上海交通大學
SHANGHAI JIAO TONG UNIVERSITY
IVM
Image, Video, and Multimedia Communications Laboratory

Digital Image Processing

Hongkai Xiong
熊红凯
http://ivm.sjtu.edu.cn

电子工程系
上海交通大学

Today

- Image Compression

Uncompressed Video

- A two hour SD video

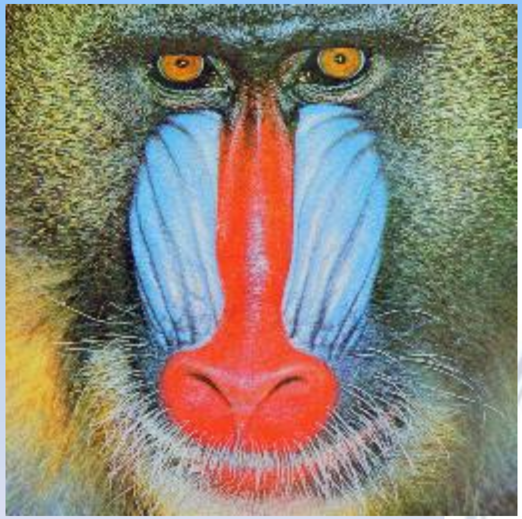
$$30\frac{frames}{\text{sec}}\times(720\times480)\frac{pixels}{frame}\times3\frac{bytes}{pixel}=31104000bytes/\text{sec}$$

$$31104000\frac{bytes}{\text{sec}}\times\left(60^{2}\right)\frac{\text{sec}}{hr}\times2hrs\cong2.24\times10^{11}bytes\approx224GB$$

- Limits on transmission and storage

Color Image Compression

- The following is a demonstration of JPEG/JPEG2000 Compression of Color Image



bit per pixel

Original Image --- 24 bpp

Color Image Compression

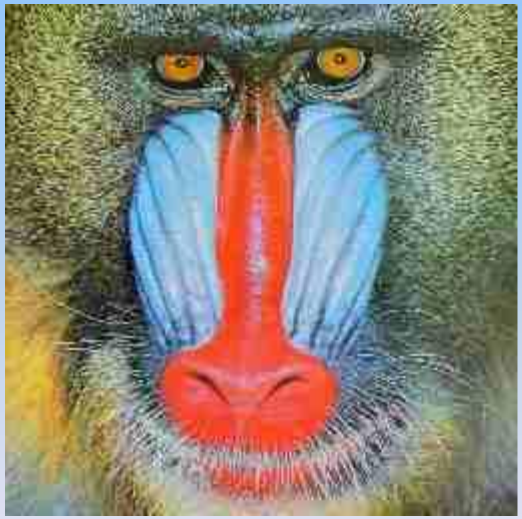
- The following is a demonstration of JPEG/JPEG2000 Compression of Color Image

JPEG --- 2 bpp

JPEG2000 --- 2 bpp

Color Image Compression

- The following is a demonstration of JPEG/JPEG2000 Compression of Color Image



JPEG --- 1 bpp

JPEG2000 --- 1 bpp

Color Image Compression

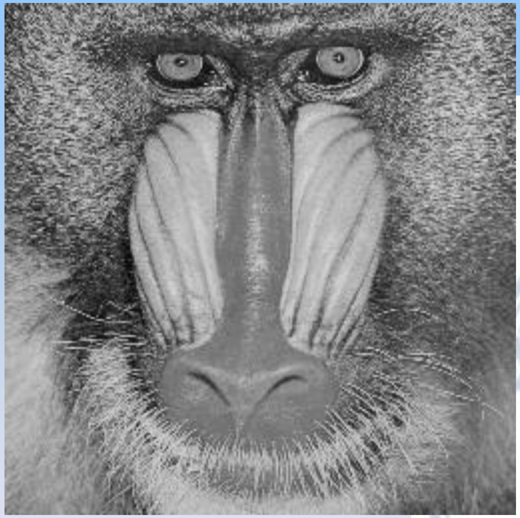
- The following is a demonstration of JPEG/JPEG2000 Compression of Color Image

JPEG --- 0.5 bpp

JPEG2000 --- 0.5 bpp

Gray-Scaled Image Compression

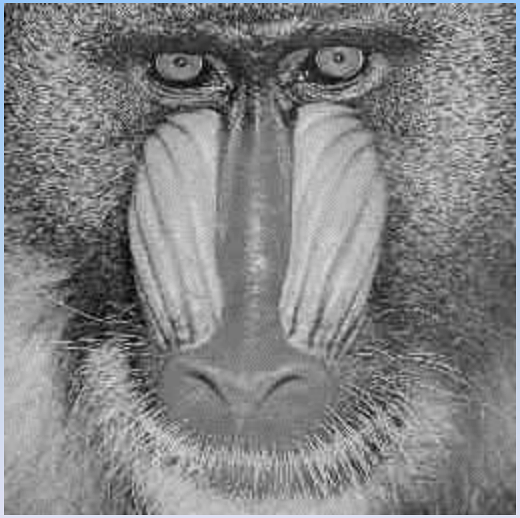
- The following is a demonstration of JPEG/JPEG2000 Compression of Gray-Scaled Image



Original Image --- 8 bpp

Gray-Scaled Image Compression

- The following is a demonstration of JPEG/JPEG2000 Compression of Gray-Scaled Image

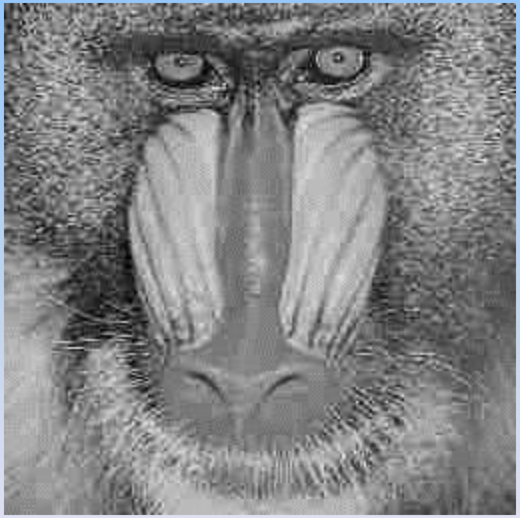


JPEG --- 1 bpp

JPEG2000 --- 1 bpp

Gray-Scaled Image Compression

- The following is a demonstration of JPEG/JPEG2000 Compression of Gray-Scaled Image



JPEG --- 0.67 bpp

JPEG2000 --- 0.67 bpp

Gray-Scaled Image Compression

- The following is a demonstration of JPEG/JPEG2000 Compression of Gray-Scaled Image

JPEG --- 0.27 bpp

JPEG2000 --- 0.27 bpp

Why Do We Need Compression?

- Requirements may outstrip the anticipated increase of storage space and bandwidth
- For data storage and data transmission
  - DVD
  - Video conference
  - Printer
- The bit rate of uncompressed digital cinema data exceeds 1 Gbps

Why Do We Need Compression?

- Image compression plays a major role in many important and diverse applications
  - televideo-conferencing
  - remote sensing
  - document and medical imaging
  - facsimile transmission (FAX)
  - control of remotely piloted vehicles in military, space and hazardous waste management applications
  - digital television
  - cell phone
  - VCD & DVD players
  - VOD (comic.sjtu.edu.cn)
  - DC & DV
  - video surveillance system
  - databases of fingerprints & human faces
  - …

Why Can We Compress?

- Coding redundancy
  - natural binary code leads to waste of bits
  - variable-length coding (VLC)
- Spatial and temporal redundancy
  - Spatial redundancy
    - Neighboring pixels are not independent but correlated
    - predictive coding, transform coding, run-length coding
  - Temporal redundancy
    - correlation between adjacent frames
    - motion estimation (ME), motion compensation (MC), predictive coding

Why Can We Compress?

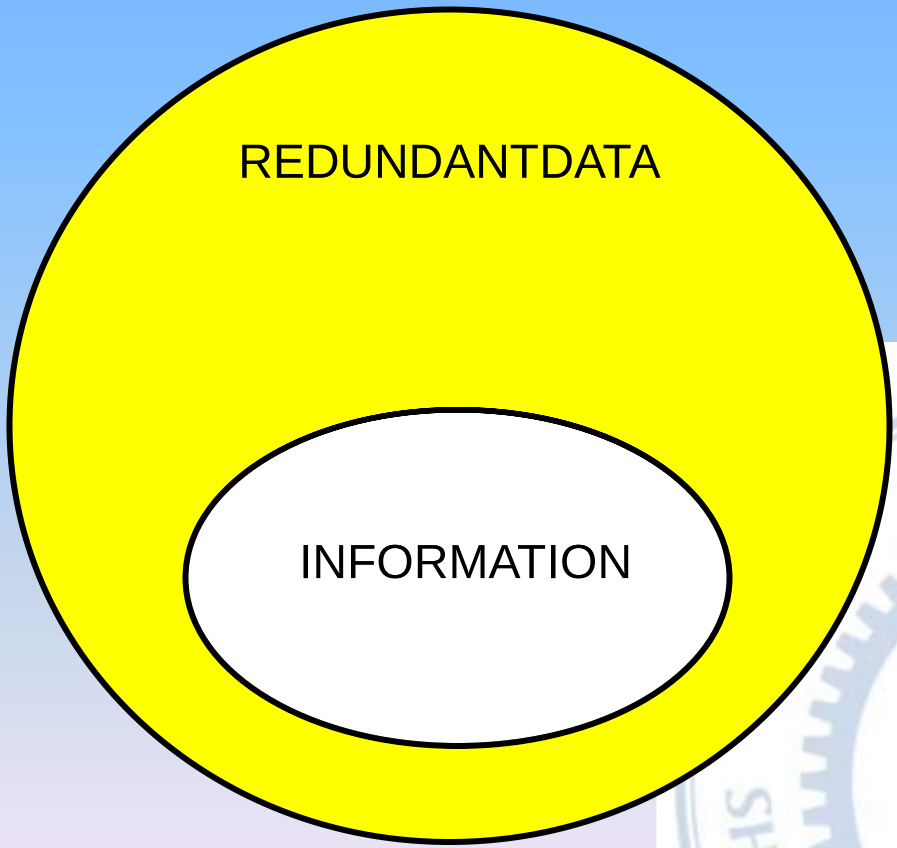
- Irrelevant information
  - Psychovisual redundancy
    - certain information simply has less relative importance than other information in normal visual processing
    - second generation image compression, lossy image compression
  - Knowledge redundancy
    - priori knowledge of image
    - model coding

