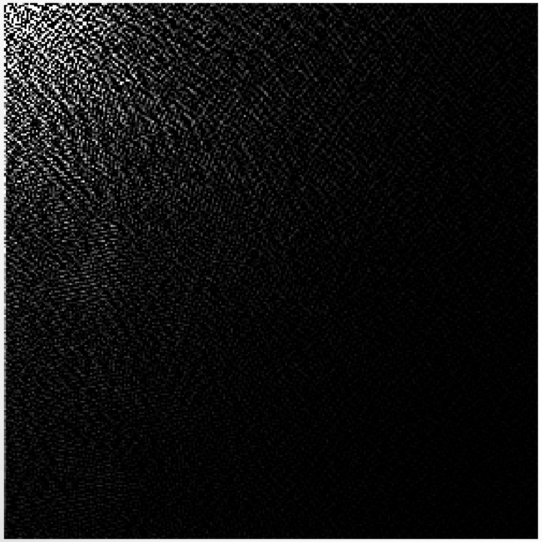
Image Compression -- JPEG

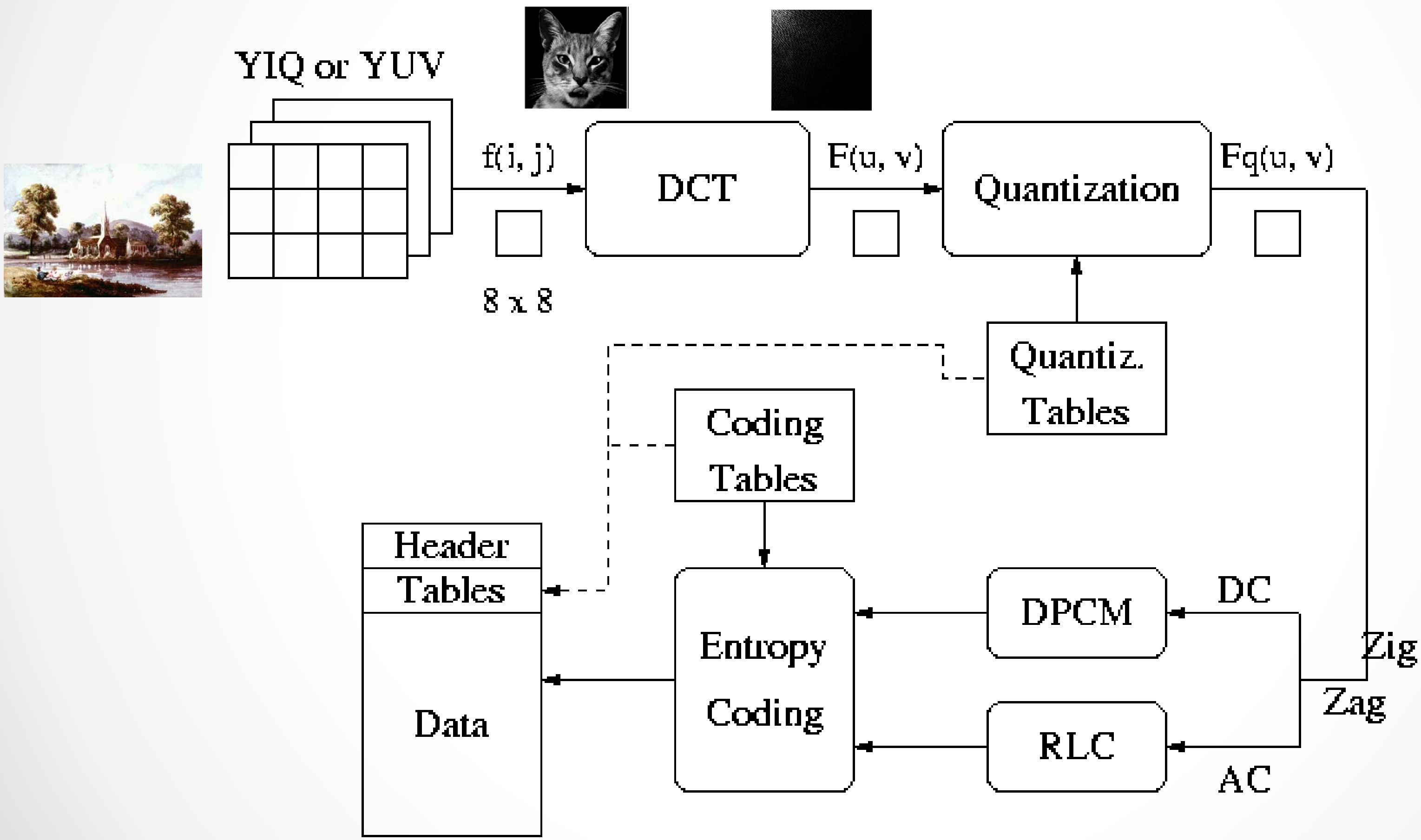
- **What is JPEG?**
- "Joint Photographic Expert Group". Voted as international standard in 1992.
- Works with color and grayscale images, e.g., satellite, medical, ...
- **Motivation**
- The *compression ratio* of lossless methods (e.g., Huffman, Arithmetic, LZW) is not high enough for image and video compression, especially when the distribution of pixel values is relatively flat.
- JPEG uses *transform coding*, it is largely based on the following observations:

Image Compression -- JPEG



- Observation 1: A large majority of useful image contents change relatively slowly across images, i.e., it is unusual for intensity values to alter up and down several times in a small area, for example, within an 8 x 8 image block. Translate this into the spatial frequency domain, it says that, generally, lower spatial frequency components contain more information than the high frequency components which often correspond to less useful details and noises.
- Observation 2: Psychophysical experiments suggest that humans are less likely to notice the loss of higher spatial frequency components than loss of lower frequency components.

Encoder & Decoder



Major Steps

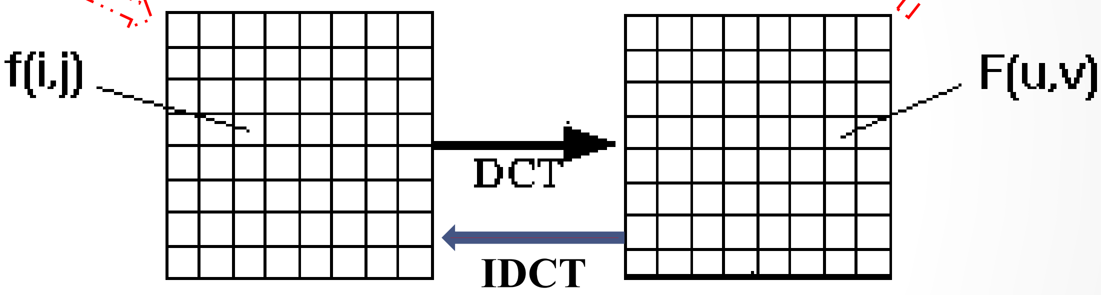
- Discrete Cosine Transformation (DCT)
- Quantization
- Zigzag Scanning
- DPCM on DC component
- RLE on AC Components
- Entropy Coding

Discrete Cosine Transform (from previous slides)

76	68	65	65	65	65	68	73
76	68	66	66	64	65	68	73
75	67	66	66	64	65	68	72
74	68	65	65	64	65	68	73
73	67	65	65	64	65	68	73
73	67	65	65	64	65	67	73
73	70	66	63	63	66	70	73
73	70	66	63	63	66	70	73

-483.1250	1.7102	25.5989	-0.2148	11.3750	3.1852	3.3324	-0.4426
3.5185	2.2448	1.1681	1.8343	0.1998	0.6538	-0.3247	0.2546
-0.2590	0.4080	0.3384	-0.1283	0.8562	0.1920	0.2866	0.3167
0.2695	-0.3552	0.2529	0.6294	0.3285	-1.0440	-0.1421	0.1350
-0.3750	-0.7855	0.4339	0.1022	-0.3750	-0.6576	-0.5856	-0.0507
-0.1464	0.0721	0.0876	0.4864	0.3698	-0.3669	-0.2240	0.3948
-0.2986	0.0187	-0.4634	0.2122	-0.2194	0.0268	0.1616	-0.0938
-0.2979	-0.2150	-0.5027	0.0962	0.1672	0.6272	0.1019	0.4927

