AY 2024/2025 Semester 1

EE6427 Video Signal Processing

Part 2 Image Compression & JPEG Standard

Dr Yap Kim Hui

Room: S2-B2b-53
Tel: 6790 4339
Email: ekhyap@ntu.edu.sg

References

- Ze-Nian Li, Mark S. Drew, Jiangchuan Liu, Fundamentals of Multimedia, Springer, 2022.
- R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4nd edition, Pearson, 2018.
- Saeed Vaseghi, Multimedia Signal Processing, Wiley, 2007.
- Fred Halsall, Multimedia communications: applications, networks, protocols and standards, Addison-Wesley, 2001.
- Yun Q. Shi and Huifang Sun, Image and video compression for multimedia engineering: fundamentals, algorithms, and standards, CRC Press, 2000.

Part 2 Outline

- Terms and Concepts
- Entropy Coding
- Image & Video Compression Basics
- Transform-based Coding / Compression
- Discrete Cosine Transform (DCT)
- JPEG Standard

Terms and Concepts

What is Compression / Coding?

- To reduce the total number of bits needed to represent certain information.

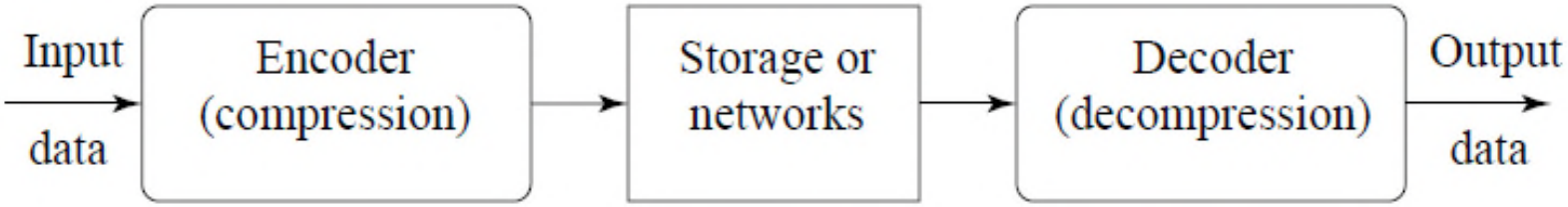


Fig. 7.1: A General Data Compression Scheme.

Compression Ratio

- A metric to indicate how much the data has been compressed.

$$compression\ ratio = \frac{B_0}{B_1}$$

B_0 – number of bits before compression
B_1 – number of bits after compression

Basics of Information Theory

- The *entropy* η of an information *source* with alphabet $S = \{s_1, s_2, \ldots, s_n\}$ is:

$$\eta = H(S) = \sum_{i=1}^{n} p_i \log_2 \frac{1}{p_i} \tag{7.2}$$

$$= -\sum_{i=1}^{n} p_i \log_2 p_i \tag{7.3}$$

p_i – probability that symbol s_i will occur in S.

Entropy Coding

Huffman Coding

- A type of Variable Length Coding (VLC).
- Higher frequency patterns (symbols) are assigned shorter codewords.
- Average number of bits used to represent each pattern/ symbol will be reduced.

Huffman Coding (1)

Example:

String of character:

AAAABBCD

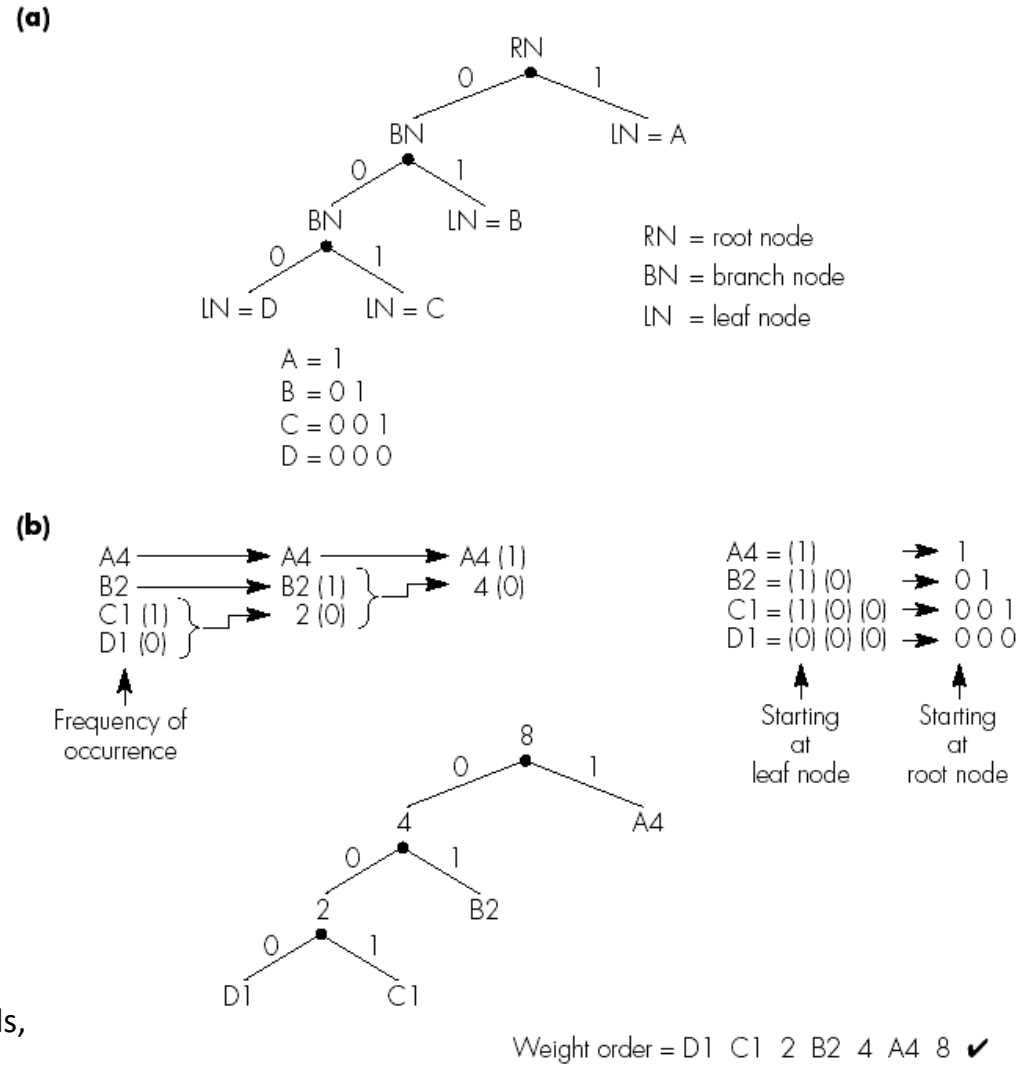


Figure source:

Fred Halsall, Multimedia communications: applications, networks, protocols and standards, Addison-Wesley, 2001

Huffman Coding (2)

- A Huffman code tree is a binary tree with branches assigned value 0 or 1.
- Tree structure
  - Root node: the root of tree.
  - Branch node: the point at which a branch divides.
  - Leaf node: the termination point of a branch.
- At each branch node, 0 and 1 are assigned to the left and right branches accordingly.
- The codeword used for each character/symbol is determined by tracing the path from the root node to the leaf node.

Exercise: Huffman Coding 1

- In a compression scheme, eight symbols are used to represent different patterns in an information source. The symbols are encoded using codewords of codebook A. The codebook A and the probabilities of occurrences are given in the following table, where *m* and *n* are positive real numbers.

Symbol	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7
Probability of occurrence	*m*	*n*	0.15	0.10	0.08	0.06	0.05	0.02
Codeword of Codebook A	01	111	110	101	100	001	0001	0000

Exercise: Huffman Coding 1

a) In the first scenario, if the average number of bits/symbol for the compression scheme needs to be less than 2.86, find the required conditions for the values of *m* and *n*.

b) In the second scenario, if the value of m=0.30 and n=0.24, discuss the effectiveness of Codebook A in encoding the symbols to achieve compression. Compare Codebook A with Huffman coding, discuss whether Codebook A is optimal in performing compression in this case.

Solution

Exercise: Huffman Coding 2

(a) In a compression scheme, a data source consists of eight symbols, with the probability distribution given in Table 1.

Table 1

Symbol	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Probability of occurrence	0.02	0.05	0.08	0.10	0.14	0.16	0.19	0.26

(i) Design a suitable set of Huffman codewords for the eight symbols. Clearly show all the key steps and calculations.

(8 Marks)

(ii) A student originally uses 8 bits to represent each symbol in an uncompressed scheme. Find the compression ratio of the Huffman coding scheme developed in part (i) when compared with the original uncompressed scheme.

(6 Marks)

(iii) Find the entropy of the data source. Briefly discuss whether it is possible to design a codeword set which can achieve a target of less than 2.5 bits/symbol.

(5 Marks)

Solution

Image & Video Compression Basics

Why Image/Video Compression?

- Image/video compression is necessary for two reasons:
  - To reduce the storage requirement of images/videos.
  - To reduce the bitrate requirement to transmit them over the network.

Why Possible to Compress?

- Image and video can be compressed because of two types of redundancies:
  - Statistical Redundancy
    - Spatial redundancy
    - Temporal redundancy
    - Coding redundancy
  - Psycho-visual Redundancy
    - Frequency masking
    - Color masking

Spatial Redundancy

- Spatial redundancy refers to the statistical correlation between pixels within an image (or more specifically, within a small image neighborhood).
- It is also called intraframe redundancy.

Figure source: https://helpx.adobe.com/sg/photoshop/using/convert-color-image-black-white.html

Temporal Redundancy

- Temporal redundancy refers to the statistical correlation between pixels from successive frames in a video sequence.
- Therefore, it is also called interframe redundancy.

Coding Redundancy

- Coding redundancy focus on the representation of information, i.e., coding itself.
- Consider the following example:

TABLE 1.1
An Illustrative Example

Symbol	Occurrence Probability	Code 1	Code 2
a_1	0.1	000	0000
a_2	0.2	001	01
a_3	0.5	010	1
a_4	0.05	011	0001
a_5	0.15	100	001

Figure source:
Yun Q. Shi and Huifang Sun, Image and video compression for multimedia engineering: fundamentals, algorithms, and standards, CRC Press, 2000

- uniform-length coding: 3 bits/symbol
- variable-length coding

$$= 4 \times 0.1 + 2 \times 0.2 + 1 \times 0.5 + 4 \times 0.05 + 3 \times 0.15 = 1.95 \ \textit{bits per symbol}$$

Psychovisual Redundancy

- Frequency masking
  - Human is less sensitive to noise or distortion in high frequency components.
- Color masking
  - Human is more sensitive to luminance / luma (brightness) component than chrominance / chroma (color) components.

What Colour is the Dress?

Answer?

Figure source: https://www.9news.com.au/world/photo-finally-solves-the-black-and-blue-white-and-gold-dress-debate/15465485-dad8-45d2-8da0-8d20558b5013

Lossy and Lossless Compression

- Lossless compression:
  - Use in important data (e.g., medical images)
  - Reconstructed image is identical to the original image after decompression.

- Lossy compression:
  - For media such as images or video, it is not necessary to display more information than the human ear or eye can perceive.
  - Compression techniques may discard data with little perceived difference by humans.
  - Reconstructed image is not identical to the original image after decompression.

Distortion Measures / Metrics

- The three most commonly used distortion measures in image compression are:

  - *Mean Squared Error* (MSE) σ_d^2 ,

$$\sigma_d^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2 \qquad (8.1)$$

  where x_n, y_n, and N are the input data sequence, reconstructed data sequence, and length of the data sequence respectively.