Why Can We Compress?

a b c

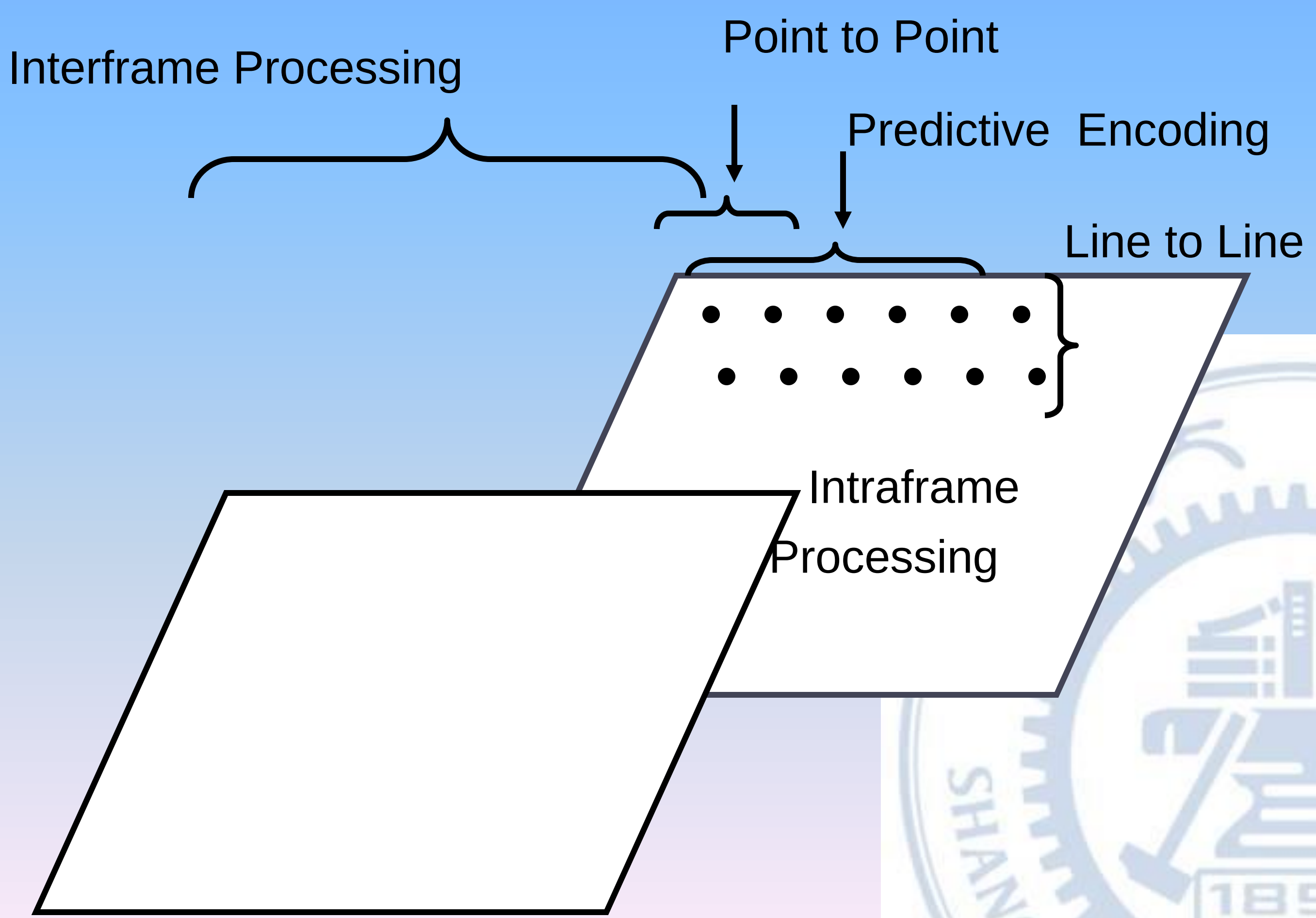
FIGURE 8.1 Computer generated 256 × 256 × 8 bit images with (a) coding redundancy, (b) spatial redundancy, and (c) irrelevant information. (Each was designed to demonstrate one principal redundancy but may exhibit others as well.)

Information vs Data



DATA = INFORMATION + REDUNDANT DATA

Interframe and Intraframe Processing



Fundamentals

- If we let b and b' denote the number of bits (or information-carrying units) in two representation of the same information, the relative data redundancy R of the representation with b is

$$R = 1 - \frac{1}{C}$$

where C, commonly called the compression ratio, is defined as

$$C = \frac{b}{b'}$$

Fundamentals

- Coding redundancy
  - Example 8.1

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	–	8	–	0

TABLE 8.1 Example of variable-length coding.

$$L_{avg} = 1.81$$

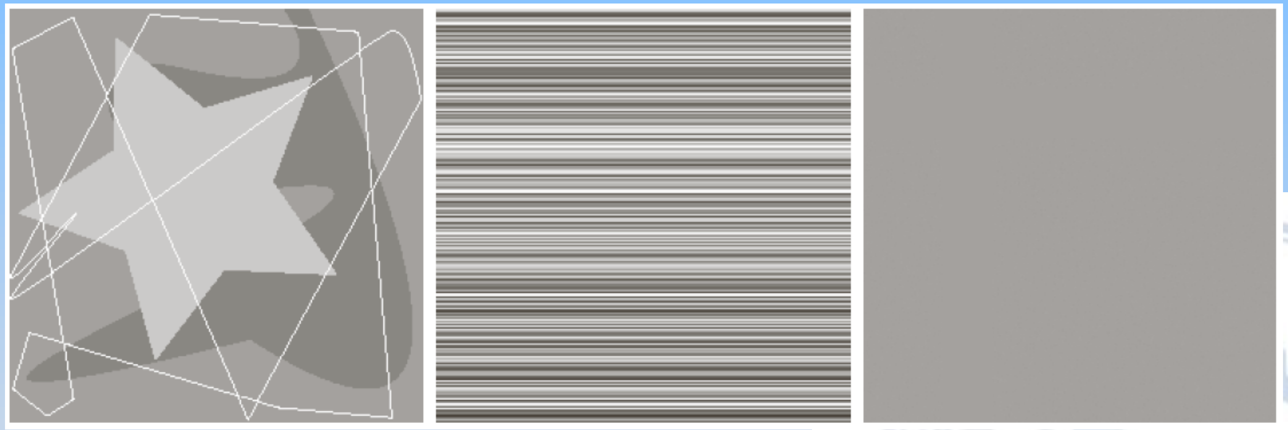
$$C = \frac{8}{L_{avg}} = 4.42$$

$$R = 1 - \frac{1}{C} = 0.774$$

77.4% of the data in the original 8-bit 2-D intensity array is redundant

Fundamentals

- Spatial and temporal redundancy



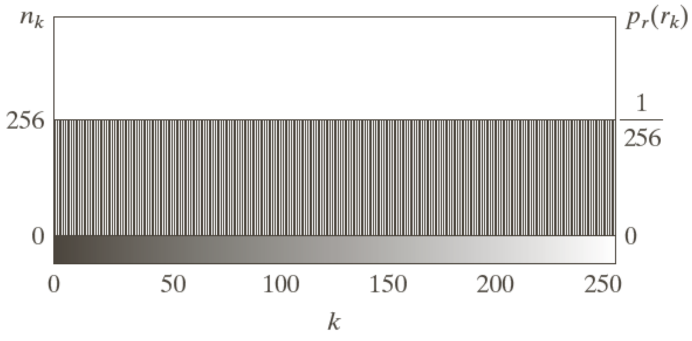
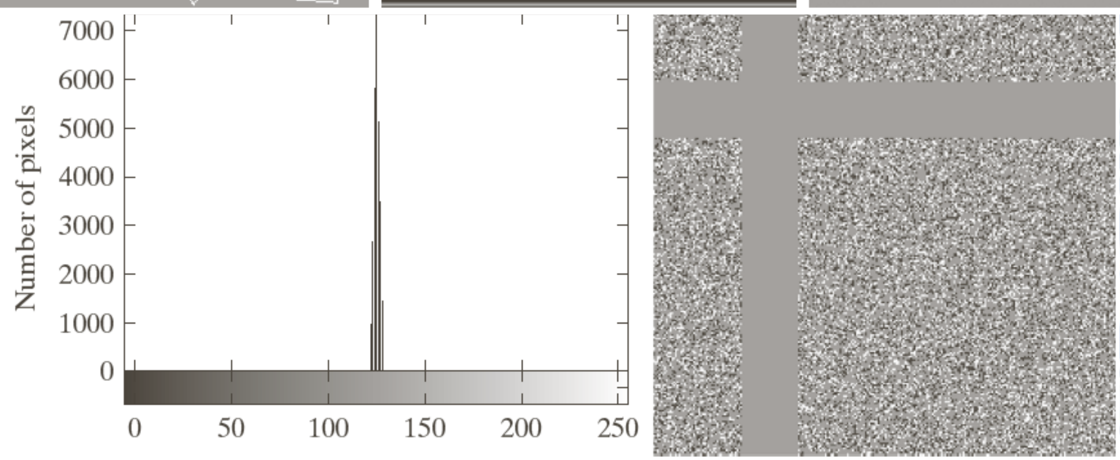


FIGURE 8.2 The intensity histogram of the image in Fig. 8.1(b).

