STEVENS
Institute of Technology

LSB and JPEG Based Steganography

Outline

- Overview of txt, audio, image and video
- Steganography
  - Definition and properties
  - Problem formulation and early methods
  - LSB embedding method
  - Introduction to JPEG
  - JPEG based steganography

Text

- Unformatted text: also known as plaintext and enables pages to be created which comprise strings of fixed-sized characters form a limited character set;
- Formatted text: also known as richtext and enables pages and complete document to be created which comprise of strings of characters of different styles, size, and shape with tables, graphics, and images inserted at appropriate points;
- Hypertext: enables an integrated set of documents to be created which have defined linkages between them.

Images

- A two dimensional matrix of individual picture elements – known as pixels or sometimes pels – each of which have a range of values with it.
  - Computer graphics or simply graphics: BMP (bit-map format), GIF (graphical interchange format), TIFF (tagged image file format) and etc.
  - Digitized documents: produced by the scanner associated with a fax machine
  - Digitized pictures: digitizing continuous-tone monochromatic images

Some concepts in digitized pictures

- Raster-scan: the pattern of image detection and reconstruction in most television set
- Pixel depth: the number of bits per pixel
- Aspect ratio: the number of pixels per scanned line and the number of lines per frame

Audio

- Speech signal:
  - Used in a variety of interpersonal applications including telephony and video telephony.
  - The bandwidth is from 50Hz through to 10kHz
  - The sampling rate is 20ksps(2*10kHZ)
- Music signal:
  - Used in applications such as CD and broadcast television
  - The bandwidth is from 15HZ through to 20kHz and
  - The sampling rate is 40ksps(2*20kHZ).

Video

- Video (or moving images, moving pictures) is a sequence of image frames which are displayed at a rapid succession to create an impression of motion.
- The time interval between each two successive frames is usually a constant. The number of frames displayed per second is called frame rate (frames per second, fps)
- Frame rates in 10 ~ 15 fps provide basic motion impression. Frame rates in 16 ~ 30 fps appear smooth in motion.
- Current movies are at 24 fps, NTSC TV is at 30 fps, PAL and SECAM TV at 25 fps. Future HDTV can get 60 fps.

Broadcast television

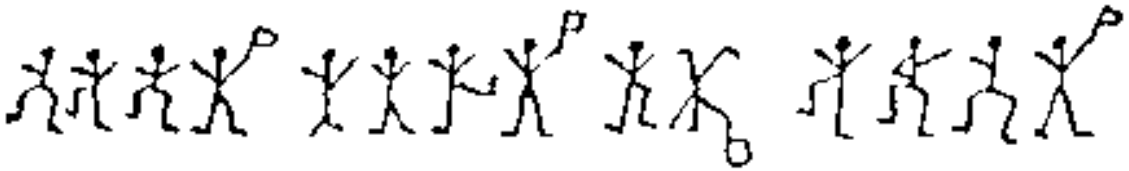
- Color signals
  - brightness: represents the amount of energy that stimulates the eye and varies on a gray scale from black (lowest) through to white (highest).
  - hue: represents the actual color of the source
  - saturation: represents the strength of vividness of the color
  - luminance: refers to the brightness of a source
  - chrominance: refers to the hue and saturation

What is Steganography?

- Embed information in such a way, its very existence is concealed.
- Goal
  - Hide information in undetectable way both perceptually and statistically.
  - Security, prevent extraction of the hidden information.
- Different concept than cryptography, but use some of its basic principles.

Early steganography

- Pictographs: e.g., Sherlock Holmes's Dancing Men.



"Come Here At Once"

An Example: Null-Cipher

- Message sent by a German spy during World war-I:

PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE. GRAVE SITUATION AFFECTING INTERNATIONAL LAW. STATEMENT FORESHADOWS RUIN OF MANY NEUTRALS. YELLOW JOURNALS UNIFYING NATIONAL EXCITEMENT IMMENSELY.

Null Cipher-Solved!

- Message sent by a German spy during World war-I:

PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE. GRAVE SITUATION AFFECTING INTERNATIONAL LAW. STATEMENT FORESHADOWS RUIN OF MANY NEUTRALS. YELLOW JOURNALS UNIFYING NATIONAL EXCITEMENT IMMENSELY.

Pershing sails from NY June I.

