CMPT 365 Multimedia Systems

Media Compression
- Image

Spring 2017

Edited from slides by Dr. Jiangchuan Liu

Facts about JPEG

- JPEG - Joint Photographic Experts Group
- International standard: 1992
- Most popular format
  - Other formats (.bmp) use similar techniques
- **Lossy** image compression
  - **transform coding** using the DCT
- JPEG 2000
  - New generation of JPEG – well, never succeeds
  - DWT (*Discrete Wavelet Transform)*

Three 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 8x8 image block.

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

Compression Ratio: 7.7

Compression Ratio: 33.9

Observations

- **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.

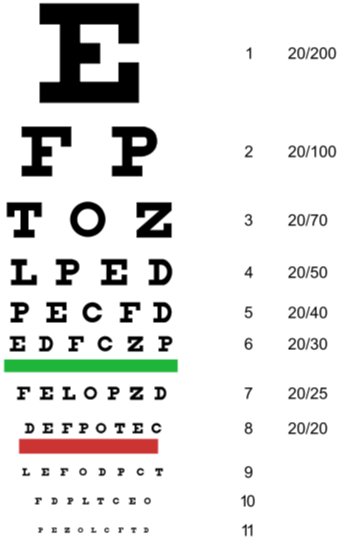
    - the spatial redundancy can be reduced by largely reducing the high spatial frequency contents.

Compression Ratio: 7.7

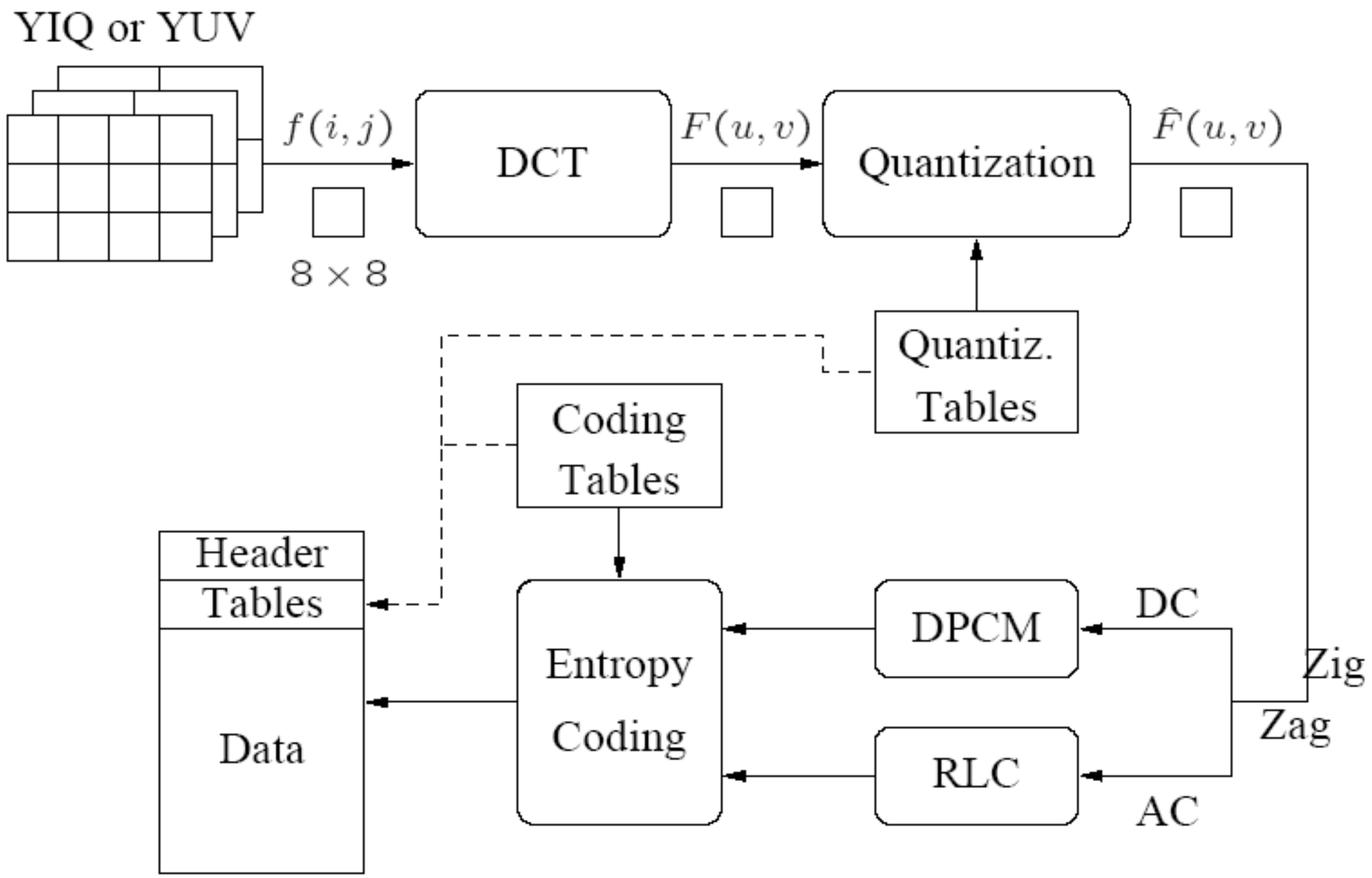
Compression Ratio: 33.9

Observations

- **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.



