Image Compression — JPEG

Dr. Xiqun Lu

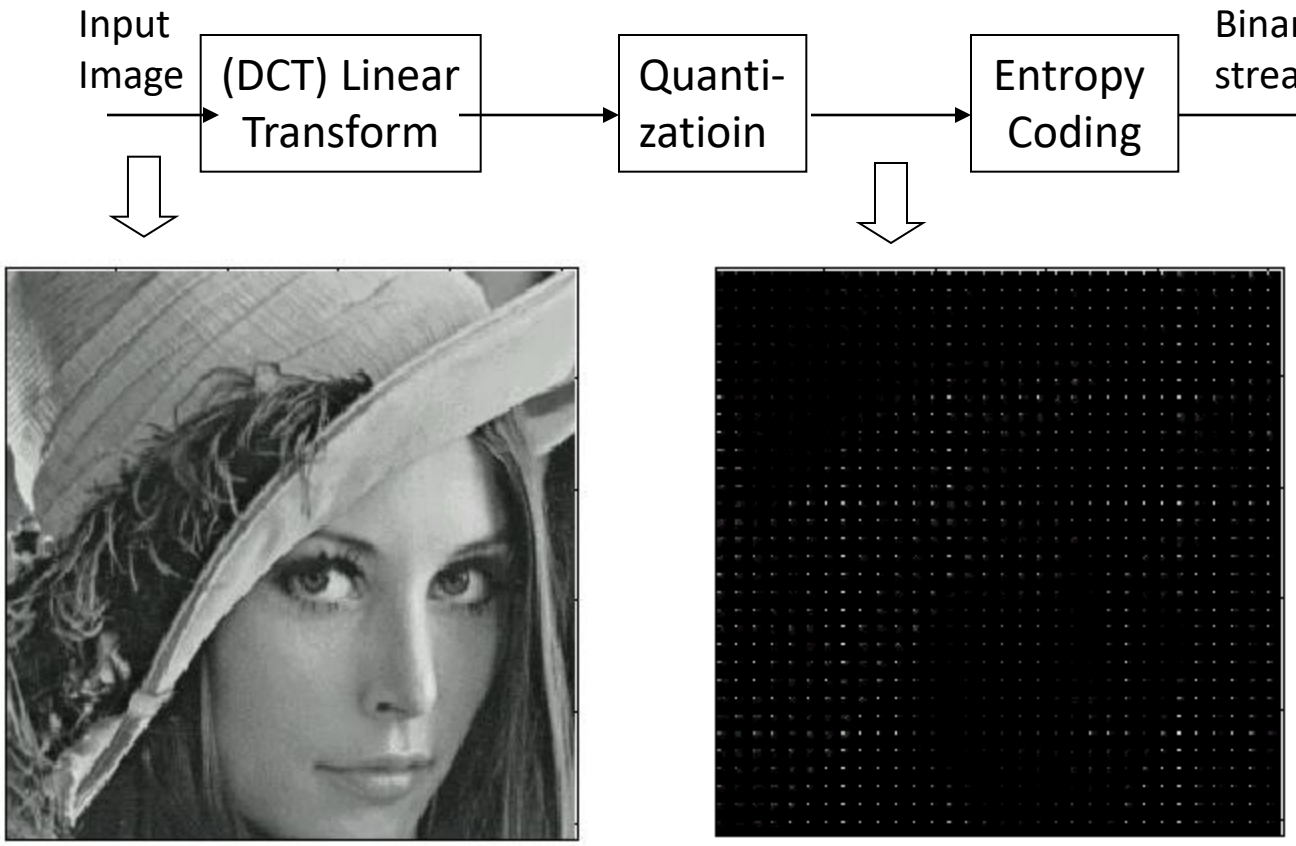
College of Computer Science

Zhejiang University

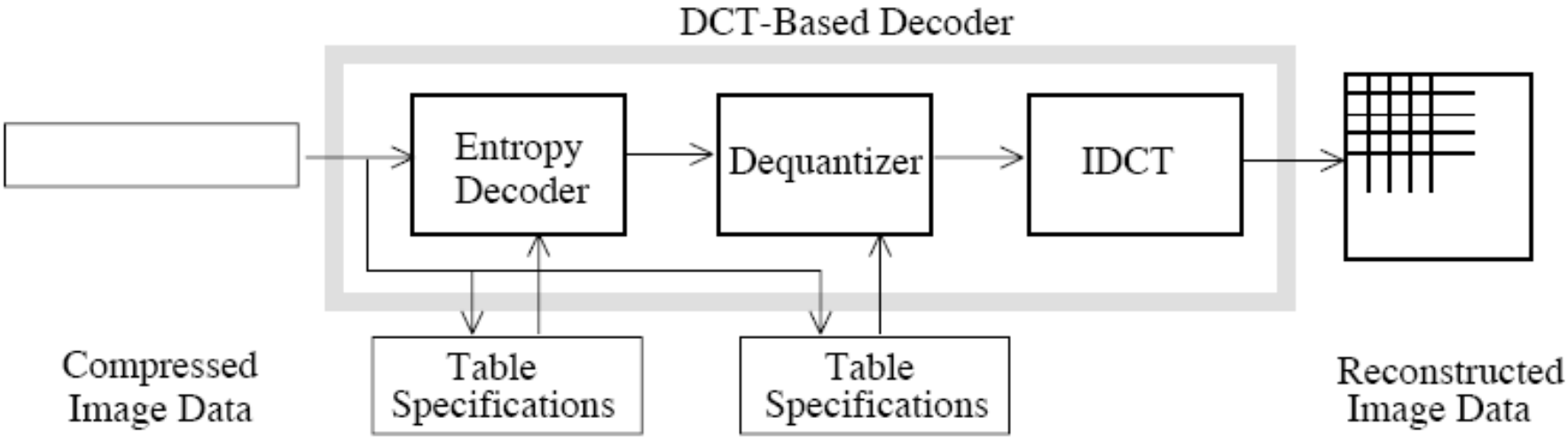
Overview of JPEG [1]

- JPEG is a standardized compression method for full-color and gray-scale images.
- JPEG is intended for compressing "**real-world**" scenes, but **NOT** for
  - line drawings
  - Cartoons
  - other non-realistic images
- JPEG can be **lossy or lossless**
  - Lossy: DCT-based method — the output image is not exactly identical to the input image.
  - Lossless: Predictive-based method

Encoder of Sequential Mode

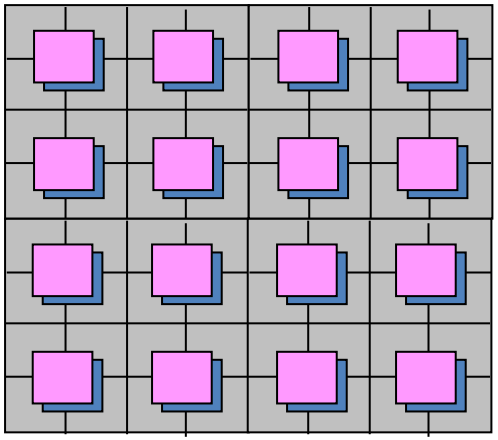


Decoder of Sequential Mode



Pre-Processing

- Color sub-sampling
  - A color image is converted from RGB to YUV color space.
  - Sub-sample U-V planes: 4:1:1 scheme.
  - For every 16 by 16 block of a color image, six 8 by 8 blocks are encoded.
- Level shifting: Each pixel value is subtracted by 128, so it ranges (–128, 127).



