Digital Image Processing

Image Compression

Importance of Image Compression

- Typical image resolution/depth
  – 1024 x 1024 x 24
  – Typical image size in bytes: 3 MB
- Typical video rate/resolution/depth
  – 30 fps, 720×576 , 24 bits
  – Typical video size for 1 minute: More than 2 GB

Data Redundancy

- Redundancy in information theory is the number of bits used to transmit a message minus the number of bits of actual information in the message. Informally, it is the amount of wasted "space" used to transmit certain data

Data Compression

- **data compression or source coding** is the process of encoding information using fewer bits (or other information-bearing units) than an un-encoded representation would use through use of specific encoding schemes.

Compression Methods

- Lossless Methods
- Lossy Methods

Run Length Coding

- In this method ***runs*** of data (that is, sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count.
- Useful in relatively simple graphic images such as icons, line drawings, and animations.
- Example,
  WWWWWWWWWWWWBWWWWWWWWWW
  WWBBBWWWWWWWWWWWWWWWWWW
  WWWWWWBWWWWWWWWWWWWWW

  is stored as : 12W1B12W3B24W1B14W

Huffman Coding

- Huffman coding is an entropy encoding algorithm used for lossless data compression.
- The term refers to the use of a variable-length code table for encoding a source symbol

Example of Variable Length Coding

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
r_k for $k \neq 87, 128, 186, 255$	0	—	8	—	0

Huffman Coding Example

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

Huffman Coding Example

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

Arithmetic Coding

- The output from an arithmetic coding process is a single number less than 1 and greater than or equal to 0
- This single number can be uniquely decoded to create the exact stream of symbols that went into its construction.

Example

- Encode the string “BILL GATES”

Character	Probability
SPACE	1/10
A	1/10
B	1/10
E	1/10
G	1/10
I	1/10
L	2/10
S	1/10
T	1/10

Arithmetic Coding Example

Character	Probability	Range
SPACE	1/10	0.00 - 0.10
A	1/10	0.10 - 0.20
B	1/10	0.20 - 0.30
E	1/10	0.30 - 0.40
G	1/10	0.40 - 0.50
I	1/10	0.50 - 0.60
L	2/10	0.60 - 0.80
S	1/10	0.80 - 0.90
T	1/10	0.90 - 1.00

Recovering Data

New Character	Low value	High Value
	0.0	1.0
B	0.2	0.3
I	0.25	0.26
L	0.256	0.258
L	0.2572	0.2576
SPACE	0.25720	0.25724
G	0.257216	0.257220
A	0.2572164	0.2572168
T	0.25721676	0.2572168
E	0.257216772	0.257216776
S	0.2572167752	0.2572167756

Lossy Compressions

- JPEG
- JPEG2000
- MPEG (Video Compreession)
- H264 (Video Compression)

Steps followed in JPEG

- Color space transformation
- Downsampling
- Block splitting
- Discrete cosine transform
- Quantization
- Entropy coding

Color space transformation

- First, the image should be converted from RGB into YCbCr color space
- The compression is more efficient as the brightness information, which is more important to the eventual perceptual quality of the image, is confined to a single channel, more closely representing the human visual system.

Downsampling

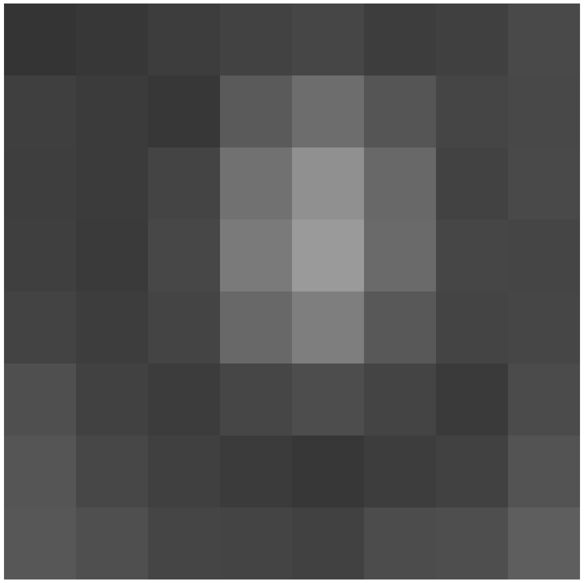
- Due to the densities of color- and brightness-sensitive receptors in the human eye, humans can see considerably more fine detail in the brightness of an image (the Y component) than in the color of an image (the Cb and Cr components).
- Downsample Cb and Cr components only
  4:2:0

Block splitting

- After subsampling, each channel must be split into 8×8 blocks of pixels.