JPEG Diagram



JPEG Steps

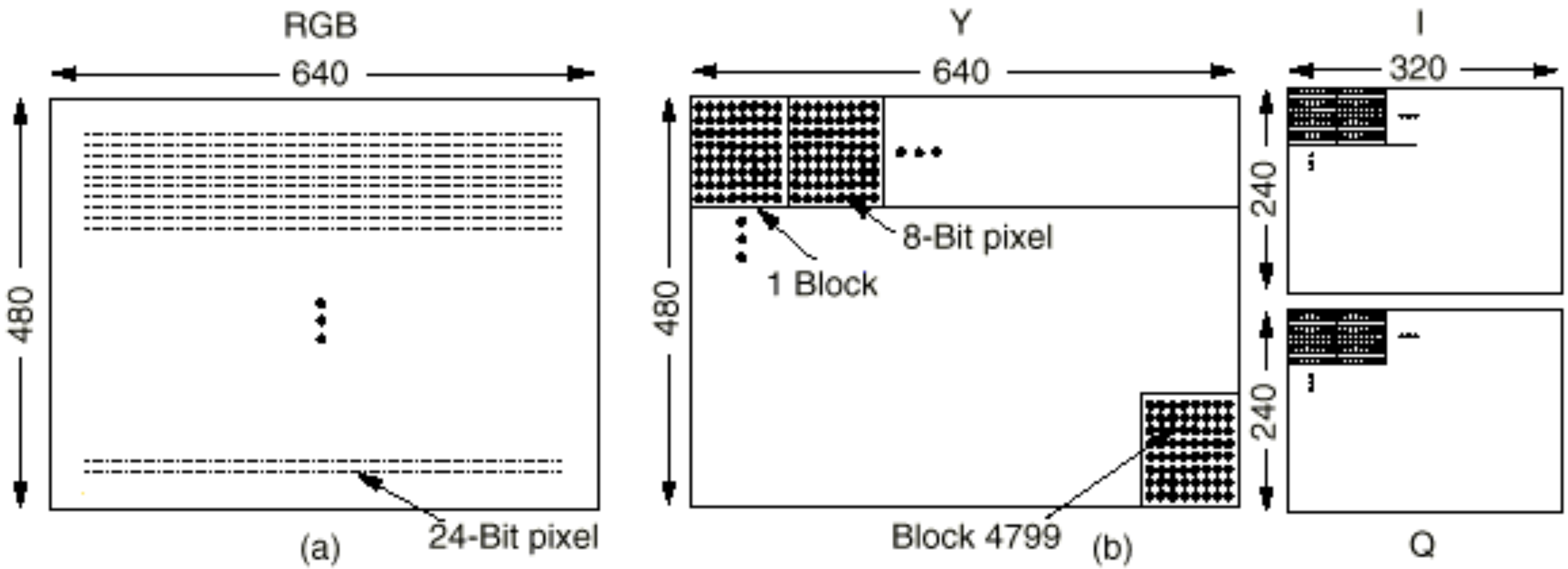
1 Block Preparation
  - RGB to YUV (YIQ) planes

2 Transform
  - 2D Discrete Cosine Transform (DCT) on 8x8 blocks.

3 Quantization
  - Quantized DCT Coefficients (lossy).

4 Encoding of Quantized Coefficients
  - Zigzag Scan
  - Differential Pulse Code Modulation (DPCM) on DC component
  - Run Length Encoding (RLE) on AC Components
  - Entropy Coding: Huffman or Arithmetic

