GNR602
Advanced Methods in Satellite Image Processing

Prof. B. Krishna Mohan
CSRE, IIT Bombay
bkmohan@csre.iitb.ac.in

Slot 13

Lecture 12-13 Image Compression

Contents of the Lecture

Principles of Image Compression

Lossy v/s Lossless Image Compression

JPEG Image Compression

Rate Controlled JPEG Compression

Acknowledgement: Some of the slides are prepared by former M.Tech. students and some are downloaded from various internet sources

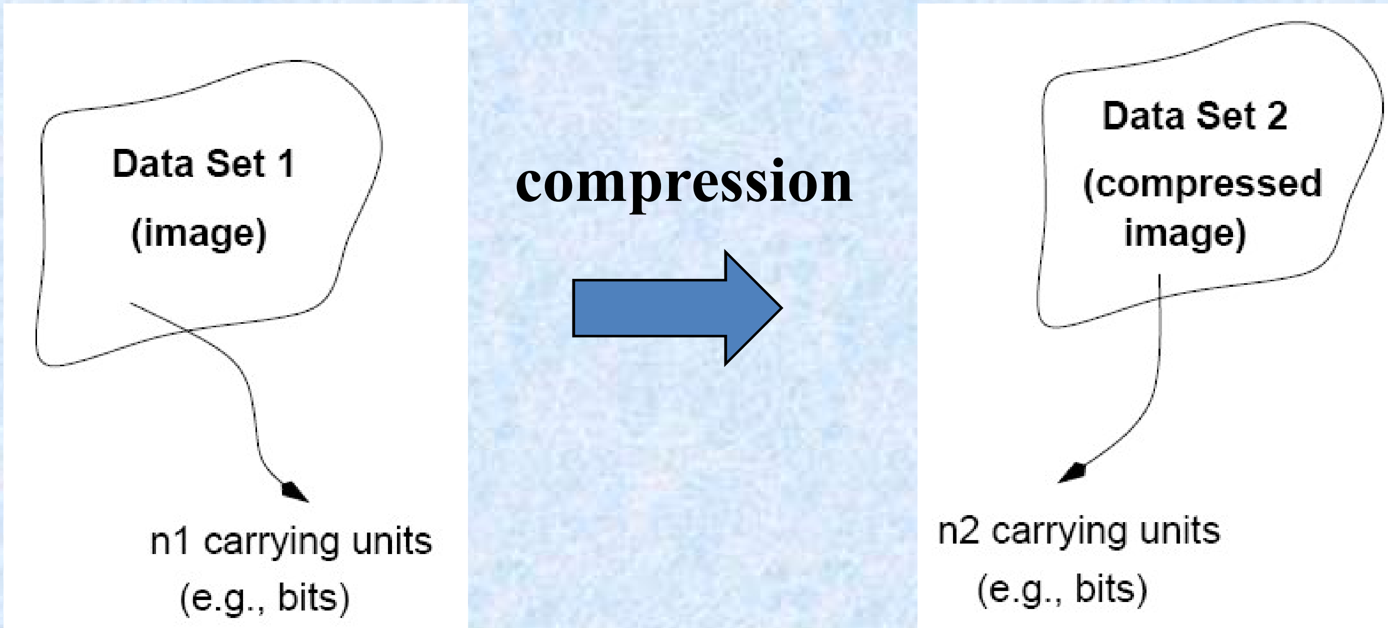
What is Data and Image Compression?

- Data compression is the representation of information in a compact form.
- Data is a sequence of symbols taken from a discrete alphabet.
- Still image data, that is a collection of 2-D arrays (one for each color plane) of values representing intensity (color) of the point in corresponding spatial location (pixel).

Why is an image compressible?

- Statistical redundancy:
  1) Spatial correlation
     a) Local - Pixels at neighboring locations have similar intensities.
     b) Global - Reoccurring patterns.
  2) Spectral correlation – between color planes (color/multispec)
  3) Temporal correlation – between consecutive frames (video).
- Tolerance to fidelity:
  1) Perceptual redundancy.
  2) Limitation of rendering hardware.

Compression Ratio



Compression ratio: $C_R = \frac{n_1}{n_2}$

This ppt file: original size in: 12,780 KB
Compressed file size using ZIP: 12,149 KB
Compressed file size using RAR: 8,864 KB

Relevant Data Redundancy

$$R_D = 1 - \frac{1}{C_R}$$

Example:

If $C_R = \frac{10}{1}$, then $R_D = 1 - \frac{1}{10} = 0.9$

(90% of the data in dataset 1 is redundant)

if $n_2 = n_1$, then C_R=1, R_D=0

if $n_2 \ll n_1$, then $C_R \rightarrow \infty$, $R_D \rightarrow 1$

Types of Data Redundancy

(1) Coding Redundancy

(2) Interpixel Redundancy

(3) Psychovisual Redundancy

- Data compression attempts to reduce one or more of these redundancy types.

Coding - Definitions

- Code: a list of symbols (letters, numbers, bits etc.)
- Code word: a sequence of symbols used to represent some information (e.g., gray levels).
- Code word length: number of symbols in a code word.

Example: (binary code, symbols: 0,1, length: 3)

0: 000	4: 100
1: 001	5: 101
2: 010	6: 110
3: 011	7: 111

Coding - Definitions (cont'd)

N x M image

r_k: k-th gray level

$l(r_k)$: # of bits for r_k

$P(r_k)$: probability of r_k

Average # of bits: $L_{avg} = E(l(r_k)) = \sum_{k=0}^{L-1} l(r_k)P(r_k)$

Total # of bits: NML_{avg}

Expected value: $E(X) = \sum_x xP(X = x)$

Coding Redundancy

- <u>Case 1:</u> $l(r_k)$ = constant length

Example:

r_k	$p_r(r_k)$	*Code 1*	$l_1(r_k)$
$r_0 = 0$	0.19	000	3
$r_1 = 1/7$	0.25	001	3
$r_2 = 2/7$	0.21	010	3
$r_3 = 3/7$	0.16	011	3
$r_4 = 4/7$	0.08	100	3
$r_5 = 5/7$	0.06	101	3
$r_6 = 6/7$	0.03	110	3
$r_7 = 1$	0.02	111	3

Assume an image with $L = 8$

Assume $l(r_k) = 3$, $L_{avg} = \sum_{k=0}^{7} 3P(r_k) = 3 \sum_{k=0}^{7} P(r_k) = 3$ bits

Total number of bits: $3NM$

Coding Redundancy (cont'd)

- Case 2: $l(r_k)$ = variable length

Table 6.1 Variable-Length Coding Example variable length

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

$$L_{avg} = \sum_{k=0}^{7} l(r_k)P(r_k) = 2.7 \text{ bits}$$

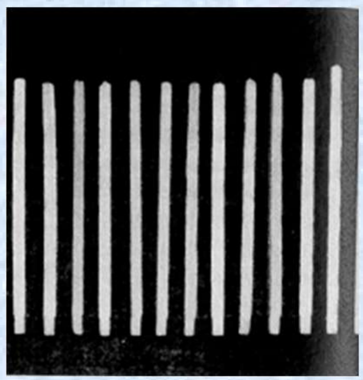
$$C_R = \frac{3}{2.7} = 1.11 \text{ (about 10\%)}$$

$$R_D = 1 - \frac{1}{1.11} = 0.099$$

Total number of bits: 2.7NM

Interpixel redundancy

- Interpixel redundancy implies that pixel values are correlated (i.e., a pixel value can be reasonably predicted by its neighbors).



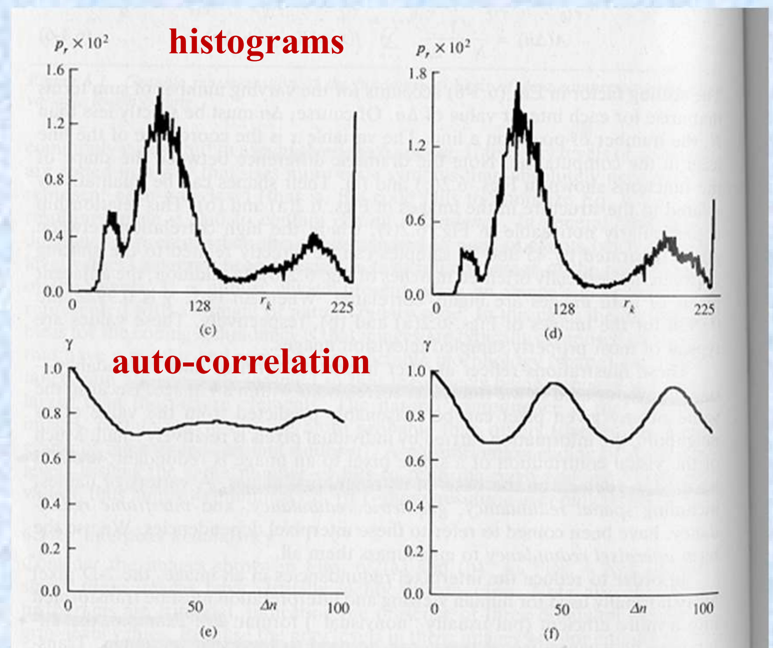


Figure 6.2 Two images and their gray-level histograms and normalized autocorrelation

$$f(x) \circ g(x) = \int_{-\infty}^{\infty} f(x) g(x+a)\, da$$

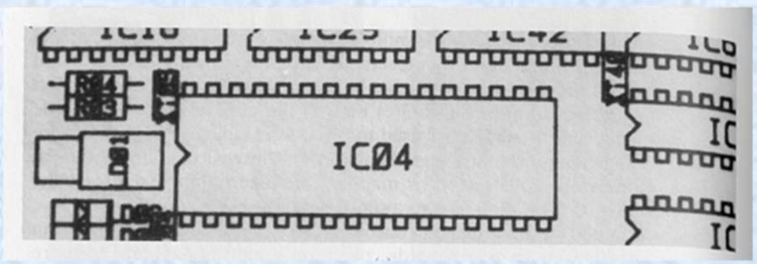
auto-correlation: f(x)=g(x)

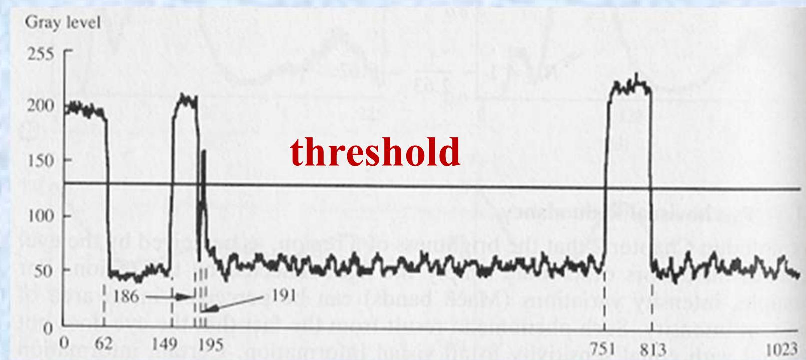
Interpixel redundancy (cont’d)

- To reduce interpixel redundancy, some kind of transformation must be applied on the data (e.g., thresholding, DFT, DWT)

Example:

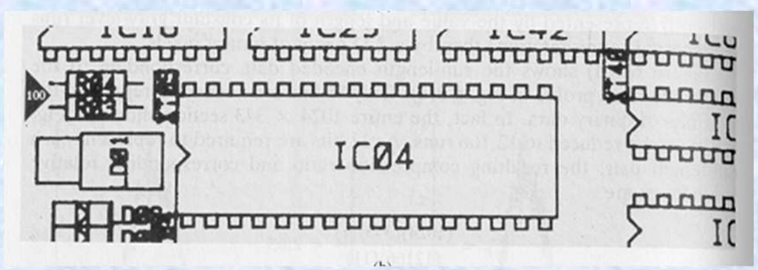
original





11 ……………0000……………………..11…..000…..

thresholded



Run-length encoding:

(1,63) (0,87) (1,37) (0,5) (1,4) (0, 556) (1,62) (0,210)

Using 11 bits/pair: **(1+10) bits/pair**

88 bits are required (compared to 1024 !!)

