Image Compression Standards

Lecturer：Jun Xiao
(肖俊）
College of Software and Technology

Content

- The JPEG Standard
- The JPEG2000 Standard*

1. The JPEG Standard

Introduction

- JPEG : Joint Photographic Experts Group
  - Original name
    - The committee of the International Organization for Standardization (ISO)
  - The first international static image compression standard Published in 1992： ISO 10918-1
- Because of its pleasing properties, JPEG gained great success only several years after published
  - Almost 80 percents of images on web are compressed by the JPEG standards

Introduction

- JPEG is a lossy image compression method. It employs a transform coding method using the DCT (Discrete Cosine Transform).

- An image is a function of i and j (or conventionally x and y) in the spatial domain. The 2D DCT is used as one step in JPEG in order to yield a frequency response which is a function F(u, v) in the spatial frequency domain, indexed by two integers u and v.

Observations for JPEG Image Compression

- The effectiveness of the DCT transform coding method in JPEG relies on 3 major observations:

Observation 1: Useful image contents change relatively slowly across the image, i.e., it is unusual for intensity values to vary widely several times in a small area, for example, within an 8×8 image block.

- much of the information in an image is repeated, hence “spatial redundancy”.

Observations for JPEG Image Compression

Observation 2: Psychophysical experiments suggest that humans are much less likely to notice the loss of very high spatial frequency components than the loss of lower frequency components.

- **the spatial redundancy can be reduced by largely reducing the high spatial frequency contents.**

Observation 3: Visual acuity (accuracy in distinguishing closely spaced lines) is much greater for gray ("black and white") than for color.

- chroma subsampling (4:2:0) is used in JPEG.