Four 8×8 blocks of luminance pixels, plus two 8×8 sub-sampled chrominance components makes a 16 by 16 **macro-block**

2D DCT

- 8×8 DCT

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

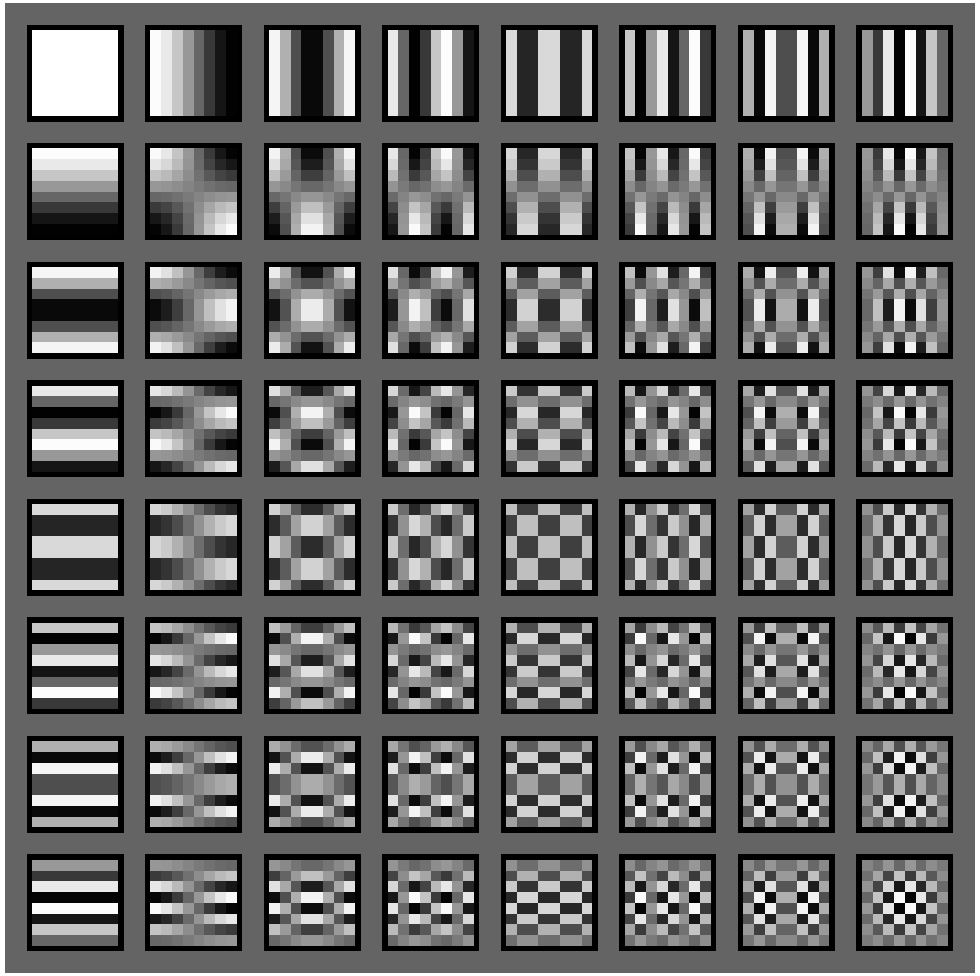
$$f(x,y) = \frac{1}{4}[\sum_{x=0}^{7}\sum_{y=0}^{7} C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16}]$$

Where $C(u), C(v) = 1/\sqrt{2}$ *for* $u, v = 0$

$C(u), C(v) = 1$ *otherwise*

The FDCT takes each 8×8 as its input and decomposes it into 64 orthogonal basis signals. Each contains one of the 64 unique 2D "spatial frequencies" which comprise the input signal's "**spectrum**".