Psychovisual redundancy

- The human eye is more sensitive to the lower frequencies than to the higher frequencies in the visual spectrum.
- Idea: discard data that is perceptually insignificant!

Psychovisual redundancy (cont’d)

Example: quantization

256 gray levels

16 gray levels

16 gray levels + random noise

C=8/4 = 2:1

add a small pseudo-random number
to each pixel prior to quantization

Measuring Information

- A key question in image compression is:

  "What is the minimum amount of data that is sufficient to describe completely an image without loss of information?"

- How do we measure the information content of an image?

Measuring Information (cont'd)

- We assume that information generation is a probabilistic process.
- Idea: associate information with probability!

A random event *E* with probability *P*(*E*) contains:

$$I(E) = log(\frac{1}{P(E)}) = -log(P(E)) \text{ units of information}$$

Note: I(E)=0 when P(E)=1

How much information does a pixel value contain?

- Suppose that gray level values are generated by a random process, then r_k contains:

$$I(r_k) = -\log(P(r_k))$$

units of information!

(assume statistically independent random events)

How much information does an image contain?

- Average information content of an image:

$$E = \sum_{k=0}^{L-1} I(r_k)\,\mathrm{P}(r_k)$$

using $I(r_k) = -\,log(P(r_k))$

Entropy: $H = -\sum_{k=0}^{L-1} P(r_k) log(P(r_k))$ **units/pixel (e.g., bits/pixel)**

Redundancy

- **Redundancy:** (data vs info)

$$R = L_{avg} - H$$

where: $$L_{avg} = E(l(r_k)) = \sum_{k=0}^{L-1} l(r_k)P(r_k)$$

Note: if $L_{avg} = H$, then $R=0$ (no redundancy)

Entropy Estimation

- It is not easy to estimate H reliably!

image

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

Gray Level	*Count*	*Probability*
21	12	3/8
95	4	1/8
169	4	1/8
243	12	3/8

Entropy Estimation (cont'd)

- First order estimate of H:

$$H = -\sum_{k=0}^{3} P(r_k) log(P(r_k)) = 1.81 \text{ bits/pixel}$$

L_{avg} = 8 bits/pixel R= L_{avg}- H

The first-order estimate provides only a <u>lower-bound</u> on the compression that can be achieved.

Estimating Entropy (cont'd)

- Second order estimate of H:
  - Use relative frequencies of pixel blocks :

image

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

Gray Level Pair	*Count*	*Probability*
(21, 21)	8	1/4
(21, 95)	4	1/8
(95, 169)	4	1/8
(169, 243)	4	1/8
(243, 243)	8	1/4
(243, 21)	4	1/8

$$H = 2.5/2 = 1.25 \text{ bits/pixel}$$

Differences in Entropy Estimates

- Differences between higher-order estimates of entropy and the first-order estimate indicate the presence of interpixel redundancy!
- Need to apply some transformation to deal with interpixel redundancy!

Differences in Entropy Estimates (cont'd)

- Example: consider pixel differences

21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243
21	21	21	95	169	243	243	243

21	0	0	74	74	74	0	0
21	0	0	74	74	74	0	0
21	0	0	74	74	74	0	0
21	0	0	74	74	74	0	0

Gray Level or Difference	*Count*	*Probability*
0	16	1/2
21	4	1/8
74	12	3/8

Differences in Entropy Estimates (cont'd)

- What is the entropy of the pixel differences image?

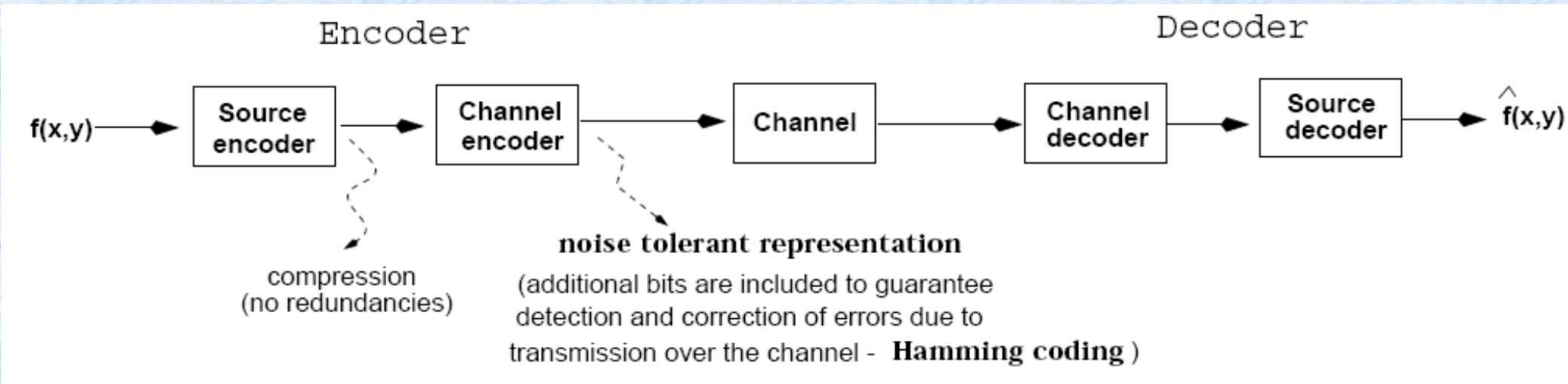
$$H = -\sum_{k=0}^{2} P(r_k)log(P(r_k)) = 1.41 \text{ bits/pixel}$$

(better than the entropy of the original image H=1.81)

- **An even better transformation should be possible since the second order entropy estimate is lower:**

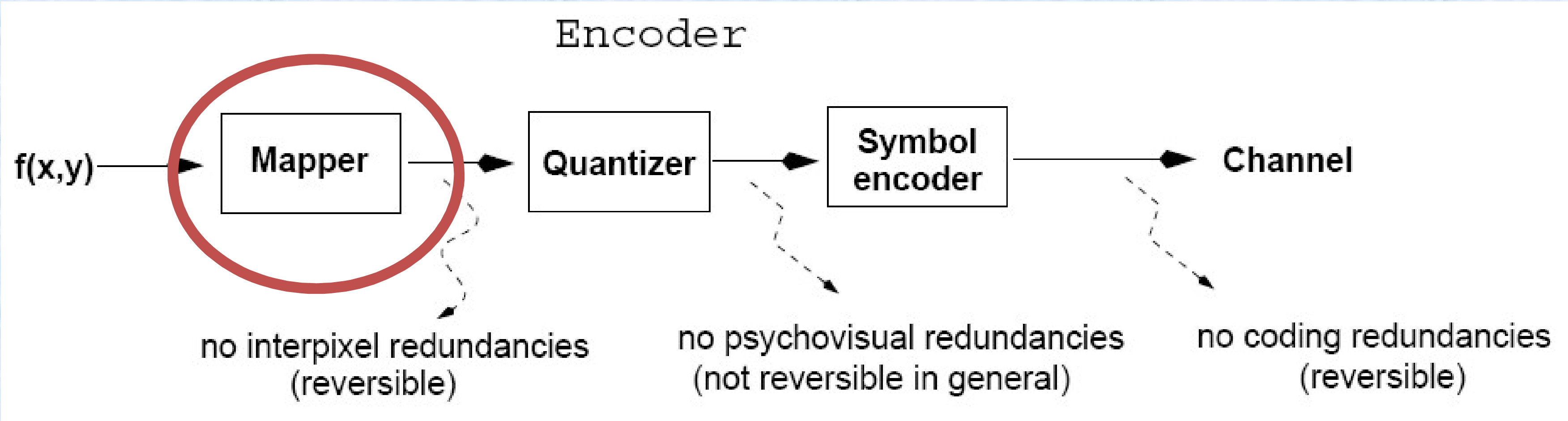
$$1.41 \text{ bits/pixel} > 1.25 \text{ bits/pixel}$$

Image Compression Model



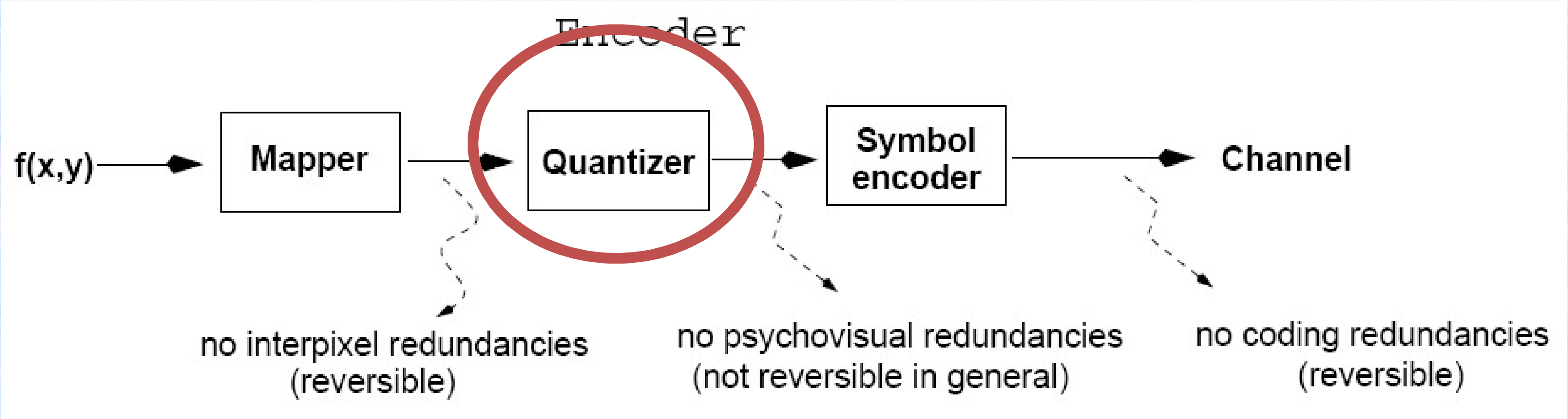
We will focus on the Source Encoder/Decoder only.