1.1 Main Steps in JPEG Image Compression

（1） Transform RGB to YIQ or YUV and subsample color

（2） Perform DCT on image blocks

（3） Apply Quantization

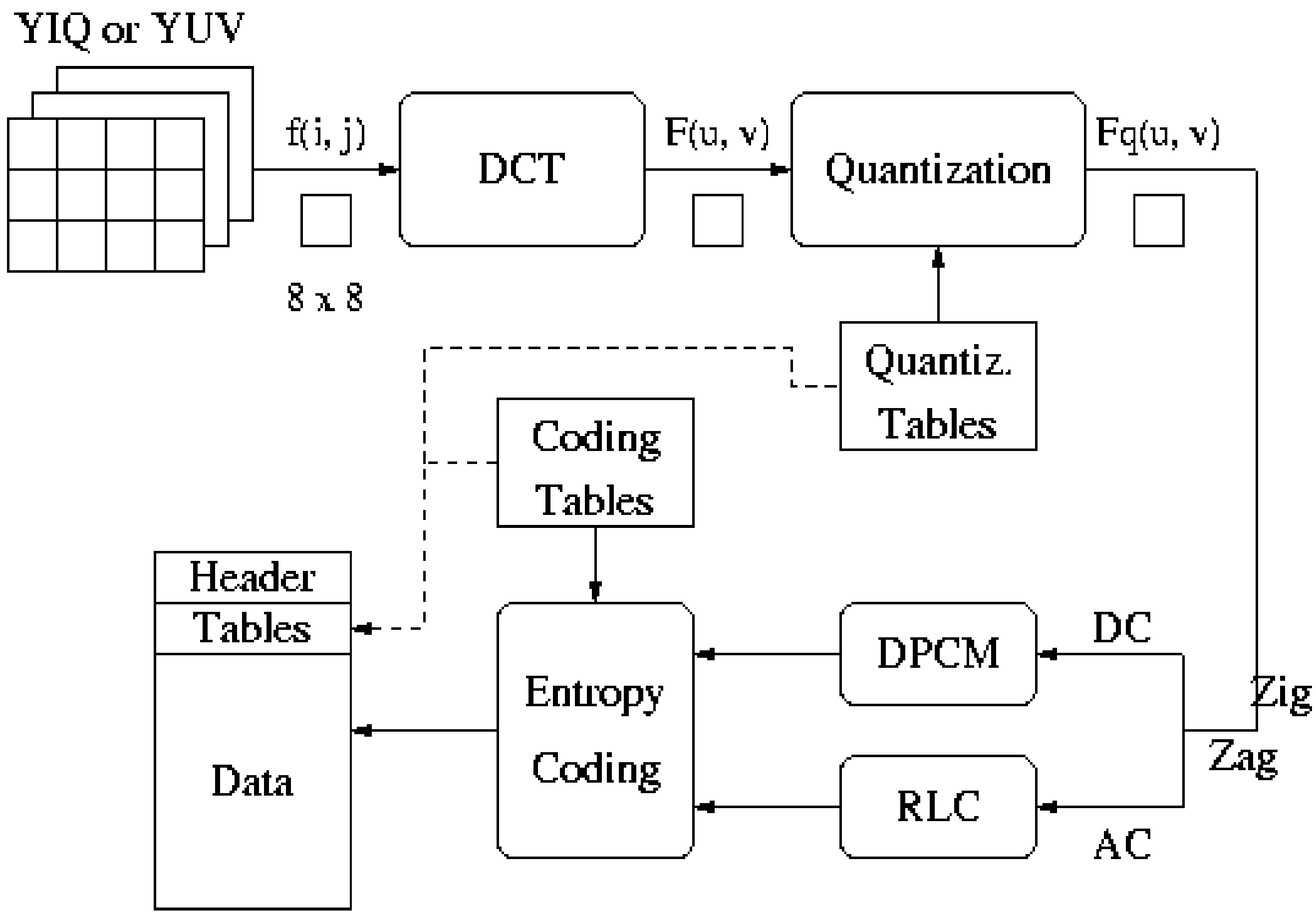
（4） Zigzag Ordering

（5） DPCM on DC coefficients

（6） RLE on AC coefficients

（7） Perform entropy coding

1.1 Main Steps in JPEG Image Compression

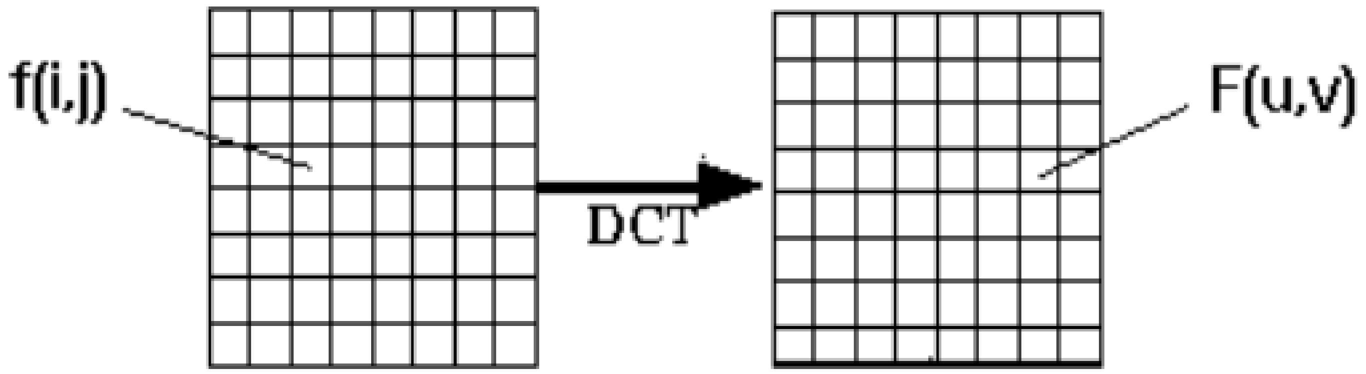


Block diagram for JPEG encoder

1.1 Main steps: DCT

DCT (Discrete Cosine Transformation)

Each image is divided into 8 × 8 blocks. The 2D DCT is applied to each block image $f(i, j)$, with output being the DCT coefficients $F(u, v)$ for each block.



1.1 Main steps: DCT

- Why the block size is 8×8?
  - Compromise between accuracy and computation

- Removing blocking artifacts is an important concern of researcher

- Using blocks, however, has the effect of isolating each block from its neighboring context. This is why JPEG images look choppy (“blocky”) when a high *compression ratio* is specified by the user.

1.1 Main steps: Quantization

$$\hat{F}(u,v) = round\left(\frac{F(u,v)}{Q(u,v)}\right) \tag{9.1}$$

- $F(u, v)$ 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.
  - **The quantization step is the main source for loss in JPEG compression.**
  - **The entries of $Q(u, v)$ tend to have larger values towards the lower right corner. This aims to introduce more loss at the higher spatial frequencies — a practice supported by Observations 1 and 2.**
  - **Table 9.1 and 9.2 show the default $Q(u, v)$ values obtained from psychophysical studies with the goal of maximizing the compression ratio while minimizing perceptual losses in JPEG images.**

1.1 Main steps: Quantization

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

1.1 Main steps: Quantization

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.

1.1 Main steps: Quantization

$$
\begin{matrix}
32 & 6 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
$$

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

$$
\begin{matrix}
512 & 66 & -10 & 0 & 0 & 0 & 0 & 0 \\
-12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-14 & 0 & 16 & 0 & 0 & 0 & 0 & 0 \\
-14 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
$$

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

$$
\begin{matrix}
199 & 196 & 191 & 186 & 182 & 178 & 177 & 176 \\
201 & 199 & 196 & 192 & 188 & 183 & 180 & 178 \\
203 & 203 & 202 & 200 & 195 & 189 & 183 & 180 \\
202 & 203 & 204 & 203 & 198 & 191 & 183 & 179 \\
200 & 201 & 202 & 201 & 196 & 189 & 182 & 177 \\
200 & 200 & 199 & 197 & 192 & 186 & 181 & 177 \\
204 & 202 & 199 & 195 & 190 & 186 & 183 & 181 \\
207 & 204 & 200 & 194 & 190 & 187 & 185 & 184
\end{matrix}
$$

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

$$
\begin{matrix}
1 & 6 & -2 & 2 & 7 & -3 & -2 & -1 \\
-1 & 4 & 2 & -4 & 1 & -1 & -2 & -3 \\
0 & -3 & -2 & -5 & 5 & -2 & 2 & -5 \\
-2 & -3 & -4 & -3 & -1 & -4 & 4 & 8 \\
0 & 4 & -2 & -1 & -1 & -1 & 5 & -2 \\
0 & 0 & 1 & 3 & 8 & 4 & 6 & -2 \\
1 & -2 & 0 & 5 & 1 & 1 & 4 & -6 \\
3 & -4 & 0 & 6 & -2 & -2 & 2 & 2
\end{matrix}
$$

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

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

1.1 Main steps: Quantization

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. 9.2: JPEG compression for a smooth image block.