Fundamentals

- Irrelevant information



a b

FIGURE 8.3 (a) Histogram of the image in Fig. 8.1(c) and (b) a histogram equalized version of the image.

Fundamentals

- Measuring image information
  - According to information theory, a random event E with probability is said to contain

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

    units of information.
  - The average information per source output, called the entropy of the source, is

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

Fundamentals

- Measuring image information
  - If an image is considered to be the output of an imaginary zero-memory "intensity source", the intensity source's entropy becomes

$$\tilde{H} = -\sum_{k=0}^{L-1} p_r(r_k) \log p_r(r_k)$$

  - It is not possible to code the intensity values of the imaginary source with fewer than $\tilde{H}$ bit/pixels

Fundamentals

- Noiseless coding theorem
  - also called Shannon's first theorem
  - defines the minimum average code word length per source symbol that can be achieved

$$\lim_{n\to\infty}\left[\frac{L_{avg,n}}{n}\right] = H$$

Fundamentals

- Terms of image compression
  - coding efficiency

$$\eta \triangleq \frac{H(x)}{L_{avg}}$$

  - coding redundancy

$$r \triangleq 1-\eta = \frac{L_{avg}-H(x)}{L_{avg}}$$

  - compression ratio

$$C \triangleq \frac{n}{n_d} \quad n_d \geq H(x)$$

  - maximum compression ratio of lossless compression

$$C_{\max} = \frac{n}{H(x)}$$

Fidelity Criteria

- Root-mean-square error

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y) \right]^2 \right]^{\frac{1}{2}}$$

- Mean-square signal-to noise ratio

$$SNR_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x,y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\hat{f}(x,y) - f(x,y) \right]^2}$$

Fidelity Criteria

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.

TABLE 8.2 Rating scale of the Television Allocations Study Organization. (Frendendall and Behrend.)

Image Compression Models

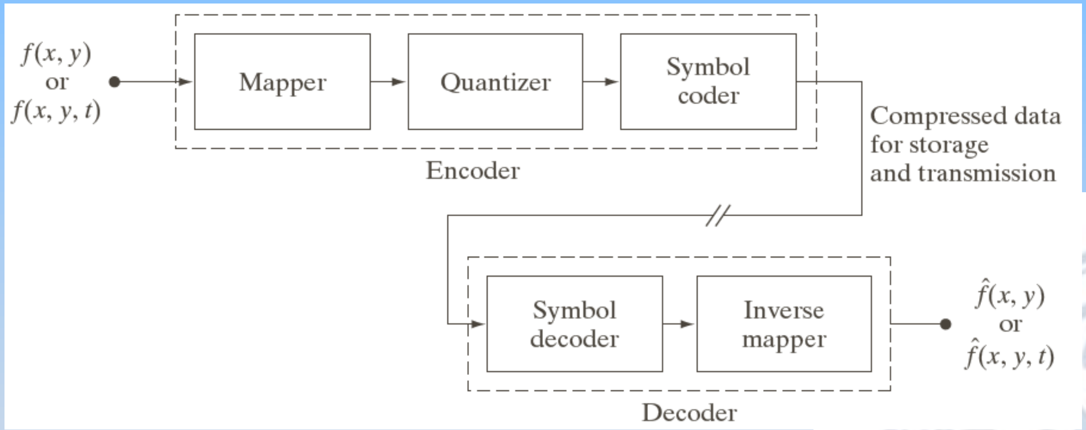


FIGURE 8.5 Functional block diagram of a general image compression system.

Image Formats, Containers, and Compression Standards

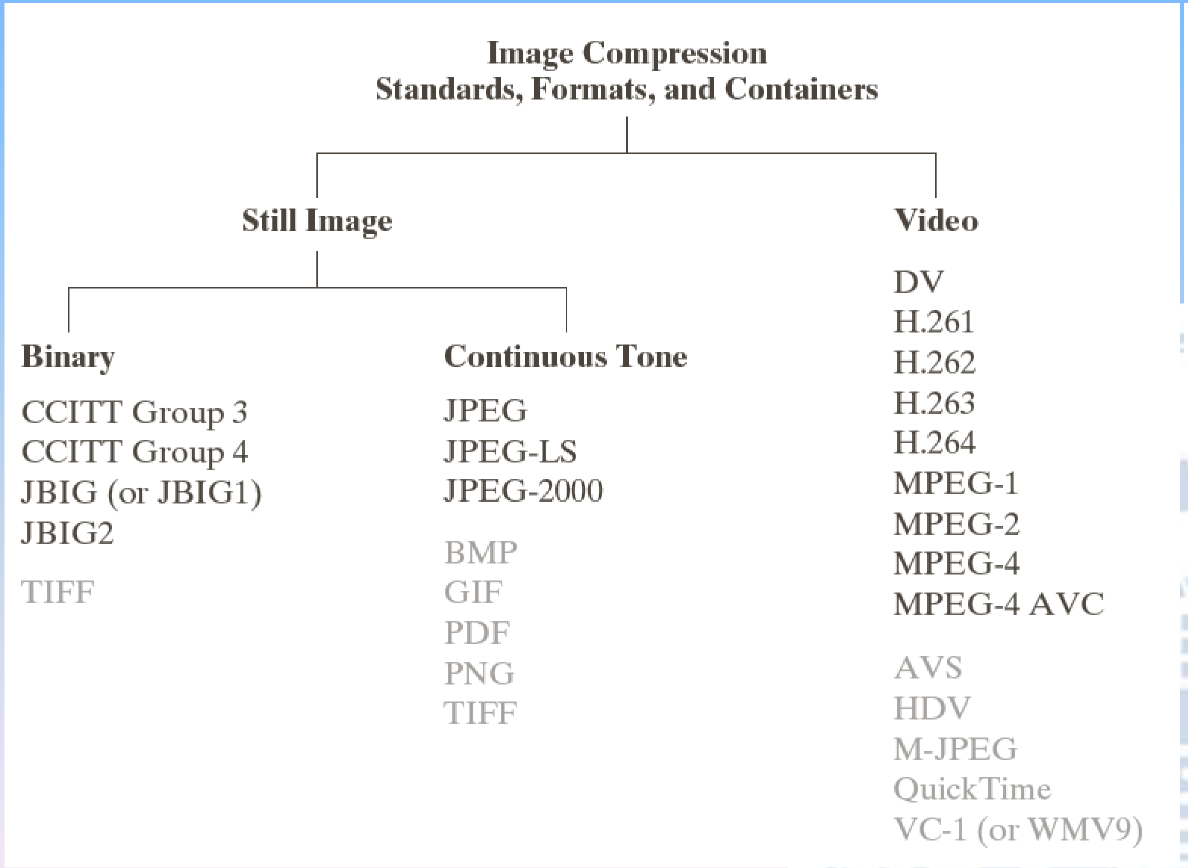


FIGURE 8.6 Some popular image compression standards, file formats, and containers. Internationally sanctioned entries are shown in black; all others are grayed.

Name	Organization	Description
Bi-Level Still Images		
CCITT Group 3	ITU-T	Designed as a facsimile (FAX) method for transmitting binary documents over telephone lines. Supports 1-D and 2-D run-length [8.2.5] and Huffman [8.2.1] coding.
CCITT Group 4	ITU-T	A simplified and streamlined version of the CCITT Group 3 standard supporting 2-D run-length coding only.
JBIG *or* JBIG1	ISO/IEC/ ITU-T	A *Joint Bi-level Image Experts Group* standard for progressive, lossless compression of bi-level images. Continuous-tone images of up to 6 bits/pixel can be coded on a bit-plane basis [8.2.7]. Context sensitive arithmetic coding [8.2.3] is used and an initial low resolution version of the image can be gradually enhanced with additional compressed data.
JBIG2	ISO/IEC/ ITU-T	A follow-on to JBIG1 for bi-level images in desktop, Internet, and FAX applications. The compression method used is content based, with dictionary based methods [8.2.6] for text and halftone regions, and Huffman [8.2.1] or arithmetic coding [8.2.3] for other image content. It can be lossy or lossless.
Continuous-Tone Still Images		
JPEG	ISO/IEC/ ITU-T	A *Joint Photographic Experts Group* standard for images of photographic quality. Its lossy *baseline coding system* (most commonly implemented) uses quantized discrete cosine transforms (DCT) on 8 × 8 image blocks [8.2.8], Huffman [8.2.1], and run-length [8.2.5] coding. It is one of the most popular methods for compressing images on the Internet.
JPEG-LS	ISO/IEC/ ITU-T	A lossless to near-lossless standard for continuous tone images based on adaptive prediction [8.2.9], context modeling [8.2.3], and Golomb coding [8.2.2].
JPEG-2000	ISO/IEC/ ITU-T	A follow-on to JPEG for increased compression of photographic quality images. Arithmetic coding [8.2.3] and quantized discrete wavelet transforms (DWT) [8.2.10] are used. The compression can be lossy or lossless.

