1)
  - Arithmetic coding uses the probability distribution of the data (histogram), so it can theoretically achieve the maximum compression specified by the entropy
  - It works by successively subdividing the interval between 0 and 1, based on the placement of the current pixel value in the probability distribution

ERROR-FREE COMPRESSION

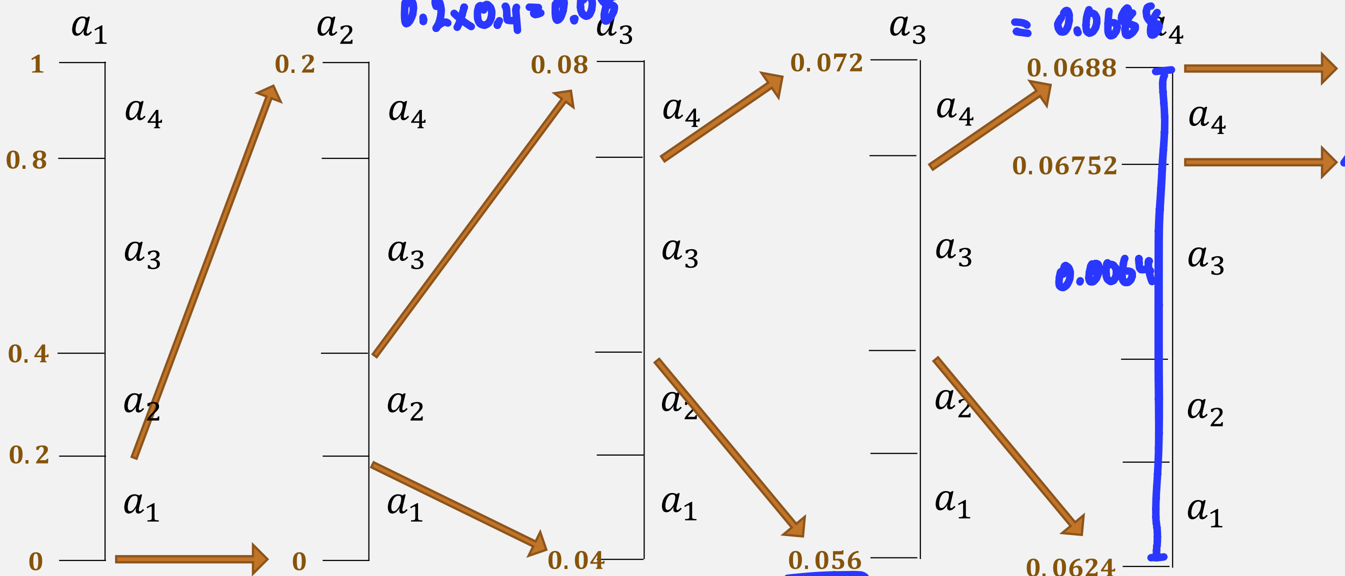
- Example: Arithmetic coding
  - A = $\{a_1, a_2, a_3, a_3, a_4\}$; P($a_1$)=0.2, P($a_2$)=0.2, P($a_3$) = 0.4, P($a_4$) = 0.2;

Encoding sequence

0.04×0.8+0.04 = =0.072
0.032 +0.04

0.016×0.4+0.056
= 0.0688

0.2×0.4=0.08



0.0064×0.8 + 0.0624 = 0.06752

Select any value in [0.6752,0.0688)
e.g., select -> 0.06787109375
has binary value → 0.00010001011

so we need 11 bits
for 5 pixels : 2.2 bits/pixel
but fixed bit used just 2
entropy = 1.92

0.2×0.2 =0.04

0.04×0.4 +0.04
= 0.016 +0.04 =0.056

0.016×0.4 + 0.056
= 0.0624

LZW CODING

- 2) LZW is called Lempel-Ziv Welch coding, which assigns fixed-length code words to variable length sequences of source symbols but require no a priori knowledge of the probablility of occurrence of the symbols to be endcoded.
- Attack an image's interpixel redundancies
- Integrated into a variety of main stream imaging file formats, including the graphic interchange format (GIF), tagged image file format (TIFF) and the portable document format (PDF)

repetitive pattern : so good for Image have redundancy pattern

- A codebook or "dictionary" containing the source symbols to be coded is constructed. For 8-bit grayscale image, the first 256 values are assigned for 0,… 255. Gray-level sequences that are not in the dictionary are assigned to the next location (256 and upper)

using 9 bits 0-255 : remain gray

256 ++ : allocate by dictionary

LZW CODING

only use 90 bits for this picture

- Consider the 4x4, 8-bit image of a vertical edge:

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

processing pixels in left-to-right and top-to-bottom manner

- A 512-word dictionary with the following starting content is assumed:

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 coding example

ERROR-FREE COMPRESSION

Bit-plane decomposition

The gray levels of an m-bit gray-scale image can be represented in the form of the base 2 polynomial

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \cdots + a_1 2^1 + a_0 2^0.$$

Based on this property, a simple method of decomposing the image into a collection of binary images is to separate the m coefficients of the polynomial into m 1-bit *bit planes*.

- 3) Bit-plane coding

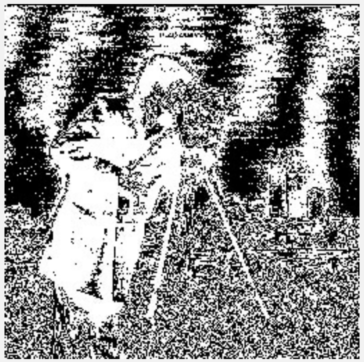

a) Original image


b) Bit plane 7, the most significant bit

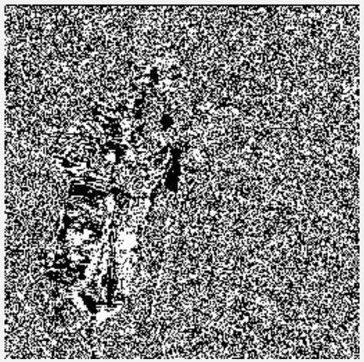

c) Bit plane 6

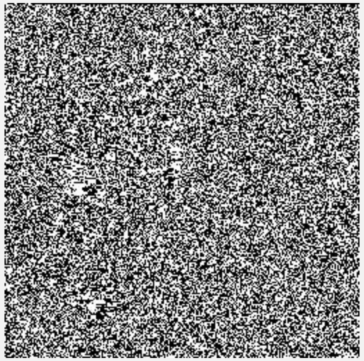

d) Bit plane 5


e) Bit plane 4


f) Bit plane 3


g) Bit plane 2


h) Bit plane 1


i) Bit plane 0, the least significant bit

Use bit-plane coding along with constant area coding and run-length coding

ERROR-FREE COMPRESSION

- Run-length coding (RLC) → BMP, TIFF
- RLC works by counting adjacent pixels with the same gray level value called the run-length, which is then encoded and stored

  ตัวติดกันเป็นก้อนๆ

  - RLC works best for binary, two-valued, images

    ไม่ต้องไปผ่าน Thresholding

  - RLC can also work with complex images that have been preprocessed by thresholding to reduce the number of gray levels to two
  - RLC can be implemented in various ways, but the first step is to define the required parameters. Horizontal RLC (counting along the rows) or vertical RLC (counting along the columns) can be used
  - In basic horizontal RLC, the number of bits used for the encoding depends on the number of pixels in a row

    นับเป็น row ๆ ไป

  - If the row has 2^n pixels, then the required number of bits is n, so that a run that is the length of the entire row can be encoded

ERROR-FREE COMPRESSION

Binary

- Example: Run-length coding (RLC)
  - A binary image of 8x8 pixels, required 3-bit for each run-length coded word. To apply this RLC, use horizontal RLC. For binary image, the convention used is that the first number is the number of zeros in the run.