1.1 Main steps: Quantization

−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

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

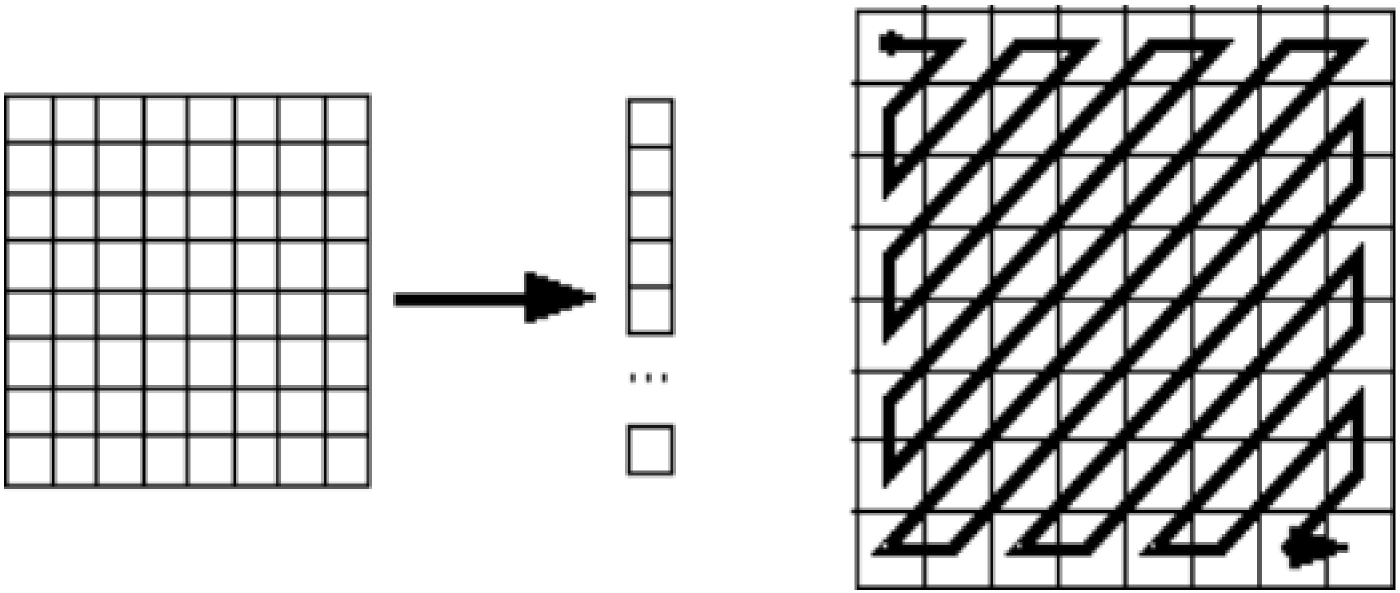
Fig. 9.3 (cont'd): JPEG compression for a textured image block.

1.1 Main steps: Quantization

- Conclusions:
  - **Reducing the total number of bits needed**
  - **The main source for information loss**
  - **Introduce more loss for quickly changing image areas**

1.1 Main steps: Zigzag Scan

- Turns the 8×8 matrix into a 64 vector
  - Lower frequency components are at the front part of the vector
  - The higher frequency component at the rear part



1.1 Main steps: RLE on AC Coefficients

- The 1 x 64 size vector contains long runs of zeros
- RLE (Run-length Coding):
  - (*skip, value*)
  - *skip: number of zeros, value: the next nonzero value*
  - (0,0): the end of a block

DC

(32,6,-1,-1,0,-1,0,0,0,-1,0,0,1,0,0,…,0)

(0,6) (0,-1) (0,-1) (1,-1) (3,-1) (2,1) (0,0)

1.1 Main steps: DPCM on DC Coefficients

- The DC coefficients are coded separately from the AC ones.
  - The values of the DC coefficients for various blocks could be large and different
  - The DC coefficient is unlikely to change drastically within a short distance
  - This makes DPCM an ideal scheme for coding the DC coefficients
  - DPCM for the DC coefficients in JPEG is carried out on the entire image at once

1.1 Main steps: DPCM on DC Coefficients

- Coding the difference with the DC of the previous 8×8 block
  - DPCM（Differential Pulse Code Modulation）

$$d_i = DC_{i+1} - DC_i$$

$$d_0 = DC_0$$

$$150, 155, 149, 152, 144 \Rightarrow 150, 5, -6, 3, -8$$

1.1 Main steps: Entropy Coding (1)

- DC is represented by a pair of symbols
  - (size, amplitude)
  - SIZE indicates how many bits are needed for representing the coefficient
  - AMPLITUDE contains the actual bits

Size	Amplitude
1	-1, 1
2	-3, -2, 2, 3
3	-7..-4, 4..7
4	-15..-8, 8..15
. . . .	
10	-1023..-512, 512..1023

- e.g. : (150, 5, -6, 3, -8) $\longrightarrow$

  (8, 10010110), (3, 101), (3, 001), (2, 11), (4, 0111)

1.1 Main steps: Entropy Coding (1)

- e.g. : (150, 5, -6, 3, -8) →

  (8, 10010110), (3, 101), (3, 001), (2, 11), (4, 0111)
  - Size is Huffman coded
  - Amplitude is not Huffman coded
- Huffman table can be customized and stored in image header, otherwise, a default Huffman table is used.

- AC Coefficient -- two symbols:
  - Symbol_1: (RUNLENGTH, *SIZE*)
  - Symbol_2: (AMPLITUDE)
- Symbol_1 using Huffman coding, Symbol_2 is not

1.2 JPEG Modes

- Sequential Mode
- Progressive Mode.
- Hierarchical Mode.
- Lossless Mode

1.2 JPEG Mode : Sequential

- The Default JPEG mode
- Each image is encoded in a single left-to-right, top-to-bottom scan
- “Motion JPEG” video coded uses baseline sequential JPEG

1.2 JPEG Mode: Progressive (1)

Progressive JPEG delivers low quality versions of the image quickly, followed by higher quality passes.

1. Spectral selection: Takes advantage of the “spectral” (spatial frequency spectrum) characteristics of the DCT coefficients: higher AC components provide detail information.