(*Continues*)

TABLE 8.3 Internationally sanctioned image compression standards. The numbers in brackets refer to sections in this chapter.

TABLE 8.3 (*Continued*)

Name	Organization	Description
Video		
DV	IEC	*Digital Video*. A video standard tailored to home and semiprofessional video production applications and equipment—like electronic news gathering and camcorders. Frames are compressed independently for uncomplicated editing using a DCT-based approach [8.2.8] similar to JPEG.
H.261	ITU-T	A two-way videoconferencing standard for ISDN (*integrated services digital network*) lines. It supports non-interlaced 352 × 288 and 176 × 144 resolution images, called CIF (*Common Intermediate Format*) and QCIF (*Quarter CIF*), respectively. A DCT-based compression approach [8.2.8] similar to JPEG is used, with frame-to-frame prediction differencing [8.2.9] to reduce temporal redundancy. A block-based technique is used to compensate for motion between frames.
H.262	ITU-T	See MPEG-2 below.
H.263	ITU-T	An enhanced version of H.261 designed for ordinary telephone modems (i.e., 28.8 Kb/s) with additional resolutions: SQCIF (*Sub-Quarter* CIF 128 × 96), 4CIF (704 × 576), and 16CIF (1408 × 512).
H.264	ITU-T	An extension of H.261–H.263 for videoconferencing, Internet streaming, and television broadcasting. It supports prediction differences within frames [8.2.9], variable block size integer transforms (rather than the DCT), and context adaptive arithmetic coding [8.2.3].
MPEG-1	ISO/IEC	A *Motion Pictures Expert Group* standard for CD-ROM applications with non-interlaced video at up to 1.5 Mb/s. It is similar to H.261 but frame predictions can be based on the previous frame, next frame, or an interpolation of both. It is supported by almost all computers and DVD players.
MPEG-2	ISO/IEC	An extension of MPEG-1 designed for DVDs with transfer rates to 15 Mb/s. Supports interlaced video and HDTV. It is the most successful video standard to date.
MPEG-4	ISO/IEC	An extension of MPEG-2 that supports variable block sizes and prediction differencing [8.2.9] within frames.
MPEG-4 AVC	ISO/IEC	MPEG-4 Part 10 *Advanced Video Coding* (AVC). Identical to H.264 above.

Name	Organization	Description
Continuous-Tone Still Images		
BMP	Microsoft	*Windows Bitmap*. A file format used mainly for simple uncompressed images.
GIF	CompuServe	*Graphic Interchange Format*. A file format that uses lossless LZW coding [8.2.4] for 1- through 8-bit images. It is frequently used to make small animations and short low resolution films for the World Wide Web.
PDF	Adobe Systems	*Portable Document Format*. A format for representing 2-D documents in a device and resolution independent way. It can function as a container for JPEG, JPEG 2000, CCITT, and other compressed images. Some PDF versions have become ISO standards.
PNG	*World Wide Web Consortium* (W3C)	*Portable Network Graphics*. A file format that losslessly compresses full color images with transparency (up to 48 bits/pixel) by coding the difference between each pixel's value and a predicted value based on past pixels [8.2.9].
TIFF	Aldus	*Tagged Image File Format*. A flexible file format supporting a variety of image compression standards, including JPEG, JPEG-LS, JPEG-2000, JBIG2, and others.
Video		
AVS	MII	*Audio-Video Standard*. Similar to H.264 but uses exponential Golomb coding [8.2.2]. Developed in China.
HDV	Company consortium	*High Definition Video*. An extension of DV for HD television that uses MPEG-2 like compression, including temporal redundancy removal by prediction differencing [8.2.9].
M-JPEG	Various companies	*Motion JPEG*. A compression format in which each frame is compressed independently using JPEG.
Quick-Time	Apple Computer	A media container supporting DV, H.261, H.262, H.264, MPEG-1, MPEG-2, MPEG-4, and other video compression formats.
VC-1 WMV9	SMPTE Microsoft	The most used video format on the Internet. Adopted for HD and *Blu-ray* high-definition DVDs. It is similar to H.264/AVC, using an integer DCT with varying block sizes [8.2.8 and 8.2.9] and context dependent variable-length code tables [8.2.1]—but no predictions within frames.

TABLE 8.4
Popular image compression standards, file formats, and containers, not included in Table 8.3.

Source Coding

- To achieve less average length of bits per pixel of the image.
- Assigns short descriptions to the more frequent outcomes and long descriptions to the less frequent outcomes
- Entropy Coding Methods
  - Huffman Coding
  - Arithmetic Coding
  - Run Length Coding
  - Dictionary Codes
    - Lempel-Ziv77
    - Lempel-Ziv 78

Variable Length Code

- A variable-length code is a code which maps source symbols to a variable number of bits.
- Variable-length codes can allow lossless data compression and still be read back symbol by symbol. With the right coding strategy. An independent and identically-distributed source may be compressed almost arbitrarily close to its entropy. This is in contrast to fixed length coding methods, for which data compression is only possible for large blocks of data, and any compression beyond the logarithm of the total number of possibilities comes with a finite probability of failure.

Huffman Coding

- The most popular technique for removing coding redundancy
- The optimal variable-length coding
- When coding the symbols of an information source individually Huffman coding yields the smallest possible number of code symbols per source symbol
- Basic principle
  - assign the shortest possible code words to the most probable symbols

Huffman Coding

- The first step is to create a series of source reductions by ordering the probabilities of the symbols
  - combine 2 lowest probability symbols into 1 single symbol
  - use the new single symbol to replace the 2 symbols in the next source reduction
  - until a reduced source with 2 symbols is reached

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				

FIGURE 8.7 Huffman source reductions.

Huffman Coding

- The second step is to code each reduced source
  - start with the smallest source and working back to the original source
  - symbols 0 and 1 are assigned to the 2 symbols on the right

Original source			Source reduction							
Symbol	Probability	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								

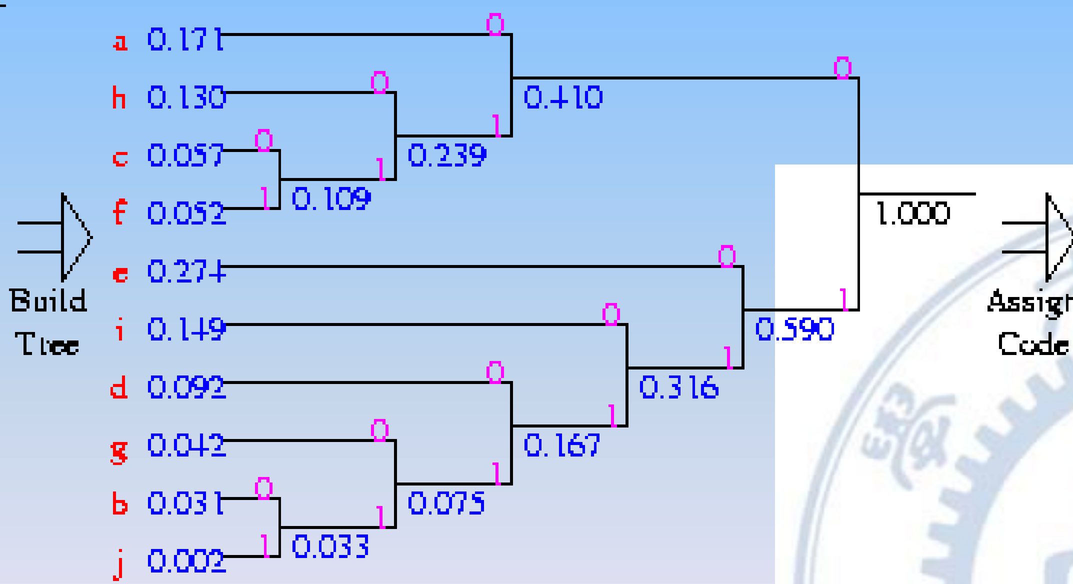
FIGURE 8.8 Huffman code assignment procedure.

Huffman Coding

X	p(X)
a	0.171
b	0.031
c	0.057
d	0.092
e	0.274
f	0.052
g	0.042
h	0.130
i	0.149
j	0.002

Huffman Coding

X	p(X)
a	0.171
b	0.031
c	0.057
d	0.092
e	0.274
f	0.052
g	0.042
h	0.130
i	0.149
j	0.002



X	Code
a	0 0
b	1 1 1 1 1 0
c	0 1 1 0
d	1 1 1 0
e	1 0
f	0 1 1 1
g	1 1 1 1 0
h	0 1 0
i	1 1 0
j	1 1 1 1 1 1

Huffman Coding

- Huffman's procedure creates the optimal code for a set of symbols
- After the code has been created, coding and/or decoding is accomplished in a simple lookup table manner
- Huffman code is an instantaneous uniquely decodable block code
  - each code word in a string of code symbols can be decoded without referencing succeeding symbols
  - any string of code symbols can be decoded in only one way
  - each source symbol is mapped into a fixed sequence

Prefix Code

- Prefix code is a type of variable-length code distinguished by the prefix property, which states that in the alphabet, there is no valid code word is a prefix of any other ones. The prefix code is meaningful, since it ensures the unique decoding of the code words.
- On the other hand, the length of code word is constrained for prefix code. Typically, they commit the Kraft-MacMillan inequality $\sum_{i=1}^{n} 2^{-l_i} \leq 1$, where n is the number of code words and l_i is the length of ith code word.
- Huffman code is the prefix code.