Basic Images of 8 × 8 DCT

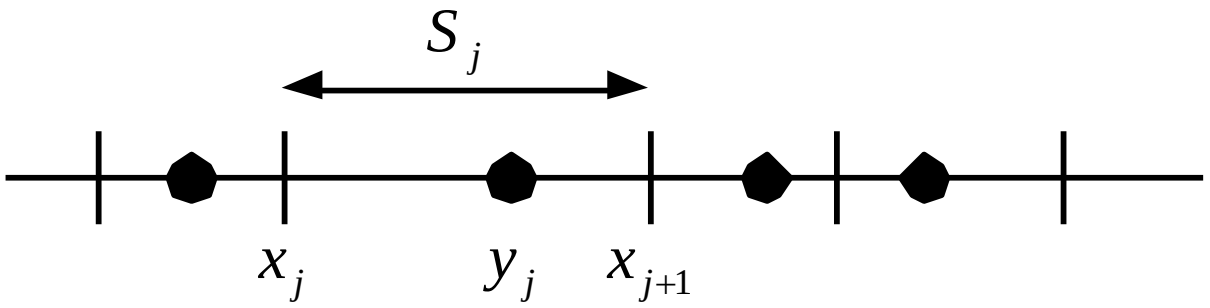


The Idea Behind the DCT

- Because sample values typically **vary slowly** from point to point across an image, the FDCT processing step lays the foundation for achieving data compression by **concentrating most of the signal energy in the lower spatial frequencies**.
- In principle, the DCT introduces **no loss** to the source image samples if the FDCT and IDCT could be computed with perfect accuracy.
  - However, the equations to compute the coefficient of DCT contain “cosine” functions, so no physical implementation can compute them with perfect accuracy.

Quantization

- **Lossy:** to discard information not visually significant!
- Many-to-one mapping



Quantization will introduce quantization **error (or noise)**

DPCM of DC Coefficients

- DC coding: All DC coefficients of each 8 by 8 blocks of the entire image are combined to make a sequence of DC coefficients.
- Next, DPCM is applied:

  $DiffDC(block_i) = DC(block_i) - DC(block_{i-1})$

- Then DiffDCs will be encoded using Huffman entropy

1216	1232	1224
1248	1248	1208

Example:

- Original:

  $1216 \rightarrow 1232 \rightarrow 1224 \rightarrow 1248 \rightarrow 1248 \rightarrow 1208$

- After DPCM:

  $1216 \rightarrow +16 \rightarrow -8 \rightarrow +24 \rightarrow 0 \rightarrow -40$

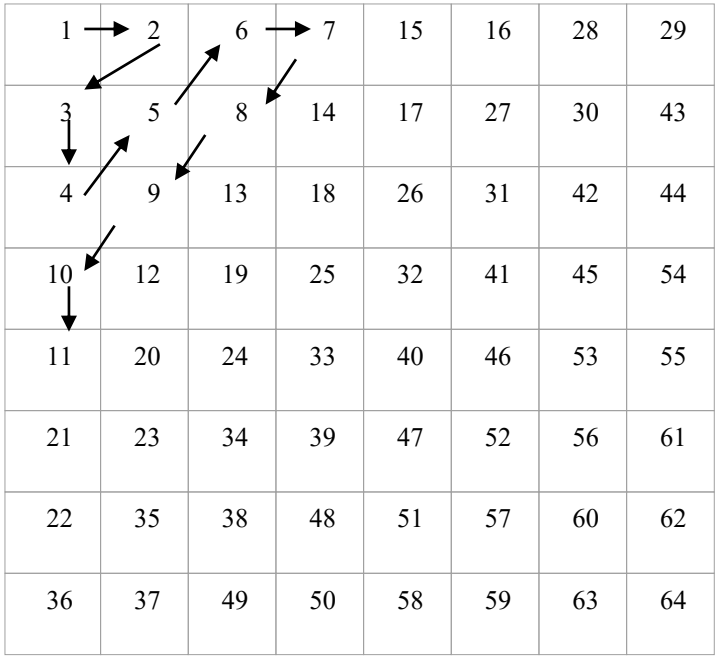
AC Coefficients

- AC coefficients are first weighted with a *quantization matrix:*

  $C(i,j)/q(i,j) = C_q(i,j)$

  Then quantized.
- Then they are scanned in a zig-zag order into a 1D sequence to be subject to AC Huffman encoding.
- This ordering helps to facilitate entropy coding by placing low-frequency coefficients before high-frequency coefficients.
- Question: Given a 8 by 8 array, how to convert it into a vector according to the zig-zag scan order? What is the algorithm?