Scan 1: Encode DC and first few AC components, e.g., AC1, AC2.
Scan 2: Encode a few more AC components, e.g., AC3, AC4, AC5.
...
Scan k: Encode the last few ACs, e.g., AC61, AC62, AC63.

1.2 JPEG Mode: Progressive (2)

2. **Successive approximation**: Instead of gradually encoding spectral bands, all DCT coefficients are encoded simultaneously but with their most significant bits (MSBs) first.

Scan 1: Encode the first few MSBs, e.g., Bits 7, 6, 5, 4.
Scan 2: Encode a few more less significant bits, e.g., Bit 3.
...
Scan m: Encode the least significant bit (LSB), Bit 0.

1.2 JPEG Mode: Hierarchical(1)

- The encoded image at the lowest resolution is basically a compressed low-pass filtered image, whereas the images at successively higher resolutions provide additional details (differences from the lower resolution images).

- Similar to Progressive JPEG, the Hierarchical JPEG images can be transmitted in multiple passes progressively improving quality.

1.2 JPEG Mode: Hierarchical(2)

1. Reduction of image resolution:

   Reduce resolution of the input image f (e.g., 512×512) by a factor of 2 in each dimension to obtain f_2 (e.g., 256 × 256). Repeat this to obtain f_4 (e.g., 128 × 128).

2. Compress low-resolution image f_4:

   Encode f_4 using any other JPEG method (e.g., Sequential, Progressive) to obtain F_4.

3. Compress difference image d_2:

   (a) Decode F_4 to obtain $\widetilde{f_4}$. Use any interpolation method to expand $\widetilde{f_4}$ to be of the same resolution as f_2 and call it $E(\widetilde{f_4})$.

   (b) Encode difference $d_2 = f_2 - E(\widetilde{f_4})$ using any other JPEG method (e.g., Sequential, Progressive) to generate D_2.

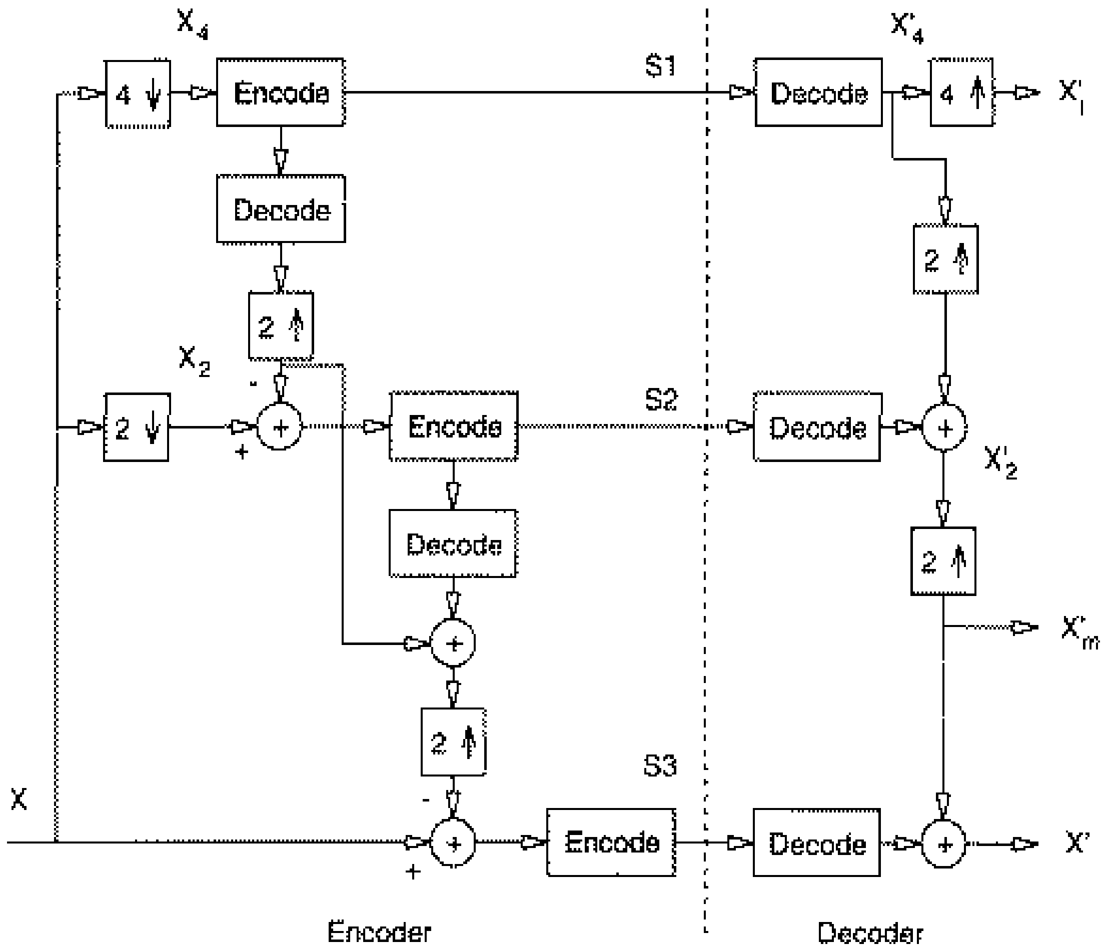
4. Compress difference image d_1:

   (a) Decode D_2 to obtain $\widetilde{d_2}$; add it to E($\widetilde{f_4}$) to get $\widetilde{f_2} = E(\widetilde{f_4}) + \widetilde{d_2}$ which is a version of f_2 after compression and decompression.

   (b) Encode difference $d_1 = f - E(\widetilde{f_2})$ using any other JPEG method (e.g., Sequential, Progressive) to generate D_1.