  - *Signal to Noise Ratio* (SNR), in decibel units (dB),

$$SNR = 10\log_{10}\frac{\sigma_x^2}{\sigma_d^2} \qquad (8.2)$$

  where σ_x^2 is the average square value of the original data sequence and σ_d^2 is the MSE.

  - *Peak Signal to Noise Ratio* (PSNR),

$$PSNR = 10\log_{10}\frac{x_{peak}^2}{\sigma_d^2} \qquad (8.3)$$

Transform-based Coding/Compression

Transform Coding

- Why transform coding?
  - To convert the data into a form which is more suitable for compression.
- Transformation produces good properties for compression:
  - Offer energy compaction.
  - Offer redundancy reduction (i.e., reduce the correlation between transform coefficients).
- A reversible process
  - Original signal can be obtained by applying the inverse transform.

Transform-based Image Compression

- A typical transform-based image compression system consists of the following steps: image partitioning, transform, quantization, and coding.

a
b

FIGURE 8.21
A block transform coding system:
(a) encoder;
(b) decoder.

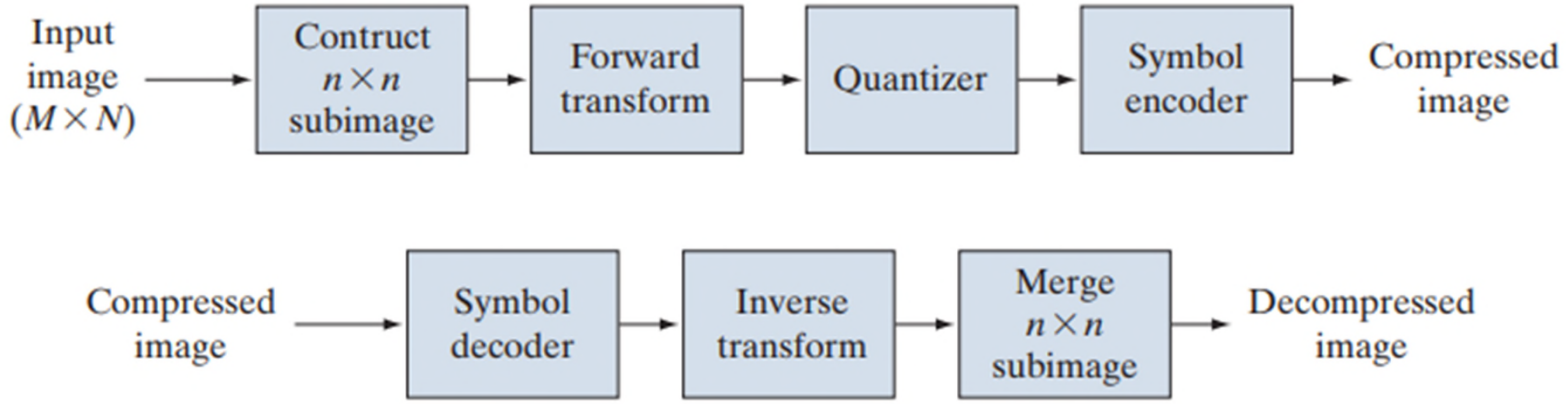


Figure source:
R. C. Gonzalez and R. E. Woods, Digital Image Processing, 4th edition, Prentice Hall, 2018.

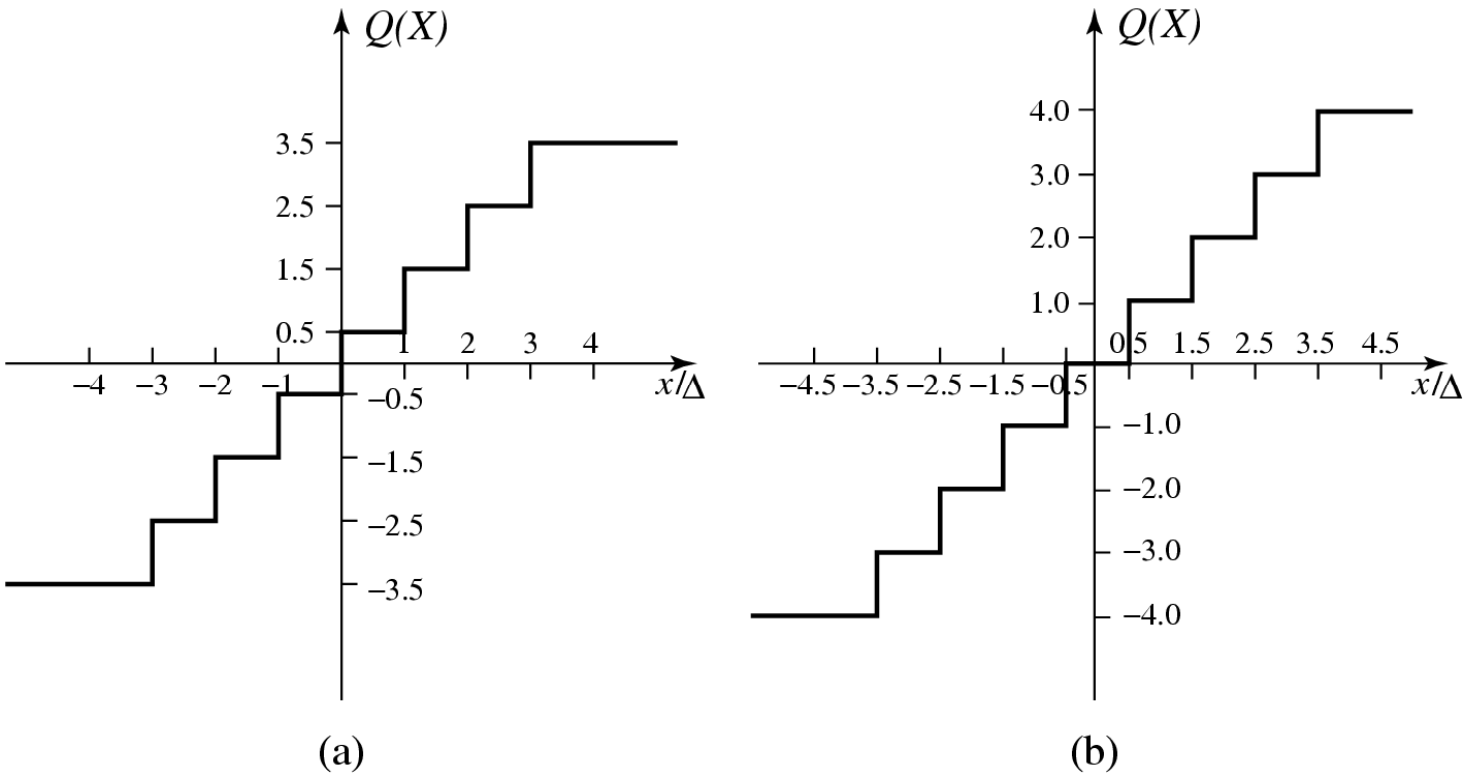
Transform

- Apply a one-to-one transform to the input image data.
- The transformed output is in a representation which is more suitable for efficient compression than the raw image data.
- Unitary mappings such as DCT packs the signal energy into a small number of coefficients.
- Some well-known transforms:
  - Discrete Fourier Transform (DFT)
  - Discrete Cosine Transform (DCT)
  - Discrete Wavelet Transform (DWT)
  - Etc.

Quantization

- Generate a limited number of symbols from the transform coefficients.
- An irreversible many-to-one mapping, causing information loss.

Scaler Quantization



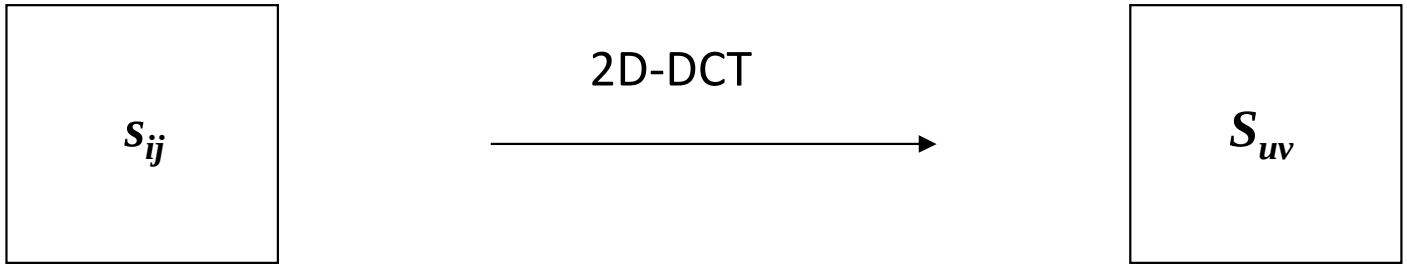
- **Fig. 8.2:** Uniform Scalar Quantizers: (a) Midrise, (b) Midtread.

Coding

- Assign a codeword or binary bitstream to each symbol at the output of the quantizer.
- May employ fixed-length or variable-length coding.
- Variable Length Coding (VLC) or entropy coding assigns codewords in such a way as to minimize the average length of the binary representation of the symbols.
- This is achieved by assigning shorter codewords to more probable symbols, which is the fundamental principle of entropy coding (e.g., Huffman coding).

Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT)



$$S_{uv} = \alpha(u)\alpha(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_{ij} \cos\frac{(2i+1)u\pi}{2N} \cos\frac{(2j+1)v\pi}{2N} \qquad u, v = 0, \ldots, N-1$$

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

Why DCT?

- 2D-DCT is a popular transform used in transform-based image compression.
- It can offer the following:
  - Energy compaction for transform coefficients
  - Redundancy reduction amongst transform coefficients
- Pro: good compression results, basis functions are fixed and not image-dependent.
- Con: compression is not as effective as some other transforms, e.g., Karhunen Loeve Transform.

“Lego Functions”

Transform and Basis Functions

- A transform can often be described by a change of basis functions.
- A transform involves a change of perspective on the image signal.
- Basis functions are analogous to “Lego functions”