ค่า 0 แทนได้, เพราะ: 0 เสียง, เป็นไปไม่ได้

number of zero

$8 = 2^n = \underline{3}$ bits

$$
\begin{array}{cl}
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
&
\begin{array}{l}
8 \\
0,4,4 \\
1,2,5 \\
1,5,2 \\
1,3,2,1,1 \\
2,1,2,2,1 \\
0,4,1,1,2 \\
8
\end{array}
\end{array}
$$

$\underline{3}(1+3+3+3+5+5+5+1)$

= number of bit needed

ERROR-FREE COMPRESSION

(color, #run) gray scales

- Example: Run-length coding (RLC)
  - Another way to extend basic RLC to gray level images is to include the gray level of a particular run as part of the code
  - Here, instead of a single value for a run, two parameters are used to characterize the run
  - The pair (G,L) correspond to the gray level value, G, and the run length, L
  - This technique is only effective with images containing a small number of gray levels

8 x 8 4-bit image

$$\begin{bmatrix} 10 & 10 & 10 & 10 & 10 & 10 & 10 & 10 \\ 10 & 10 & 10 & 10 & 10 & 12 & 12 & 12 \\ 10 & 10 & 10 & 10 & 10 & 12 & 12 & 12 \\ 0 & 0 & 0 & 10 & 10 & 10 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 & 0 & 1 & 0 \\ 5 & 5 & 5 & 10 & 10 & 9 & 9 & 10 \\ 5 & 5 & 5 & 4 & 4 & 4 & 0 & 0 \\ 0. & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

4 bit 3 bit

(10, 8)
(10,5)(12,3)
same
(0,3)(10,3)(0,2)
(5,3)(0,3)(1,1)(0,1)
(5,3)(10,2)(9,2)(10,1)
(5,3)(4,3)(2,2)
(0,8)