Image Compression Model (cont'd)



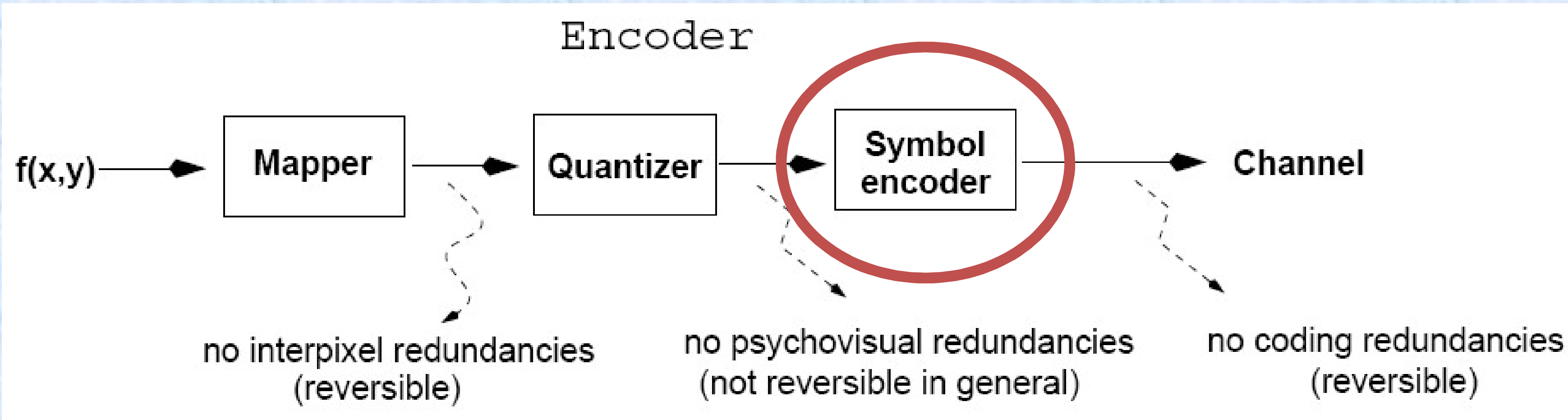
- **Mapper:** transforms data to account for interpixel redundancies.

Image Compression Model (cont'd)



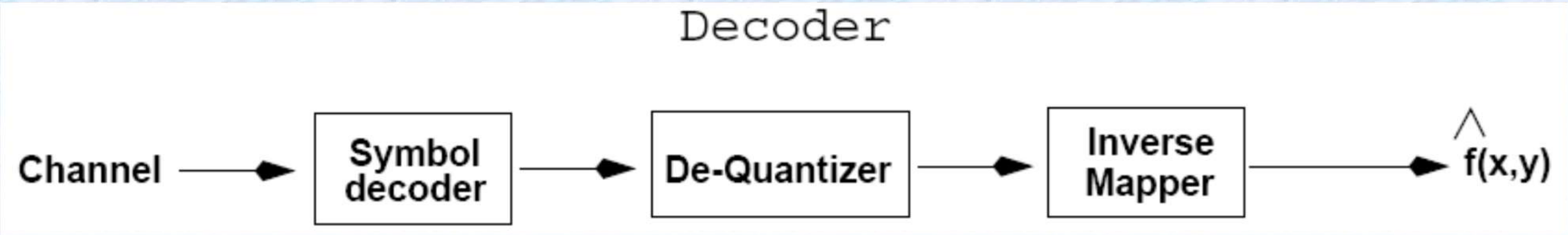
- **Quantizer:** quantizes the data to account for psychovisual redundancies.

Image Compression Model (cont'd)



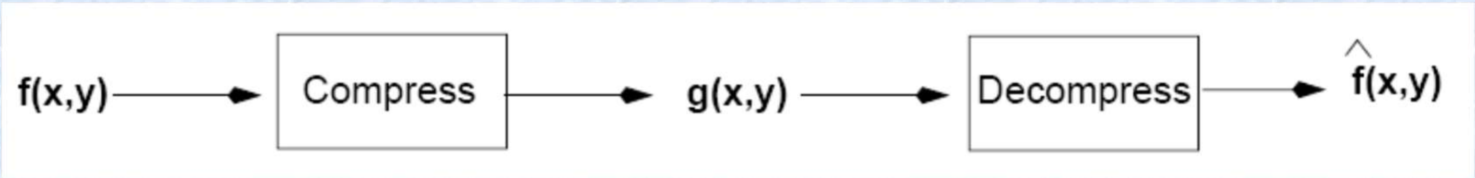
- **Symbol encoder:** encodes the data to account for coding redundancies.

Image Compression Models (cont'd)



- The decoder applies the inverse steps.
- Note that quantization is irreversible in general.

Fidelity Criteria



- How close is $f(x, y)$ to $\hat{f}(x, y)$?
- Criteria
  - Subjective: based on human observers
  - Objective: mathematically defined criteria

How are images compressed?

- Lossless compression: reversible, information preserving
- text compression algorithms,
- binary images, palette images
- Lossy compression: irreversible
- grayscale, color, video
- Near-lossless compression:
- medical imaging, remote sensing.
- **Questions to answer before compression**
- 1) Why do we need lossy compression?
- 2) When do we use only lossless compession?

Lossy vs. Lossless compression

- Achievable compression
  - Lossless: Modest 1:2 to 1:3 at best
  - Lossy: Can be as high as 1:100

Lossy vs. Lossless compression Example

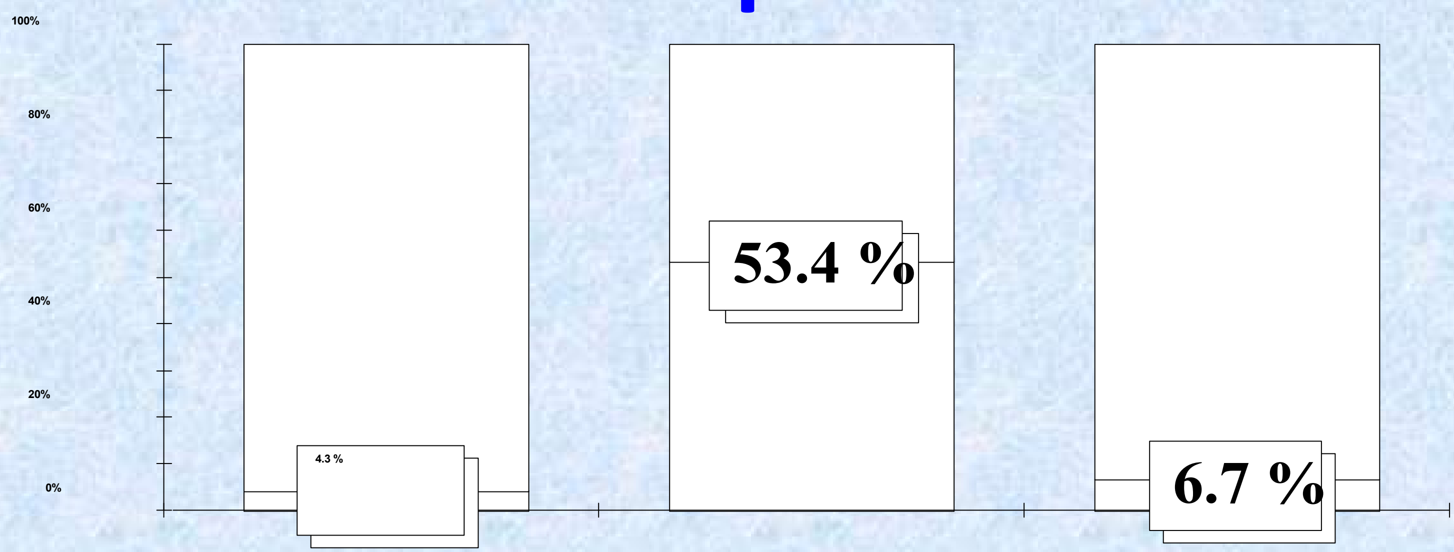


IMAGE:	**CCITT-3**	**LENA**	**LENA**
TYPE:	**binary**	**gray-scale**	**gray-scale**
METHOD:	**JBIG (lossless)**	**JPEG (lossless)**	**JPEG (lossy)**

Rate measures

Bitrate:

bits/pixel

$$\frac{\text{size of the compressed file}}{\text{pixels in the image}} = \frac{C}{N}$$

Compression ratio:

$$\frac{\text{size of the original file}}{\text{size of the compressed file}} = \frac{N \cdot k}{C}$$

Distortion measures

x_i, y_i are original and estimate after reconstruction

Mean average error (MAE): $$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - x_i\right|$$

Mean square error (MSE): $$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - x_i\right)^2$$

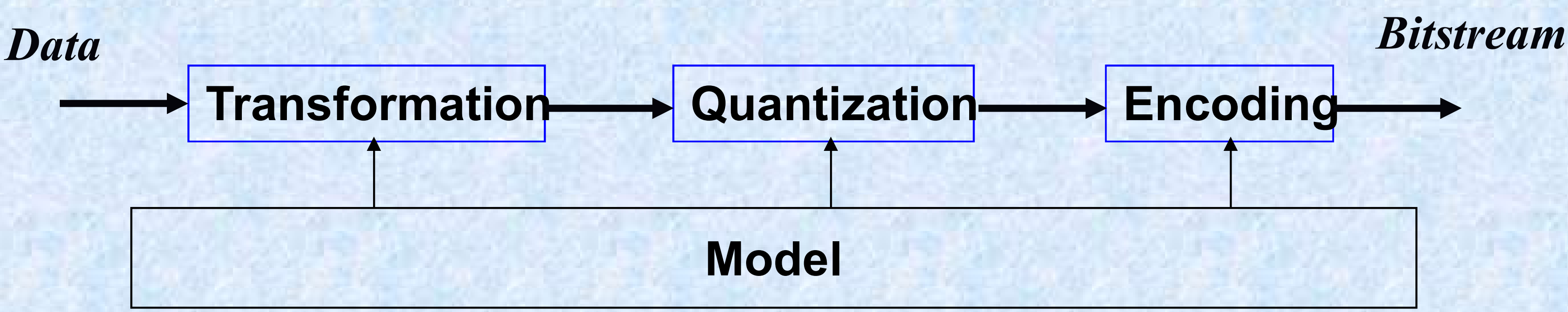
Signal-to-noise ratio (SNR): $$\mathrm{SNR} = 10\cdot\log_{10}\left[\sigma^2/\mathrm{MSE}\right]$$
(decibels)

Peak-signal-to-noise ratio (PSNR): $$\mathrm{PSNR} = 10\cdot\log_{10}\left[A^2/\mathrm{MSE}\right]$$
(decibels)

A: signal amplitude = 2^8-1=255 for 8-bits signal. σ^2 is original image variance

Lossy image compression

- DPCM: Prediction error quantization
- Block Truncation Coding (BTC)
- Vector Quantization (VQ)
- Transform Coding (DCT, JPEG)
- Subband Coding
- Wavelet Coding (JPEG2000)