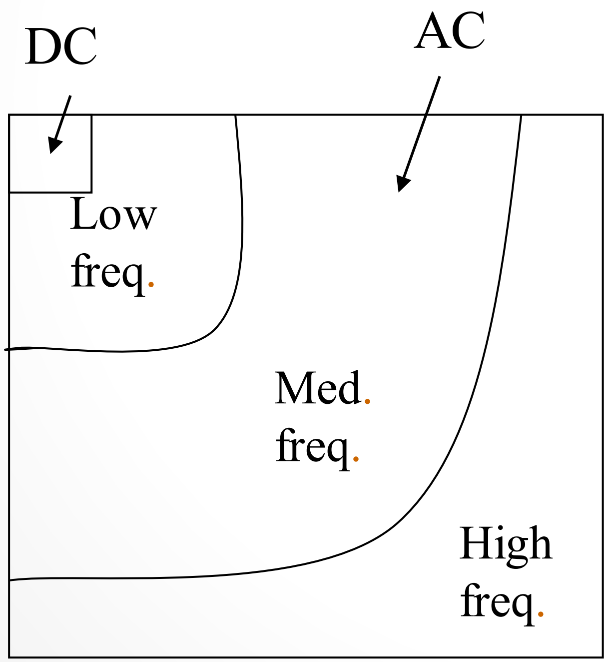
- From spatial domain to frequency domain:



- A reversible, linear transform maps the image f(i,j) into transform coefficients F(u,v), then quantized & coded
- For most natural images, a significant number of coefficients have small magnitudes and can be coarsely quantized or discarded with little distortion ---> compression

Quantization

- Human vision -- low frequencies are more important than high frequencies. Hence higher freqs. can be more coarsely quantized or discarded. The bits saved for coding high frequencies are used for lower frequencies to obtain better subjective coded images.



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

Quantization

-483.1250	1.7102	25.5989	-0.2148	11.3750	3.1852	3.3324	-0.4426
3.5185	2.2448	1.1681	1.8343	0.1998	0.6538	-0.3247	0.2546
-0.2590	0.4080	0.3384	-0.1283	0.8562	0.1920	0.2866	0.3167
0.2695	-0.3552	0.2529	0.6294	0.3285	-1.0440	-0.1421	0.1350
-0.3750	-0.7855	0.4339	0.1022	-0.3750	-0.6576	-0.5856	-0.0507
-0.1464	0.0721	0.0876	0.4864	0.3698	-0.3669	-0.2240	0.3948
-0.2986	0.0187	-0.4634	0.2122	-0.2194	0.0268	0.1616	-0.0938
-0.2979	-0.2150	-0.5027	0.0962	0.1672	0.6272	0.1019	0.4927

- **Quantization**

Fq(u,v) = round [F(u,v)/(QF*Z(u,v))]

-30	0	3	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	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Z matrix for Luminance Quantization →

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

- The numbers in the above quantization tables can be scaled up (or down) to adjust the quality factor (QF). Default of QF is 1.
- Custom quantization tables can also be put in image/scan header.

Transform Coefficients & Quantization

QF 2

constant QP 37

Quantization

JPEG compressed @ 0.58193 bpp

JPEG compressed @ 0.36827 bpp

JPEG compressed @ 0.24513 bpp

JPEG compressed @ 0.19365 bpp

Quantization and inverse quantization

-30	0	3	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	0	0	0	0	0	0
0	0	0	0	0	0	0	0

X

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

-480	0	30	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	0	0	0	0	0	0
0	0	0	0	0	0	0	0

-483.1250	1.7102	25.5989	-0.2148	11.3750	3.1852	3.3324	-0.4426
3.5185	2.2448	1.1681	1.8343	0.1998	0.6538	-0.3247	0.2546
-0.2590	0.4080	0.3384	-0.1283	0.8562	0.1920	0.2866	0.3167
0.2695	-0.3552	0.2529	0.6294	0.3285	-1.0440	-0.1421	0.1350
-0.3750	-0.7855	0.4339	0.1022	-0.3750	-0.6576	-0.5856	-0.0507
-0.1464	0.0721	0.0876	0.4864	0.3698	-0.3669	-0.2240	0.3948
-0.2986	0.0187	-0.4634	0.2122	-0.2194	0.0268	0.1616	-0.0938
-0.2979	-0.2150	-0.5027	0.0962	0.1672	0.6272	0.1019	0.4927