1.2 JPEG Mode: Hierarchical(3)

Encode an image in a hierarchy of several different resolutions



1.2 JPEG Mode: Hierarchical(4)

1. Decompress the encoded low-resolution image F_4:

   – Decode F_4 using the same JPEG method as in the encoder to obtain $\widetilde{f}_4$.

2. Restore image $\widetilde{f}_2$ at the intermediate resolution:

   – Use $E(\widetilde{f}_4)+\widetilde{d}_2$ to obtain $\widetilde{f}_2$.

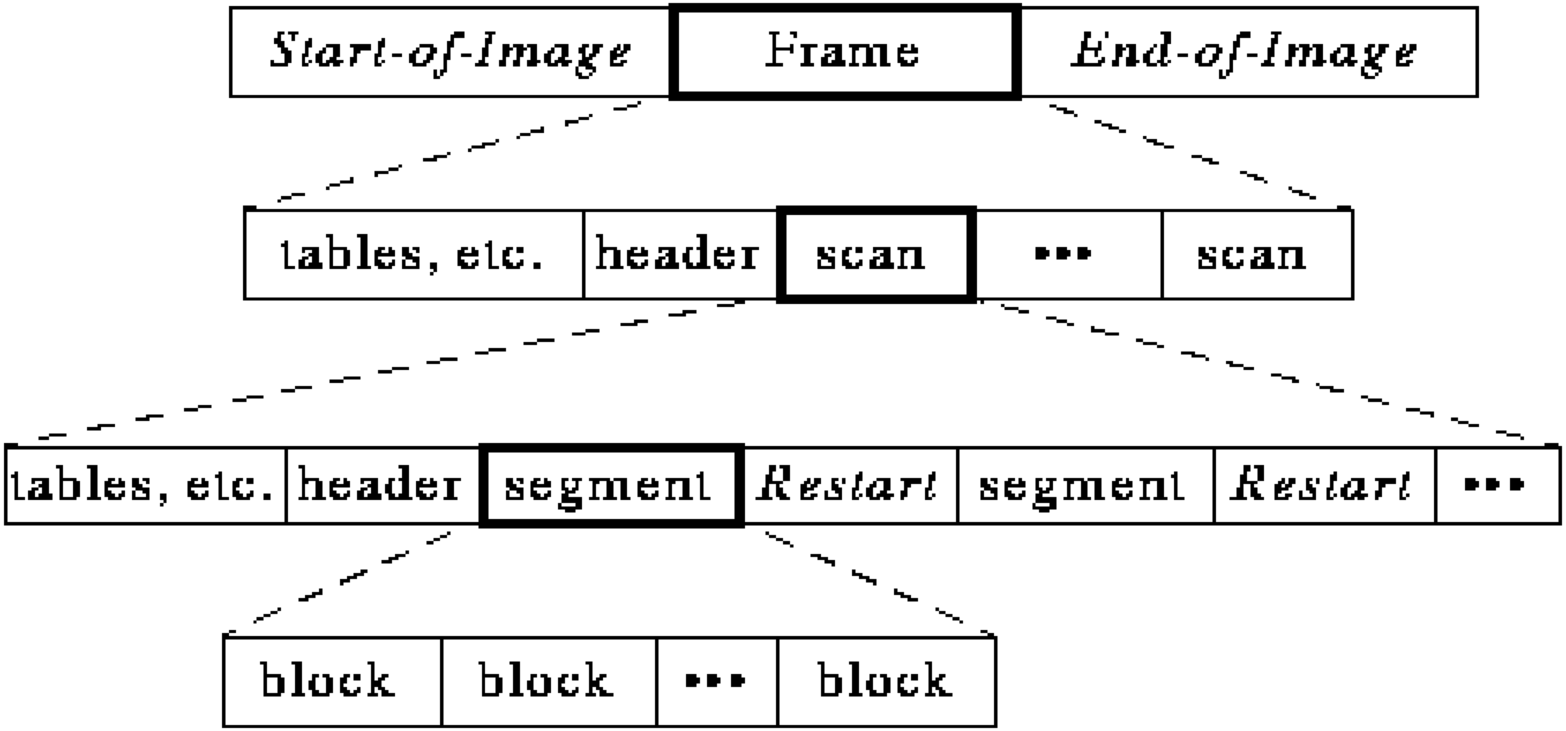
3. Restore image $\tilde{f}$ at the original resolution:

   – Use $E(\widetilde{f}_2)+\widetilde{d}_1$ to obtain $\tilde{f}$.

1.2 JPEG Mode: Lossless

- A special case of the JPEG where indeed there is no loss in its image quality
- It does not use DCT-based method! Instead, it uses a *predictive* (differential coding) method
- It's rarely used, since its compression ratio is very low compared to other lossy mode

1.3 A Glance at the JPEG Bitstream



JPEG Bitstream

A "Frame" is a picture, a "scan" is a pass through the pixels (e.g., the red component), a "segment" is a group of blocks, a "block" is an 8 x 8 group of pixels.