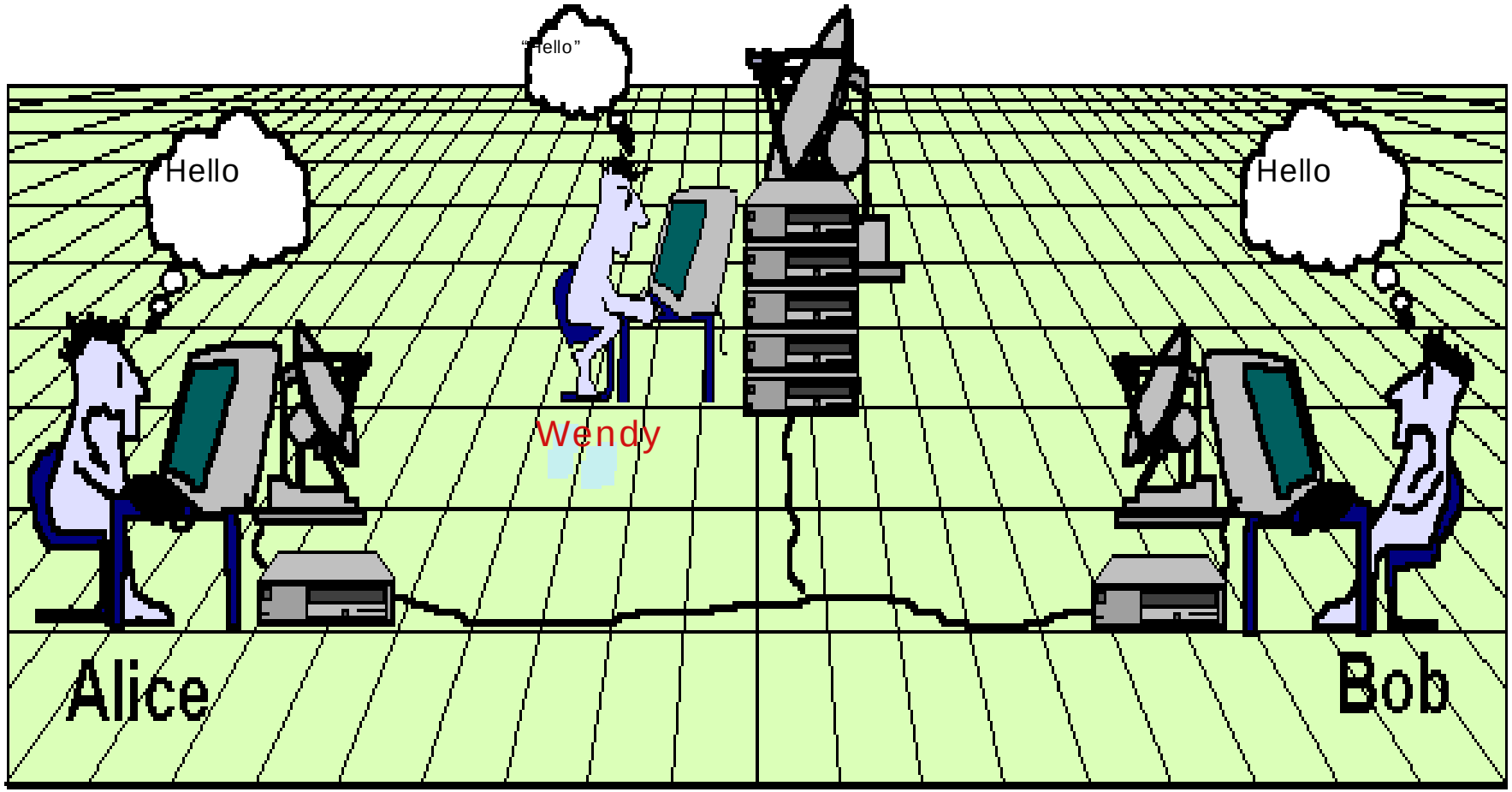
Other Old Ideas

- Pinpricks in maps.
- Tattoos on scalp.
- Dotted I's and crossed T's.
- Hidden Meanings: "Is father dead or deceased?"
- Deliberate Mispellings or Errors, e.g., errors in trivia books, logtables, etc.
- Unusual languages: e.g.,navajo, peculiar sounds used esp., in Guerilla warfare (Chenghez Khan)