Basis Functions / Images for 4x4 DCT

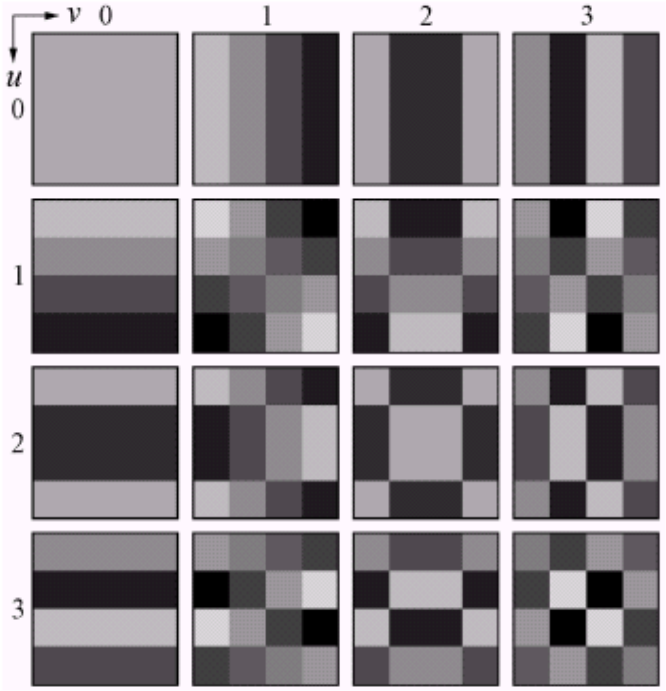
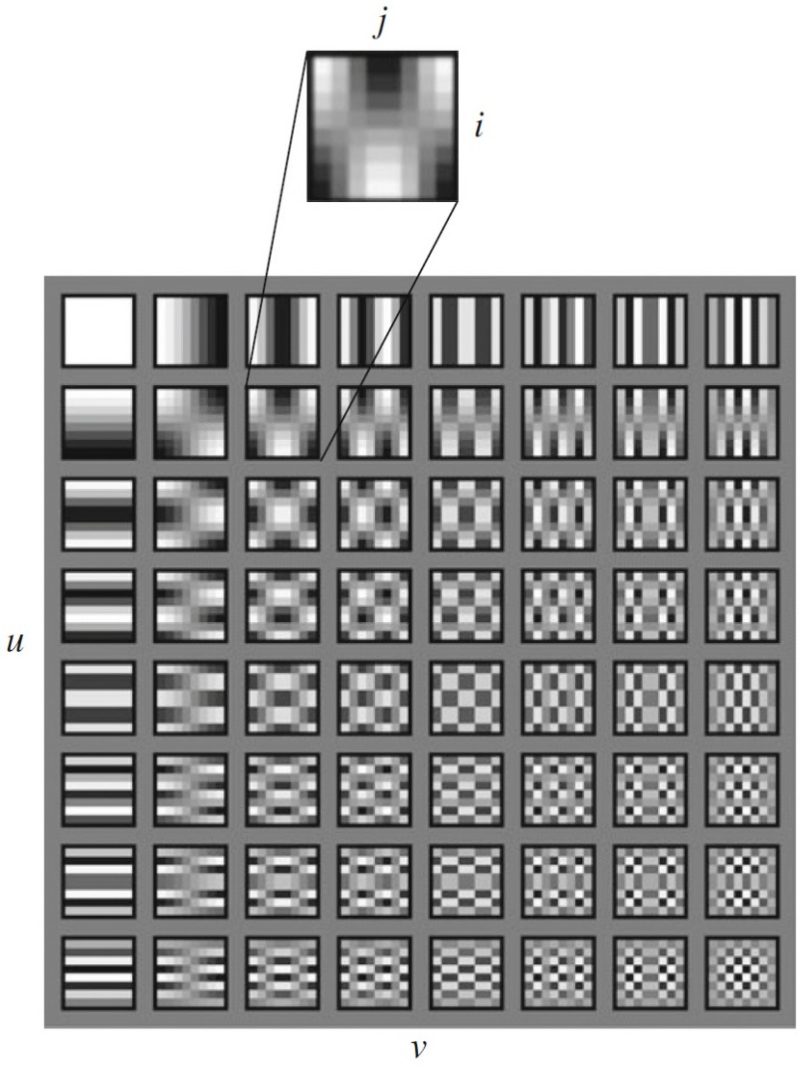


FIGURE 8.30 Discrete-cosine basis functions for $N = 4$. The origin of each block is at its top left.

Figure source:
R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd edition, Prentice Hall, 2002.

Basis Functions / Images for 8x8 DCT



Exercise: 2D-DCT (1)

The two-dimensional Discrete Cosine Transform (2-D DCT) of an $N \times N$ data matrix is given by:

$$S_{uv} = \alpha(u)\alpha(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} s_{ij} \cos\frac{(2i+1)u\pi}{2N}\cos\frac{(2j+1)v\pi}{2N} \qquad u, v = 0, \ldots, N-1$$

where

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

(a) Calculate the 2-D DCT of the following 4×4 pixel block **A**.

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 0 \\ 0 & 10 & 10 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(7 Marks)

Exercise: 2D-DCT (2)

(b) Based on your result in part (a), calculate the 2-D DCT of the following pixel block **B**.

$$\mathbf{B} = \begin{bmatrix} 20 & 20 & 20 & 20 \\ 20 & 15 & 15 & 20 \\ 20 & 15 & 15 & 20 \\ 20 & 20 & 20 & 20 \end{bmatrix}$$

(4 Marks)

Solution

2D DCT Matrix Implementation (1)

- The above factorization of a 2D DCT into two 1D DCTs can be implemented by two consecutive matrix multiplications:

$$F(u, v) = \mathbf{T} \cdot f(i, j) \cdot \mathbf{T}^T. \tag{8.27}$$

- We will name $\mathbf{T}$ the *DCT-matrix*.

$$\mathbf{T}[i, j] = \begin{cases} \frac{1}{\sqrt{N}}, & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} \cdot \cos\frac{(2j+1) \cdot i\pi}{2N}, & \text{if } i > 0 \end{cases} \tag{8.28}$$

Where i = 0, … , N-1 and j = 0, … , N-1 are the row and column indices, and the block size is N x N.

2D DCT Matrix Implementation (2)

When $N = 8$, we have:

$$\mathbf{T_8}[i, j] = \begin{cases} \frac{1}{2\sqrt{2}}, & \text{if } i = 0 \\ \frac{1}{2} \cdot \cos\frac{(2j+1)\cdot i\pi}{16}, & \text{if } i > 0. \end{cases} \tag{8.29}$$

$$\mathbf{T_8} = \begin{bmatrix} \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \frac{1}{2\sqrt{2}} & \cdots & \frac{1}{2\sqrt{2}} \\ \frac{1}{2}\cdot\cos\frac{\pi}{16} & \frac{1}{2}\cdot\cos\frac{3\pi}{16} & \frac{1}{2}\cdot\cos\frac{5\pi}{16} & \cdots & \frac{1}{2}\cdot\cos\frac{15\pi}{16} \\ \frac{1}{2}\cdot\cos\frac{\pi}{8} & \frac{1}{2}\cdot\cos\frac{3\pi}{8} & \frac{1}{2}\cdot\cos\frac{5\pi}{8} & \cdots & \frac{1}{2}\cdot\cos\frac{15\pi}{8} \\ \frac{1}{2}\cdot\cos\frac{3\pi}{16} & \frac{1}{2}\cdot\cos\frac{9\pi}{16} & \frac{1}{2}\cdot\cos\frac{15\pi}{16} & \cdots & \frac{1}{2}\cdot\cos\frac{45\pi}{16} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2}\cdot\cos\frac{7\pi}{16} & \frac{1}{2}\cdot\cos\frac{21\pi}{16} & \frac{1}{2}\cdot\cos\frac{35\pi}{16} & \cdots & \frac{1}{2}\cdot\cos\frac{105\pi}{16} \end{bmatrix}. \tag{8.30}$$

2D DCT Matrix Implementation (3)

- The 2D IDCT matrix implementation is simply:

$$f(i, j) = \mathbf{T}^T \cdot F(u, v) \cdot \mathbf{T}.$$

The DCT-matrix is orthogonal, hence,

$$\mathbf{T}^T = \mathbf{T}^{-1}.$$

Exercise: 2D-DCT Using Matrix Implementation

(a) The two-dimensional Discrete Cosine Transform (2-D DCT) matrix of an $N \times N$ pixel block is given by:

$$T(i,j) = \begin{cases} \dfrac{1}{\sqrt{N}}, & \text{if } i = 0 \\ \sqrt{\dfrac{2}{N}} \cos \dfrac{(2j+1)i\pi}{2N}, & \text{if } i > 0 \end{cases},$$

where i and j are the row and column indices, respectively.

(i) Determine the 2-D DCT matrix **T** for a 4×4 pixel block. Round your answer to 4 decimal places.

(4 Marks)

(ii) Based on your result in part (a)(i), calculate the 2-D DCT of the following pixel block **A**. Round your answer to 3 decimal places.

$$\mathbf{A} = \begin{bmatrix} 20 & 20 & 0 & 0 \\ 20 & 20 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

(6 Marks)

Solution

JPEG

Observations

- Observation 1: Image contents change relatively slowly across the image, e.g., within an 8x8 image block.
- How does it affect JPEG design?

- Observation 2: Humans are less sensitive to distortion in high spatial frequency components than low frequency components.
- How does it affect JPEG design?

- Observation 3: Human is more sensitive to intensity (luminance) component than color (chrominance) component.
- How does it affect JPEG design?

JPEG

- JPEG is a popular image compression standard.
- Address both lossy and lossless image compression.
- Compression ratio ranges from 10:1 to 20:1.
- Four modes of operation:
  - sequential DCT-based mode
  - progressive DCT-based mode
  - lossless mode
  - hierarchical mode
- We will focus on sequential DCT-based mode or baseline JPEG as it is the most popular mode in JPEG.

Main Stages in Baseline JPEG

- Image/block processing
- DCT on image blocks
- Quantization
- Entropy coding
- Frame building

JPEG Encoder

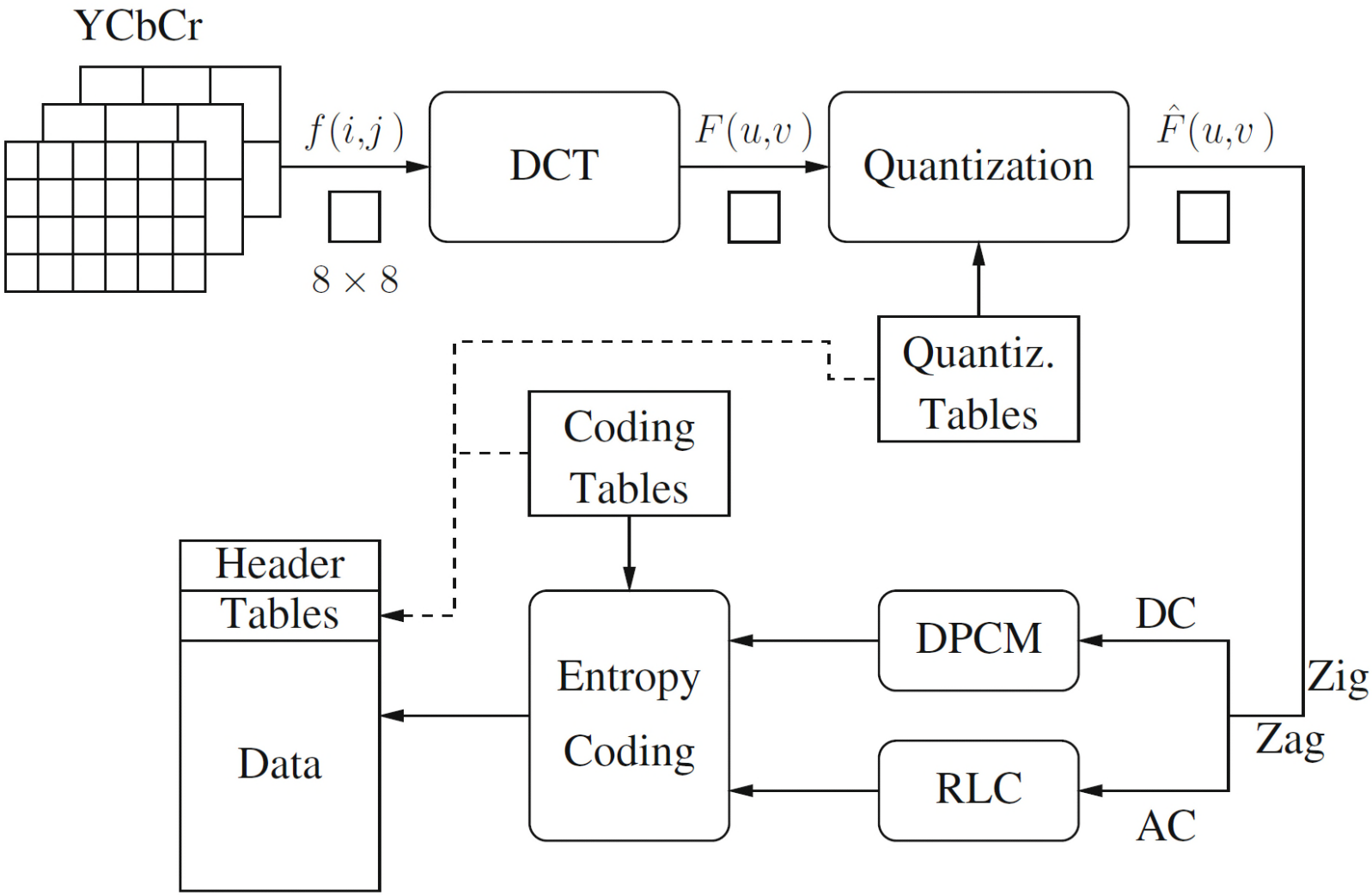


Fig. 9.1: Block diagram for JPEG encoder.