1.3 A Glance at the JPEG Bitstream

- Frame header
  - Sample precision (Bits per pixel)
  - (width, height) of image
  - Number of components
  - Unique ID (for each component)
  - Horizontal/vertical sampling factors (for each component)
  - Quantization table to use (for each component)

- Scan header
  - Number of components in scan
  - Component ID (for each component)
  - Huffman table (for each component)

2. The JPEG2000 Standard(*)

2.1 Why JPEG 2000

- A new-generation image compression standard
  - Provide both lossless compression and lossy compression in a same scheme
  - Excellent rate-distortion at low-bitrate compression
  - ROI (Region of interest) coding
  - Large image
  - Single decompression architecture
  - Transmission in noisy environments
  - Progressive transmission
  - Computer-generated imagery
  - Compound documents

2.3 Region-of-Interest coding

- Goal:
  - **Particular regions of the image may contain important information, thus should be coded with better quality than others.**

1.0bpp

0.5bpp

(ROI) can be coded with better quality than the rest of the image

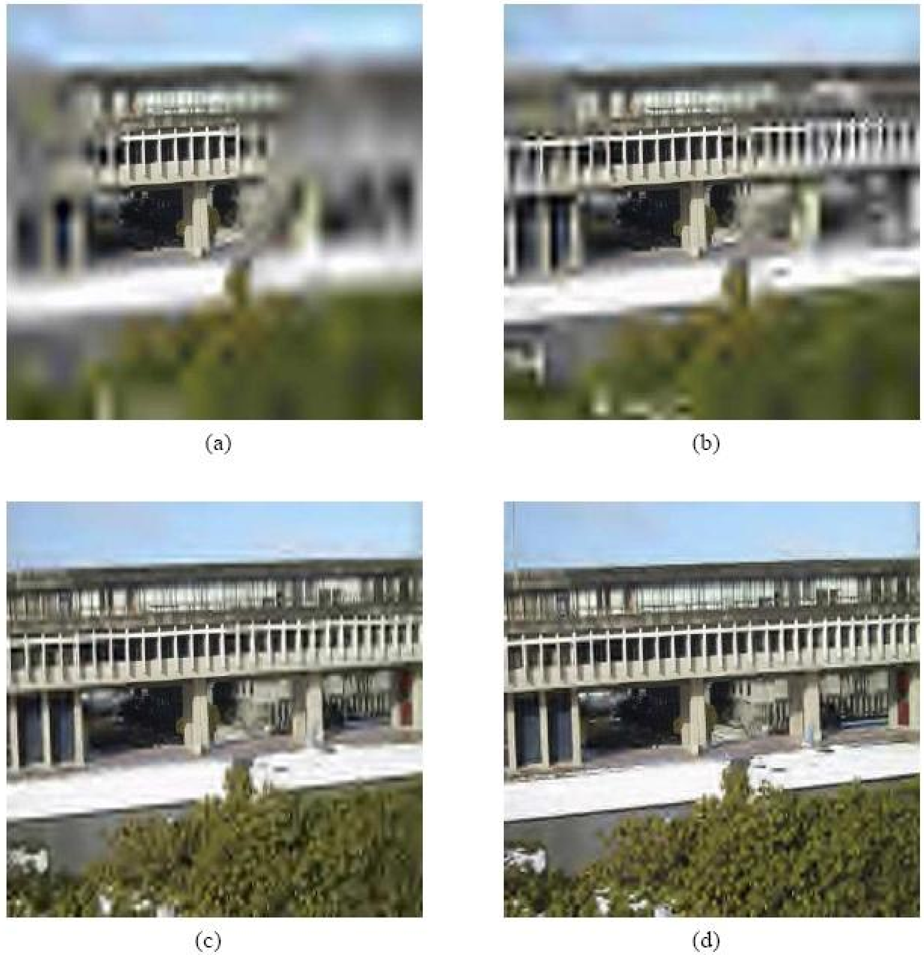
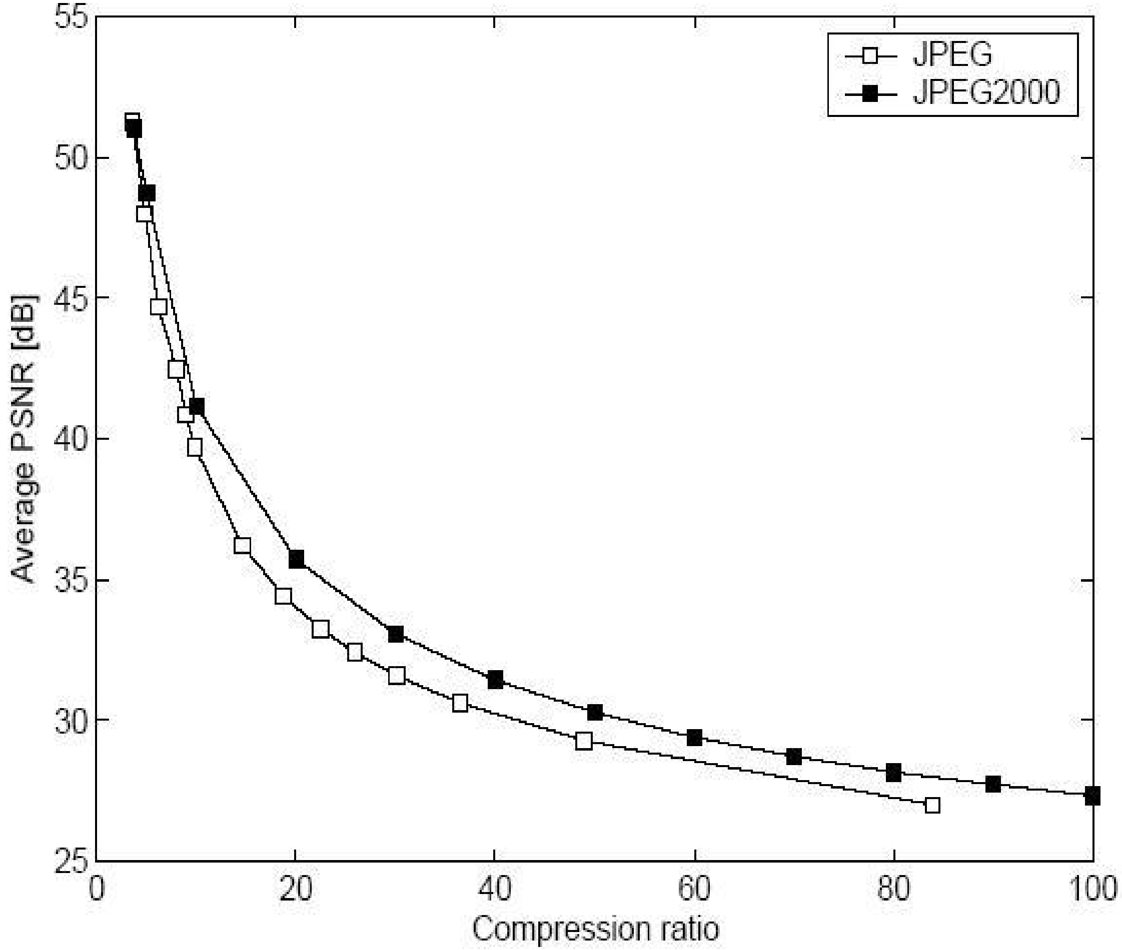
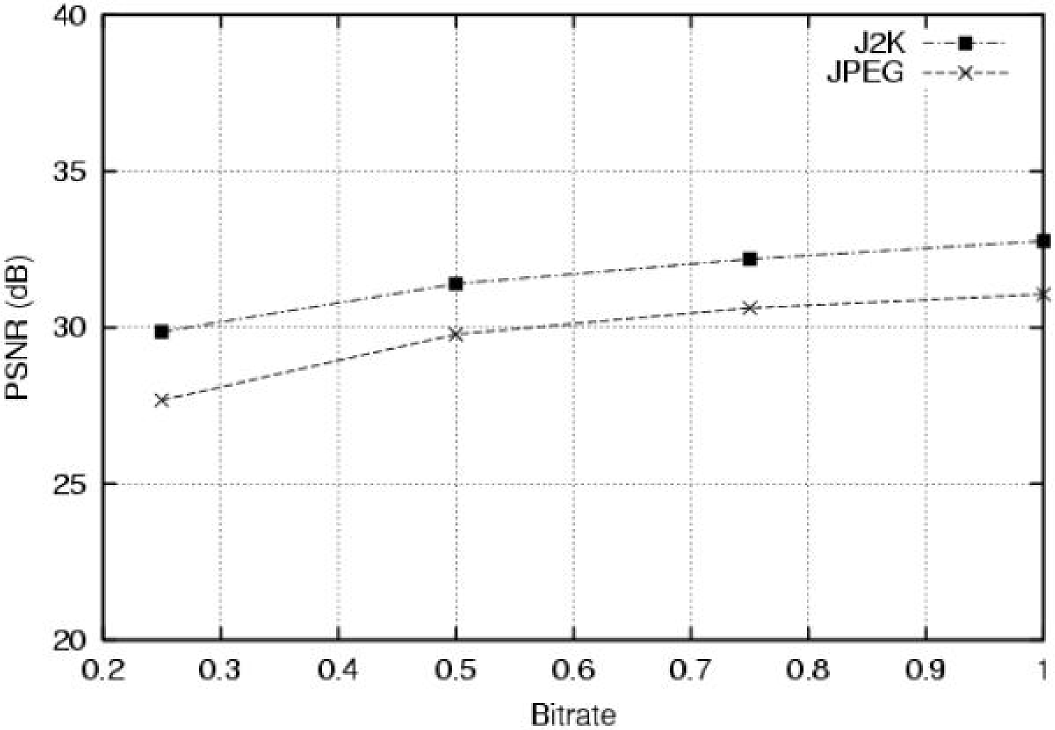