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

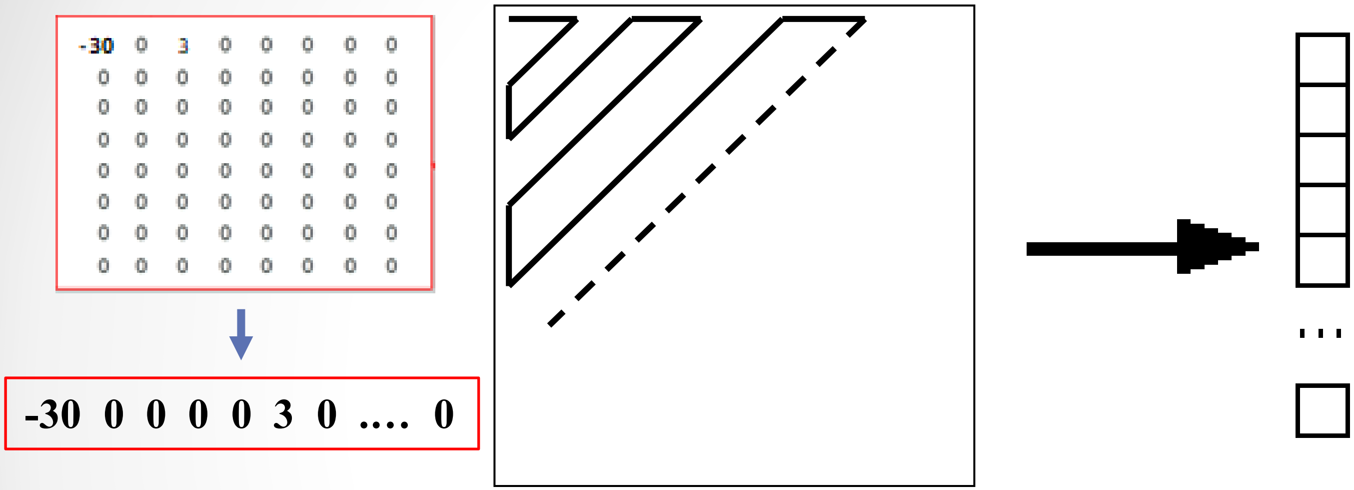
Quantization

- Eye is most sensitive to low frequencies (upper left corner), less sensitive to high frequencies (lower right corner).

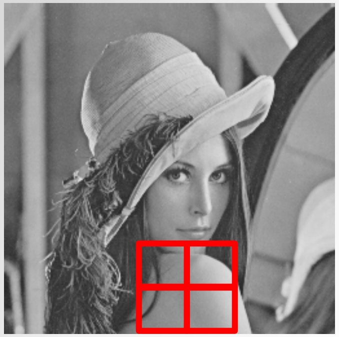
The *Chrominance Quantization Table* q(u, v)