JPEG: Block Preparation

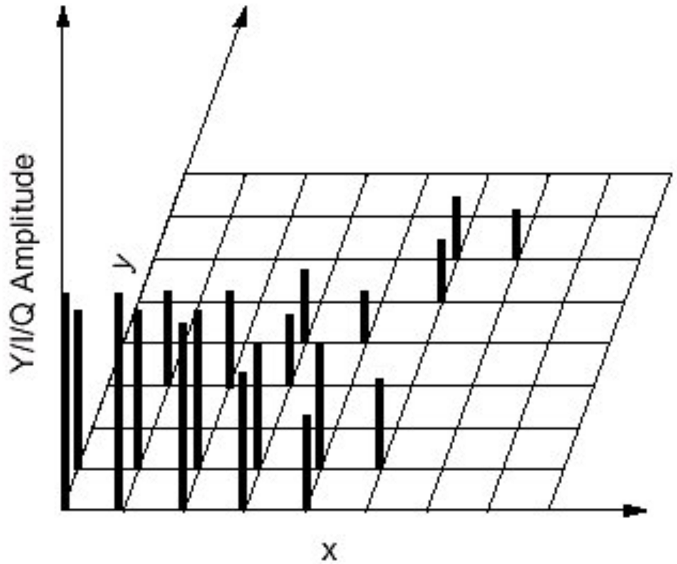


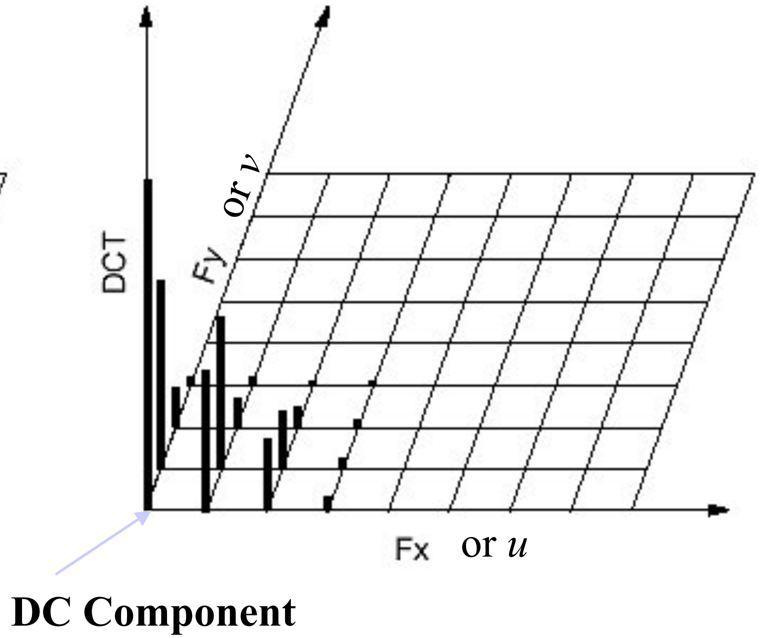
RGB Input Data **After Block Preparation**

Input image: 640 x 480 RGB (24 bits/pixel) transformed to three planes:

Y: (640 x 480, 8-bit/pixel) Luminance (brightness) plane.
U, V: (320 X 240 8-bits/pixel) Chrominance (color) planes.

8x8 DCT Example





Original values of an 8x8 block (in spatial domain)

Corresponding DCT coefficients (in frequency domain)

JPEG: Quantized DCT Coefficients

Uniform quantization:
Divide by constant N and round result.

In JPEG, each DCT F[u,v] is divided by a constant q(u,v).

The table of q(u,v) is called quantization table.

Quantization table

q(u,v)

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

DCT Coefficients

F[u,v]

150	80	40	14	4	2	1	0
92	75	36	10	6	1	0	0
52	38	26	8	7	4	0	0
12	8	6	4	2	1	0	0
4	3	2	0	0	0	0	0
2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized coefficients

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	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

Rounded F[u,v]/ Q(u,v)

Block Effect

- Using blocks, however, has the effect of isolating each block from its neighboring context.
  - choppy ("blocky") with high *compression ratio*