Arithmetic Coding

Shannon-Fano-Elias Coding

- We take **X**={1,2,…,m}, p(x)>0 for all x.
- Modified cumulative distribution function $\overline{F} = \sum_{a<x} P(a) + \frac{1}{2} P(x)$
- Assume we round off $\overline{F(x)}$ to $l(x)$, which is denoted by $\lfloor \overline{F(x)} \rfloor_{l(x)}$
- The codeword of symbol x has *l(x)* $= \left\lceil \log \frac{1}{p(x)} \right\rceil + 1$ bits
- Codeword is the binary value of $\overline{F(x)}$ with *l(x)* bits

x	*P(x)*	*F(x)*	$\overline{F(x)}$	$\overline{F(x)}$ in binary	*l(x)*	codeword
1	0.25	0.25	0.125	0.001	3	001
2	0.25	0.50	0.375	0.011	3	011
3	0.20	0.70	0.600	0.1001	4	1001
4	0.15	0.85	0.775	0.1100011	4	1100
5	0.15	1.00	0.925	0.1110110	4	1110

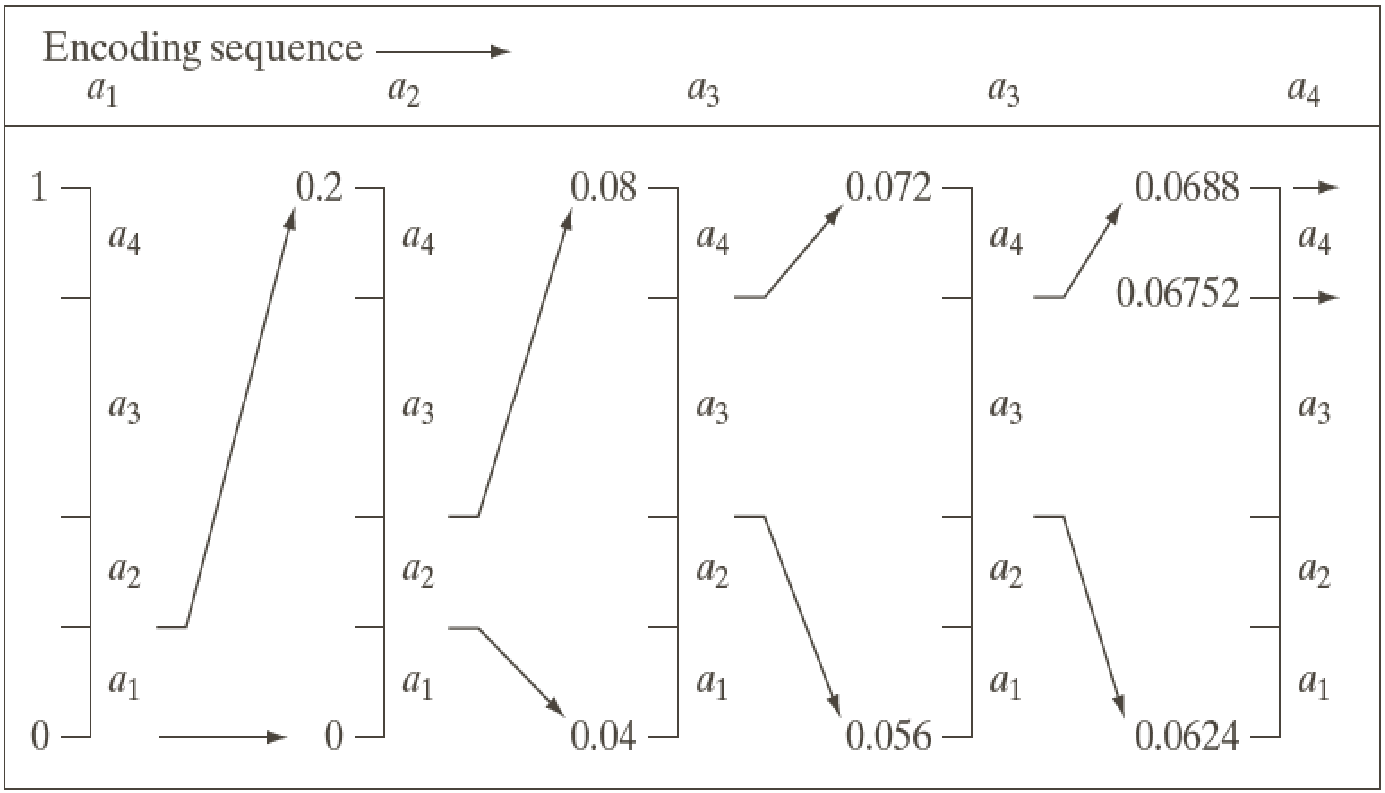
Arithmetic Coding

- **Arithmetic Coding**: a direct extension of Shannon-Fano-Elias coding calculate the probability mass function $p(x^n)$ and the cumulative distribution function $F(x^n)$ for the source sequence x^n
  - Lossless compression technique
  - Treat multiple symbols as a single data unit

Arithmetic coding procedure

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)

TABLE 8.6 Arithmetic coding example.



Encode $a_1\ a_2\ a_3\ a_3\ a_4$

[0.06752, 0.0688)

or

0.068

- The message $a_1\ a_2\ a_3\ a_3\ a_4$ is encoded using 3 decimal digits or 3/5 = 0.6 decimal digits per source symbol.

$$H = -\sum_{k=0}^{3} P(r_k) log(P(r_k))$$

- The entropy of this message is:

  $-(3 \times 0.2\log_{10}(0.2)+0.4\log_{10}(0.4))=0.5786$ digits/symbol

<u>Note:</u> finite precision arithmetic might cause problems due to truncations!

Zero-Run-Length Coding

- The notation (L,F)
  - L zeros in front of the nonzero value F
- EOB (End of Block)
  - A special coded value means that the rest elements are all zeros
  - If the last element of the vector is not zero, then the EOB marker will not be added
- An Example:
- 5 5 5 5 5 5 7 7 7 7 7 3 3 3 2 2 2 2 1 1 1 1 1 1 1
- Zero-Run-Length codes: (5，6）（7，5）（3，3）（2，4）（1，7）
- Can compress any type of data but cannot achieve high compression ratios compared to other compression methods

LZW Coding (interpixel redundancy)

- Requires no priori knowledge of pixel probability distribution values.
- Assigns fixed length code words to variable length sequences.
- Included in GIF and TIFF and PDF file formats

LZW Coding

- A codebook (or dictionary) needs to be constructed.
- Initially, the first 256 entries of the dictionary are assigned to the gray levels 0,1,2,..,255 (i.e., assuming 8 bits/pixel)

Consider a 4x4, 8 bit image

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

Initial Dictionary

Dictionary Location	Entry
0	0
1	1
.	.
255	255
256	-
511	-

LZW Coding (cont'd)

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

As the encoder examines image pixels, gray level sequences (i.e., **blocks**) that are not in the dictionary are assigned to a new entry.

Dictionary Location	Entry
0	0
1	1
.	.
255	255
256	-39-39
511	-

- Is 39 in the dictionary……..Yes
- What about 39-39………….No
- Then add 39-39 in entry 256

Example

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

Concatenated Sequence: CS = CR + P

Currently Recognized Sequence (CR)	Pixel Being Processed (P)	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		

CR = empty

If CS is found:

(1) No Output

(2) CR=CS

else:

(1) Output D(CR)

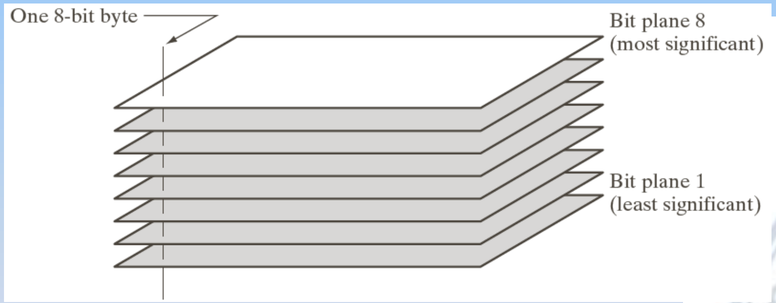
(2) Add CS to D

(3) CR=P

Bit-plane coding (interpixel redundancy)

- An effective technique to reduce inter pixel redundancy is to process each bit plane individually.

(1) Decompose an image into a series of binary images.



(2) Compress each binary image (e.g., using run-length coding)

Combining Huffman Coding with Run-length Coding

- Assuming that a message has been encoded using Huffman coding, additional compression can be achieved using run-length coding.