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

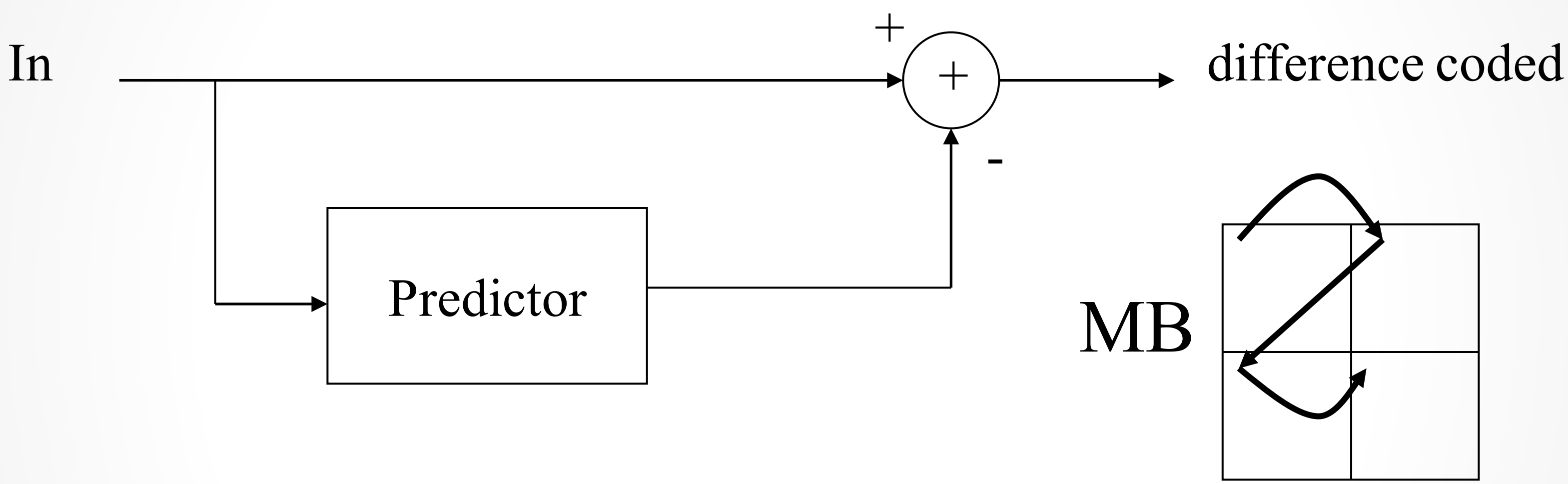
Zig-Zag Scaning



- Why? -- to group low frequency coefficients in top of vector. Increase the likelihood of grouping all nonzero coefficients together
- Maps 8 x 8 to a 1 x 64 vector.
- The reordered 1-D sequence contains long runs of 0’s, and can be run-length coded.
- The non-zero coefficients are represented by variable-length codes.



Predictive Coding (DPCM)



- DC coefficients of successive blocks often vary only slightly --> Use DPCM to code DC coefficients
- For each DC, the predictor is the DC of previous block. Hence produce small difference

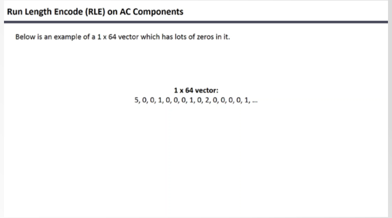
Run Length Encode (RLE) on AC Components

- 1 x 64 vector has lots of zeros in it
- Keeps *skip* and *value*, where *skip* is the number of zeros and *value* is the next non-zero component.
- Send (0,0) as end-of-block sentinel value.

Entropy Coding

- Categorize DC values into SIZE (number of bits needed to represent) and actual bits.
- Example: if DC value is 4, 3 bits are needed.
- Send off SIZE as Huffman symbol, followed by actual 3 bits.

DC Coefficient	Size	Huffman codes for Size
0	0	00
-1,1	1	010
-3,-2,2,3	2	011
-7,…,-4,4,…,7	3	100
-15,…,-8,8,…,15	4	101
-31,…,-16,16,…,31	5	110
⋮	⋮	⋮
-1023,…-512,512,…,1023	10	1111 1110
-2047,…-1024,1024,…2047	11	1 1111 1110



Entropy Coding

- For AC components two symbols are used because there is a strong correlation between the Size of a coefficient and the expected Run of zeros : Symbol_1: (*Run, Size*), Symbol_2: actual bits. Symbol_1 (*Run, Size*) is encoded using the Huffman coding, Symbol_2 is not encoded.
- Small coefficients usually follow long runs; larger coefficients tend to follow shorter runs. Huffman Tables can be custom (sent in header) or default.
- ZRL represents a run of 16 zeros which can be part of a longer run of any length.
- EOB is transmitted after the last non-zero coefficient in a 64-vector. It is omitted in case the final element of the vector is non-zero.