Source: Ze-Nian Li, Mark S. Drew, Jiangchuan Liu, Fundamental of Multimedia, Springer 2021

Image Partitioning

- Image is first partitioned into numerous 8x8 pixel blocks.

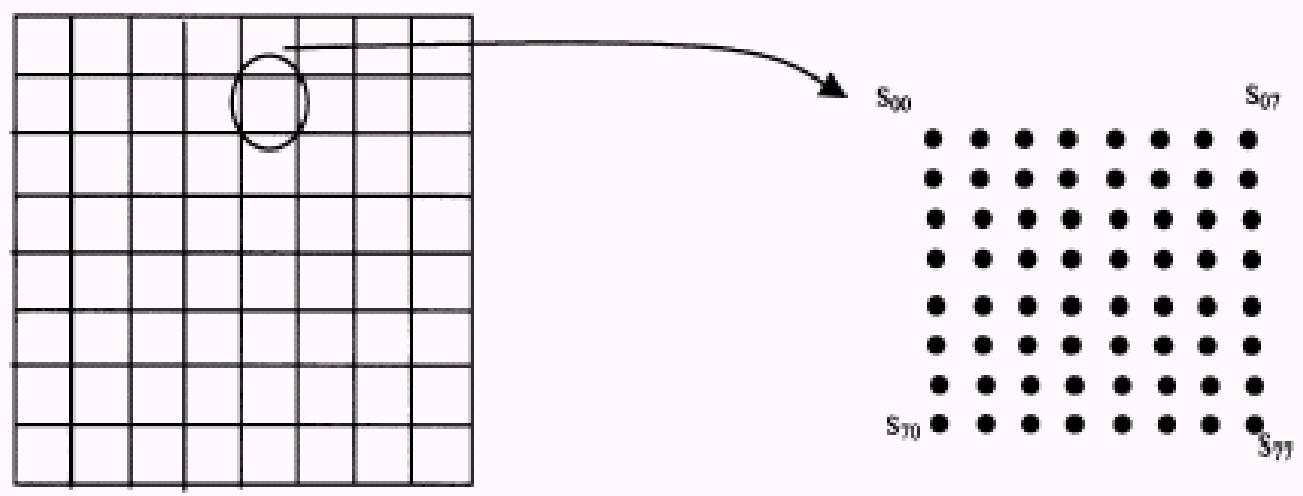
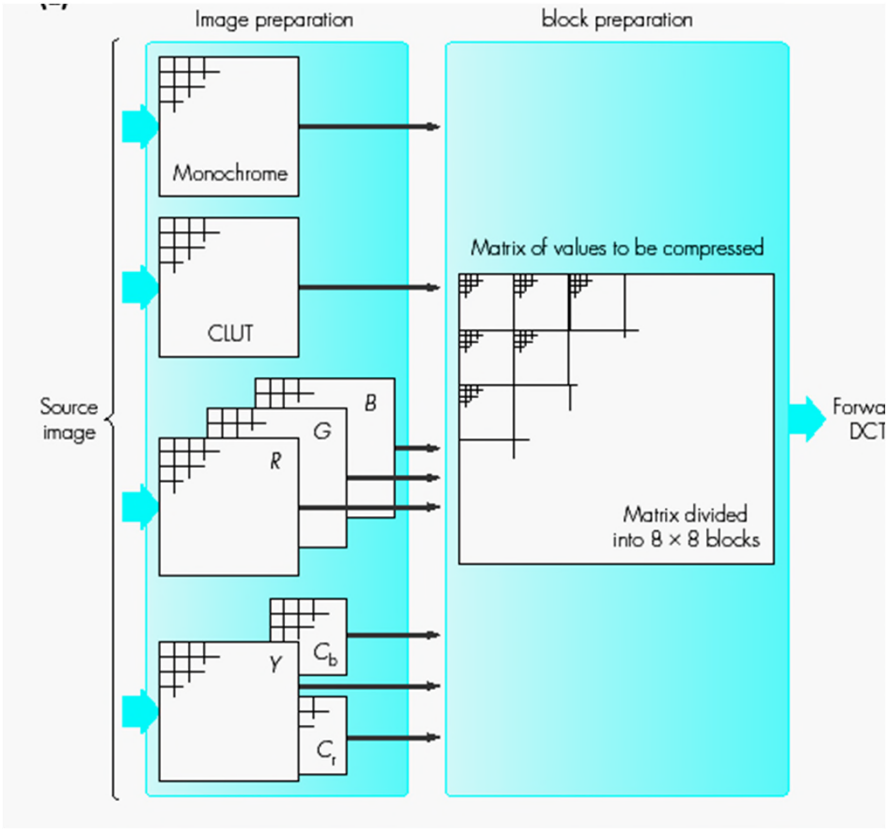


FIGURE 7.4 Partitioning to 8×8 blocks.

Figure source:
Yun Q. Shi and Huifang Sun, Image and video compression for multimedia engineering: fundamentals, algorithms, and standards, CRC Press, 2000

Image Preparation



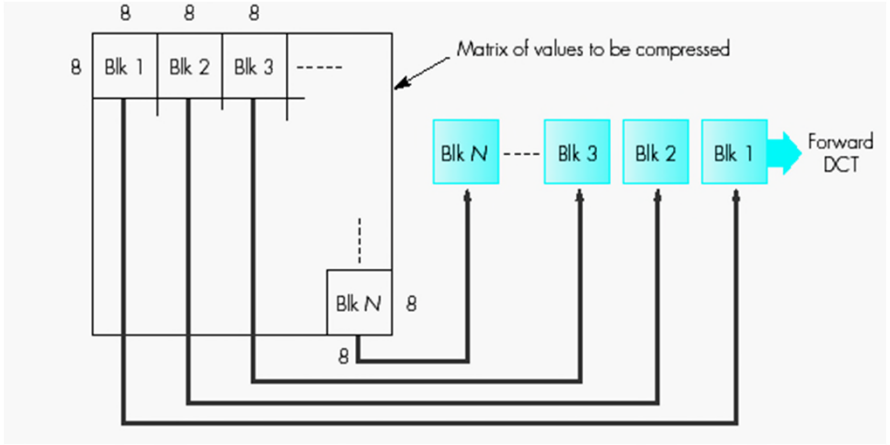


Figure source:
Fred Halsall, Multimedia communications: applications, networks, protocols and standards, Addison-Wesley, 2001

Forward DCT

- Forward 2D-DCT is applied to each 8x8 pixel block.

$$F(u,v) = \frac{C(u)\,C(v)}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} \cos\frac{(2i+1)u\pi}{16} \cos\frac{(2j+1)v\pi}{16} f(i,j)$$

$$C(\xi) = \begin{cases} \frac{\sqrt{2}}{2} & if \quad \xi = 0, \\ 1 & otherwise. \end{cases}$$

Recap: Why 2D-DCT?

- To convert image data into a form which is more suitable for compression.
- 2D-DCT yields energy compaction.
- 2D-DCT offers redundancy reduction by reducing the correlation between transform coefficients.

JPEG DCT Basis Function

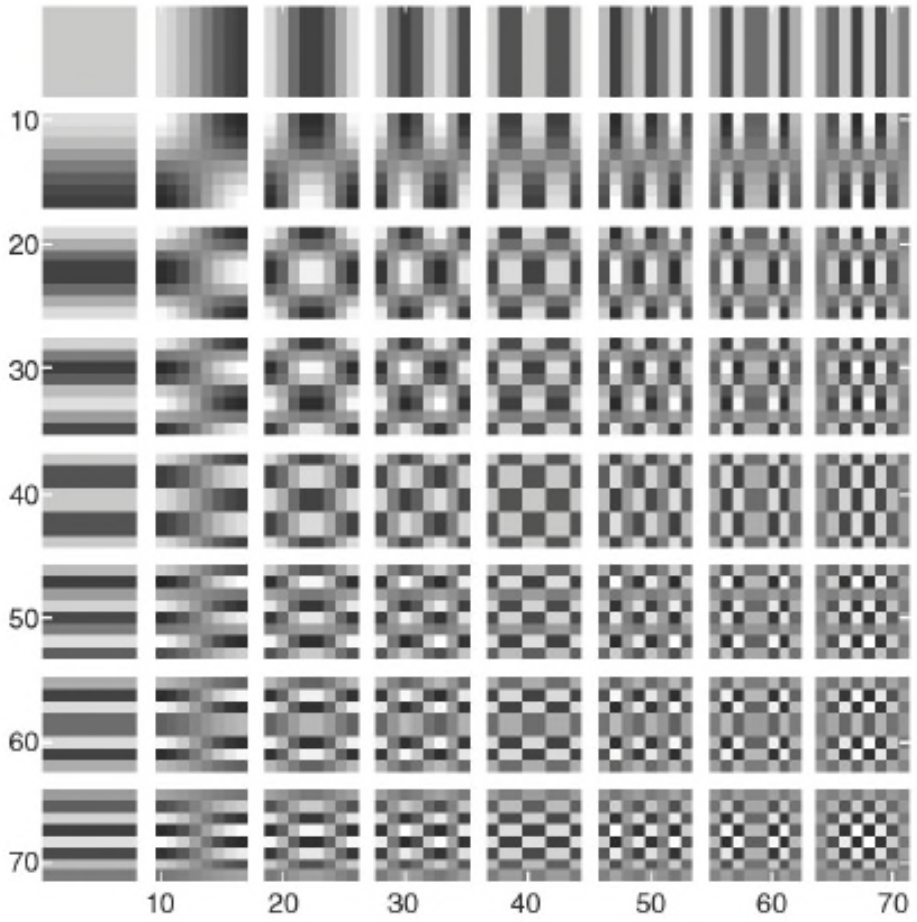
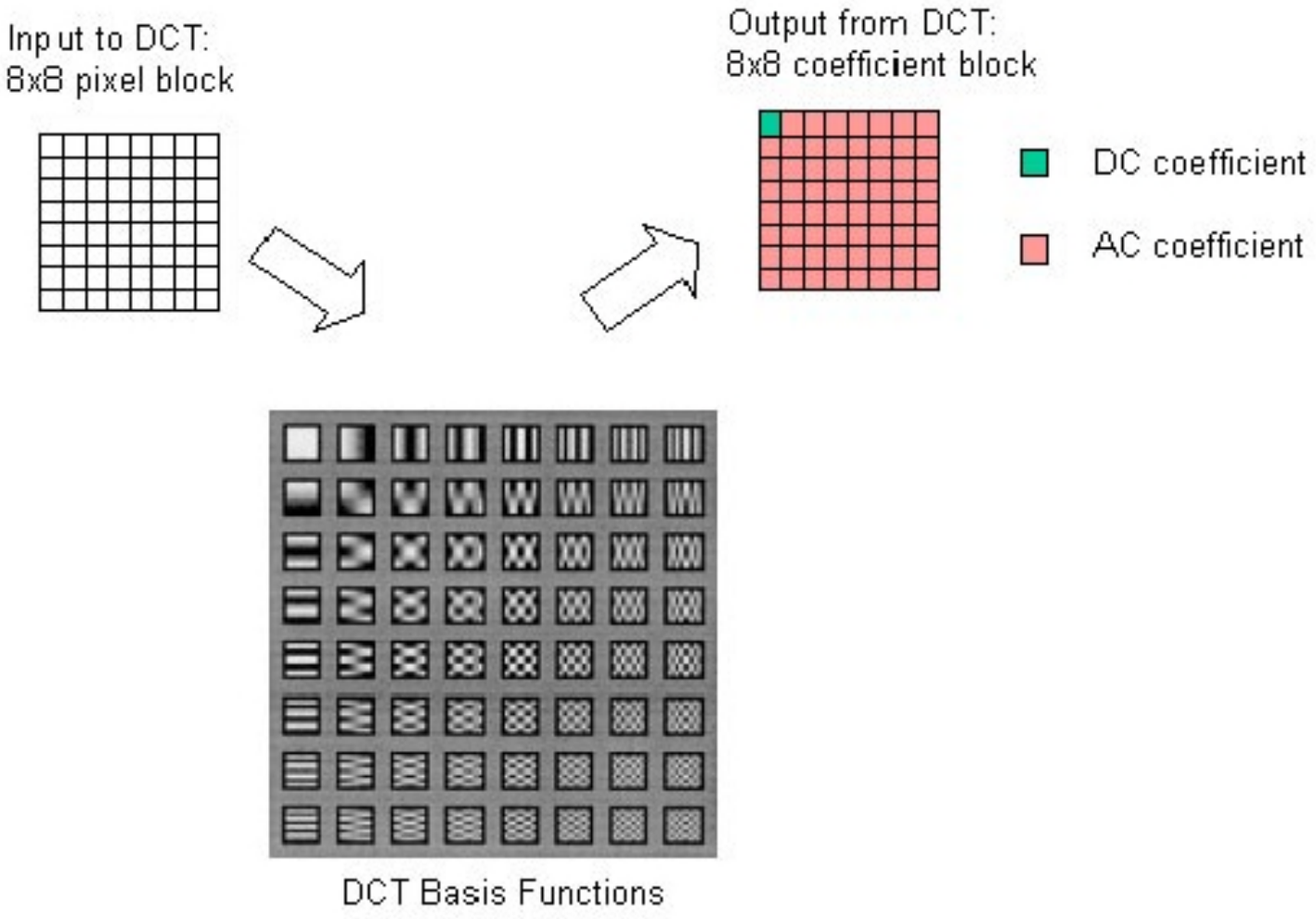