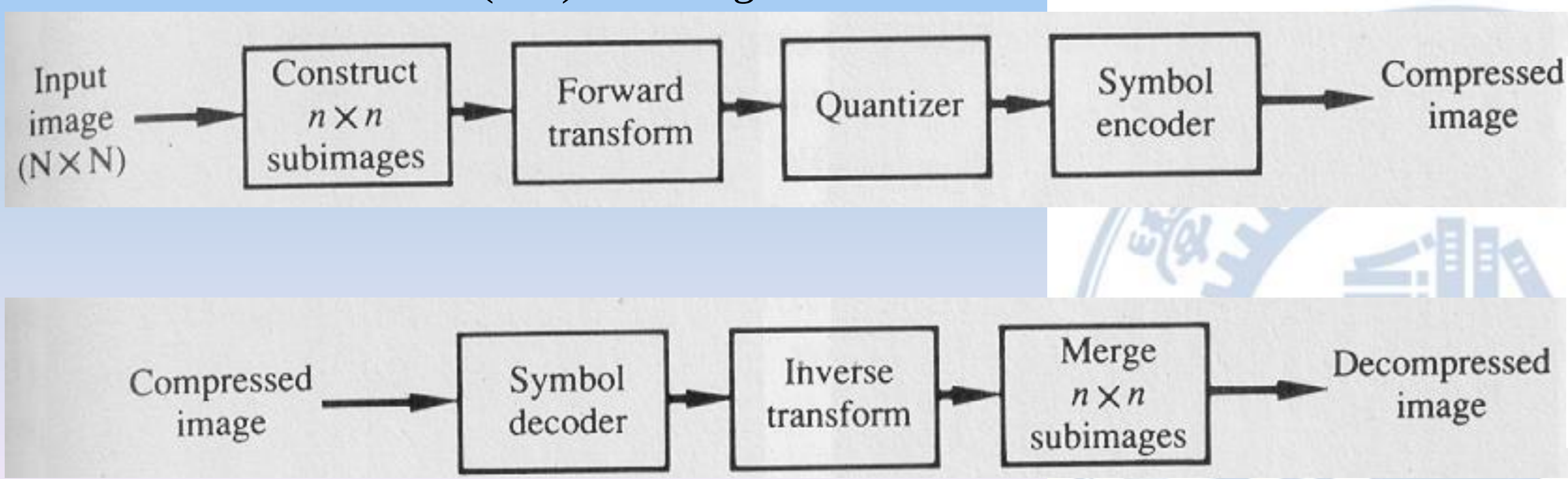
0 1 0 1 0 0 1 1 1 1 0 0

e.g., (0,1)(1,1)(0,1)(1,0)(0,2)(1,4)(0,2)

Lossy Compression

- Transform the image into a domain where compression can be performed more efficiently (i.e., reduce interpixel redundancies).

~ $(N/n)^2$ subimages



a
b

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

Reduce the Correlation between Pixels

- **Orthogonal Transform Coding**
  - KLT (Karhunen-Loeve Transform)
    - Maximal Decorrelation Process
  - DCT (Discrete Cosine Transform)
    - JPEG is a DCT-based image compression standard, which is a lossy coding method and may result in some loss of details and unrecoverable distortion.

Reduce the Correlation between Pixels

- **Subband Coding**
  - DWT (Discrete Wavelet Transform)
    - To divide the spectrum of an image into the lowpass and the highpass components, DWT is a famous example.
    - JPEG 2000 is a 2-dimension DWT based image compression standard.
- **Predictive Coding**
  - DPCM
    - To remove mutual redundancy between seccessive pixels and encode only the new information

Karhunen-Loeve Transform

- **KLT is** the optimum transform coder that is defined as the one that minimizes the mean square distortion of the reproduced data for a given number of total bits

The KLT

X: The input vector with size N-by-1

A: The transform matrix with size N-by-N

Y: The transformed vector with size N-by-1, and each components v(k) are mutually uncorrelated

C*xixj*: The covariance matrix of *xi* and *xj*

C*yiyj*: The covariance matrix of *yi* and *yj*

The transform matrix **A** is composed of the eigenvectors of the autocorrelation matrix **C***xixj*, which makes the output autocorrelation matrix **C***yiyj* be composed of the eigenvalues $\lambda_0, \lambda_1, \ldots \lambda_{N-1}$ in the diagonal direction. That is

$$
\begin{aligned}
C_{yy} &= E[(Y - E(Y))(Y - E(Y))^T] \\
&= E[YY^T]: zero - mean\ assumption \\
&= E[(Ax)(Ax)^T] = E[(Axx^T A^T)] \\
&= AE[xx^T]A^T = AC_{xx}A^T
\end{aligned}
$$

$$
C_{yy} = \begin{bmatrix} \lambda_0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_{N-1} \end{bmatrix} = AC_{xx}A^T
$$

Discrete Cosine Transform

- Why DCT is more appropriate for image compression than DFT?
  - The DCT can concentrate the energy of the transformed signal in low frequency, whereas the DFT can not
  - For image compression, the DCT can reduce the blocking effect than the DFT

Forward DCT

$$F(u,v)=\frac{2}{N}C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

for $u=0,\ldots,N-1$ and $v=0,\ldots,N-1$

where $N=8$ and $C(k)=\begin{cases}1/\sqrt{2} & \text{for } k=0\\ 1 & \text{otherwise}\end{cases}$

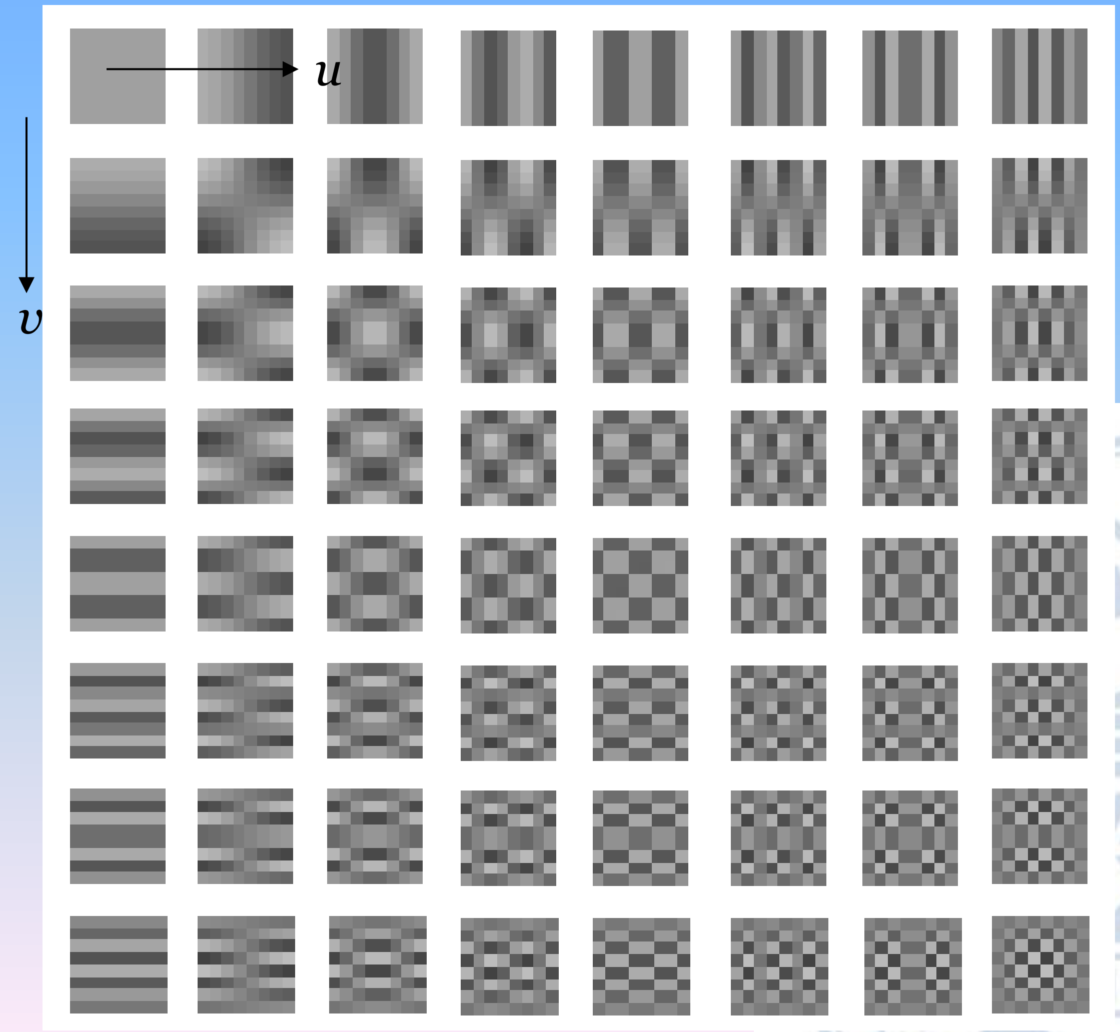
Inverse DCT

$$f(x,y)=\frac{2}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cos\left[\frac{\pi(2x+1)u}{2N}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right]$$

for $x=0,\ldots,N-1$ and $y=0,\ldots,N-1$ where $N=8$

Discrete Cosine Transform

The 8-by-8 DCT basis



a b c d

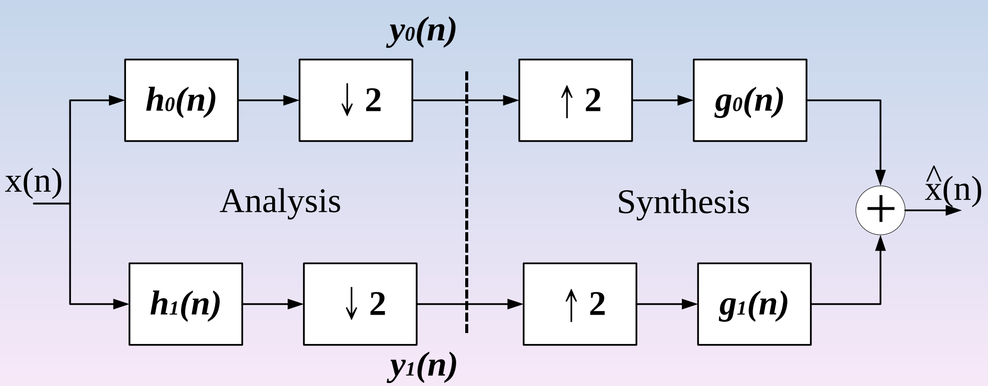
FIGURE 8.27 Approximations of Fig. 8.27(a) using 25% of the DCT coefficients and (b) 2×2 subimages, (c) 4×4 subimages, and (d) 8×8 subimages. The original image in (a) is a zoomed section of Fig. 8.9(a).