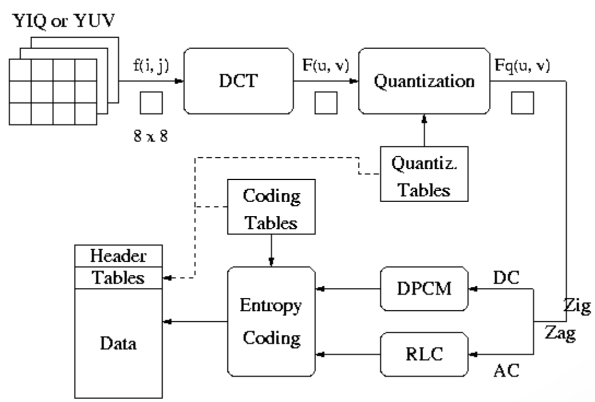
(Run,Size)	Code Word	(Run,Size)	Code Word
(0,1)	00	(0,6)	1111000
(0,2)	01	(1,3)	1111001
(0,3)	100	(5,1)	1111010
(EOB)	1010	(6,1)	1111011
(0,4)	1011	(0,7)	11111000
(1,1)	1100	(2,2)	11111001
(0,5)	11010	(7,1)	11111010
(1,2)	11011	(1,4)	111110110
(2,1)	11100		
(3,1)	111010	(ZRL)	11111111001
(4,1)	111011		