FIGURE 4.6 When N = 8: a set of the 64 basis images of the DCT.

Change of Basis in DCT (1)



Source: eetimes.com

Change of Basis in DCT (2)

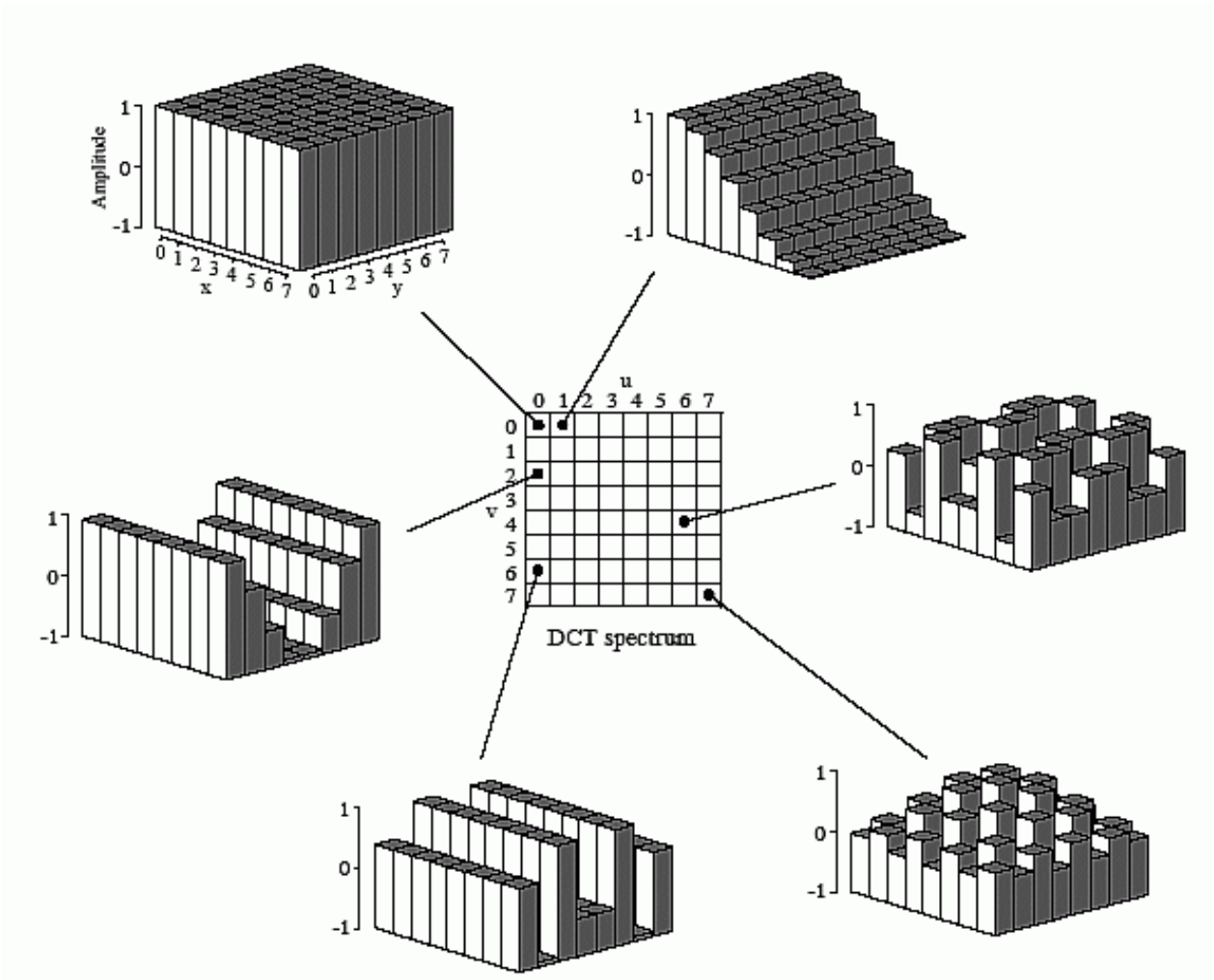
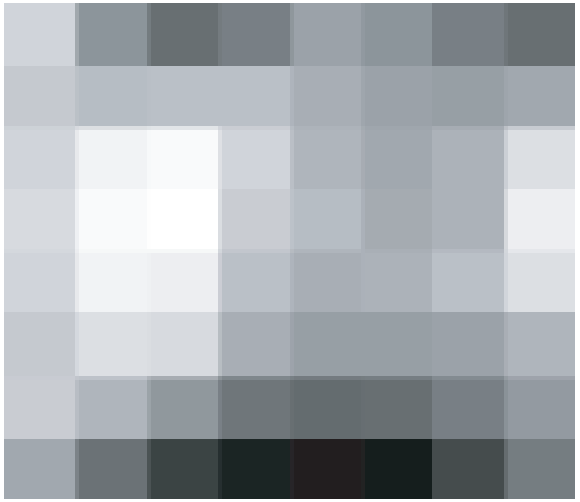


FIGURE 27-10
The DCT basis functions. The DCT spectrum consists of an 8×8 array, with each element in the array being an amplitude of one of the 64 basis functions. Six of these basis functions are shown here, referenced to where the corresponding amplitude resides.

Source: dspguide.com

Change of Basis in DCT (3)



PICTURE MATRIX

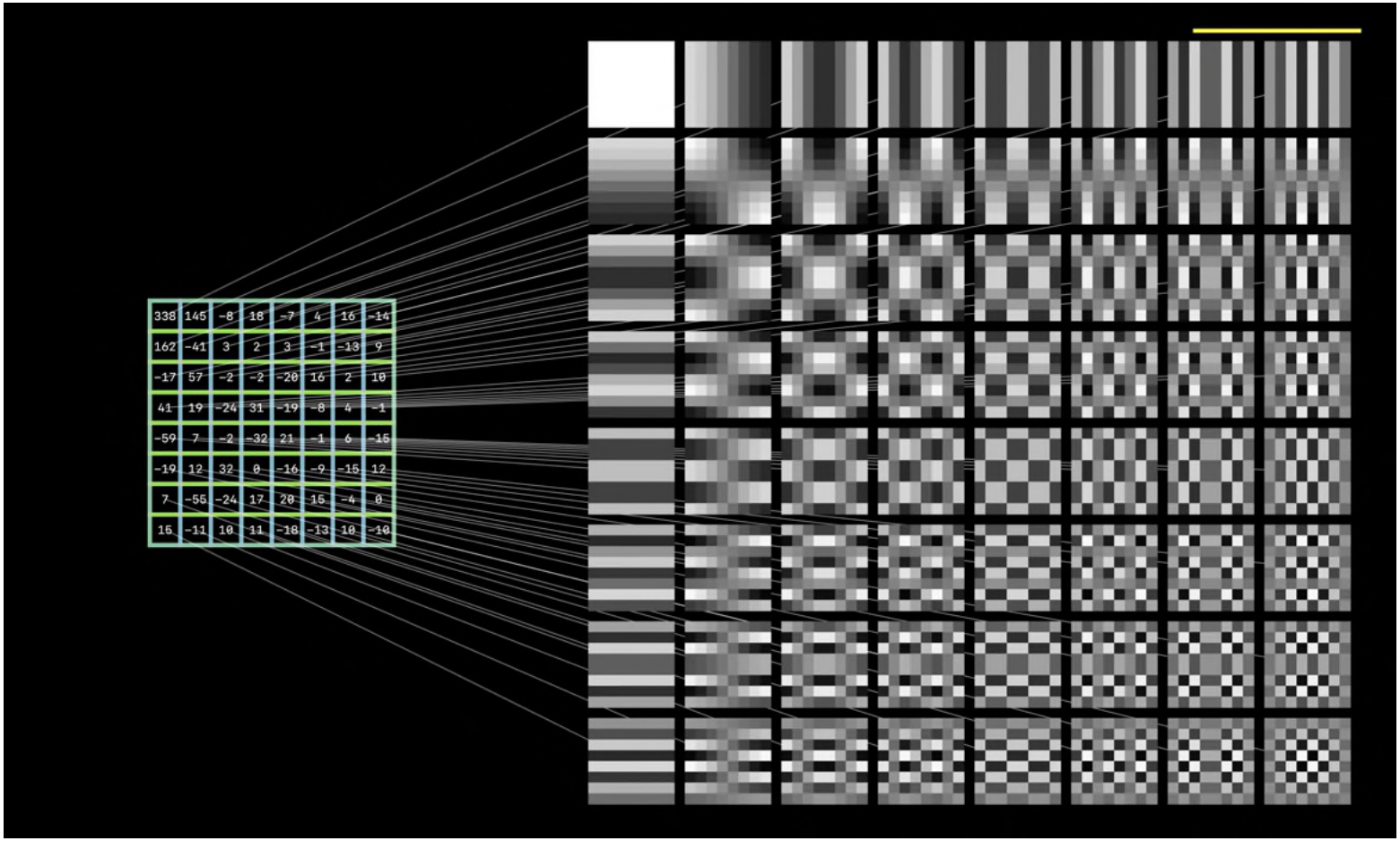
40	24	15	19	28	24	19	15
38	34	35	35	31	28	27	29
40	47	49	40	33	29	32	43
42	49	50	39	34	30	32	46
40	47	46	35	31	32	35	43
38	43	42	31	27	27	28	33
39	33	25	17	14	15	19	26
29	16	6	1	-4	0	7	18