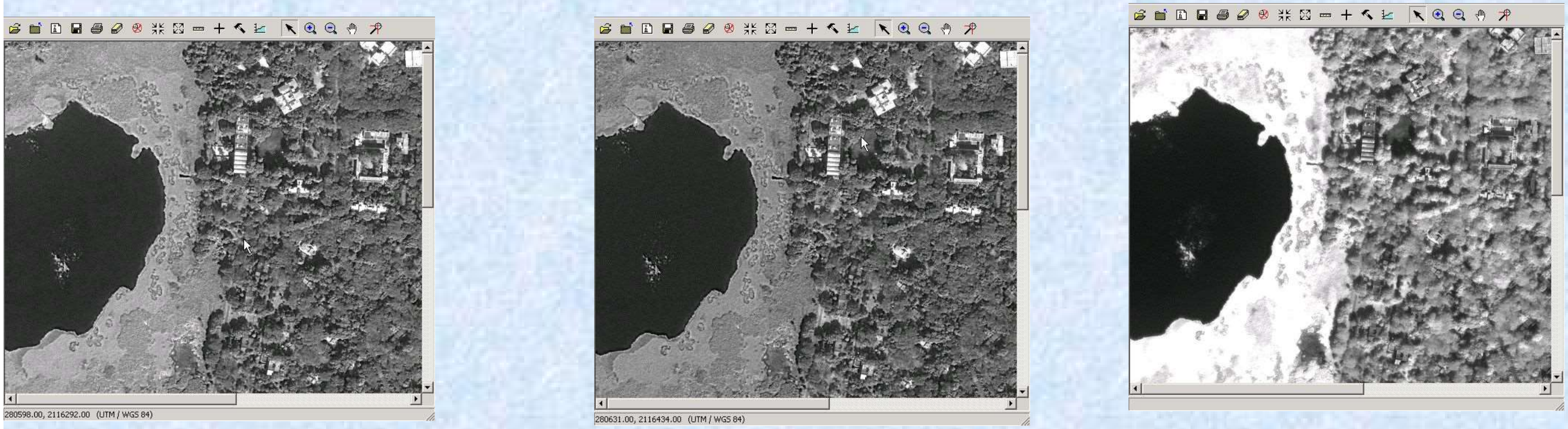
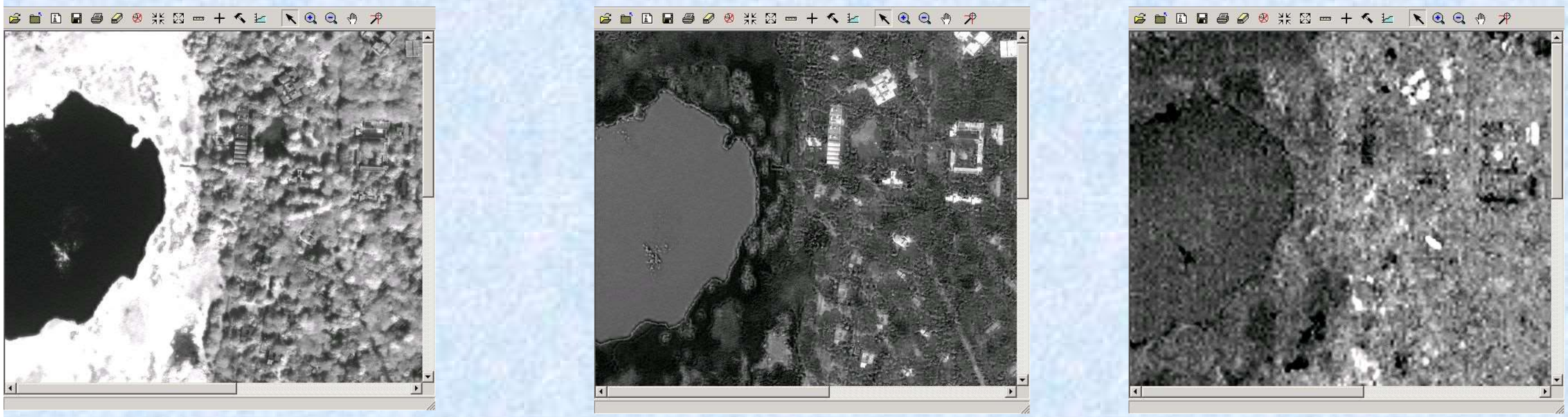
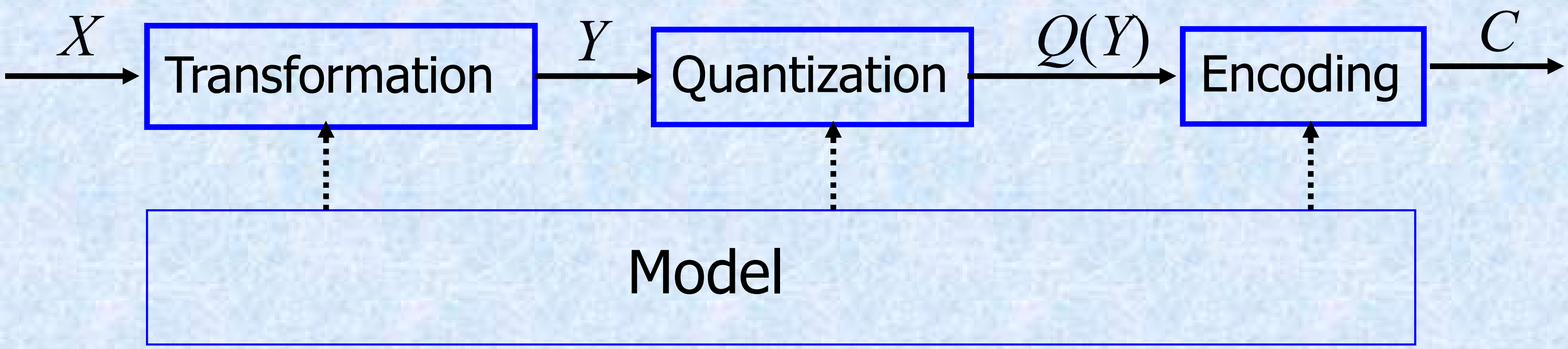


Illustration of PCT – transformation highlights low variance bands for discarding; 1st row original, 2nd row PC 1,2,3



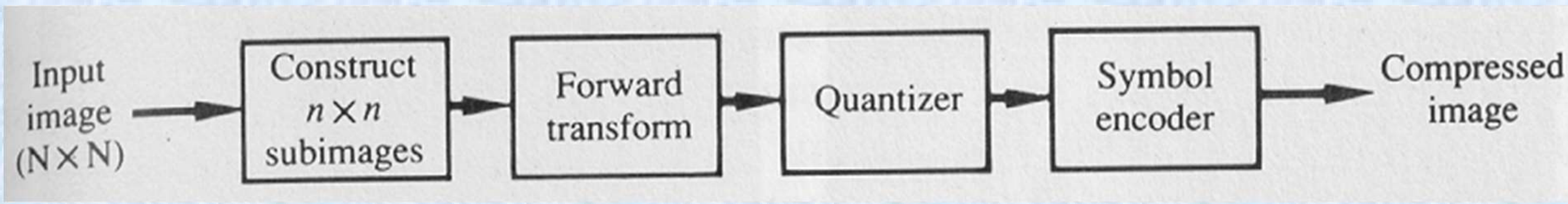
Transform Domain Compression

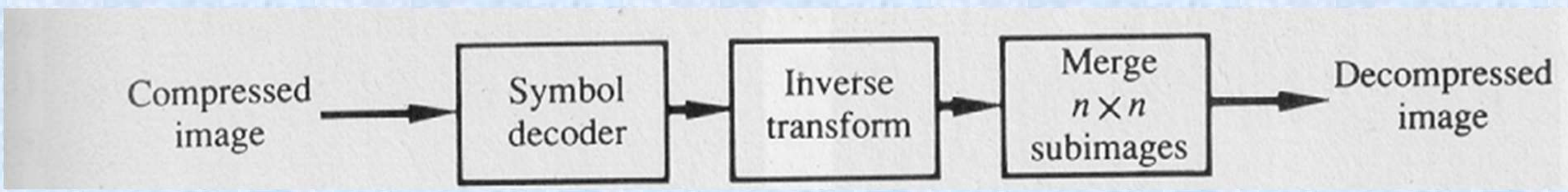
- The main goal of transformation is decorrelation of data.



Lossy Compression

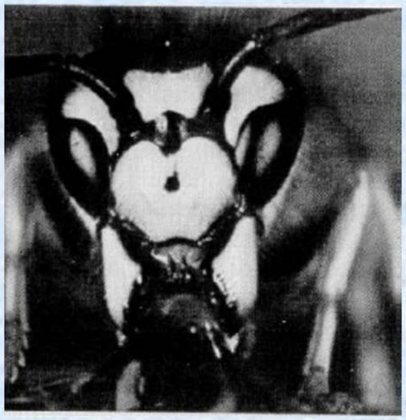
- Transform the image into some other domain to reduce interpixel redundancy.

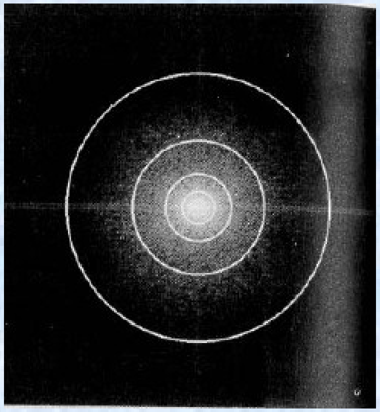




Example: Fourier Transform

$$f(x,y)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}F(u,v)e^{\frac{j2\pi(ux+vy)}{N}},\quad \text{x,y=0,1,...,N-1}$$





Note that the magnitude of the FT decreases, as *u*, *v* increase!

K << N

$$\hat{f}(x,y)=\frac{1}{N}\sum_{u=0}^{K-1}\sum_{v=0}^{K-1}F(u,v)e^{\frac{j2\pi(ux+vy)}{N}},\quad \text{x,y=0,1,...,N-1}$$

$$\sum_{x,y}(\hat{f}(x,y)-f(x,y))^2 \text{ is very small !!}$$

Transform Selection

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(u,v) h(x,y,u,v)$$

- T(u,v) can be computed using various transformations, for example:
  - DFT
  - DCT (Discrete Cosine Transform)
  - KLT (Karhunen-Loeve Transformation) or Principal Component Analysis (PCA)
- JPEG uses DCT for handling interpixel redundancy.

DCT (Discrete Cosine Transform)

Forward: $$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)cos(\frac{(2x+1)u\pi}{2N})cos(\frac{(2y+1)v\pi}{2N}),$$

u, v=0,1,...,N-1

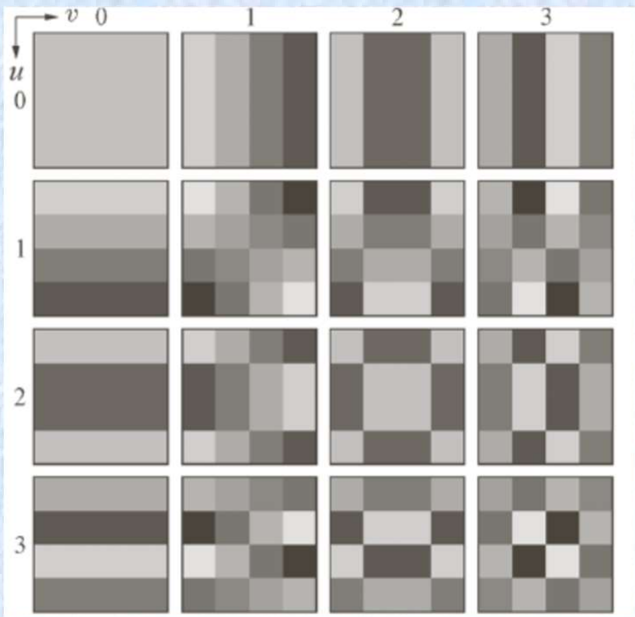
Inverse: $$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)cos(\frac{(2x+1)u\pi}{2N})cos(\frac{(2y+1)v\pi}{2N}),$$

x, y=0,1,...,N-1

$$\alpha(u) = \begin{cases} \sqrt{1/N} & \text{if u=0} \\ \sqrt{2/N} & \text{if u>0} \end{cases} \qquad \alpha(v) = \begin{cases} \sqrt{1/N} & \text{if v=0} \\ \sqrt{2/N} & \text{if v>0} \end{cases}$$

DCT (cont'd)

- Basis functions for a 4x4 image (i.e., cosines of different frequencies).