Four JPEG Modes

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

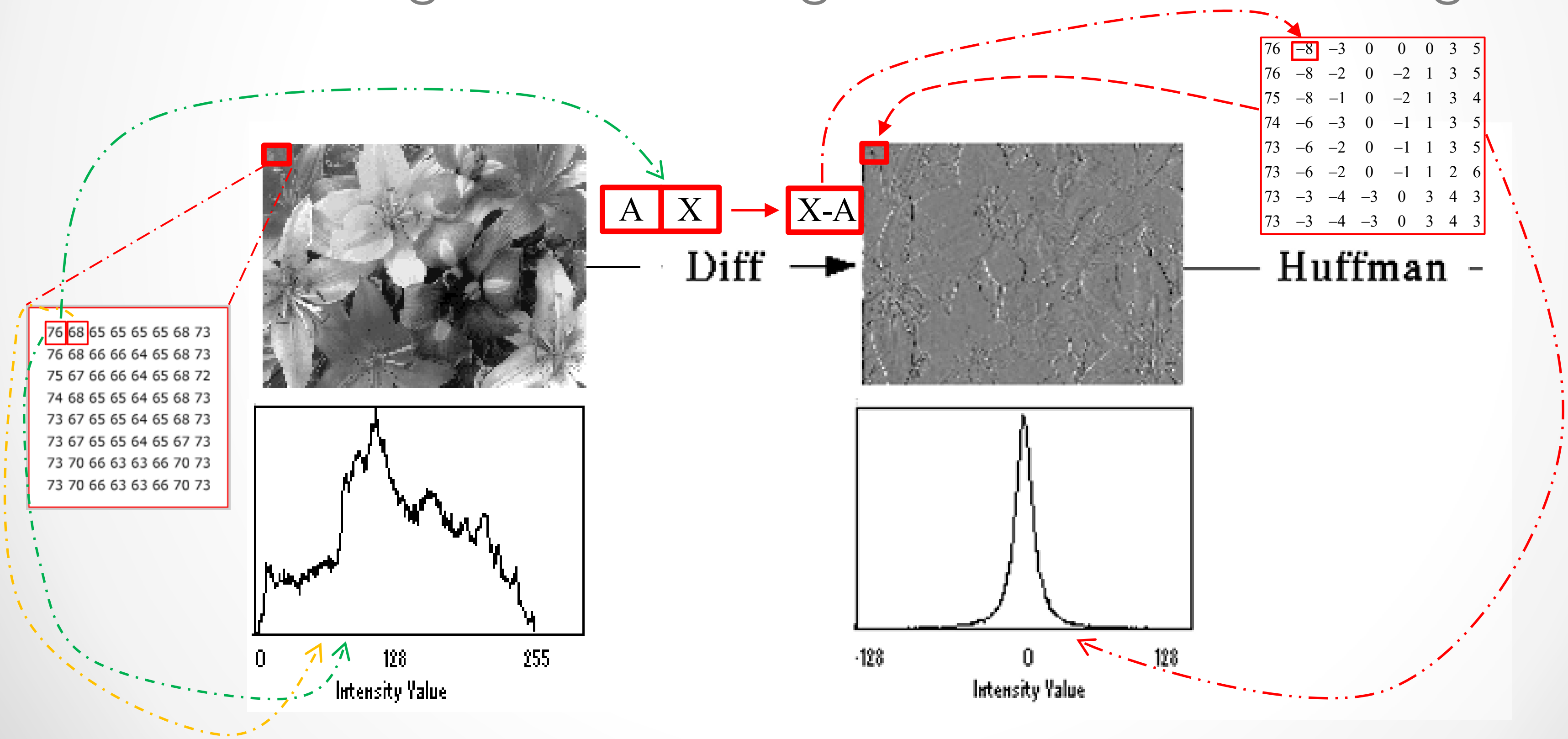
Sequential Mode

- Each image component is encoded in a single left-to-right, top-to-bottom scan. *Baseline Sequential Mode*, the one that we described above, is a simple case of the Sequential mode:
- It supports only 8-bit images (not 12-bit images)
- It uses only Huffman coding (not Arithmetic coding)



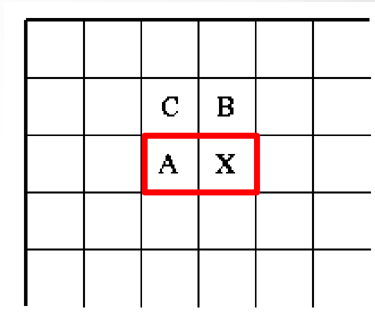
Lossless Mode

- A special case of the JPEG where indeed there is no loss. Its block diagram and histograms are in the followings.



Lossless Mode

- It does not use DCT-based method! Instead, it uses a predictive (differential coding) method.
- A predictor combines the values of up to three neighboring pixels (not blocks as in the Sequential mode) as the predicted value for the current pixel, indicated by "X" in the figure on the right.
- The encoder then compares this prediction with the actual pixel value at the position "X", and encodes the difference (prediction residual) losslessly.
- It can use any one of the seven predictors.
- Since it uses only previously encoded neighbors, the very first pixel I(0, 0) will have to use itself. Other pixels at the first row always use P1, at the first column always use P2.

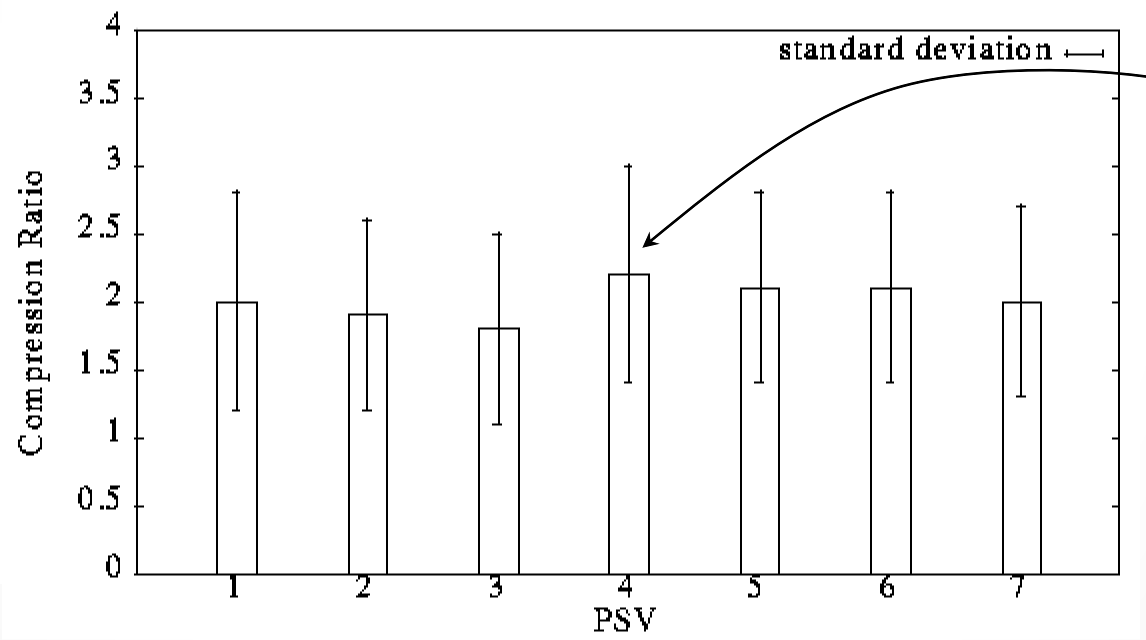


Predictor	Prediction
1	A
2	B
3	C
4	A + B - C
5	A + (B - C) / 2
6	B + (A - C) / 2
7	(A + B) / 2

Lossless Mode

- Effect of Predictor (test result with 20 images):

Predictor	Prediction
1	A
2	B
3	C
4	A + B - C
5	A + (B - C) / 2
6	B + (A - C) / 2
7	(A + B) / 2



Intensity gradually change in horizontal and vertical direction.