DCT COEFFICENTS

239	32	27	-12	3	-5	3	1
34	-3	-19	6	3	0	-1	1
-70	2	8	23	9	6	-1	-1
5	0	-6	11	-2	0	-1	1
-17	-3	6	6	3	-1	0	0
2	4	2	2	1	-2	0	1
-3	0	0	-1	-1	-1	0	0
1	-1	3	1	0	0	0	0

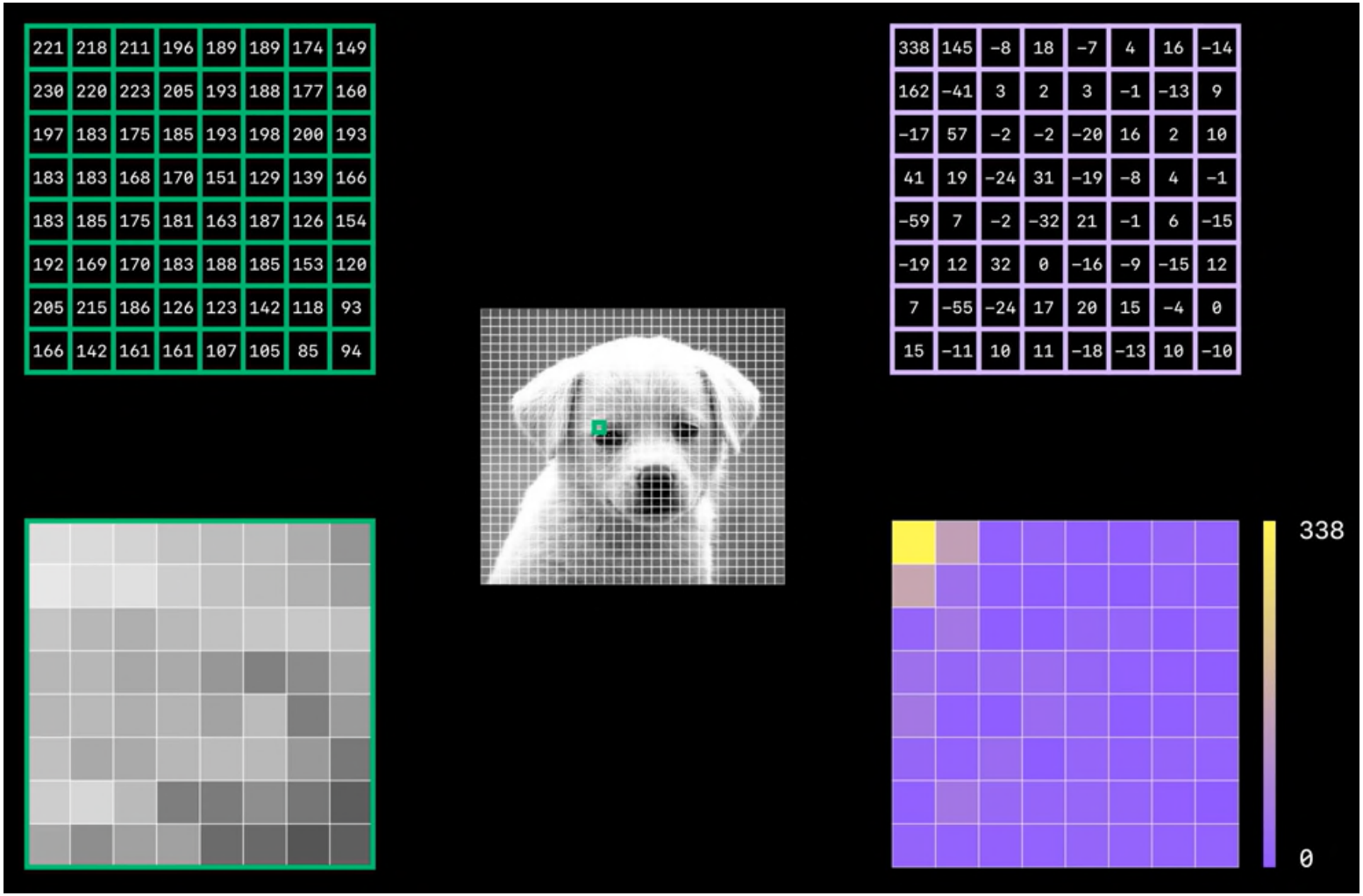
Source: cnx.org

Change of Basis in DCT (4)



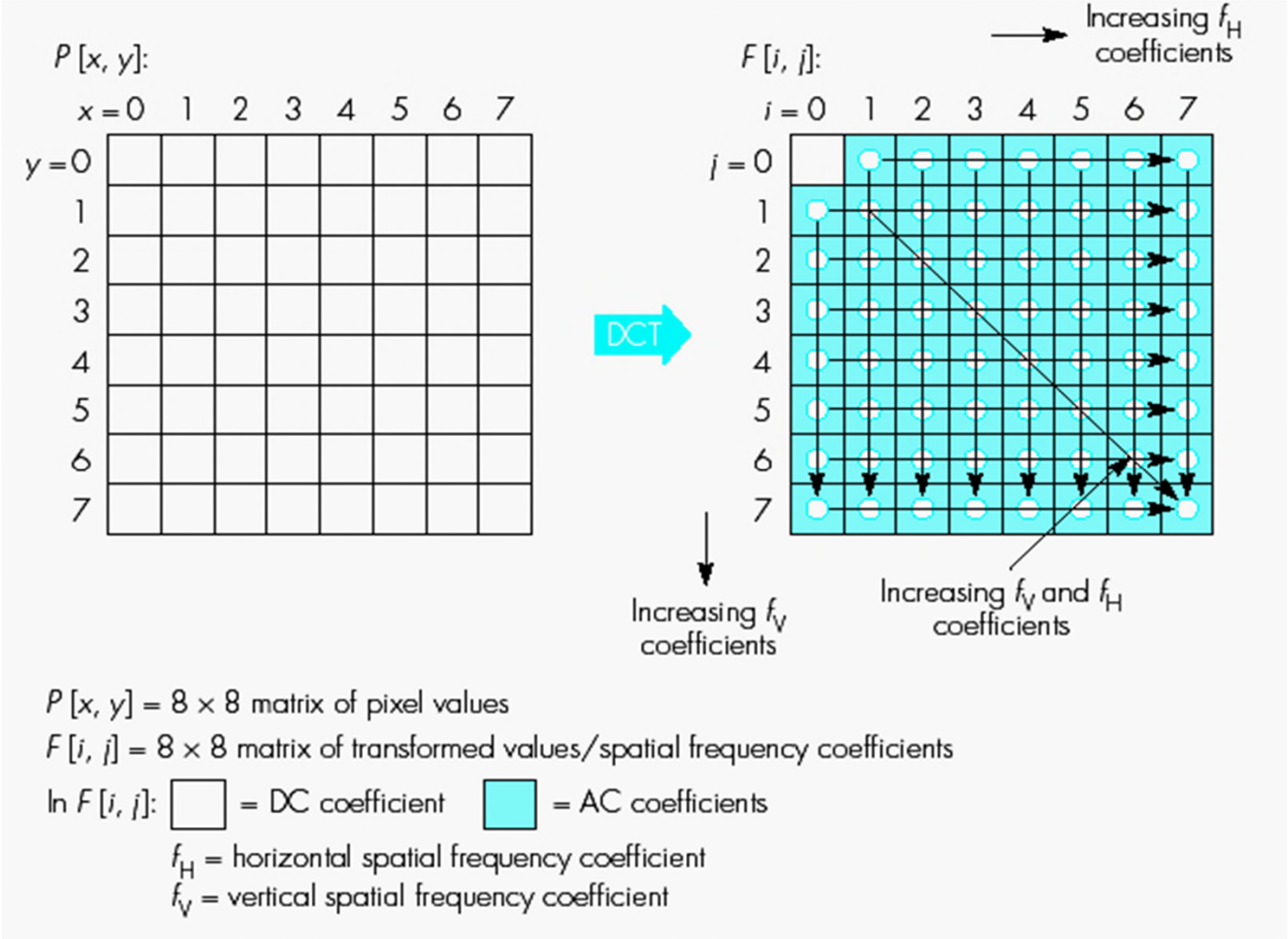
Source: Youtube Video. The Unreasonable Effectiveness of JPEG: A Signal Processing Approach

Change of Basis in DCT (5)



Source: Youtube Video. The Unreasonable Effectiveness of JPEG: A Signal Processing Approach

DCT Coefficients



Quantization

- As a finite number of bits is used to represent DCT coefficients, quantization is required to map the continuous value into discrete value.
- The coefficients are quantized using quantization tables.
- Information loss occurs during quantization.

Quantization Tables

- Human is more sensitive to DC and low AC coefficients.
- Hence, more emphasis should be taken in quantizing them.
- Smaller step sizes (quantization values) are used to quantize DC and low AC coefficients to reduce quantization error.
- Step sizes/values in quantization table are a compromise between level of compression and information loss.

Quantization (1)

- $F(u,v)$ represents a DCT coefficient, $Q(u,v)$ is a "quantization matrix" entry, and $\hat{F}(u,v)$ represents the *quantized DCT coefficients* which JPEG will use in the succeeding entropy coding.

$$\hat{F}(u,v) = round\left(\frac{F(u,v)}{Q(u,v)}\right)$$

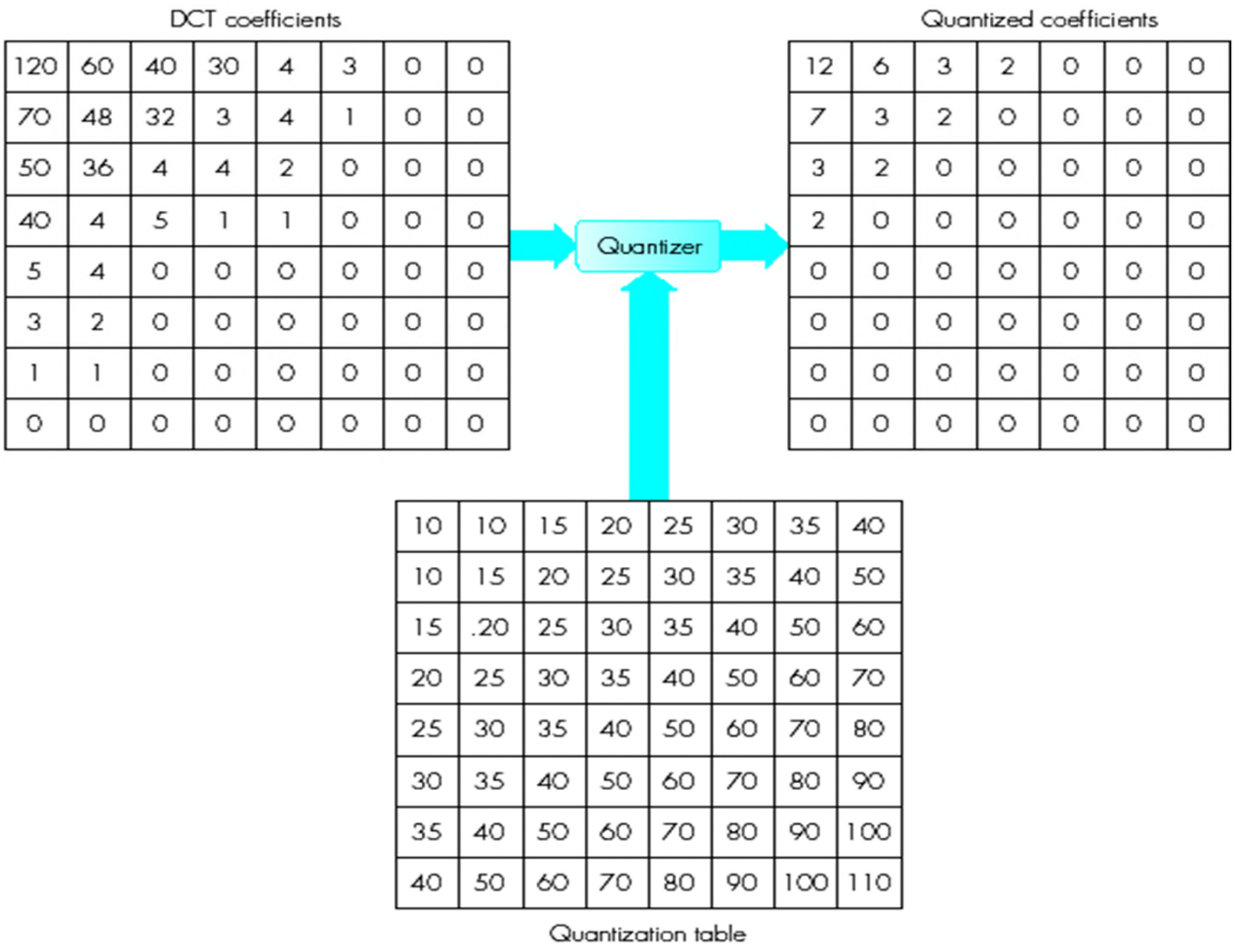
Table 9.1 The Luminance Quantization Table