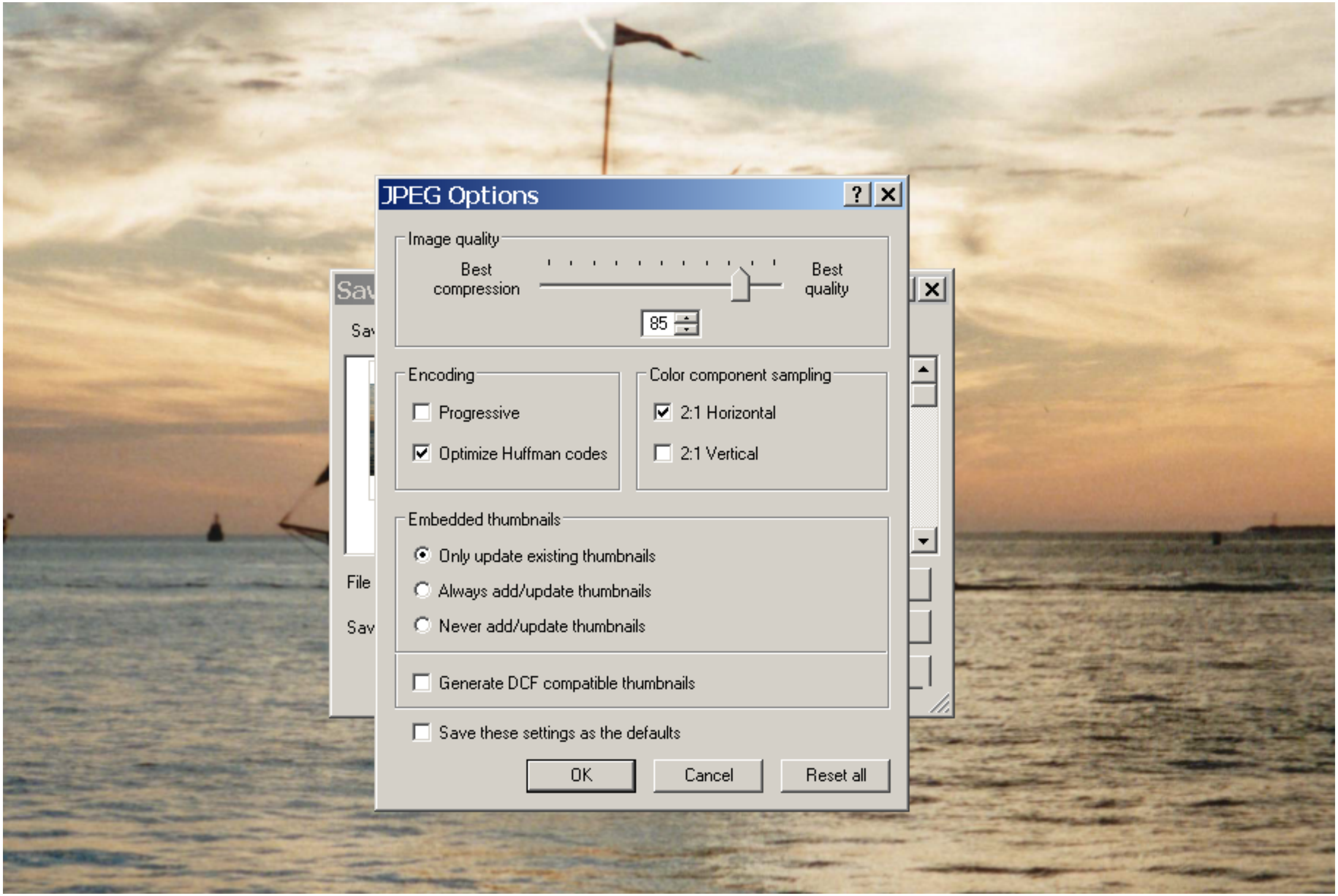
Compression Ratio: 7.7

Compression Ratio: 33.9

Compression Ratio: 60.1

More about Quantization

- **Quantization is the main source for loss**
  - $Q(u, v)$ of larger values towards lower right corner
    - More loss at the higher spatial frequencies
    - Supported by Observations 1 and 2.
  - Q(u,v) obtained from psychophysical studies
    - maximizing the compression ratio while minimizing perceptual losses


JPEG Options
Image quality
Best compression
Best quality
85
Encoding
Progressive
Optimize Huffman codes
Color component sampling
2:1 Horizontal
2:1 Vertical
Embedded thumbnails
Only update existing thumbnails
Always add/update thumbnails
Never add/update thumbnails
Generate DCF compatible thumbnails
Save these settings as the defaults
OK
Cancel
Reset all

More about Quantization

```
// qf is the user-selected compression quality
// Q is the default Quantization Matrix

// Qx is the scaled Quantization Matrix
// Q1 is a Quantization Matrix which is all 1's

if qf >= 50
    scaling_factor = (100-qf)/50;
else
    scaling_factor = (50/qf);
end
if scaling_factor != 0     // if qf is not 100
    Qx = round( Q*scaling_factor );
else
    Qx = Q1;               // no quantization
end
Qx = uint8(Qx);            // max is clamped to 255 for qf=1
```

JPEG: Zigzag Scan

Maps an 8x8 block into a 1 x 64 vector
Zigzag pattern group low frequency coefficients in top of vector.

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	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

JPEG: Encoding of Quantized DCT Coefficients

- DC Components (zero frequency)
  - DC component of a block is large and varied, but often close to the DC value of the previous block.
  - Encode the difference from previous
    - Differential Pulse Code Modulation (DPCM).
- AC components:
  - Lots of zeros (or close to zero)
  - Run Length Encoding (RLE, or RLC)
    - encode as (skip, value) pairs
    - Skip: number of zeros, value: next non-zero component
  - (0,0) as end-of-block value.

DPCM on DC coefficients

- The DC coefficients are coded separately from the AC ones. *Differential Pulse Code modulation* (*DPCM*) is the coding method.

- If the DC coefficients for the first 5 image blocks are 150, 155, 149, 152, 144, then the DPCM would produce 150, 5, -6, 3, -8, assuming $d_i = DC_{i+1} - DC_i$, and $d_0 = DC_0$.

Entropy Coding for DC coefficients

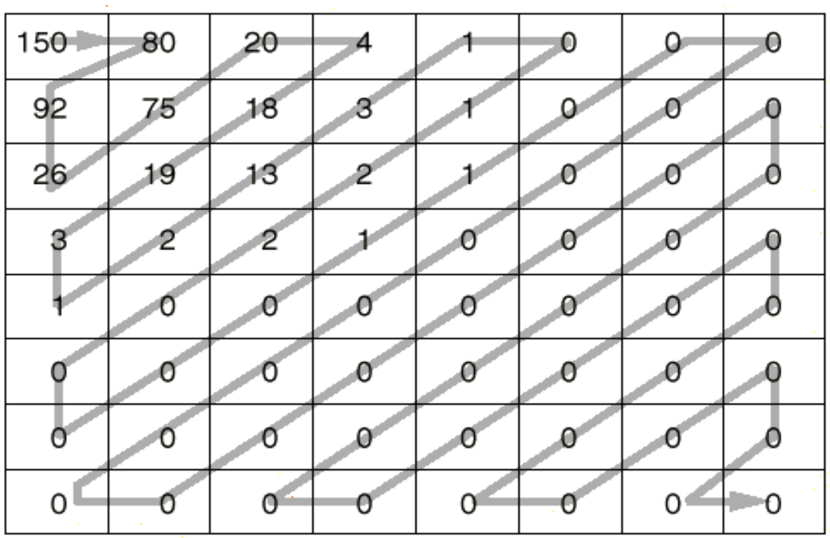
- Use DC as an example: each DPCM coded DC coefficient is represented by (SIZE, AMPLITUDE), where SIZE indicates how many bits are needed for representing the coefficient, and AMPLITUDE contains the actual bits.
- In the example we're using, codes 150, 5, −6, 3, −8 will be turned into
  - (8, 10010110), (3, 101), (3, 001), (2, 11), (4, 0111) .
- SIZE is Huffman coded since smaller SIZEs occur much more often. AMPLITUDE is not Huffman coded, its value can change widely so Huffman coding has no appreciable benefit.