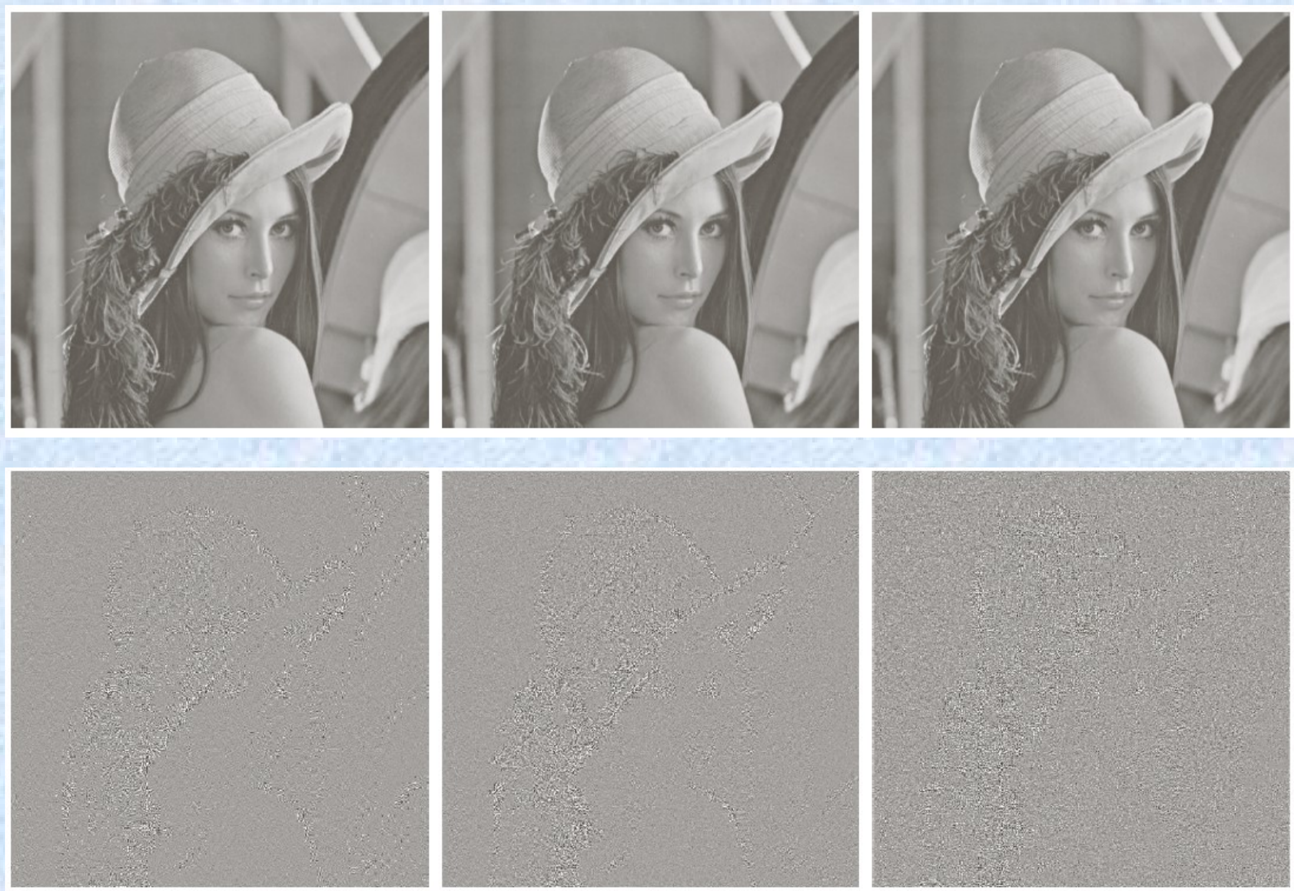
DCT (cont'd)

Using
8 x 8 sub-images
yields 64 coefficients
per sub-image.

Reconstructed images
by truncating
50% of the
coefficients

DCT is a more
compact
transformation!

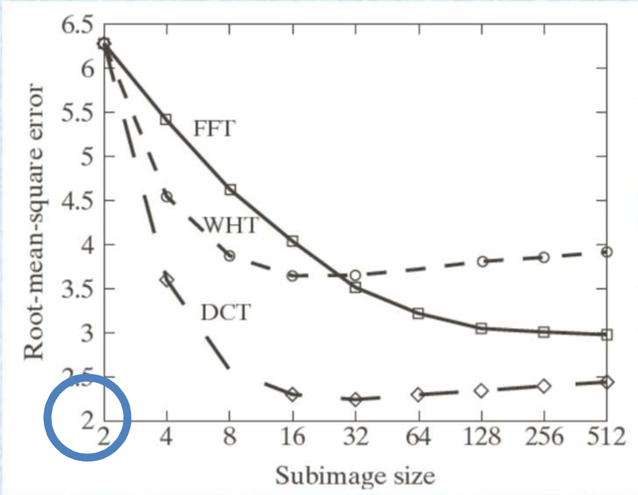
DFT **WHT** **DCT**



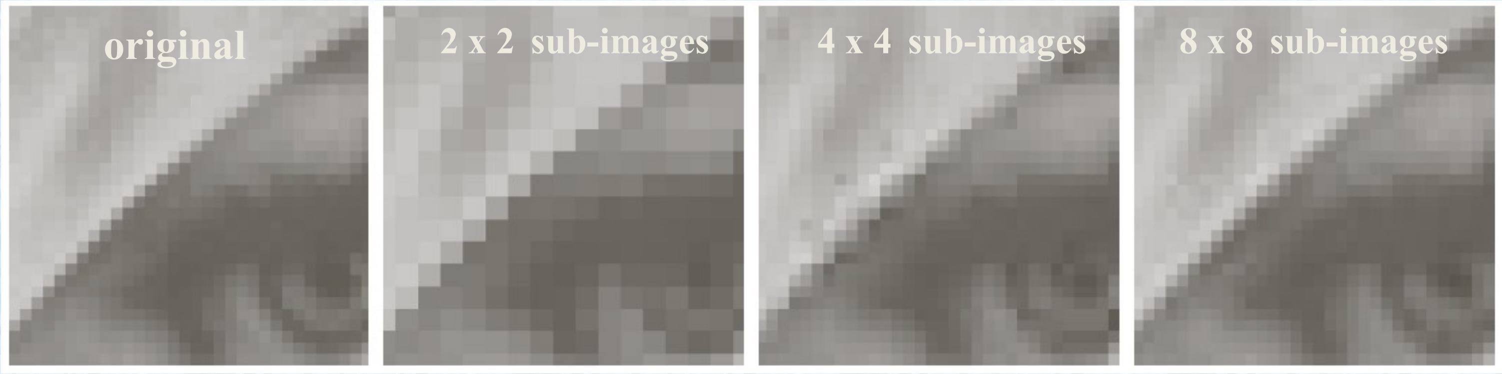
RMS error: 2.32 1.78 1.13

DCT (cont'd)

- Sub-image size selection:



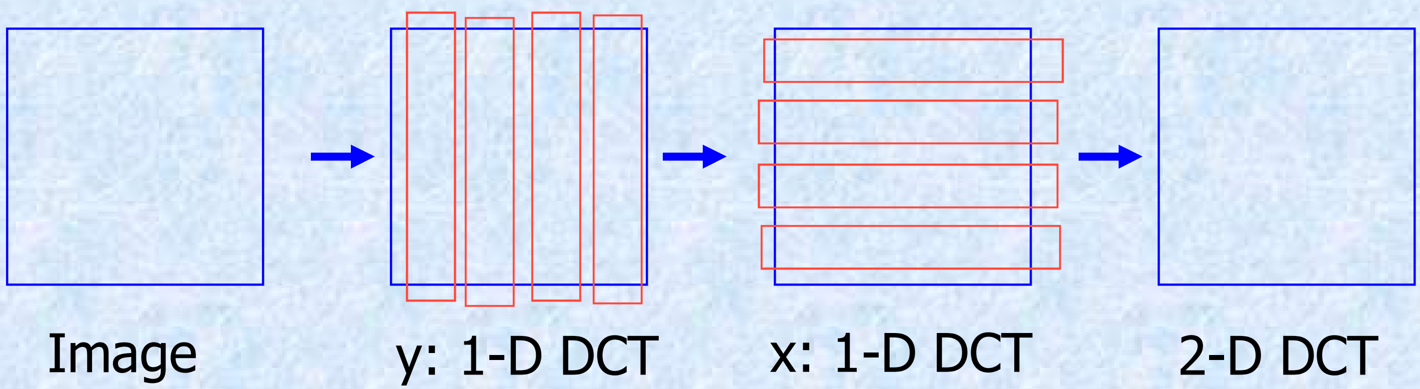
Reconstructions



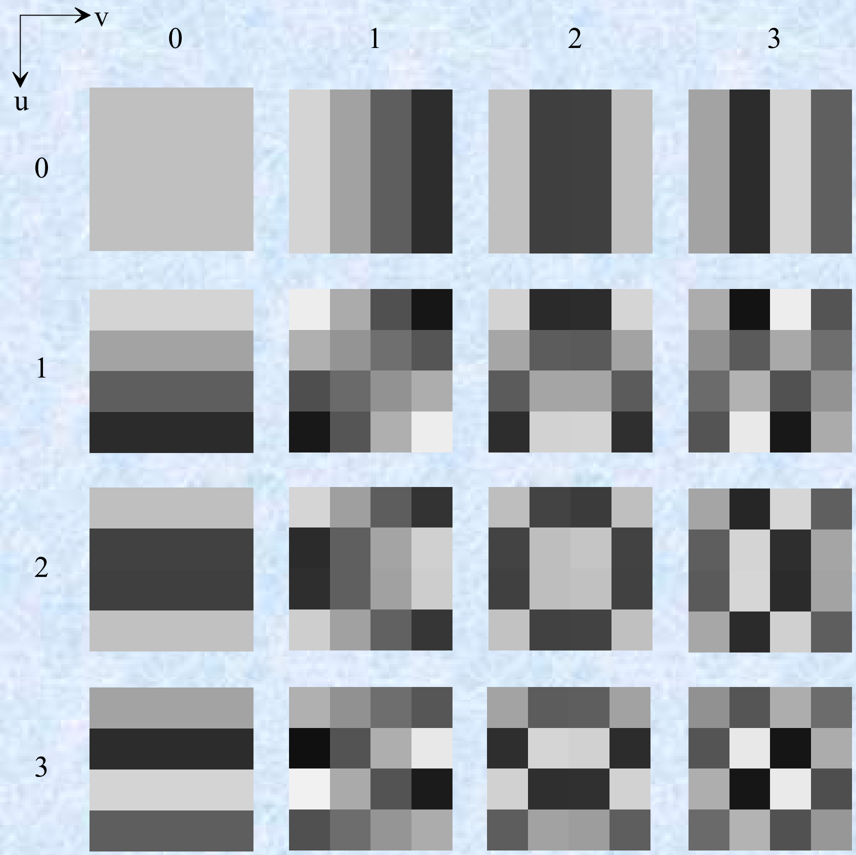
2-D DCT

$$C(k,l) = \alpha(k)\alpha(l)\sum_{j=0}^{N-1}\sum_{i=0}^{N-1} x_{i,j} \cdot \cos\left[\frac{(2i+1)k\pi}{2N}\right] \cdot \cos\left[\frac{(2j+1)l\pi}{2N}\right]$$