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

Table 9.2 The Chrominance Quantization Table

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

Quantization (2)



Quantization Effect (1)

An 8×8 block from the Y image of 'Lena'

200	202	189	188	189	175	175	175
200	203	198	188	189	182	178	175
203	200	200	195	200	187	185	175
200	200	200	200	197	187	187	187
200	205	200	200	195	188	187	175
200	200	200	200	200	190	187	175
205	200	199	200	191	187	187	175
210	200	200	200	188	185	187	186

$f(i,j)$

515	65	-12	4	1	2	-8	5
-16	3	2	0	0	-11	-2	3
-12	6	11	-1	3	0	1	-2
-8	3	-4	2	-2	-3	-5	-2
0	-2	7	-5	4	0	-1	-4
0	-3	-1	0	4	1	-1	0
3	-2	-3	3	3	-1	-1	3
-2	5	-2	4	-2	2	-3	0

$F(u,v)$

Fig. 9.2: JPEG compression for a smooth image block.

Quantization Effect (2)

32	6	-1	0	0	0	0	0
-1	0	0	0	0	0	0	0
-1	0	1	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

$\hat{F}(u,v)$

512	66	-10	0	0	0	0	0
-12	0	0	0	0	0	0	0
-14	0	16	0	0	0	0	0
-14	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

$\tilde{F}(u,v)$

199	196	191	186	182	178	177	176
201	199	196	192	188	183	180	178
203	203	202	200	195	189	183	180
202	203	204	203	198	191	183	179
200	201	202	201	196	189	182	177
200	200	199	197	192	186	181	177
204	202	199	195	190	186	183	181
207	204	200	194	190	187	185	184

$\tilde{f}(i,j)$

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

$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

Fig. 9.2 (cont'd): JPEG compression for a smooth image block.

Quantization Effect (3)

Another 8×8 block from the Y image of 'Lena'

70	70	100	70	87	87	150	187
85	100	96	79	87	154	87	113
100	85	116	79	70	87	86	196
136	69	87	200	79	71	117	96
161	70	87	200	103	71	96	113
161	123	147	133	113	113	85	161
146	147	175	100	103	103	163	187
156	146	189	70	113	161	163	197

$f(i,j)$

-80	-40	89	-73	44	32	53	-3
-135	-59	-26	6	14	-3	-13	-28
47	-76	66	-3	-108	-78	33	59
-2	10	-18	0	33	11	-21	1
-1	-9	-22	8	32	65	-36	-1
5	-20	28	-46	3	24	-30	24
6	-20	37	-28	12	-35	33	17
-5	-23	33	-30	17	-5	-4	20

$F(u,v)$

Fig. 9.3: JPEG compression for a textured image block.

Quantization Effect (4)

-5	-4	9	-5	2	1	1	0
-11	-5	-2	0	1	0	0	-1
3	-6	4	0	-3	-1	0	1
0	1	-1	0	1	0	0	0
0	0	-1	0	0	1	0	0
0	-1	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\hat{F}(u,v)$

-80	-44	90	-80	48	40	51	0
-132	-60	-28	0	26	0	0	-55
42	-78	64	0	-120	-57	0	56
0	17	-22	0	51	0	0	0
0	0	-37	0	0	109	0	0
0	-35	55	-64	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$\tilde{F}(u,v)$

70	60	106	94	62	103	146	176
85	101	85	75	102	127	93	144
98	99	92	102	74	98	89	167
132	53	111	180	55	70	106	145
173	57	114	207	111	89	84	90
164	123	131	135	133	92	85	162
141	159	169	73	106	101	149	224
150	141	195	79	107	147	210	153

$\tilde{f}(i,j)$

0	10	-6	-24	25	-16	4	11
0	-1	11	4	-15	27	-6	-31
2	-14	24	-23	-4	-11	-3	29
4	16	-24	20	24	1	11	-49
-12	13	-27	-7	-8	-18	12	23
-3	0	16	-2	-20	21	0	-1
5	-12	6	27	-3	2	14	-37
6	5	-6	-9	6	14	-47	44

$\epsilon(i,j) = f(i,j) - \tilde{f}(i,j)$

Fig. 9.3 (cont'd): JPEG compression for a textured image block.

Exercise: Basis Function and Quantization

(b) In a new image compression scheme, a student would like to follow similar steps in the baseline JPEG to perform grayscale image compression. However, the student proposes to partition image into multiple 4×4 pixel blocks, perform 4×4 DCT for each pixel block, followed by corresponding quantization and entropy encoding.

(i) Briefly discuss a main similarity and a main difference between the basis functions of the DCT used in this new compression scheme with the basis functions used in the baseline JPEG compression.

(5 Marks)

(ii) Write down a suitable quantization table for this new compression scheme. Briefly justify your answer.

(5 Marks)

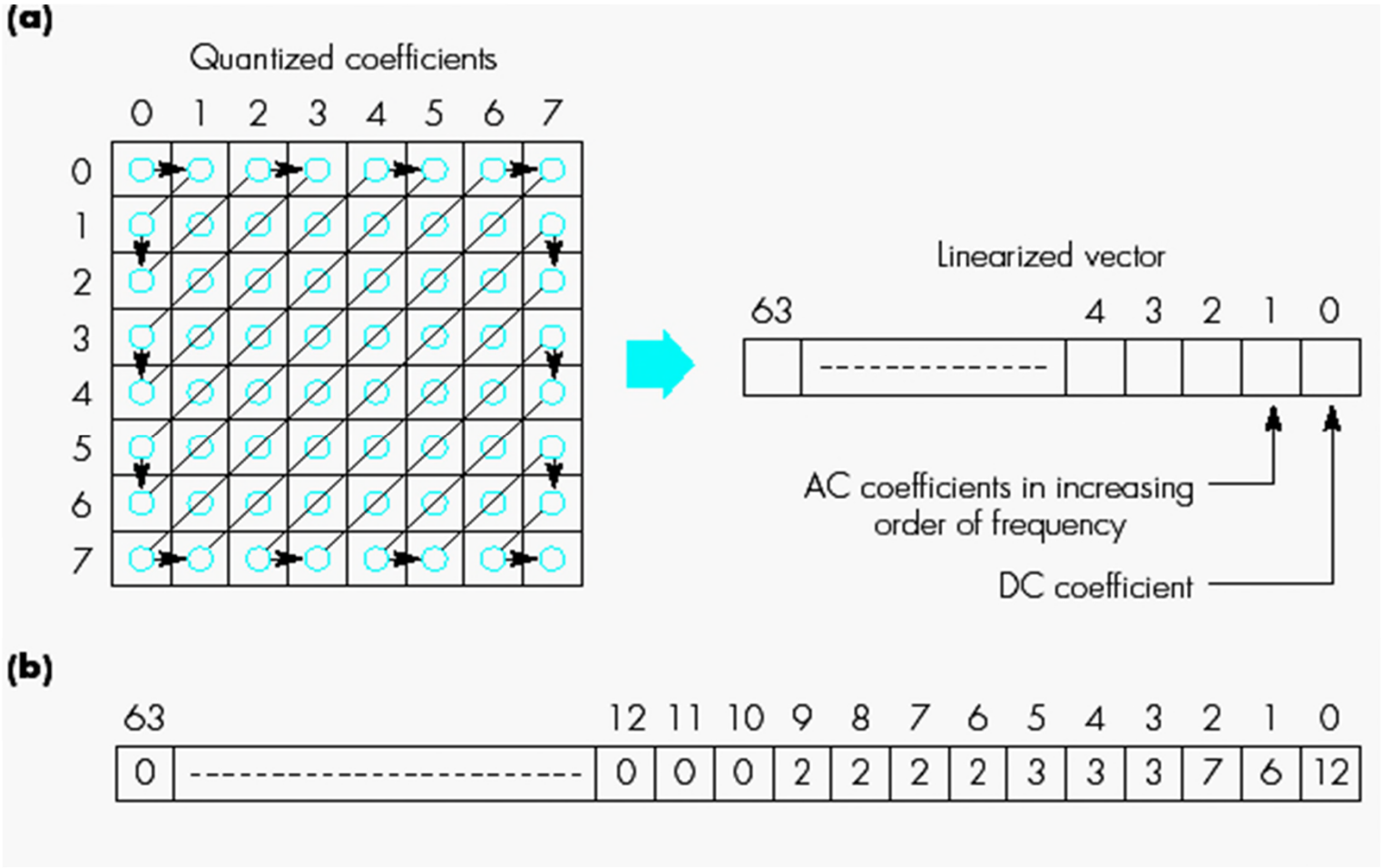
Solution

Entropy Encoding

- Entropy encoding comprises 4 steps:
  - Vectoring (zig-zag scanning)
  - Differential encoding / Differential Pulse Code Modulation (DPCM)
  - Run-length encoding (RLE)
  - Huffman encoding

- Vectoring
  - Zig-zag scanning
  - Represent the quantized 2D matrix in 1D vector.
  - Aim to exploit the presence of large number of zeros in the quantized matrix.

Zig-zag Scanning



Differential Encoding of DC Coefficients

- DC coefficient is treated separately from 63 AC coefficients.
- DC coefficient reflects average intensity of pixel block.
- DC coefficients are encoded using differential coding.
- The difference (prediction error) between DC coefficients of current block and previous block is encoded.
- Differential coding is used as average intensity between 2 consecutive blocks is similar.