Why ZigZag Scan

- RLC aims to turn the block values into sets

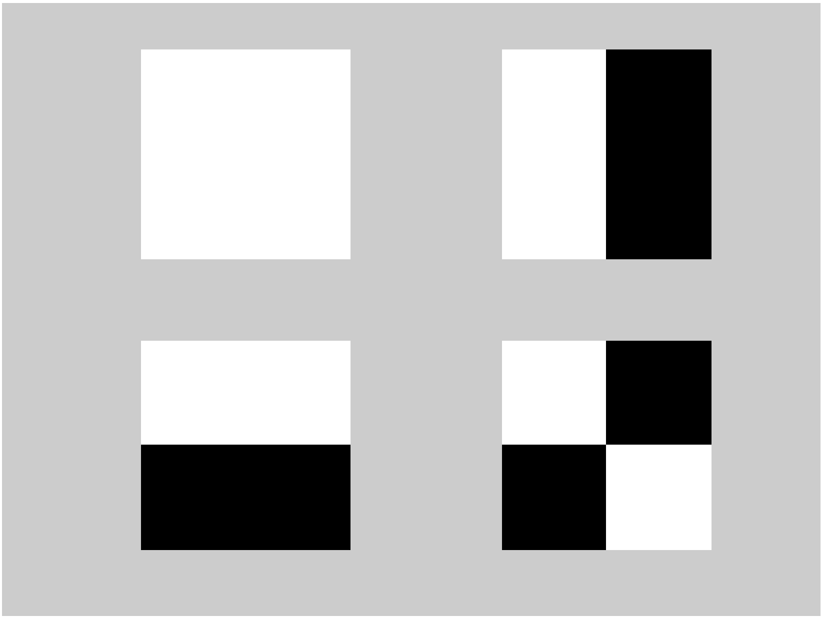
  <#-zeros-to-skip , next non-zero value>.