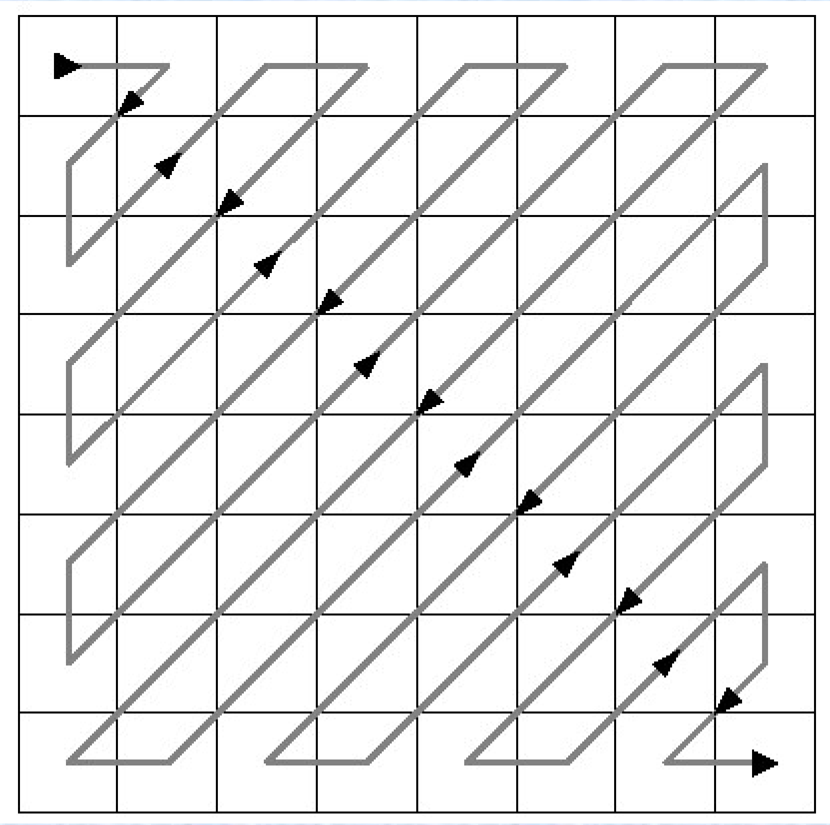
- The 2-D DCT is performed as two sequential 1-D DCTs
- Complexity of DCT is O(N logN) instead of O(N^4)



2-D DCT: N=4



Zig-zag DCT coefficients ordering



Where is compression?

- DCT is reversible transformation:

$$Y=TX \rightarrow X=T^{-1}Y$$

- Where is compression?

Data decorrelation and energy compactness

$\rightarrow$ Quantization (lossy operation)

$\rightarrow$ Statistical encoding

Example

Original Pixels

55	55	109	109	109	109	109	109
55	55	109	109	109	109	109	109
55	55	109	109	109	109	109	109
55	55	109	109	109	109	109	109
55	55	55	55	55	55	55	55
55	55	55	55	55	55	55	55
55	55	55	55	55	55	55	55
55	55	55	55	55	55	55	55

Discrete Cosine Transform

Unquantized Coefficients

602	-69	-50	-24	0	16	21	14
147	-63	-45	-22	0	15	19	12
0	0	0	0	0	0	0	0
-52	22	16	8	0	-5	-7	-4
0	0	0	0	0	0	0	0
34	-15	-11	5	0	3	4	3
0	0	0	0	0	0	0	0
-29	12	9	4	0	-3	-4	-2

Quantize

Quantized Coefficients

75	-8	-6	-3	0	2	2	1
18	-7	-5	-2	0	1	2	1
0	0	0	0	0	0	0	0
-6	2	2	1	0	0	0	0
0	0	0	0	0	0	0	0
4	-1	-1	0	0	0	0	0
0	0	0	0	0	0	0	0
-3	1	1	0	0	0	0	0

Inverse DCT

Reconstructed Pixels

59	59	105	107	107	110	107	107
56	59	105	108	107	108	106	107
56	62	105	110	107	107	106	107
57	64	100	108	107	106	103	102
54	50	61	58	55	56	59	60
55	52	56	56	54	55	57	55
55	55	56	57	54	54	56	55
52	55	56	59	54	53	56	55

JPEG

- JPEG = Joint Photographic Experts Group
- Lossy coding of continuous tone still images (color and grayscale)
- Based on Discrete Cosine Transform (DCT):

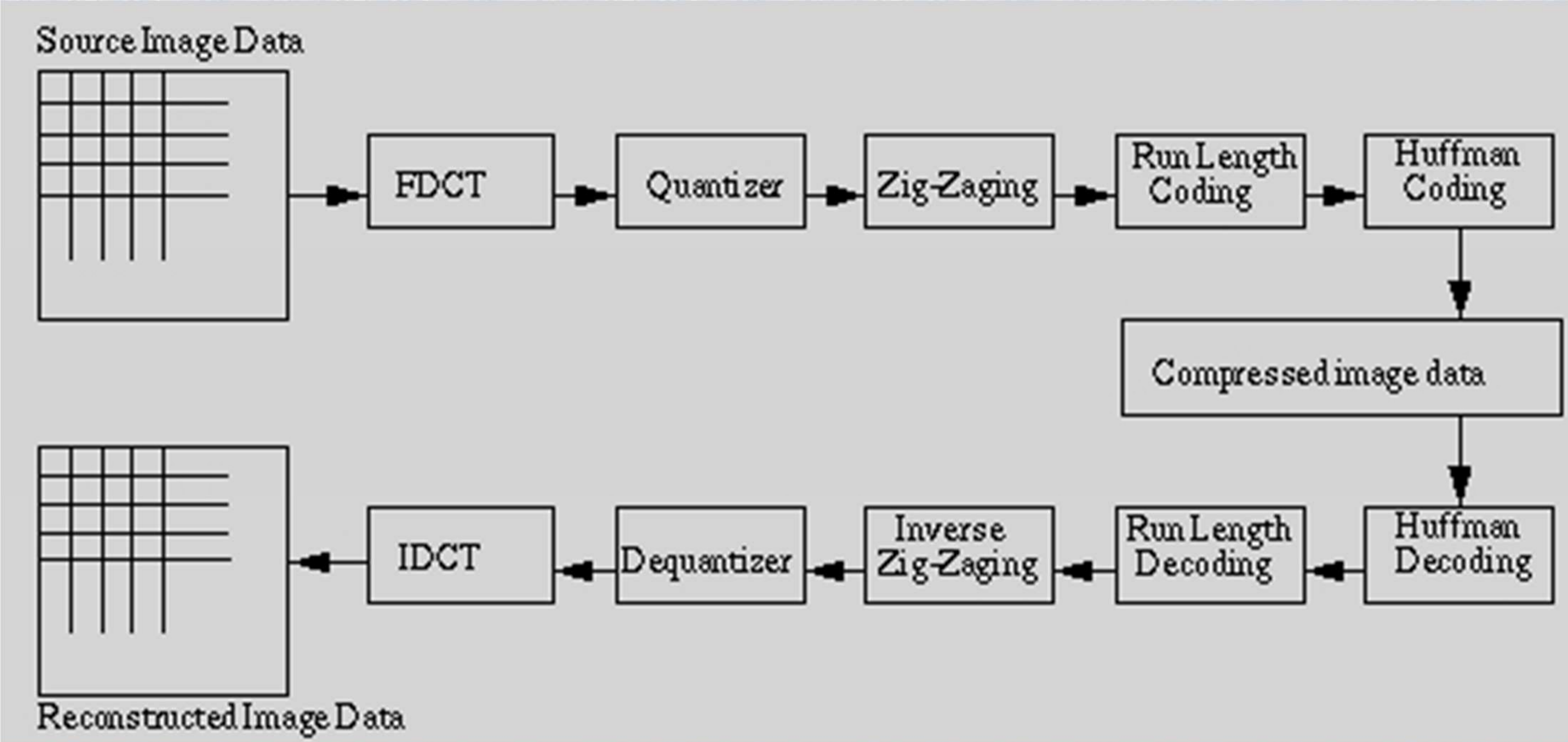
  0) Image is divided into block $N \times N$