coding redundancy : Huffman, Arithmetic

Interpixel : LZW, RLC

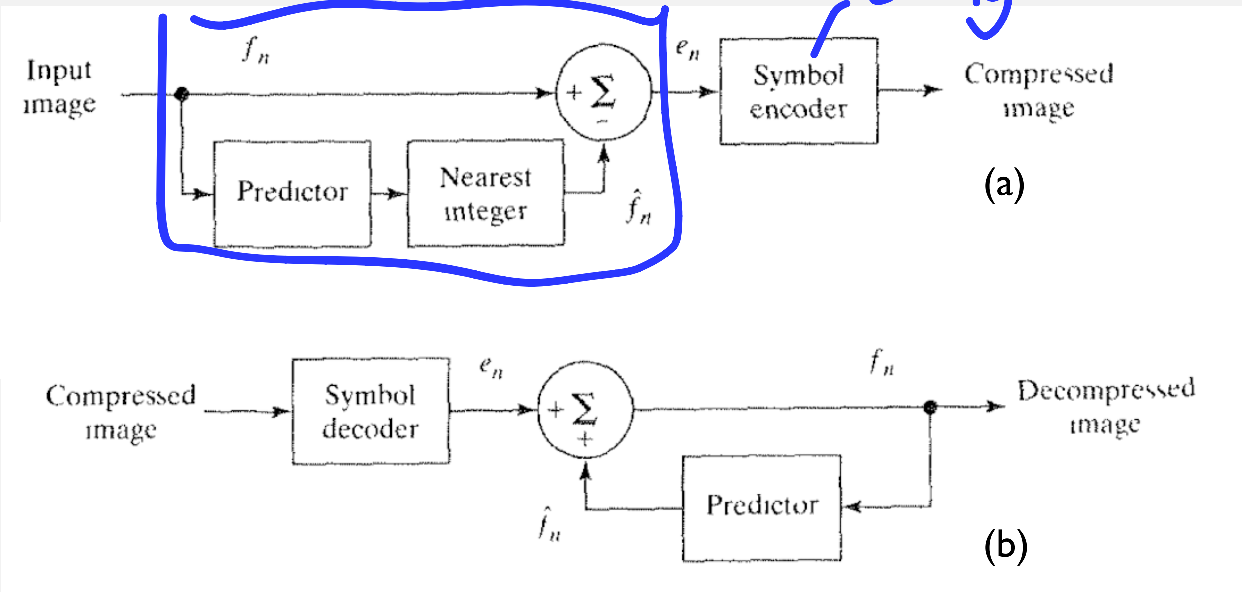
psychovisual : Quantization, DCT

ERROR-FREE COMPRESSION

general form

- 4) Lossless predictive coding

Interpixel



lossless predictive coding model:
(a) encoder,
(b) decoder

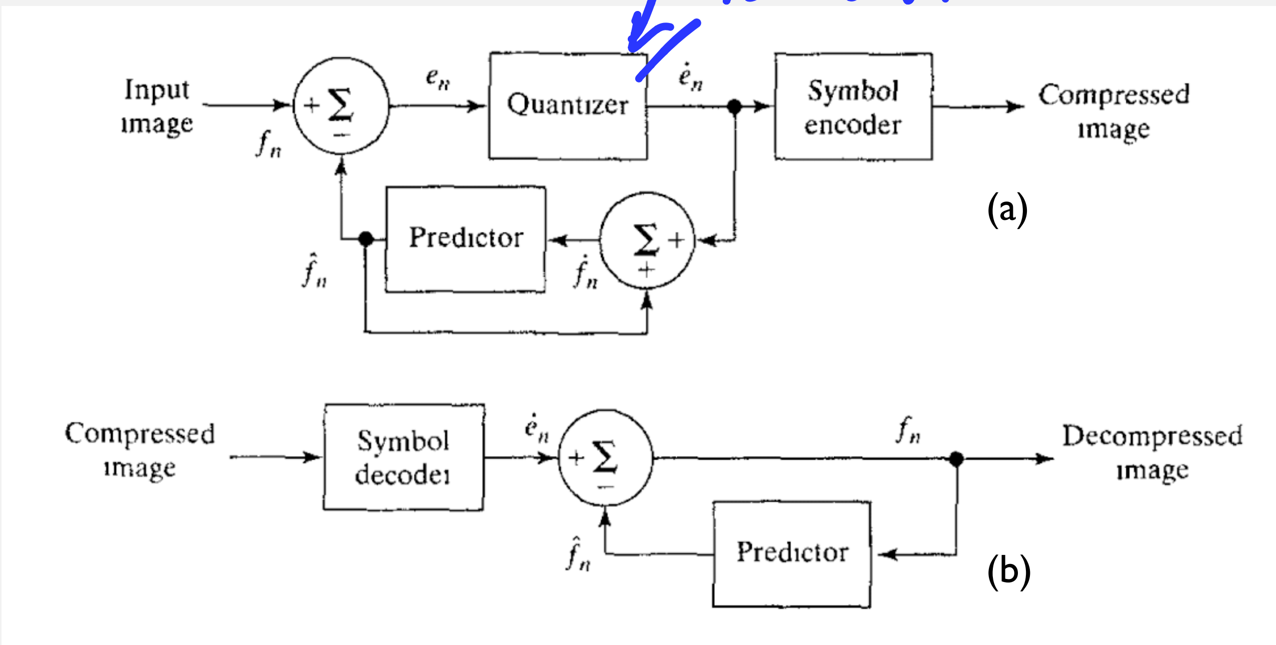
$$e_n = f_n - \hat{f}_n,$$

Prediction error – coded using var-length code
To generate the next element of compressed data stream

LOSSY COMPRESSION

- 1) Lossy predictive coding

reduce bit



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

Add quantizer which absorbs the nearest integer function, map prediction error to limited range of output

LOSSY COMPRESSION

- 2) Transform coding
  - Transform selection
    - Any of the previously defined transforms can be used, frequency (e.g. Fourier) or sequency (e.g. Walsh/Hadamard), but it has been determined that the discrete cosine transform (DCT) is optimal for most images

      ↳ real number only PT

      มันมีแค่ real so reconstruct ไม่ได้ 100% → lossy
    - The JPEG2000 algorithms uses the wavelet transform, which has been found to provide even better compression

LOSSY COMPRESSION

แปลงเป็นใน Freq Domain แล้วส่งเฉพาะส่วน LF
เพราะ HF มีค่า 0 ซะส่วนใหญ่

- 2) Transform coding
  - An image f(x,y) of size NxN whose, forward discrete transform T(u,v) can be expressed in terms of the general relation

Transform function

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

  - Inverse transform:

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

$g(x,y,u,v)$ and $h(x,y,u,v)$ are forward and inverse transformation kernels.