- ZigZag scan is more effective

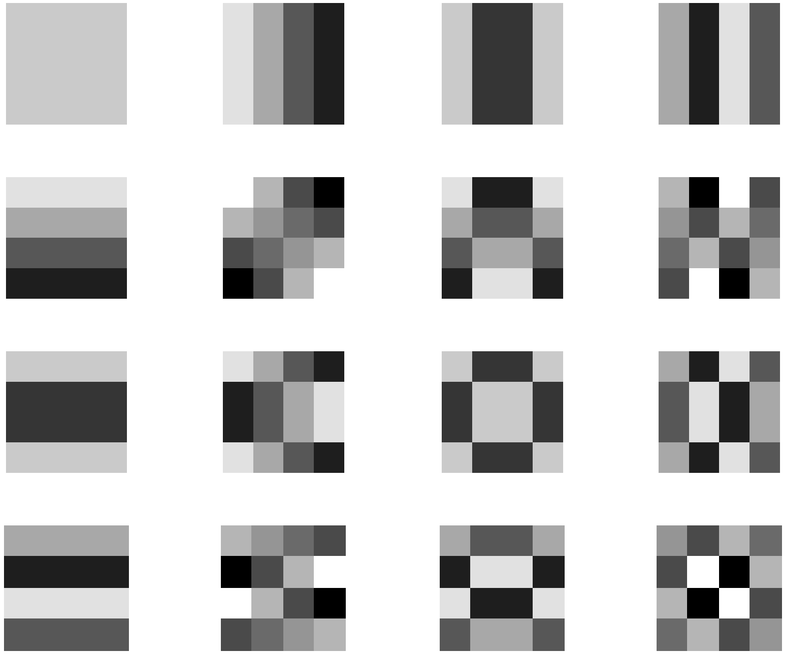


Recall: 2-D DCT Basis Matrices

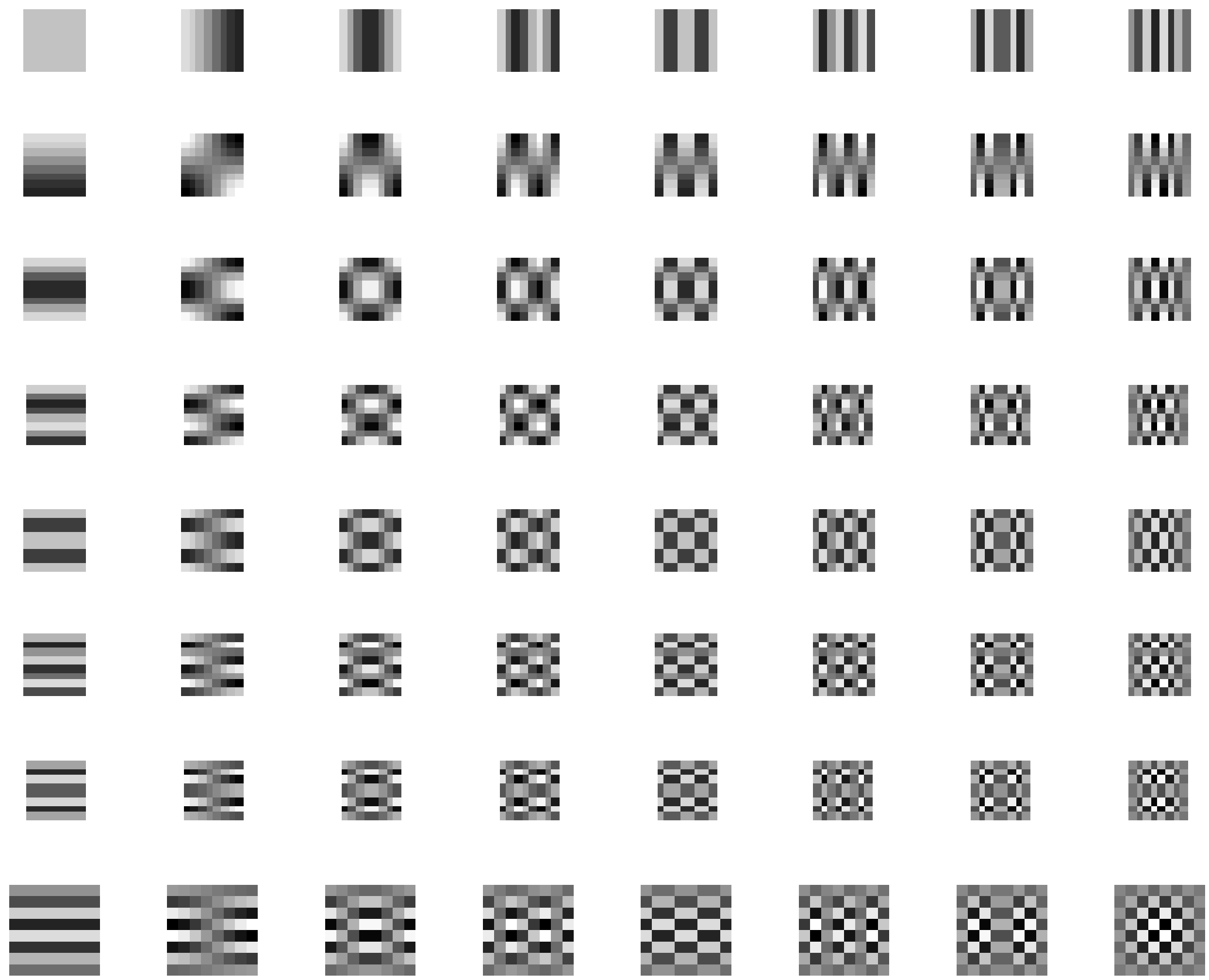
For 2-point DCT



For 4-point DCT



Recall: 2-D DCT Basis Matrices: 8-point DCT



Runlength Encoding (RLE)

A typical 8x8 block of quantized DCT coefficients.
Most of the higher order coefficients have been quantized to 0.

12	34	0	54	0	0	0	0
87	0	0	12	3	0	0	0
16	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
0	0	0	0	0	0	0	0

Zig-zag scan: the sequence of DCT coefficients to be transmitted:
12 34 87 16 0 0 54 0 0 0 0 0 0 12 0 0 3 0 0 0 .....
DC coefficient (12) is sent via a separate Huffman table.
Runlength coding remaining coefficients:
34 | 87 | 16 | 0 0 54 | 0 0 0 0 0 0 12 | 0 0 3 | 0 0 0 .....
(0,34),(0,87),(0,16),(2,54),(6,12),(2,3)...

- Further compression: statistical (entropy) coding

Entropy Coding

- Huffman/arithmetic coding
- Lossless

JPEG Modes

- Sequential Mode
  - default JPEG mode, implicitly assumed in the discussions so far. Each graylevel image or color image component is encoded in a single left-to-right, top-to-bottom scan.
- Progressive Mode.
- Hierarchical Mode.
- Lossless Mode

Progressive Mode

- Progressive
  - Delivers low quality versions of the image quickly, followed by higher quality passes.

- **Method 1. Spectral selection**

  - higher AC components provide detail texture 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.