Fig. 9.11: Region of interest (ROI) coding of an image using a circularly shaped ROI. (a) 0.4 bpp, (b) 0.5 bpp, (c) 0.6bpp, and (d) 0.7 bpp.

2.4 Comparison for JPEG and JPEG2000



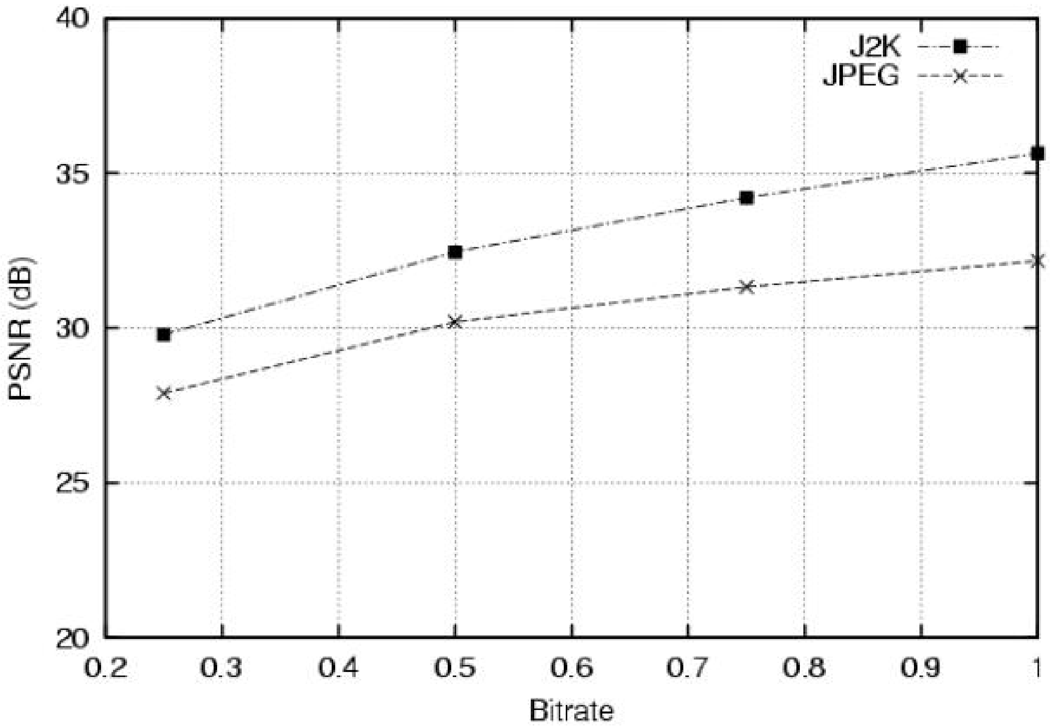
2.4 Comparison for JPEG and JPEG2000



(a)

Fig. 9.12: Performance comparison for JPEG and JPEG2000 on different image types. (a): Natural images.