  1) The blocks are transformed with 2-D DCT

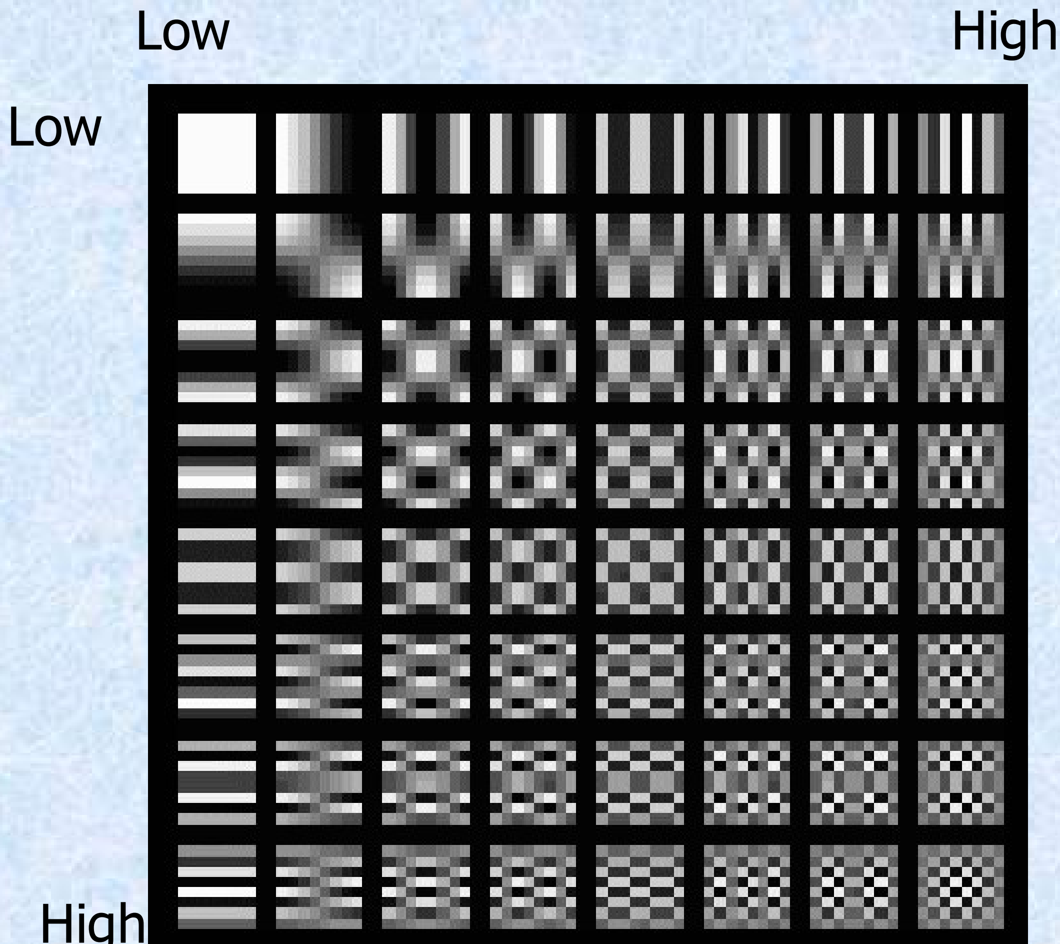
  2) DCT coefficients are quantized

  3) The quantized coefficients are encoded

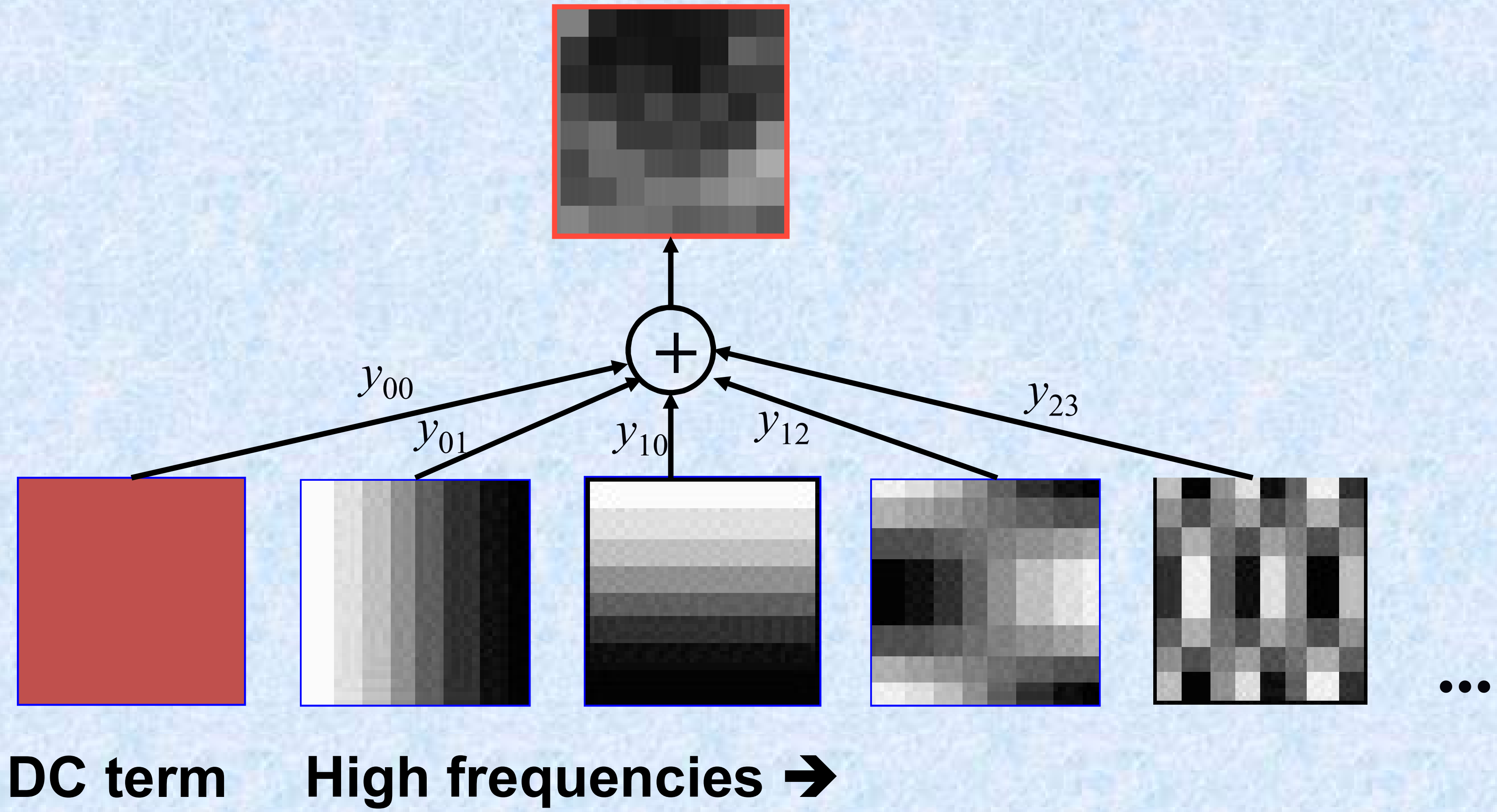
JPEG: Encoding and Decoding



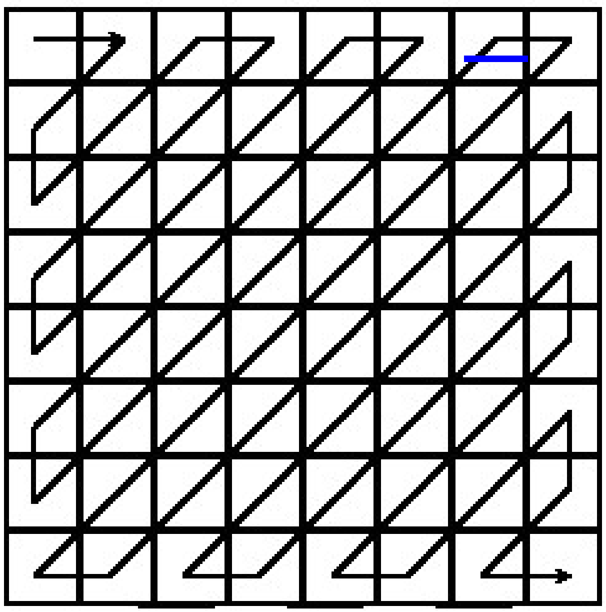
2-D DCT basis functions: N=8



2-D Transform Coding



Zig-zag ordering of DCT coefficients



0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Converting a 2-D matrix into a 1-D array, so that the frequency (horizontal and vertical) increases in this order and the coefficents variance are decreasing in this order.

JPEG Steps

4. Quantize the coefficients (i.e., reduce the amplitude of coefficients that do not contribute a lot).

$$C_q(u,v) = Round\left[\frac{C(u,v)}{Q(u,v)}\right]$$

Q(u,v): quantization table

One way to generate the quantization table

- Quantization Table Q[i][j]

```
for i=0 to n;
  for j=0 to n;
    Q[i,j]= 1 + (1+i+j)*quality;
  end j;
end i;
```

$$1 \leq quality \leq 25$$

(best - low compression) (worst - high compression)

Example

(d) Quantization table
(quality = 2)

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

Example of DCT for image block

Original block									Transformed block							
139	144	149	153	155	155	155	155		235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
144	151	153	156	159	156	156	156		-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
150	155	160	163	158	156	156	156		-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
159	161	162	160	160	159	159	159		-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
159	160	161	162	162	155	155	155		-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
161	161	161	161	160	157	157	157		1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
162	162	161	163	162	157	157	157		-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
162	162	161	161	163	158	158	158		-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

Default quantization matrix Q

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

Examples: $236/16 \rightarrow 15$
$-22/11 \rightarrow -2$

$$y_q(k,l)=\text{round}[y(k,l)/Q(k,l)]$$

Quantization of DCT coefficients: Example

Quantization matrix							
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

Quantized coefficients							
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	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

Ordered DCT coefficients: 15,0,-2,-1,-1,-1,0,0,-1,-1, 54{'0'}.

Dequantization

Quantized coefficients							
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	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

Dequantized 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

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

$$z(k,l)=y_q(k,l)\cdot Q(k,l)$$

Examples: $236/16 \rightarrow 15$
$-22/11 \rightarrow -2$

Inverse DCT

Dequantized 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

Decompressed block

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	156
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

Original

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

Encoding of quantized DCT coefficients

Quantized coefficients							
15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	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

- Ordered data: 15,0,-2,-1,-1,-1,0,0,-1,-1, 54{'0'}.
- Encoding: • DC: ? • AC: ?

Encoding of quantized DCT coefficients

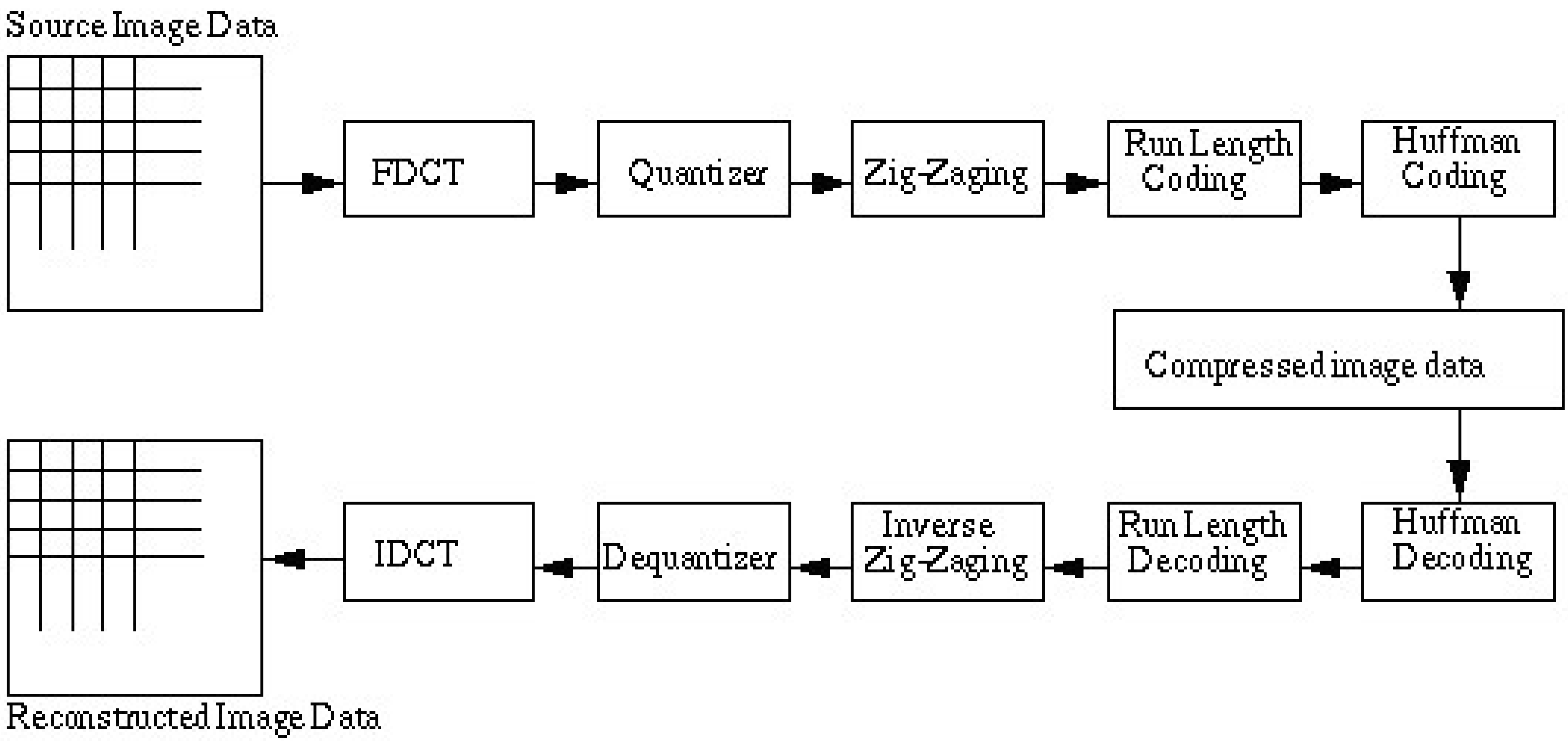
- DC coefficient for the current block is predicted of that of the previous block, and error is coded using Huffman coding
- AC coefficients:

(a) Huffman code, arithmetic code for non-zeroes
(b) run-length encoding: (number of ’0’s, non-’0’-symbol)

Suitability of JPEG Method

- The disadvantage with the present JPEG compression technique is that the compression rate is arbitrary.
- As the main source of compression is the quantization table, an optimal quantization table can help to achieve a desired compression rate with good image quality.