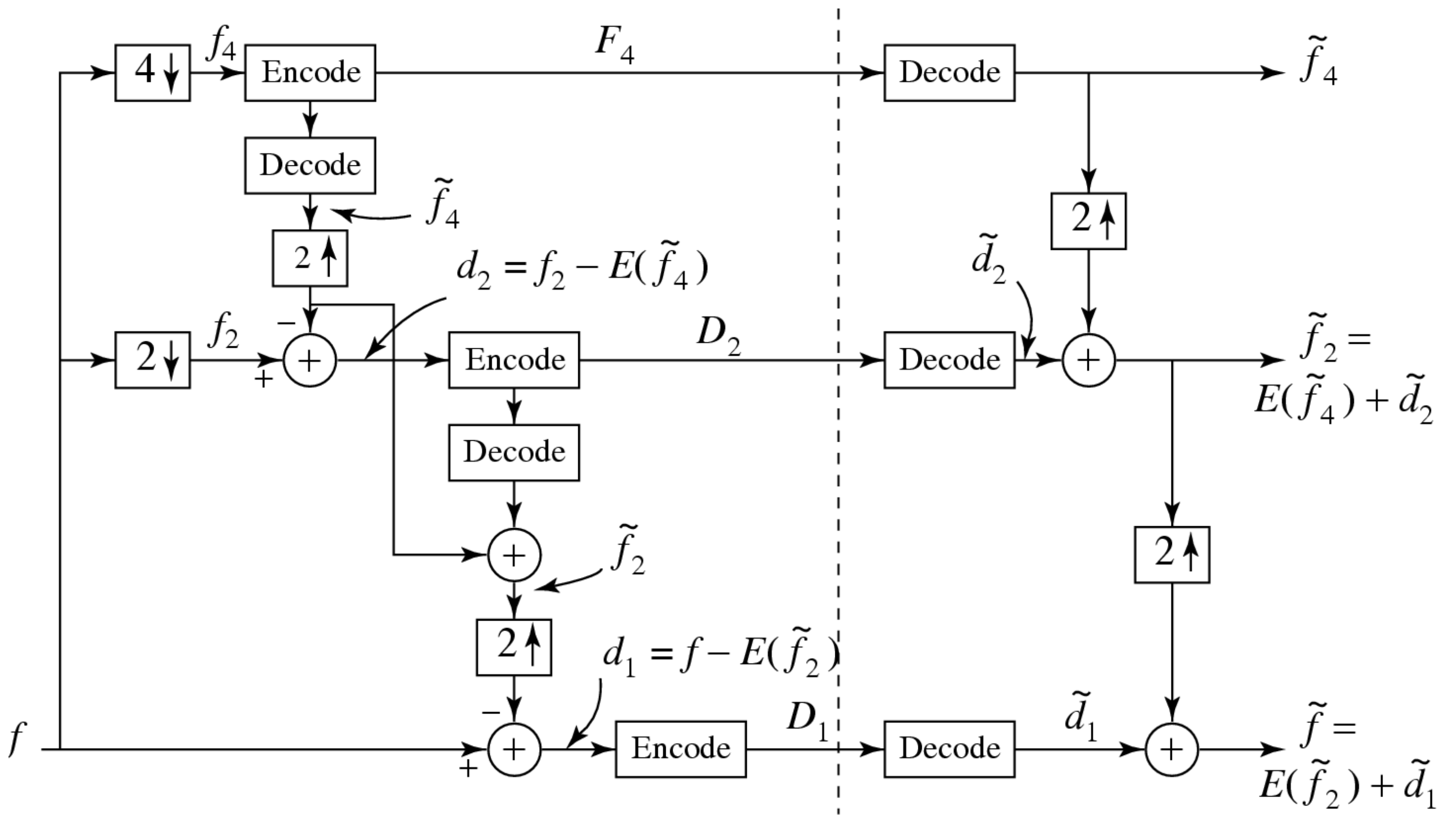
Progressive Mode cont'd

- **Method 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.

Hierarchical Mode

- Encoding
  - First, lowest resolution picture (using low-pass filter)
  - Then, successively higher resolutions
    - additional details (encoding differences)
- Transmission:
  - transmitted in multiple passes
  - progressively improving quality
  - Similar to Progressive JPEG



- **Fig. 9.5: Block diagram for Hierarchical JPEG.**

Algorithm 9.1: Three-Level Hierarchical JPEG Encoder

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 $\tilde{f}_4$. Use any interpolation method to expand $\tilde{f}_4$ to be of the same resolution as f_2 and call it $E(\tilde{f}_4)$.

   (b) Encode difference $d_2 = f_2 - E(\tilde{f}_4)$ 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 $\tilde{d}_2$; add it to $E(\tilde{f}_4)$ to get $\tilde{f}_2 = E(\tilde{f}_4) + \tilde{d}_2$, which is a version of f_2 after compression and decompression.

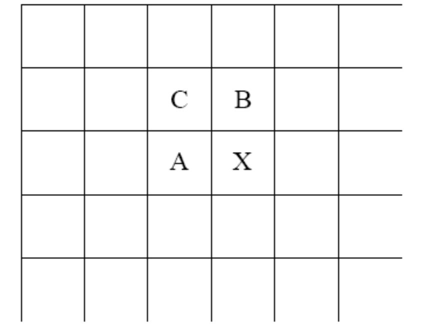
   (b) Encode difference $d_1 = f - E(\tilde{f}_2)$ using any other JPEG method (e.g., Sequential, Progressive) to generate D_1.

Algorithm 9.2: Three-Level Hierarchical JPEG Decoder

1. **Decompress the encoded low-resolution image $\boldsymbol{F_4}$.** Decode F_4 using the same JPEG method as in the encoder, to obtain $\tilde{f}_4$.

2. **Restore image $\boldsymbol{\tilde{f}_2}$ at the intermediate resolution.** Use $E(\tilde{f}_4) + \tilde{d}_2$ to obtain $\tilde{f}_2$.

3. **Restore image $\boldsymbol{\tilde{f}}$ at the original resolution.** Use $E(\tilde{f}_2) + \tilde{d}_1$ to obtain $\tilde{f}$.



Lossless Mode

- **Using prediction and entropy coding**
- **Forming a differential prediction:**
  - A predictor combines the values of up to three neighboring pixels as the predicted value for the current pixel
  - Seven schemes for combination
- **Encoding:**
  - The encoder compares the prediction with the actual pixel value at the position `X' and encodes the difference using entropy coding