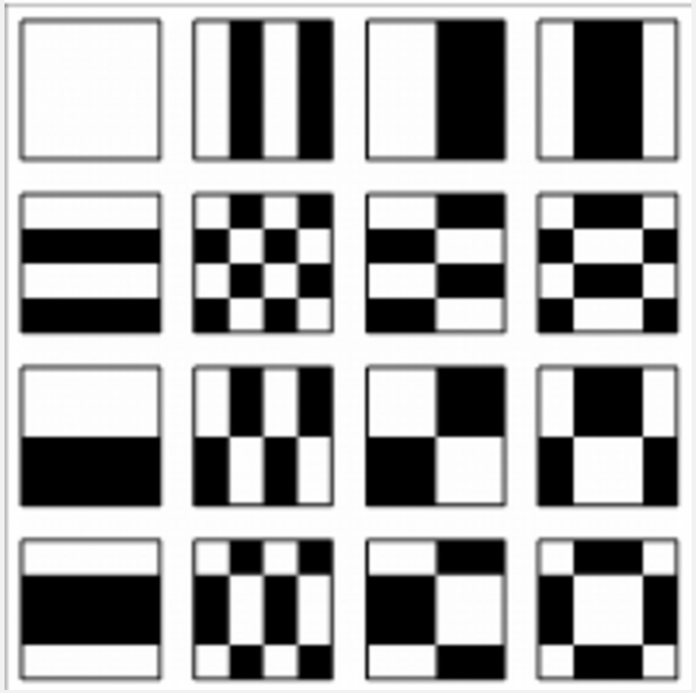
LOSSY COMPRESSION

- 2) Transform coding

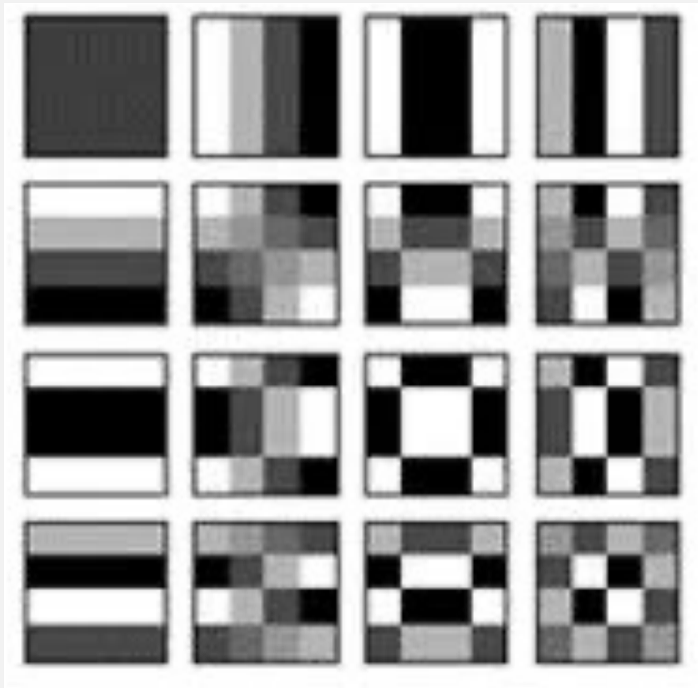
only 0,1

DCT basis

Walsh Hadamard basis function



discrete cosine basis function (4x4)