JPEG LOSSY COMPRESSION



Performance of JPEG algorithm

8 bpp

0.6 bpp

0.37 bpp

0.22 bpp

Rate control in JPEG Method

- When is a precise compression rate essential?
  - Maximum quality to be preserved
  - Full capacity of communication channel to be utilized
  - Data rate should not exceed the channel capacity

Key requirement in space applications

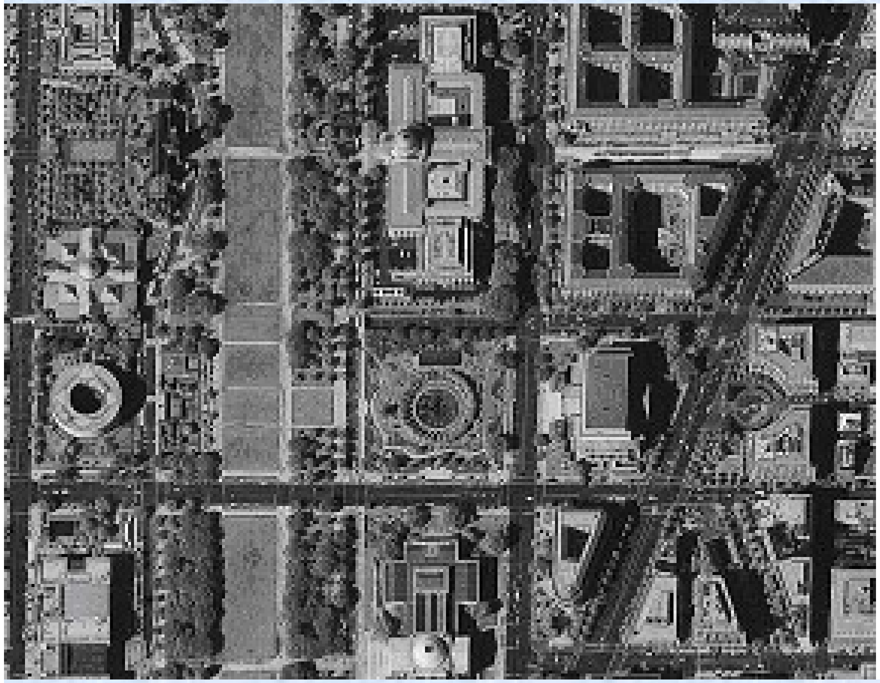
How to control rate of compression with JPEG?

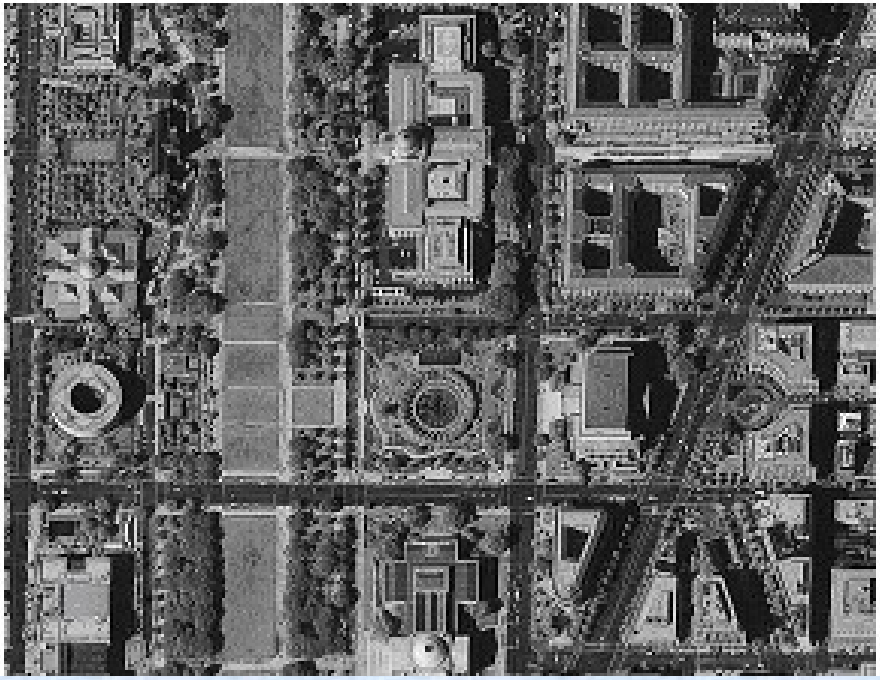
- Quality factor is the only way to compress less or more with JPEG.
- How can the desired rate of compression be specified in JPEG?
- What is the implication when the rate of compression is user specified on JPEG technique?

Quality Factor in JPEG

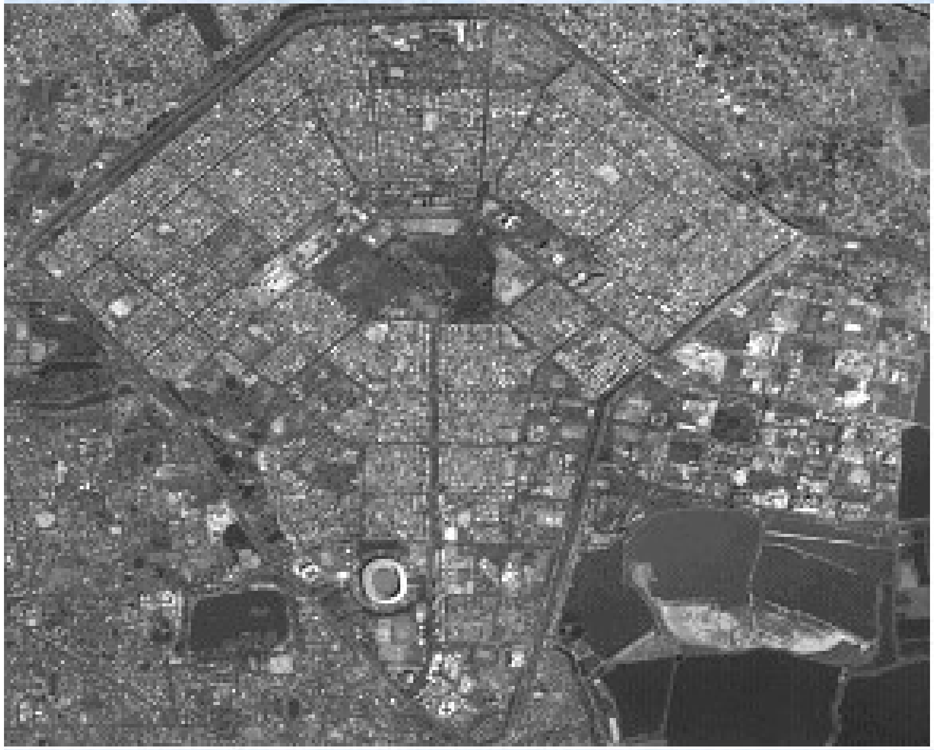
- The information loss happens during the quantization phase
- In normal case quantization table coefficients are scaled up or down based on user-specified
- The precise rate cannot be predicted based on the quality factor

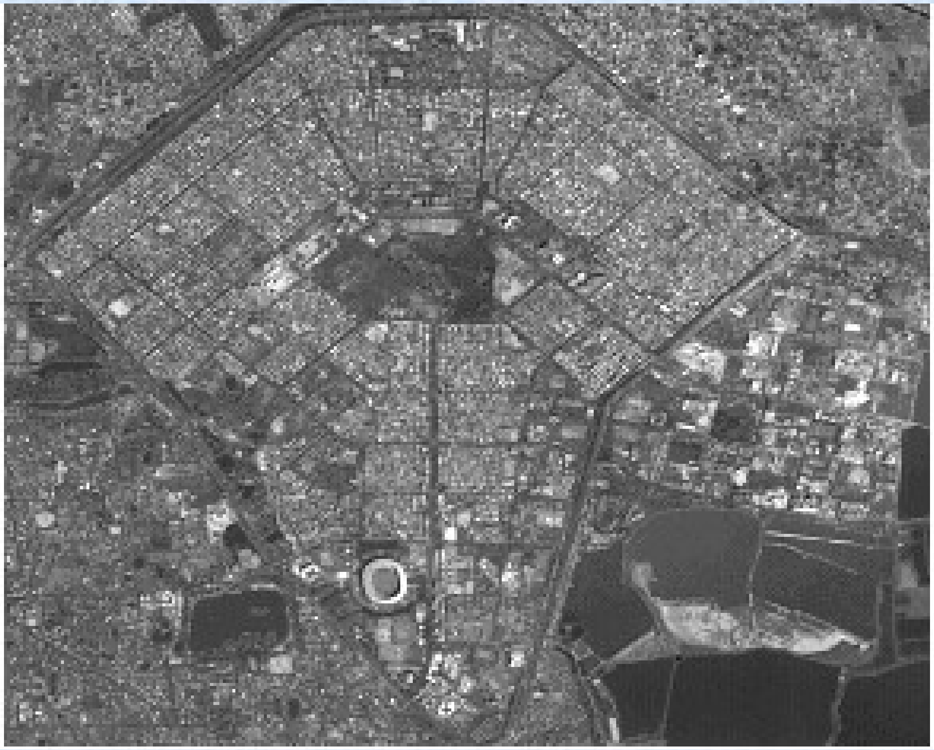
Uncompressed CR = 3.5





Uncompressed **CR =3.5**





Uncompressed **CR = 3.5**

Uncompressed

CR = 3.5

Uncompressed CR = 3.5

POWAI (8 BIT)

(Large Size)

POWAI (8 BIT); (Large Size) CR=6

Original

CR = 6

Sample Quantization Table CR=6

2	8	9	13	13	21	22	41
8	10	12	13	21	22	40	43
10	12	15	20	23	40	44	55
11	15	20	23	39	46	54	55
17	17	25	35	46	54	56	83
17	25	35	49	51	57	63	88
28	32	50	51	59	61	104	115
30	50	50	59	61	105	112	127

Slide 82

Uncompressed MUMBAI(11 BIT) CR = 3.5

1	1	1	1	1	2	3	5
1	1	1	1	2	3	5	5
1	1	2	2	3	5	5	7
1	2	2	3	5	5	6	7
2	2	3	4	6	6	7	23
2	3	4	6	6	7	20	32
3	4	6	6	9	16	38	68
4	6	6	12	15	65	67	71

CR = 6 **CR = 9.5**

2	8	9	13	13	21	22	41
8	10	12	13	21	22	40	43
10	12	15	20	23	40	44	55
11	15	20	23	39	46	54	55
17	17	25	35	46	54	56	83
17	25	35	49	51	57	63	88
28	32	50	51	59	61	104	115
30	50	50	59	61	105	112	127

Quant. Tables

Summary

- JPEG compressors and decompressors can be used for compressing remotely sensed data as the Compression ratio (CR) will seldom be high and hence JPEG gives satisfactory performance in terms of maintaining image characteristics.
- As JPEG works with 8 by 8 block of pixels at a time the implementation in hardware for JPEG is amenable to onboard satellite platforms
- The JPEG process maintains a small Mean Squared Error (MSE) at different compression ratios tested in the range from 3.0 to 15.0.

Contd…