Zig-Zag scan order



Baseline Encoding Example

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

(a) source image samples

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

(b) forward DCT coefficients

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

(c) quantization table

(1, 2) (–2)
(0, 1) (–1)
(0, 1) (–1)
(0, 1) (–1)
(2, 1) (–1)
(0, 0) % EOB

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-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	0	0	0	0	0	0	0

(d) normalized quantized coefficients

240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	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

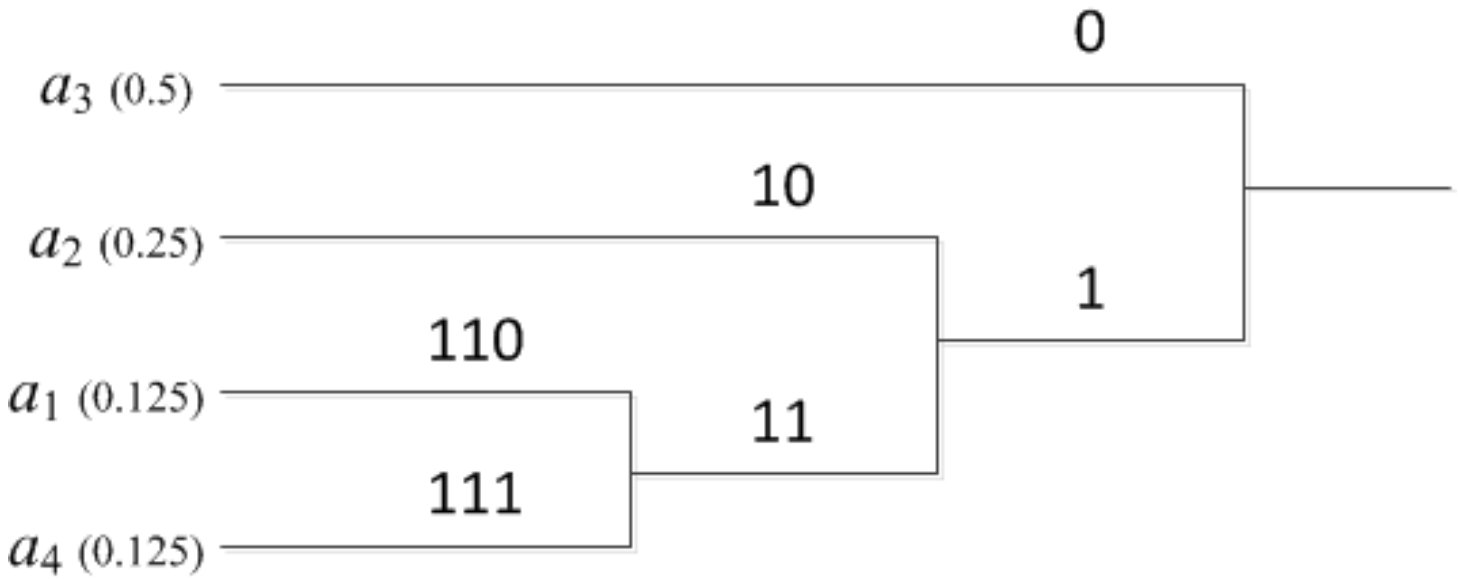
(e) denormalized quantized coefficients

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	164	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

(f) reconstructed image samples

Huffman Coding

- Suppose we have a discrete letter set $\{a_1, a_2, a_3, a_4\}$, and the occurrence probability of each letter is $P_1 = 0.125$, $P_2 = 0.25$, $P_3 = 0.5$, $P_4 = 0.125$.



No code is the prefix of other codes.

- Entropy $-\sum_i P_i \log_2(P_i)$

References

- [1] G. K. Wallace "The JPEG still picture compression standard", Communications of the ACM, 34(4):30-44, April 1991.

Thank You!

Dr. Xiqun Lu

xqlu@zju.edu.cn