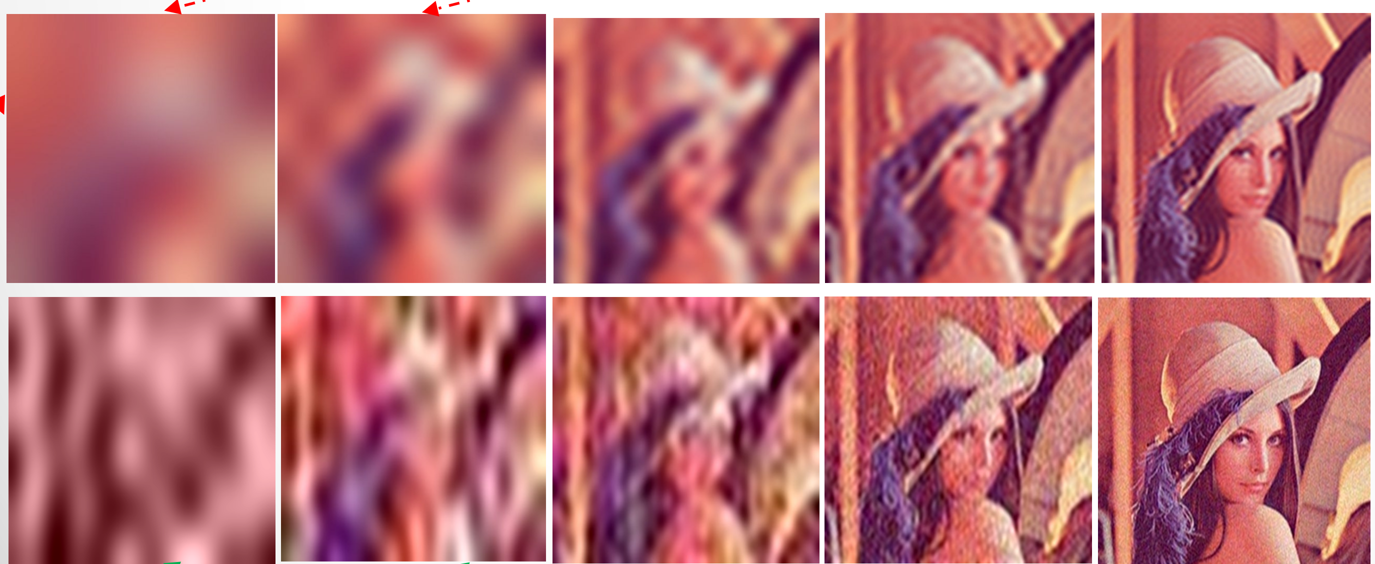
- Predictors (4-7) always do better than predictors (1-3).

Progressive Mode

- Goal: display low quality image and successively improve.
- Two ways to successively improve image:
  1. *Spectral selection*: Send DC component and first few AC coefficients first, then gradually some more ACs.
  2. *Successive approximation*: send DCT coefficients MSB (most significant bit) to LSB (least significant bit).

17	2	0	0	0	0	0	0
4	1	0	0	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	0	0	0	0	0	0	0

17	2	4	0	1	0	0	0	1	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	1	0	0	0	0	0	0	0		0
0	1	0	0	0	0	0	0	0	0		0
1	0	0	0	1	0	0	0	1	0	......	0



Hierarchical Mode

- A Three-level Hierarchical JPEG Encoder
  - Down-sample by factors of 2 in each dimension, e.g., reduce 640 x 480 to 320 x 240
  - Code smaller image using another JPEG mode (Progressive, Sequential, or Lossless).
  - Decode and up-sample encoded image
  - Encode difference between the up-sampled and the original using Progressive, Sequential, or Lossless.
- It can be repeated multiple times.
- Good for viewing high resolution image on low resolution display.

Three Level Hierarchical Mode