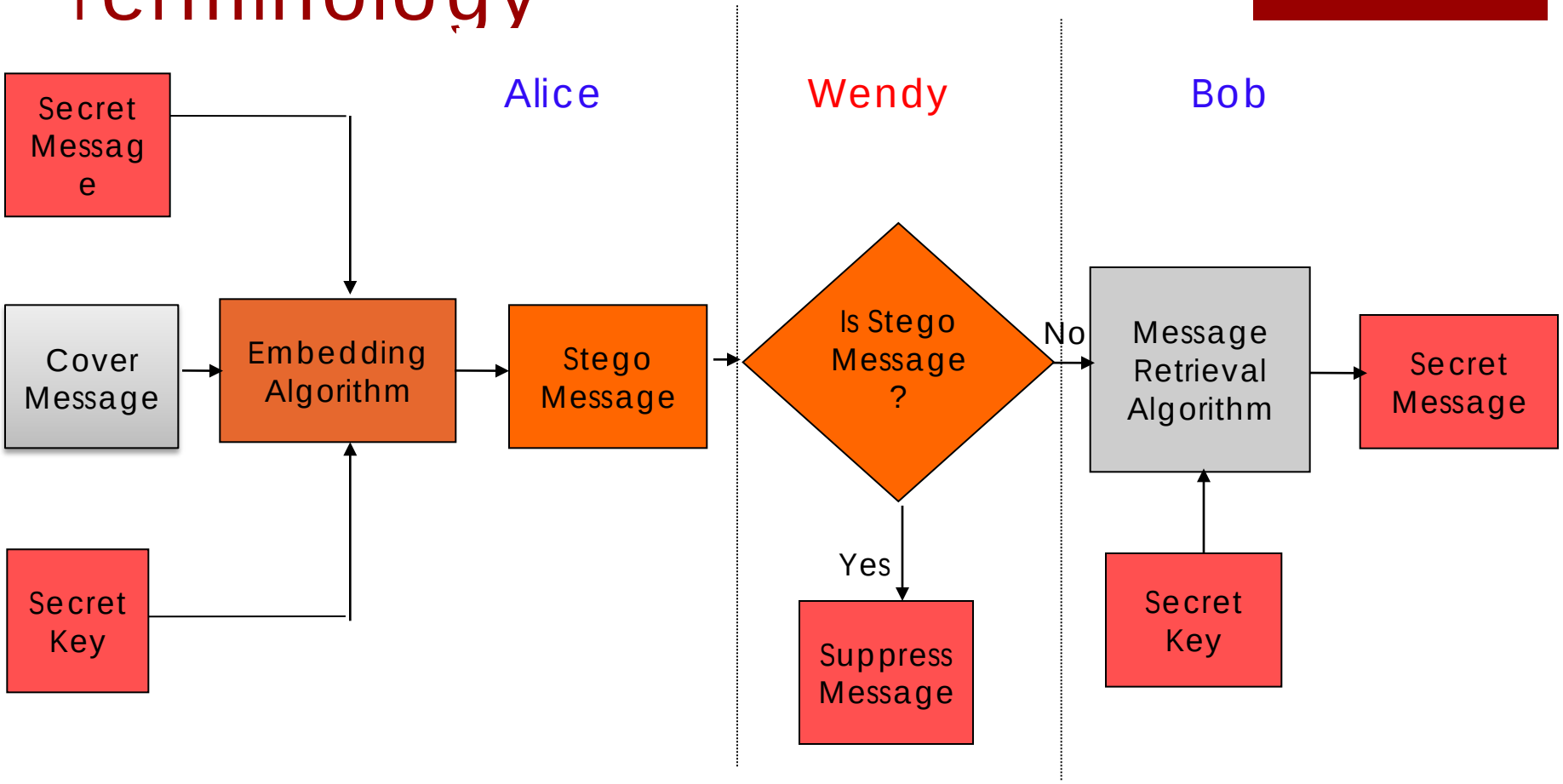
The prisoners problem

- Alice and Bob are in jail and wish to hatch an escape plan.
- Alice's and Bob's communication pass through Willy.
- Alice's and Bob's goal is to hide their ciphertext in innocuous looking way so that Willy will not become suspicious.
- If Willy is a passive warden he will not do any thing to Alice's and Bob's communication.
- If Willy is an active warden he will alter the data being sent between Alice and Bob.

Problem Formulation



Terminology



Steganography Techniques

- Substitution methods
  - Bit plane methods
  - Palette-based methods
- Signal Processing methods
  - Transform methods
  - Spread spectrum techniques
- Coding methods
  - Quantizing, dithering
  - Error correcting codes
- Statistical methods – use hypothesis testing
- Cover generation methods - fractals

Stego-system Criteria

- Cover data should not be significantly modified ie perceptible to human perception system
- The embedded data should be directly encoded in the cover & not in wrapper or header
- Embedded data should be immune to modifications to cover
- Distortion cannot be eliminated so error-correcting codes need to be included whenever required

Places to Hide Information:

- Images
- Audio files
- Text
- Video

We focus on Images as cover media. Though most ideas apply to video and audio as well.

Steganography in Text

- Soft Copy Text
  - Encode data by varying the number of spaces after punctuation
  - Slight modifications of formatted text will be immediately apparent to anyone reading the text
  - Use of White Space (tabs & spaces) is much more effective and less noticeable
  - This is most common method for hiding data in text
- Hard Copy Text
  - Line Shift Coding
    - Shifts every other line up or down slightly in order to encode data
  - Word Shift Coding
    - Shifts some words slightly left or right in order to encode data
- Some methods that can be used with either hard or soft copy text
  - Feature Coding
  - Syntactic
  - Semantic

Steganography in Images

- Way images are stored:
- Array of numbers representing RGB values for each pixel
- Common images are in 8-bit/pixel and 24-bit/pixel format.
- 24-bit images have lot of space for storage but are huge and invite compression
- 8-bits are good options.
- Proper selection of cover image is important.
- Best candidates: gray scale images .
- Cashing on limitations of perception in human vision

Steganography: Bit plane Methods

- Image: replace least significant bit (LSB) of image intensity with message bit
- Replace lowest 3 or 4 LSB with message bits or image data (assume 8 bit values)
- Data is hidden in “noise” of image
- Can hide surprisingly large amounts of data this way
- Very fragile to any image manipulation

Bit plane Methods

- Variations include:
  - Using a permutation of pixel locations at which to hide the bits.
  - Put bits at only certain locations in image where there is “significant” variation and change in gray-value would not be visually perceptible

Least Significant Bit (LSB) Embedding Method

- Consider a 24 bit picture
- Data to be inserted: character '*A*': (10000011)
- Host pixels: 3 pixel will be used to store one character of 8-bits
- The pixels which would be selected for holding the data are chosen on the basis of the key which can be a random number.
- Ex:

00100111 11101001 11001000

00100111 11001000 11101001

11001000 00100111 11101001

Embedding '*A*'

00100111 11101000 11001000

00100110 11001000 11101000

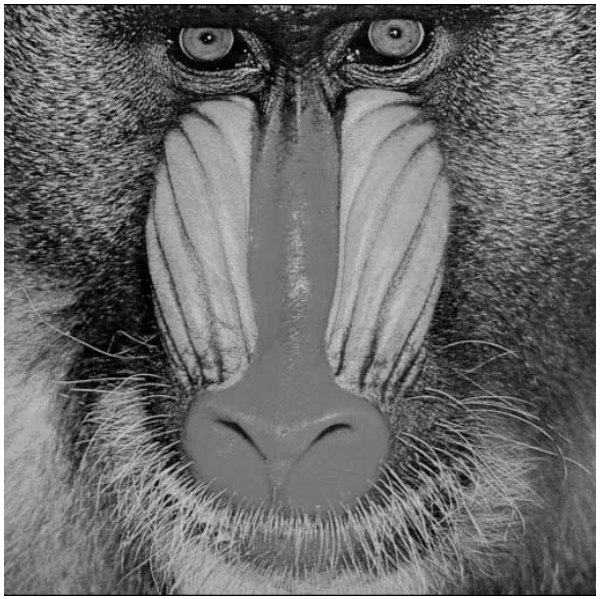
11001001 00100111 11101001

- According to researchers on an average only 50% of the pixels actually change from 0-1 or 1-0.