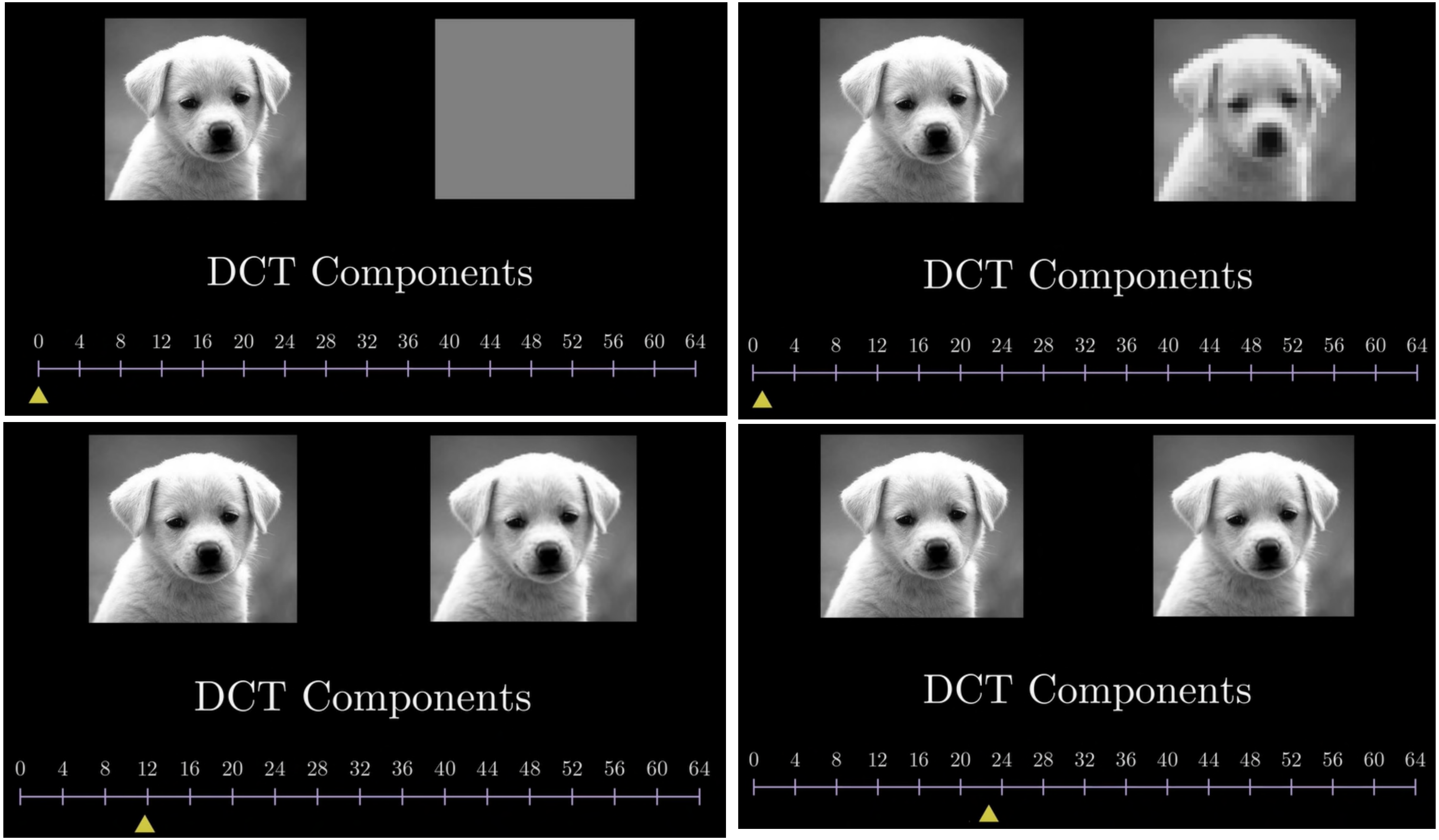
Run-Length Encoding (RLE)

- The AC coefficients are encoded using run-length encoding.
- Run-length encoding exploits the presence of a large number of zeros in the AC values by trying to group zeros together.
- Run-length encoding syntax uses a string made up of pairs of values.
- Each pair has (skip, value) format:
  - Skip – number of zeros in the run
  - Value – the next non-zero coefficient

Huffman Encoding

- After differential coding and run-length coding, some symbols/patterns occur more often than the others.
- Hence, Huffman coding is used to compress these symbols.
- High compression can be achieved by replacing frequent patterns with shorter codewords, and vice versa.
- Huffman coding is used in both differential coding and run-length coding.

Image Quality vs DCT Components



Source: Youtube Video. The Unreasonable Effectiveness of JPEG: A Signal Processing Approach

Frame Building

- JPEG defines syntax for bitstream relating to image/frame.
- The role of frame builder is to encapsulate all the information relating to an encoded image in this format.

JPEG Bitstream Format

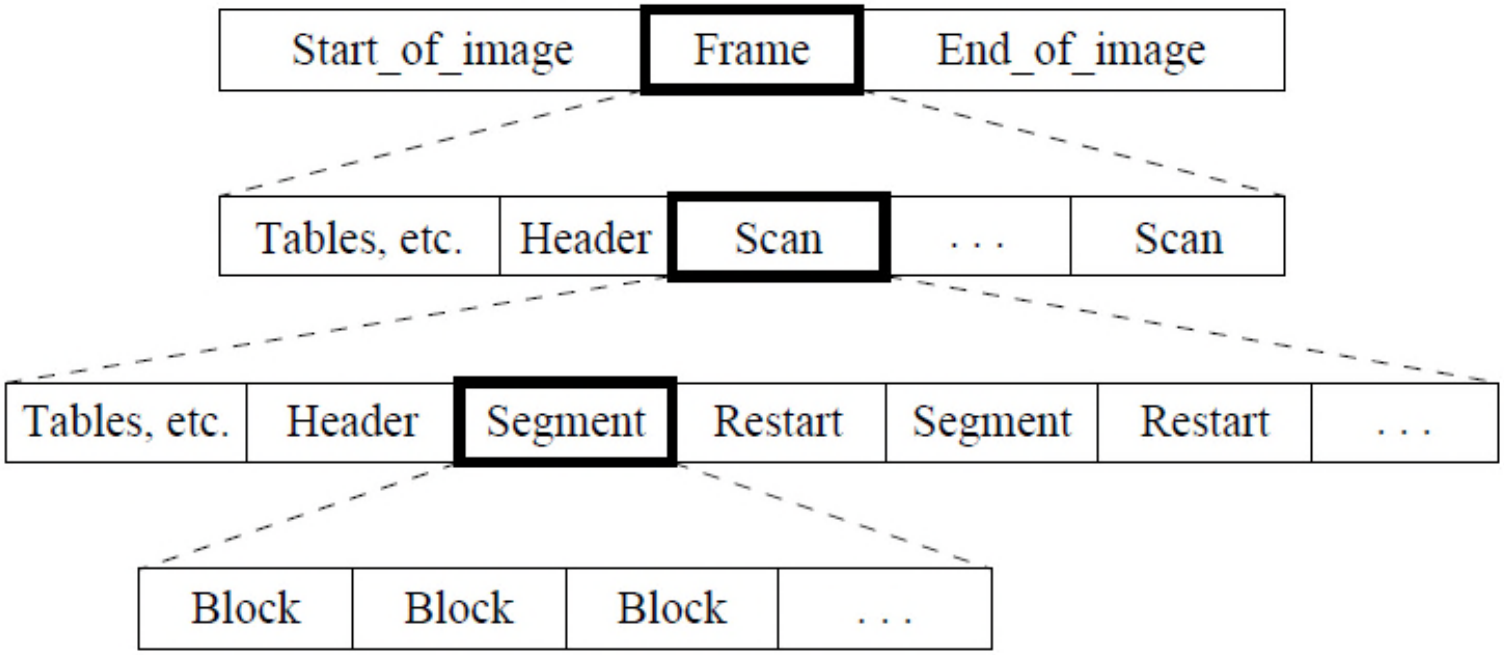
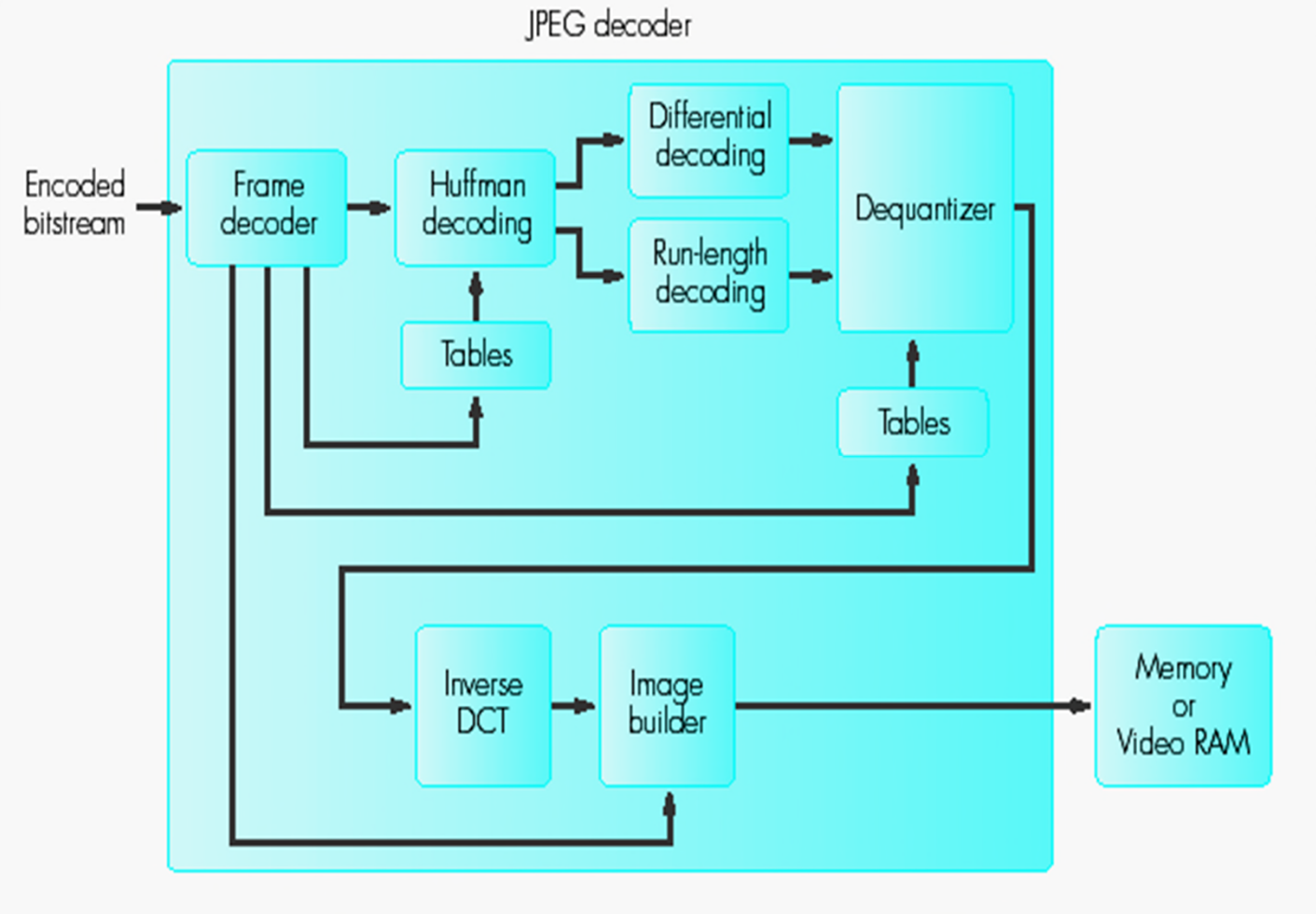


Fig. 9.6: JPEG bitstream.

JPEG Decoder



JPEG Decoder

- JPEG decoder extracts the control information and tables, and then passes them to the image builder.
- Huffman decoding is used to obtain the DC and AC symbols.
- Differential decoding is used to obtain the DC coefficients.
- Run-length decoding is used to obtain the AC coefficients.
- The DC and AC coefficients are put together and dequantized using quantization tables.
- Inverse DCT is used to transform DCT coefficients to pixel block.
- Image builder reconstructs the image from all the pixel blocks.

Exercise: JPEG Decoder

(b) Draw a simple block diagram of the baseline JPEG decoder. Clearly label all the key components in the diagram.

(6 Marks)

Solution

Part 2 Summary

- This part covers the following:
  - Terms and Concepts
  - Entropy Coding
  - Image & Video Compression Basics
  - Transform-based Coding / Compression
  - Discrete Cosine Transform (DCT)
  - JPEG Standard