LOSSY COMPRESSION

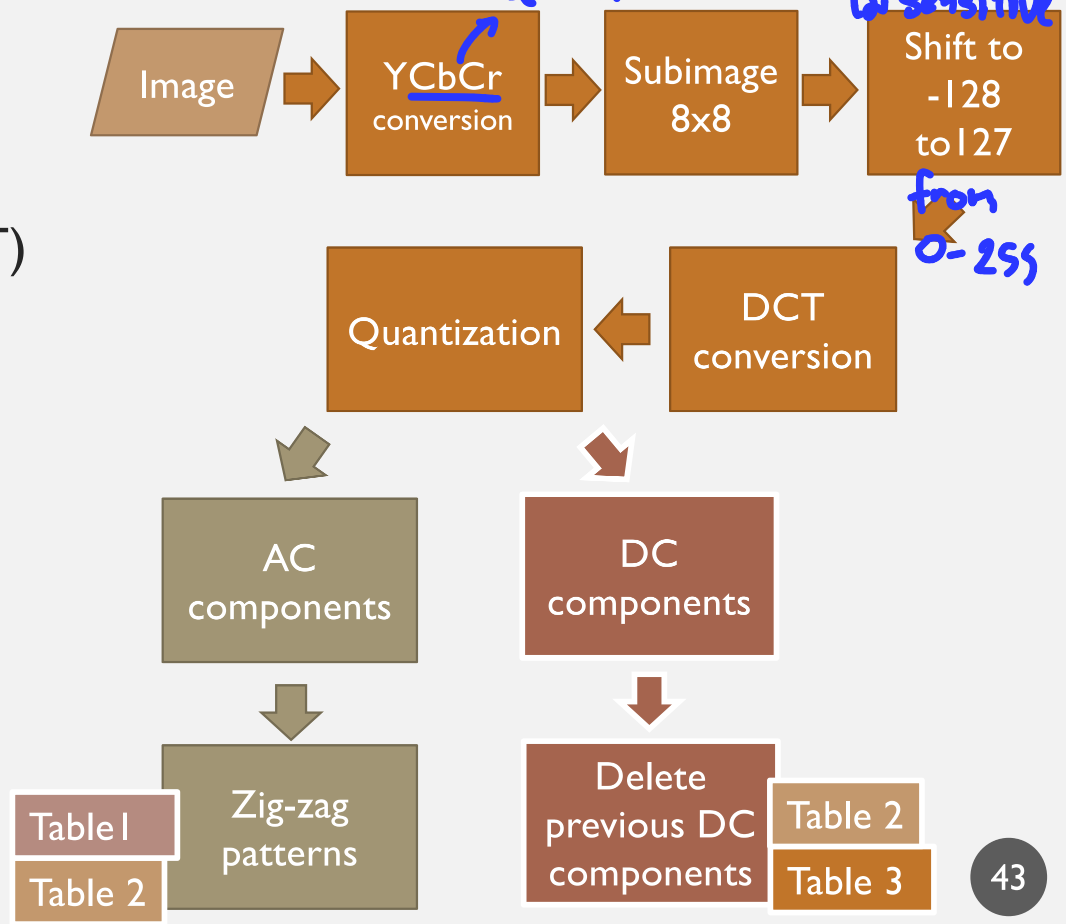
- 2) Transform coding
  - Transform selection

  - Transform a 8x8 image using Hadamard Basis

JPEG – JOINT PHOTOGRAPHIC EXPERTS GROUP

reduce all 3 redundancies

- JPEG – Joint Photographic Experts Group
  - ISO/IEC JTC1 SC29 WG1
  - Formed in 1986 by ISO and CCITT (ITU-T)
  - Became international standard in 1991
  - Compression ratio 10-50
  - Digital compression and coding of continuous-tone still Images (Grayscale or color)



LOSSY COMPRESSION

- JPEG

Consider compression and reconstruction of the following 8 x 8 subimage with JPEG baseline standard

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	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

−128

Level shifting

The original image consists of 256 or 2^8 possible gray levels, so the coding process begins by level shifting the pixels of the original subimage by -128 or -2^7 gray levels. The resulting shifted array is

−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

LOSSY COMPRESSION

- JPEG

The shifted level subimage

−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

DCT

The shifted level subimage is then transformed using DCT for N = 8, becomes

ค่อนข้างสูงใน LF

−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

LF

HF

น้อย

LOSSY COMPRESSION

- JPEG

JPEG recommended normalization array is used to quantize the transformed array, the scaled and truncated coeffficients are

Quantization

DC

AC

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & 0 & 0 & 0 \\ 1 & -2 & -4 & 0 & 0 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Normalized and transformed image $(\hat{T})$

The DC coefficient is computed as

$$\hat{T}(0,0) = \text{round}\left[\frac{T(0,0)}{Z(0,0)}\right] = \text{round}\left[\frac{-415}{16}\right] = -26.$$

Where $\hat{T}(0,0)$ is the noramlized coefficients, T is the transformed image, Z is the typical normalization array.

16	11	10	16	124	140	151	161
12	12	14	19	126	158	160	155
14	13	16	24	140	157	169	156
14	17	22	29	151	187	180	162
18	22	37	56	168	109	103	177
24	35	55	64	181	104	113	192
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	199

Luminance

y

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Chrominance

CbCr

Typical JPEG quantizer (Z)

shows a typical normalization array. This array, which has been used extensively in the JPEG[†] standardization efforts (see Section 8.6.2), weighs each coefficient of a transformed subimage according to heuristically determined perceptual or psychovisual importance.