Example: LSB Encoding

Original Image

Watermark

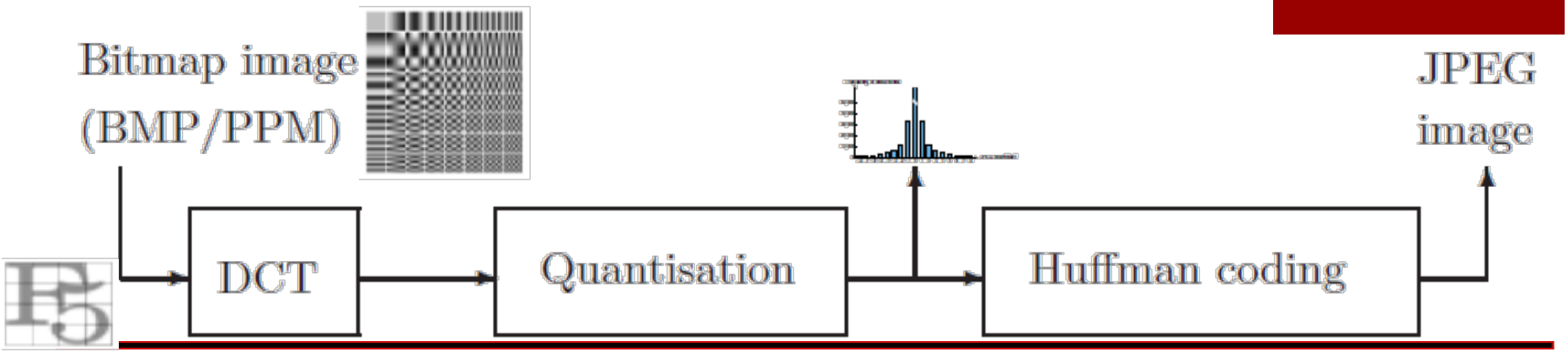


Replace 4 and 7 LSBs of Original

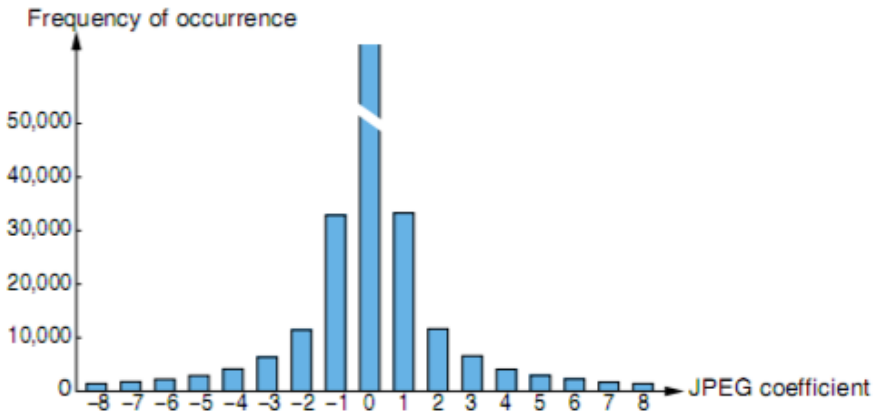
4 LSBs Watermarked

7 LSBs Watermarked

JPEG Mechanism



Histogram for JPEG coefficient after quantization

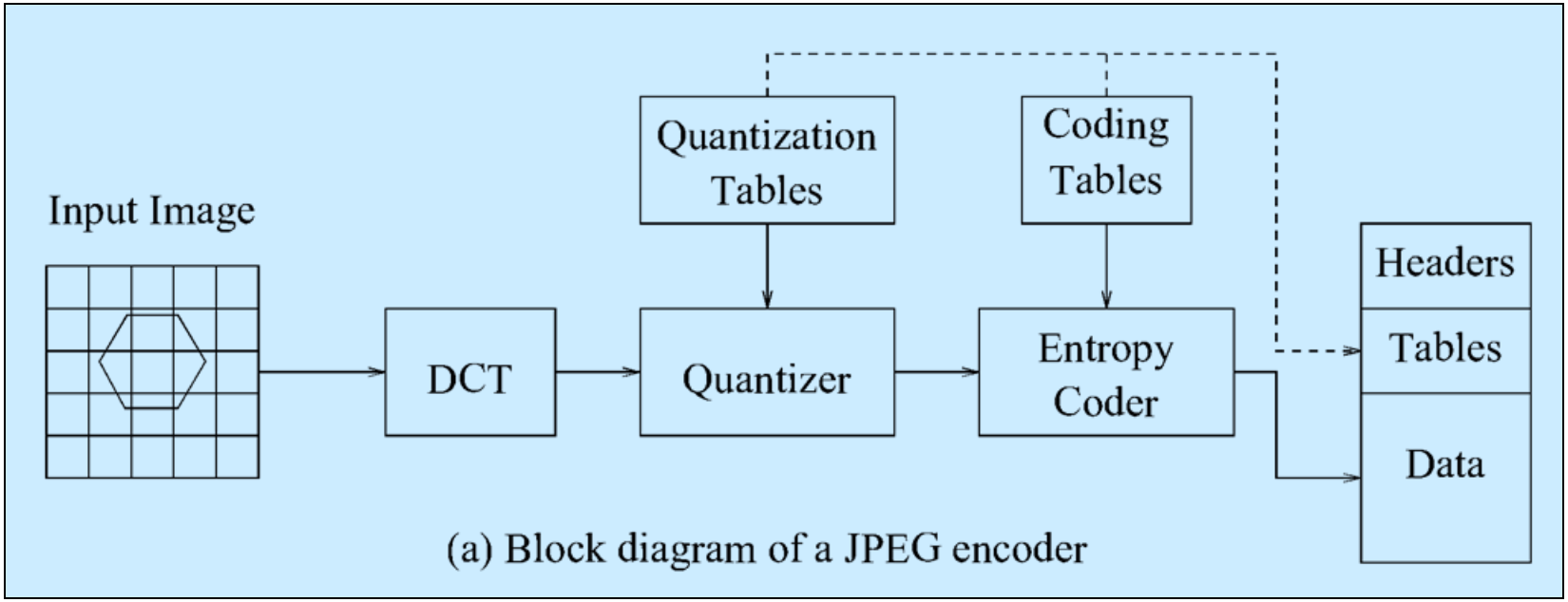


1. The coefficient's frequency of occurrence decreases with increasing absolute value.
2. The difference between two bars in the middle of the histogram is greater than the difference between two bars near the ends.

The JPEG Standard

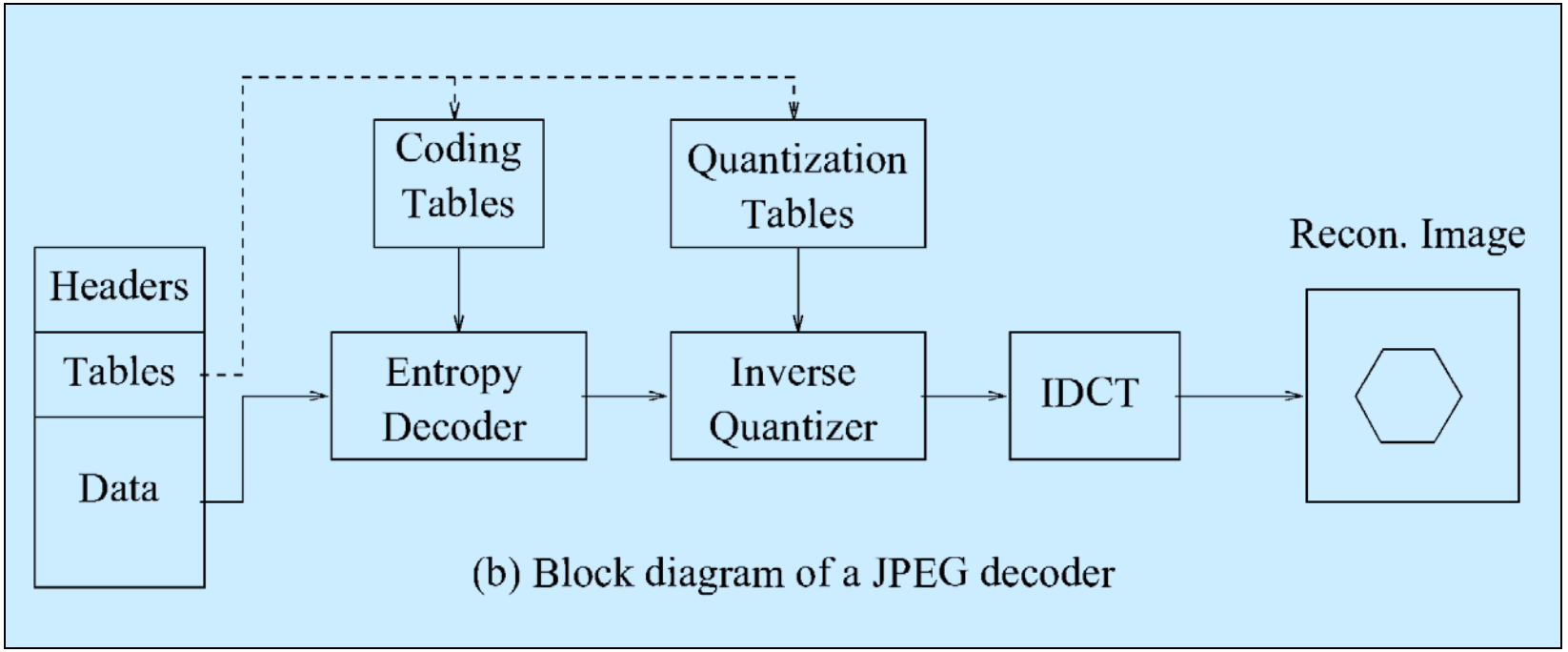
- JPEG: Joint Photographic Experts Group (ISO10918, ITU-T T.81).
- Requirements:
  - Generic still image compression
  - Modest to low software/hardware complexity,
  - Sequential, Progressive and layered coding.
- Features:
  - Psychovisual-based quantization,
  - Sequential, progressive/ hierarchical coding modes,
  - Relatively low memory requirement.