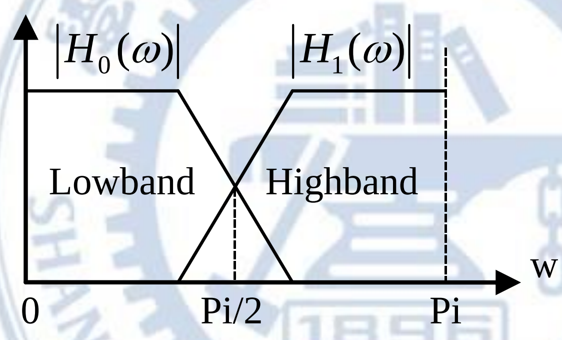
a b
c d

FIGURE 8.28 Approximations of Fig. 8.9(a) using 12.5% of the 8×8 DCT coefficients: (a)—(b) threshold coding results; (c)—(d) zonal coding results. The difference images are scaled by 4.

Discrete Wavelet Transform

- Subband Coding
  - The spectrum of the input data is decomposed into a set of bandlimitted components, which is called subbands
  - Ideally, the subbands can be assembled back to reconstruct the original spectrum without any error
- The input signal will be filtered into lowpass and highpass components through analysis filters
- The human perception system has different sensitivity to different frequency band
  - The human eyes are less sensitive to high frequency-band color components
  - The human ears is less sensitive to the low-frequency band less than 0.01 Hz and high-frequency band larger than 20 KHz





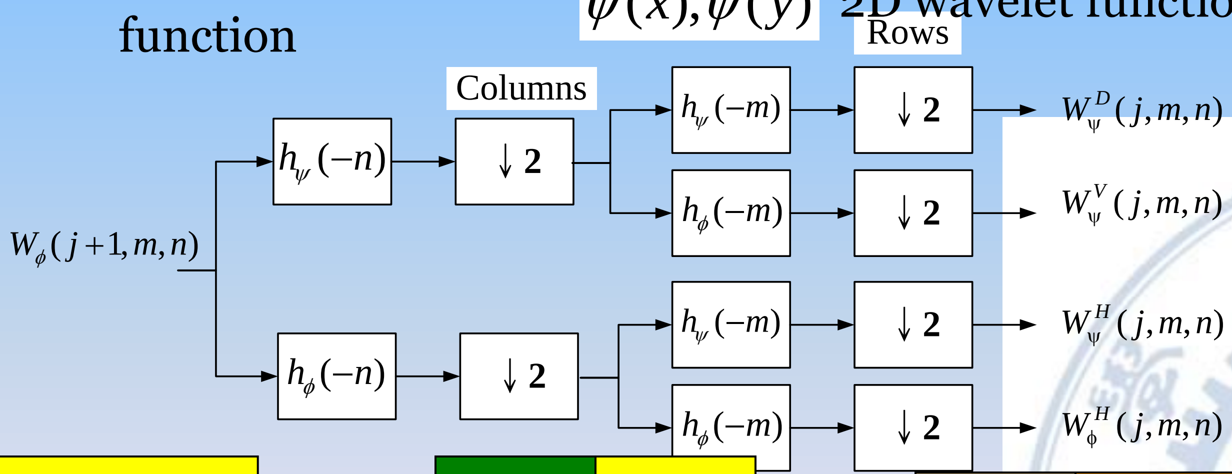
Discrete Wavelet Transform

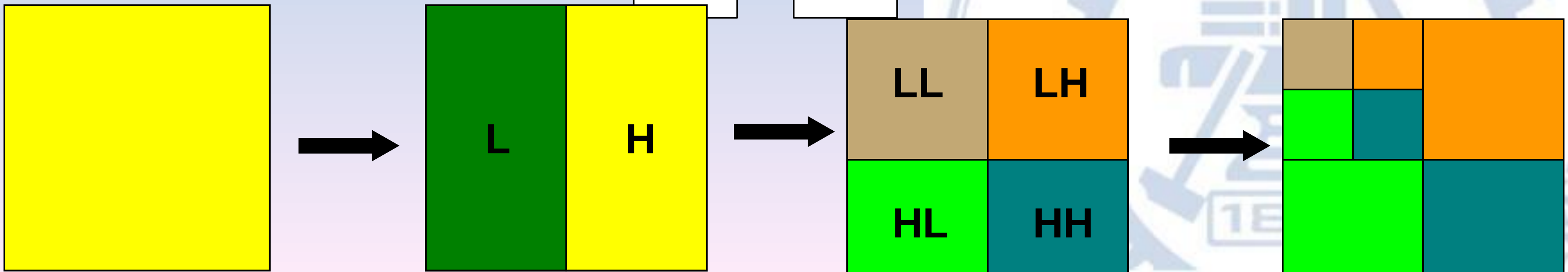
- 1D DWT applied alternatively to vertical and horizontal direction line by line
- The LL band is recursively decomposed, first vertically, and then horizontally

1D scaling function $\phi(x), \phi(y)$ 2D scaling function $\phi(x,y) = \phi(x)\phi(y)$

1D wavelet function $\psi(x), \psi(y)$ 2D wavelet function

$$\psi^D(x,y) = \psi(x)\psi(y)$$
$$\psi^H(x,y) = \psi(x)\phi(y)$$
$$\psi^V(x,y) = \phi(y)\psi(x)$$





Standards

- JPEG
- MPEG

Why Do We Need International Standards?

- International standardization is conducted to achieve inter-operability .
  - Only syntax and decoder are specified.
  - Encoder is not standardized and its optimization is left to the manufacturer.
- Standards provide state-of-the-art technology that is developed by a group of experts in the field.
  - Not only solve current problems, but also anticipate the future application requirements.

What Is JPEG?

- "Joint Photographic Expert Group". Voted as international standard in 1992.
- Works with color and grayscale images, e.g., satellite, medical, ...
- Lossy and lossless

JPEG
(Intraframe coding)

- First generation JPEG uses DCT+Run length Huffman entropy coding.
- Second generation JPEG (JPEG2000) uses wavelet transform + bit plane coding + Arithmetic entropy coding.

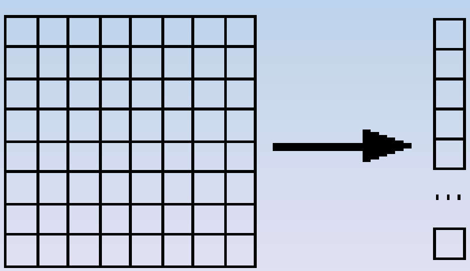
Why DCT Not DFT?

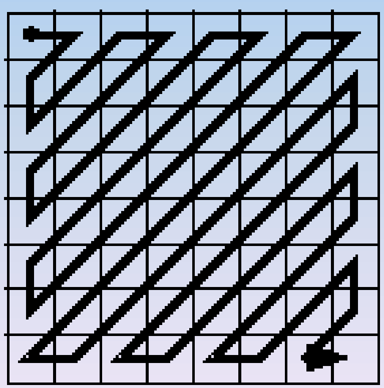
- DCT is similar to DFT, but can provide a better approximation with fewer coefficients

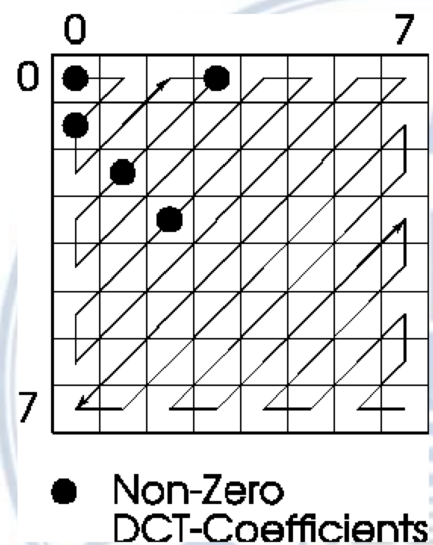
- The coefficients of DCT are real valued instead of complex valued in DFT.

Zig-zag Scan DCT Blocks

- Why? -- To group low frequency coefficients in top of vector.
- Maps 8 x 8 to a 1 x 64 vector.