AC

RLC

LOSSY COMPRESSION

- JPEG - Table

Table I

JPEG default AC code

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
0/0	**1010 (= EOB)**	**4**			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	111111111000000	17
0/3	100	6	8/3	1111111110110111	19
0/4	1011	8	8/4	1111111110111000	20
0/5	11010	10	8/5	1111111110111001	21
0/6	111000	12	8/6	1111111110111010	22
0/7	1111000	14	8/7	1111111110111011	23
0/8	1111110110	18	8/8	1111111110111100	24
0/9	1111111110000010	25	8/9	1111111110111101	25
0/A	1111111110000011	26	8/A	1111111110111110	26
1/1	1100	5	9/1	111111000	10
1/2	111001	8	9/2	1111111110111111	18
1/3	1111001	10	9/3	1111111111000000	19
1/4	111110110	13	9/4	1111111111000001	20
1/5	11111110110	16	9/5	1111111111000010	21
1/6	1111111110000100	22	9/6	1111111111000011	22
1/7	1111111110000101	23	9/7	1111111111000100	23
1/8	1111111110000110	24	9/8	1111111111000101	24
1/9	1111111110000111	25	9/9	1111111111000110	25
1/A	1111111110001000	26	9/A	1111111111000111	26
2/1	11011	6	A/1	111111001	10
2/2	11111000	10	A/2	1111111111001000	18
2/3	1111110111	13	A/3	1111111111001001	19
2/4	1111111110001001	20	A/4	1111111111001010	20
2/5	1111111110001010	21	A/5	1111111111001011	21
2/6	1111111110001011	22	A/6	1111111111001100	22
2/7	1111111110001100	23	A/7	1111111111001101	23
2/8	1111111110001101	24	A/8	1111111111001110	24
2/9	1111111110001110	25	A/9	1111111111001111	25
2/A	1111111110001111	26	A/A	1111111111010000	26
3/1	111010	7	B/1	111111010	10
3/2	111110111	11	B/2	1111111111010001	18
3/3	11111110111	14	B/3	1111111111010010	19
3/4	1111111110010000	20	B/4	1111111111010011	20
3/5	1111111110010001	21	B/5	1111111111010100	21
3/6	1111111110010010	22	B/6	1111111111010101	22
3/7	1111111110010011	23	B/7	1111111111010110	23
3/8	1111111110010100	24	B/8	1111111111010111	24
3/9	1111111110010101	25	B/9	1111111111011000	25
3/A	1111111110010110	26	B/A	1111111111011001	26
4/1	111011	7	C/1	1111111010	11
4/2	1111111000	12	C/2	1111111111011010	18
4/3	1111111110010111	19	C/3	1111111111011011	19
4/4	1111111110011000	20	C/4	1111111111011100	20
4/5	1111111110011001	21	C/5	1111111111011101	21
4/6	1111111110011010	22	C/6	1111111111011110	22
4/7	1111111110011011	23	C/7	1111111111011111	23
4/8	1111111110011100	24	C/8	1111111111100000	24
4/9	1111111110011101	25	C/9	1111111111100001	25
4/A	1111111110011110	26	C/A	1111111111100010	26

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
5/1	1111010	8	D/1	11111111010	12
5/2	1111111001	12	D/2	1111111111100011	18
5/3	1111111110011111	19	D/3	1111111111100100	19
5/4	1111111110100000	20	D/4	1111111111100101	20
5/5	1111111110100001	21	D/5	1111111111100110	21
5/6	1111111110100010	22	D/6	1111111111100111	22
5/7	1111111110100011	23	D/7	1111111111101000	23
5/8	1111111110100100	24	D/8	1111111111101001	24
5/9	1111111110100101	25	D/9	1111111111101010	25
5/A	1111111110100110	26	D/A	1111111111101011	26
6/1	1111011	8	E/1	111111110110	13
6/2	11111111000	13	E/2	1111111111101100	18
6/3	1111111110100111	19	E/3	1111111111101101	19
6/4	1111111110101000	20	E/4	1111111111101110	20
6/5	1111111110101001	21	E/5	1111111111101111	21
6/6	1111111110101010	22	E/6	1111111111110000	22
6/7	1111111110101011	23	E/7	1111111111110001	23
6/8	1111111110101100	24	E/8	1111111111110010	24
6/9	1111111110101101	25	E/9	1111111111110011	25
6/A	1111111110101110	26	E/A	1111111111110100	26
7/1	11111001	9	**F/0**	**111111110111**	**12**
7/2	11111111001	13	F/1	1111111111110101	17
7/3	1111111110101111	19	F/2	1111111111110110	18
7/4	1111111110110000	20	F/3	1111111111110111	19
7/5	1111111110110001	21	F/4	1111111111111000	20
7/6	1111111110110010	22	F/5	1111111111111001	21
7/7	1111111110110011	23	F/6	1111111111111010	22
7/8	1111111110110100	24	F/7	1111111111111011	23
7/9	1111111110110101	25	F/8	1111111111111100	24
7/A	1111111110110110	26	F/9	1111111111111101	25
			F/A	1111111111111110	26