JPEG Encoder



(a) Block diagram of a JPEG encoder

JPEG Decoder



(b) Block diagram of a JPEG decoder

Entropy

- Amount of information I in a symbol of occurring probability ***P*** is defined by:

$$I = -\log_2 P$$

- Entropy or self information of the source is given by:

$$H = -\sum_{i=1}^{n} P_i \log_2 P_i$$

where ***n*** is the number of different symbols in the source stream and P_i is the probability of occurrence of symbol ***i***.

- Entropy denotes the minimum average number of bits that are required to transmit a particular source stream.

- The average number of bits per codeword of the source stream is:

$$\sum_{i=1}^{n} N_i P_i$$

where N_i is the number of bits for the symbol generated by the encoding algorithm

Entropy coding

- Upon digitization, each symbol of the source is assigned a unique binary codeword that is used to represent the entire source over the channel.

- The probability of occurrence of a particular symbol rules the size of the codeword assigned to it in a VLC (variable length coding) scheme. This is termed as entropy coding.

Run Length Coding

- A digital source can possess quite a bit of redundancy within itself. Several methods of compression have been proposed and used to exploit this redundancy of the source.
- Repeated occurrence of the same character is called a run and number of repetition is called the length of the run
- Some of examples of string that can be encoded by using run length coding
  - Eeeeeee7tnnnnnnnn…
  - @e77t@@@n88…
  - 000001111000111…

Huffman Coding

- Assigns fewer bits to symbols that appear more often and more bits to the symbols that appear less often.
- Efficient when occurrence probabilities vary widely.
- Huffman codebook from the set of symbols and their occurring probabilities.
- Two properties:
  - generates compact codes
  - prefix property

Huffman Coding Example

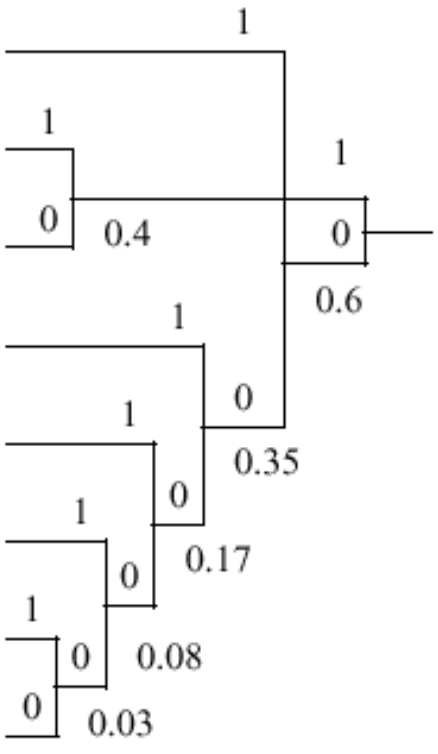
- Suppose we want to encode a source of N =8 symbols: {a,b,c,d,e,f,g,h}.
- The probabilities of these symbols are: P(a) = 0.01, P(b)=0.02, P(c)=0.05, P(d)=0.09, P(e)=0.18, P(f)=0.2, P(g)=0.2, P(h)=0.25
- If we assign 3 bits per symbol (N =23=8), the average length of the symbols is: 3
- The minimum average length we could ever achieve is equal to the entropy (according to Shannon's theorem):