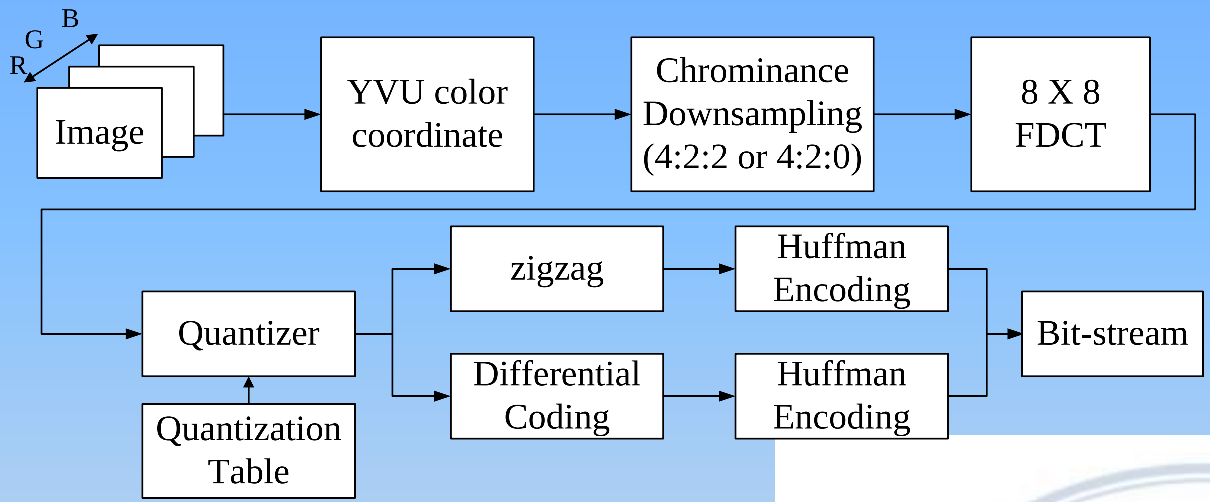




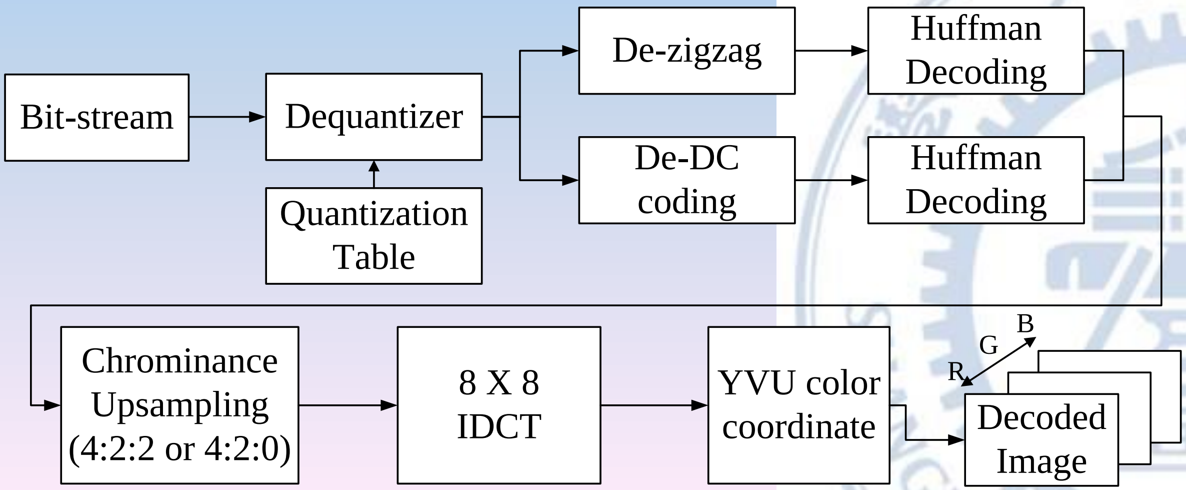


JPEG

The JPEG Encoder

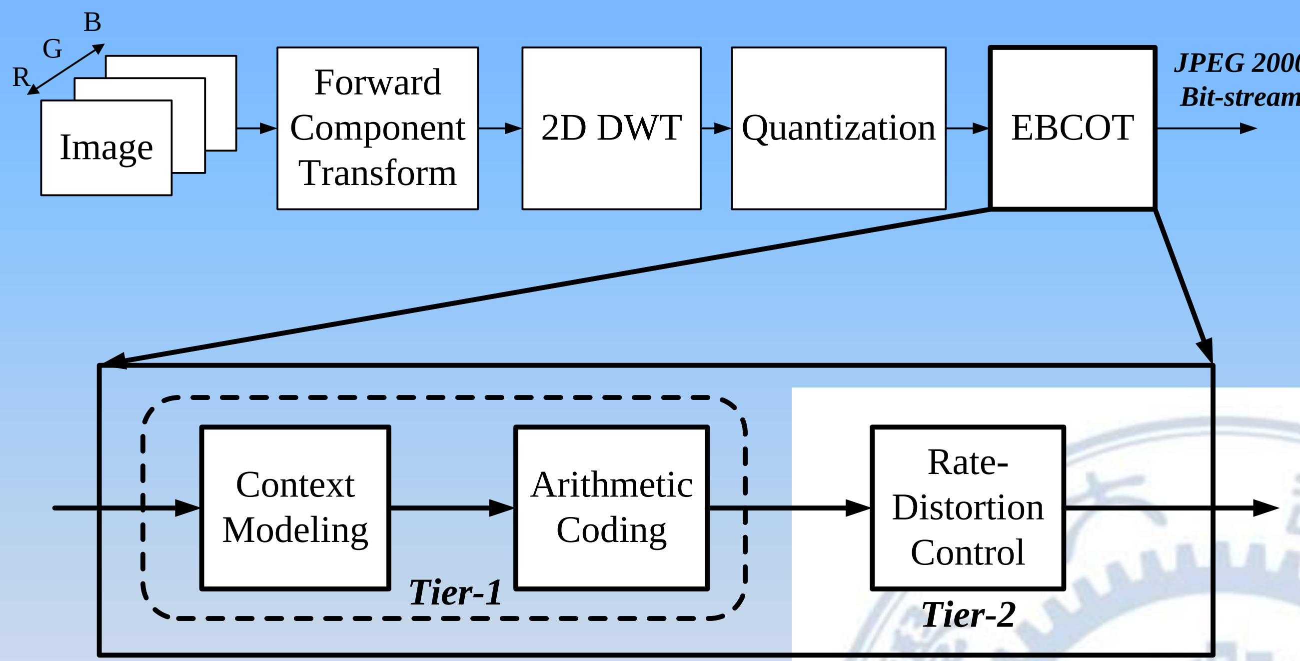


The JPEG Decoder

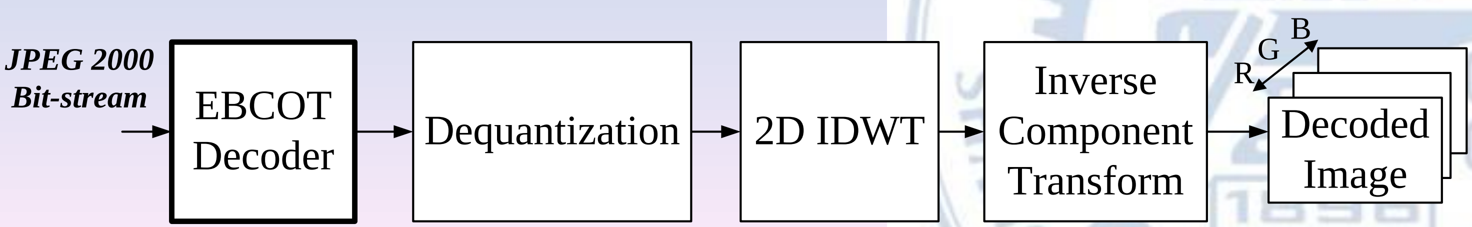


JPEG 2000

The JPEG 2000 Encoder



The JPEG 2000 Decoder



What Is MPEG ?

- "Motion Picture Expert Group", established in 1990 to create standard for delivery of audio and video.
- MPEG-1 : target VHS quality on a CD-ROM (320 x 240 + CD audio @ 1.5 Mbits/sec) .

MPEG
(Interframe Coding)

- Temporal DPCM is used to remove temporal redundancy first.
- The motion compensated error is coded with DCT+Run length Huffman entropy coding.

MPEG

- Temporal redundancy
  - Prediction along the motion trajectories (motion compensation prediction)

Motion Estimation

- The accuracy of motion estimation has a big influence on coding efficiency.
- Motion estimation is a very time-consuming work.
- Some fast algorithms are needed.

CAVLC in H.264/AVC

- Context adaptive variable length coding is adopted in H.264/AVC. It is easy to implement but commits high bit rates cost.
- CAVLC encodes the quantized DCT coefficients while other syntax elements are encoded with exp-Golomb codes.
- With a group of well-designed code word tables, CAVLC encodes the coefficients adaptively with the neighboring coded ones. The neigboring coefficient, selected as contexts, decide the exact code word table for coding.

Context Based Binary Arithmetic Coding (CABAC)

- It is noticed that the probability for '0' and '1' is fixed to p0 and p1 previously. However, it is not optimal for practical coding, especially for non-stationary data (e.g. image and video).
- To characterize the local statistics of encoding data, context based methods are introduced. They dynamically readjust the probability distribution for alphabet according to the encoded input, the so-called context. More sophisticatedly, context model is built with the data-driven methods or the analytic formulation to estimate the above distribution.

CABAC in H.264/AVC

- CABAC is adopted in the recent video coding standard H.264/AVC. Compared to CAVLC, CABAC typically provides a reduction in bit rate between 5%~15%.
- CABAC includes three steps: binarization, context modeling and coding.
  - Binarization decomposes the input syntax elements into sequence of bits, which is called bin.
  - Context modeling adaptively classifies the bins with various context models and assigns corresponding estimated probability.
  - Binary arithmetic coder (M-coder) is used for coding the estimated probability.

Homework 2:

- Problem 8.9,8.18 and the following problem:

Problem3:

Given a 3-symbol source {A,B,C} with source probabilities {0.4,0.2,0.4} , arithmetically encode the sequence BACCA, given the encode procedure, the entropy of the information, average code length and the encoding efficiency.

Requirements of Project Three now posted!

Thank You!