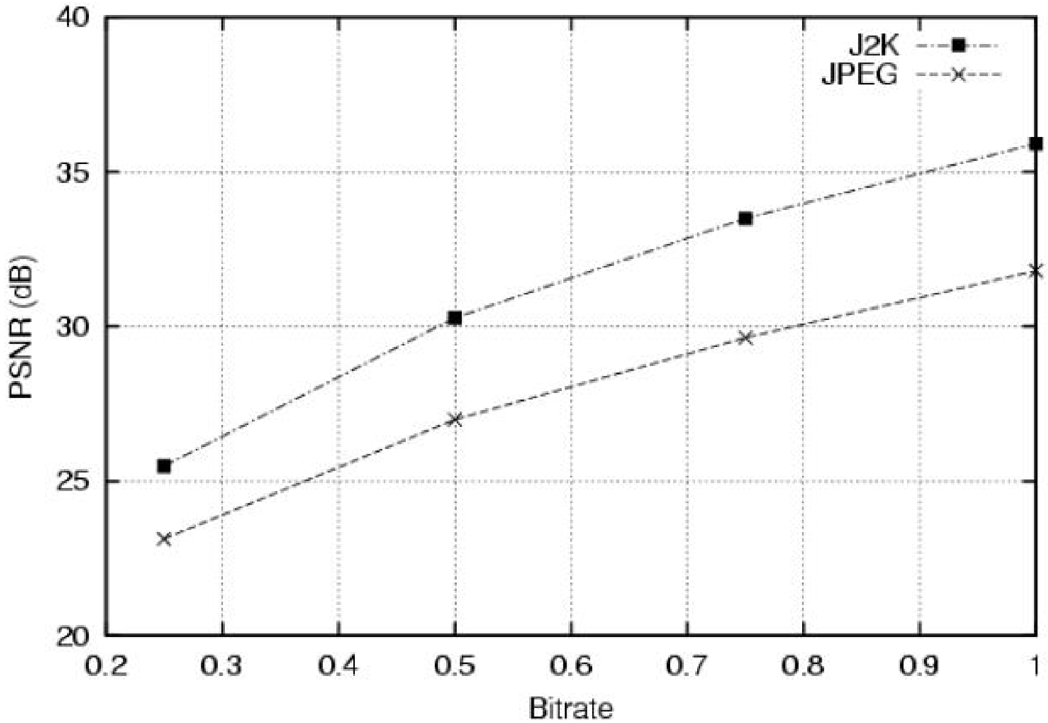
2.4 Comparison for JPEG and JPEG2000



(b)

Fig. 9.12: Performance comparison for JPEG and JPEG2000 on different image types. (b): Computer generated images.

2.4 Comparison for JPEG and JPEG2000



(c)

Fig. 9.12: Performance comparison for JPEG and JPEG2000 on different image types. (c): Medical images.

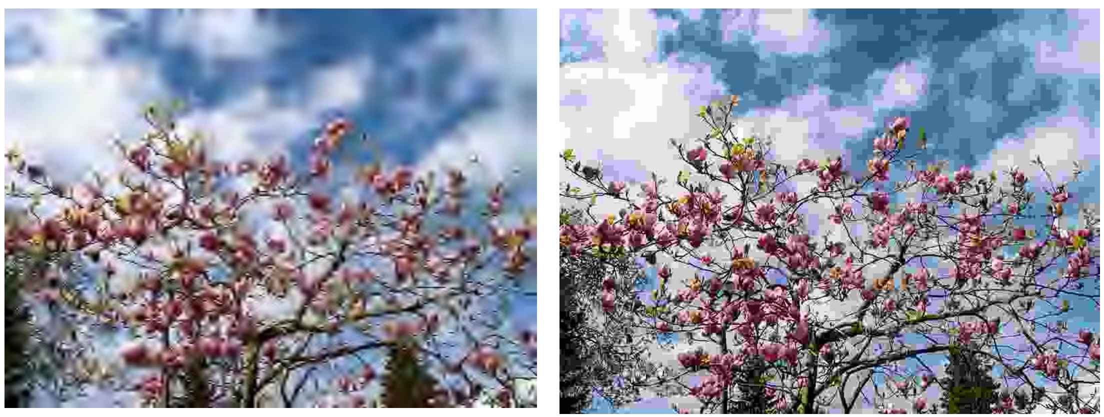
2.4 Comparison for JPEG and JPEG2000

(a)

Fig. 9.13: Comparison of JPEG and JPEG2000. (a) Original image.

2.4 Comparison for JPEG and JPEG2000

(b)



(c)

Fig. 9.13 (Cont'd): Comparison of JPEG and JPEG2000. (b) JPEG (left) and JPEG2000 (right) images compressed at 0.75 bpp. (c) JPEG (left) and JPEG2000 (right) images compressed at 0.25 bpp.

The End

Thanks！

Email: junx@cs.zju.edu.cn