Table 2

JPEG coefficient coding categories.

AC/DC

Range	DC Difference Category	AC Category
0	0	N/A
−1, 1	1	1
−3, −2, 2, 3	2	2
−7, …, −4, 4, …, 7	3	3
−15, …, −8, 8, …, 15	4	4
−31, …, −16, 16, …, 31	5	5
−63, …, −32, 32, …, 63	6	6
−127, …, −64, 64, …, 127	7	7
−255, …, −128, 128, …, 255	8	8
−511, …, −256, 256, …, 511	9	9
−1023, …, −512, 512, …, 1023	A	A
−2047, …, −1024, 1024, …, 2047	B	B
−4095, …, −2048, 2048, …, 4095	C	C
−8191, …, −4096, 4096, …, 8191	D	D
−16383, …, −8192, 8192, …, 16383	E	E
−32767, …, −16384, 16384, …, 32767	F	N/A

Table 3

JPEG default DC code (luminance).

DC

Category	Base Code	Length	Category	Base Code	Length
0	010	3	6	1110	10
1	011	4	7	11110	12
2	100	5	8	111110	14
3	00	5	9	1111110	16
4	101	7	A	11111110	18
5	110	8	B	111111110	20

LOSSY COMPRESSION

- JPEG – coefficients are..

Zig zag pattern

−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

เพื่อ encode ทั้ง LF

DC values of the previous block is -17
-26-(-17) = -9 (Cat 4 in Table 2)
Base code is 101 _ _ _ _ (Table 3)
9 -> 1001
-9 -> 0110 (1' complement of 9)
Code = 1010110

ไม่มีศูนย์ก่อนหน้า

AC -3 is in Cat 2 in Table 2, run/cat = 0/2
Base code is 01
3 -> 11, -3 -> 00 (1' complement of 3)
Code = 0100

coefficient sequence is

[−26 −3 1 −3 −2 −6 2 −4 1 −4 1 1 5 0 2 0 0 −1 2 $\boxed{0\ 0\ 0\ 0\ 0}$ −1 −1 EOB]

1010110 −9

5 zeros = 5 run

Run/cat = 5/1
Base code = 1111010
-1 -> 0 จาก 1 = 1, -1 = 0
Code = 11110100

The completely coded array:

$\boxed{1010110}$ $\boxed{0100}$ 001 0100 0101 100001 0110 100011 001 100011 001
001 100101 11100110 110110 0110 $\boxed{11110100}$ 000 1010

SUMMARY

	Quantization	Coding	Interpixel	Interband
JPEG	Quantization using DCT	Huffman coding (past – also Arithmetic coding)	Use 8x8 block Run-length coding	YCbCr color components
JPEG2000 JPEG-LS	Quantization using Wavelet Lossy+lossless	Arithmetic coding	Embeddded block coding with optional truncation (EMCOT)	YCbCr color components
PNG		DEFLATE = Huffman + LZ77		
BMP TIFF			LZW coding	
Lossy/lossless Predictive Coding DPCM	Quantizer (only lossy)	Symbol (no specified)	Predictor (based on previouse pixels)	

DPCM – Diffirential pulse-code modution – for multimedia used for further reduction of the required data bandwidth for a given signal-to-noise ratio

REFERNCES

- Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Addison- Wesley
  - Chapter 8 – Image Compression