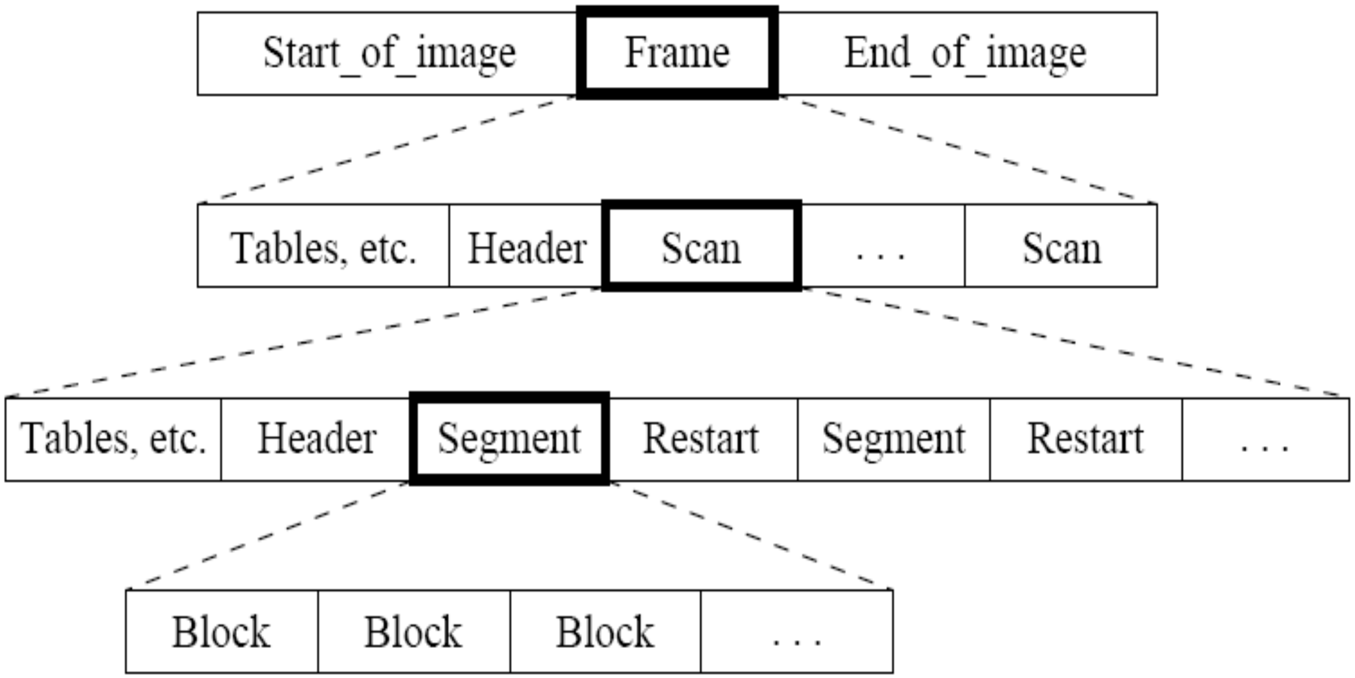
7 Predictors

Predictor	Prediction
P1	A
P2	B
P3	C
P4	A + B - C
P5	A + (B - C) / 2
P6	B + (A - C) / 2
P7	(A + B) / 2

Comparison with Other Lossless

Compression Program	Compression Ratio			
	Lena	football	F-18	flowers
Lossless JPEG	1.45	1.54	2.29	1.26
Optimal lossless JPEG	1.49	1.67	2.71	1.33
compress (LZW)	0.86	1.24	2.21	0.87
gzip (LZ77)	1.08	1.36	3.10	1.05
gzip -9 (optimal LZ77)	1.08	1.36	3.13	1.05
pack (Huffman coding)	1.02	1.12	1.19	1.00

JPEG Bitstream



JPEG 2000

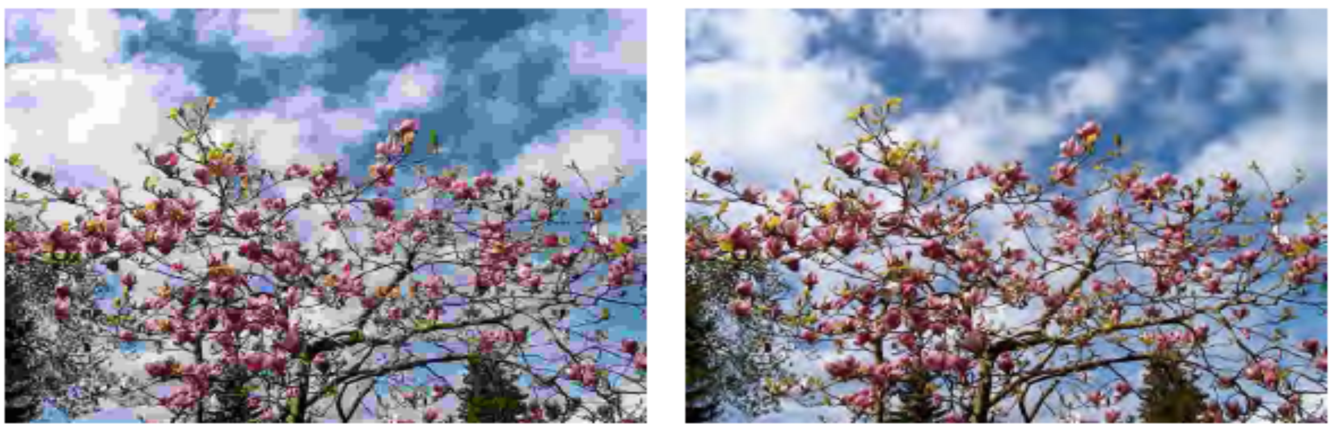
- JPEG 1992
- JPEG 2000
  - **.jp2** for ISO/IEC 15444-1
  - **.jpx** for extended part-2 specifications (ISO/IEC 15444-2)
  - Wavelet transform based
  - 20% gain in compression
- Design Goals:
  - To provide a better rate-distortion tradeoff and improved subjective image quality.
  - To provide a better rate-distortion tradeoff and improved subjective image quality.

JPEG 2000 vs JPEG

- Original image

JPEG2000 vs JPEG

(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.

Further Exploration

- **Textbook Chapter 9**
- **Other sources**
  - *The JPEG Still Image Compression Standard* by Pennebaker and Mitchell
  - *JPEG2000: Image Compression Fundamentals, Standards, and Practice* by Taubman and Marcellin
  - *Image and Video Compression Standards: Algorithms and Architectures, 2nd ed.* by Bhaskaren and Konstantinides