$$H = -\sum_{i=1}^{8} P_i \log_2 P_i = 2.582$$

- Hence, we see a saving of 0.418 bits/symbol

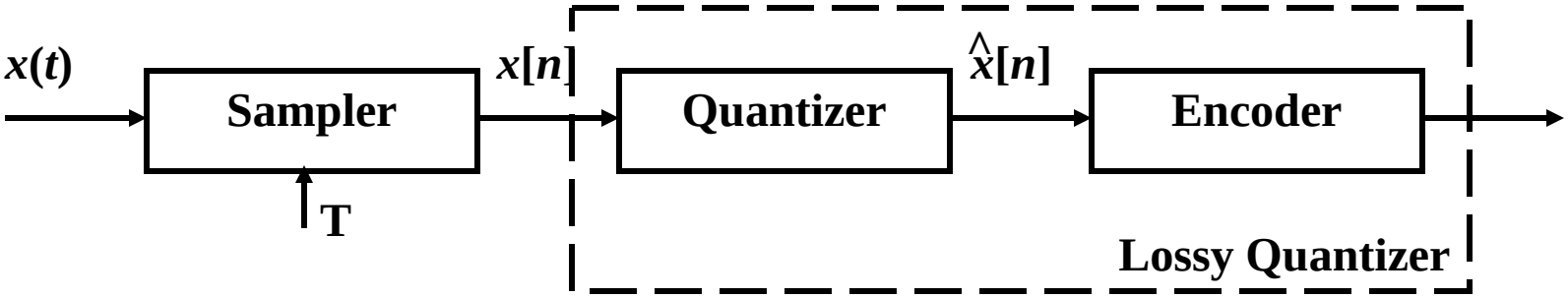
Huffman Coding Example

Original codewords	Huffman codewords	
111	01	P(h)=0.25
110	11	P(g)=0.2
101	10	P(f)=0.2
100	001	P(e)=0.18
011	0001	P(d)=0.09
010	00001	P(c)=0.05
001	000001	P(b)=0.02
000	000000	P(a)=0.01



Lossy Compression

- Coding continuous-time signals:



- $x[n]$ is usually modeled as a sequence of continuous-amplitude random variables.

Linear Transforms

- 1-D linear transform:

$$y[k] = \sum_{n=0}^{N-1} x[n] a_k[n], \quad k = 0, 1, ..., N-1$$

$$\underline{y} = \underline{A}\underline{x} \qquad \underline{x} = \underline{A}^{-1}\underline{y}$$

- Desired properties of a transform:
  - Invertible.
  - Energy preserving: $\sum_{n=0}^{N-1} x^2[n] = \sum_{k=0}^{N-1} |y[k]|^2$
  - Decorrelation: transform coefficients should be mostly un-correlated.
  - Energy compacting: the energy of all transform coefficients should be concentrated in a few elements

2-D Transforms

- 2-D transforms are in the form of

$$y[k_1,k_2]=\sum_{n=0}^{N-1}x[n_1,n_2]a_{k_1,k_2}[n_1,n_2],\quad k_1,k_2=0,1,\ldots,N-1$$

- Most useful 2-D transforms are separable:

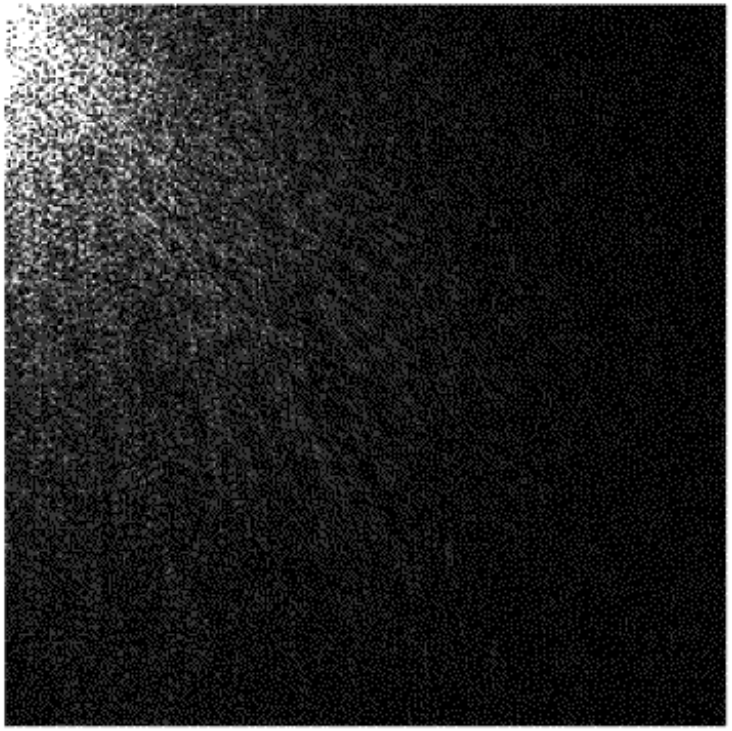
$$\boldsymbol{a_{k_1,k_2}[n_1,n_2]=a_{k_1}[n_1]a_{k_2}[n_2]}$$

$$\underline{\boldsymbol{y}}=\underline{\boldsymbol{A}}\,\underline{\boldsymbol{x}}\,\underline{\boldsymbol{A}}^{\mathrm{T}}$$

- These 2-D transform can be obtained by applying the corresponding 1-D transform to each of the rows of a 2-D signal, and then to each of the resulting columns.