Discrete cosine transform

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$



2D DCT

$$G_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} g_{x,y} \cos\left[\frac{\pi}{8}\left(x + \frac{1}{2}\right) u\right] \cos\left[\frac{\pi}{8}\left(y + \frac{1}{2}\right) v\right]$$

$$\alpha_p(n) = \begin{cases} \sqrt{\frac{1}{8}}, & \text{if } n = 0 \\ \sqrt{\frac{2}{8}}, & \text{otherwise} \end{cases}$$

DCT Transformed Image

$$\overset{\displaystyle u}{\longrightarrow}$$

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix} \Big\downarrow v$$

Quantization

- Quantization is done by simply dividing each component in the frequency domain by a constant for that component, and then rounding to the nearest integer

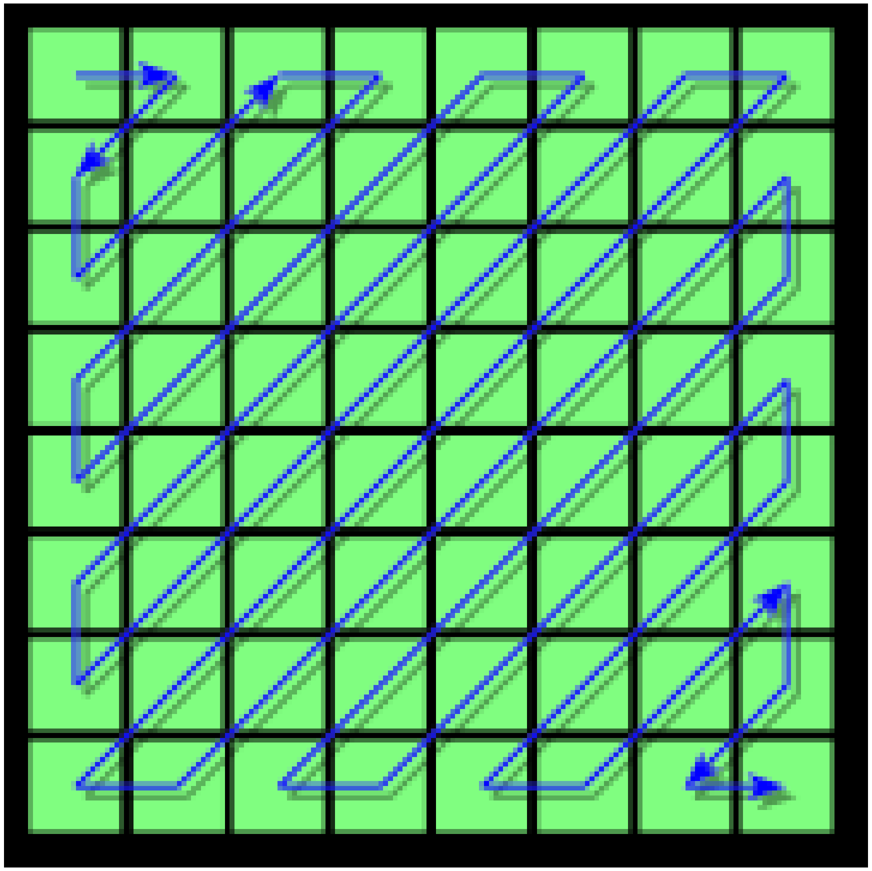
Sample Quantization Table

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Quantization Results

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Entropy coding (Zig-Zag Scanning)



Zig-Zag Scanning

−26							
−3	0						
−3	−2	−6					
2	−4	1	−4				
1	1	5	1	2			
−1	1	−1	2	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							

Variable length codes

- Variable length codes are generated for the code words of the zig-zag scan result.
- Trailing zeros are truncated by inserting an EOB symbol after the last non-zero value.

Typical compression ratios

- The compression ration depends on the quantization factors used.
- Typical ratio is 4:1