2-D Transform: Example

Original Image



2-D transform (note how energy is compacted in top left corner)

The Discrete Cosine Transform

- The 1-D discrete cosine transform (DCT) is defined as:

$$y[k] = \alpha[k] \sum_{n=0}^{N-1} x[n] \cos\left[\frac{\pi(2n+1)k}{2N}\right] \quad 0 \le k \le N-1$$

$$\text{where} \quad \alpha[0] = \sqrt{\frac{1}{N}} \qquad \alpha[k] = \sqrt{\frac{2}{N}} \quad 1 \le k \le N-1$$

- DCT belongs to the family of sinusoidal transforms. Its basis functions are sinusoidal waveforms.
- It can be computed from the DFT. Therefore it has fast algorithm derived from the FFT.

The DCT Properties

- Properties of the DCT:
  - It is data independent.
  - It has a near optimal data decorrelation capability. Its performance is especially good for first order Markov models with correlation coefficient p>0.9.
  - The DCT transform coefficients and the basis functions have frequency-domain interpretations similar to the DFT. Therefore each DCT coefficient represent a certain frequency component in the original signal.
  - Fast transform algorithm exists, O(Nlog2N).
  - Used in most image/video coding standards, e.g. JPEG, MPEG, Motion-JPEG, etc.

2-D DCT: Example (8x8)

$$X[k_1,k_2]=\frac{1}{4}\sum_{n_1=0}^{7}\sum_{n_2=0}^{7}C_{k_1}C_{k_2}x[n_1,n_2]\cos[\frac{(2n_1+1)k_1\pi}{16}]\cos[\frac{(2n_2+1)k_2\pi}{16}]$$

$$x[n_1,n_2]=\frac{1}{4}\sum_{k_1=0}^{7}\sum_{k_2=0}^{7}C_{k_1}C_{k_2}X[k_1,k_2]\cos[\frac{(2n_1+1)k_1\pi}{16}]\cos[\frac{(2n_2+1)k_2\pi}{16}]$$

$$C_{k_1},C_{k_2}=\begin{cases}\frac{1}{\sqrt{2}} & \text{for } k_1,k_2=0\\ 1 & \text{otherwise}\end{cases}$$

79	75	79	82	82	86	94	94
76	78	76	82	83	86	85	94
72	75	67	78	80	78	74	82
74	76	75	75	86	80	81	79
73	70	75	67	78	78	79	85
69	63	68	69	75	78	82	80
76	76	71	71	67	79	80	83
72	77	78	69	75	75	78	78

2-D DCT (8x8) →

619	-29	8	2	1	-3	0	1
22	-6	-4	0	7	0	-2	-3
11	0	5	-4	-3	4	0	-3
2	-10	5	0	0	7	3	2
6	2	-1	-1	-3	0	0	8
1	2	1	2	0	2	-2	-2
-8	-2	-4	1	2	1	-1	1
-3	1	5	-2	1	-1	1	-3

Input Data

- Input of the coding process: 8x8 block, 8 bits/pixel.
- Example:

$$\begin{bmatrix} 175 & 172 & 172 & 173 & 169 & 174 & 171 & 167 \\ 171 & 170 & 170 & 174 & 177 & 175 & 171 & 166 \\ 167 & 168 & 172 & 174 & 169 & 172 & 174 & 161 \\ 149 & 149 & 156 & 153 & 148 & 149 & 155 & 141 \\ 101 & 106 & 105 & 107 & 99 & 92 & 106 & 97 \\ 73 & 74 & 72 & 71 & 65 & 70 & 79 & 72 \\ 72 & 71 & 67 & 62 & 62 & 72 & 80 & 71 \\ 68 & 70 & 64 & 58 & 62 & 73 & 74 & 67 \end{bmatrix}$$

Block 8x8 2D-DCT of the Image

- 8x8 DCT: Computationally, the most demanding stage.
- Example:

$$\begin{bmatrix} 31144 & 52 & 35 & 86 & -294 & 264 & -137 & 14 \\ 8039 & 100 & -258 & 11 & 108 & 10 & 38 & -12 \\ -387 & -89 & 117 & 156 & -12 & -103 & 90 & -52 \\ -1778 & 47 & 96 & -78 & 59 & 10 & -31 & -19 \\ 164 & 80 & -7 & -43 & -8 & 0 & -6 & 12 \\ 551 & -17 & 54 & 18 & -67 & 64 & 47 & -3 \\ -152 & 4 & 6 & -16 & -30 & 24 & 15 & -34 \\ -83 & 19 & 12 & -37 & 6 & 15 & -46 & -6 \end{bmatrix}$$

Quantization of DCT coefficients

- Quantization is performed using a quantization table (the Q-table).
- Example:

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

The JPEG Standard cont.

- Let the 8x8 DCT coefficients X [k1, k2] be the input of the quantization, and Q [k1, k2] be the Q-table, the weighted DCT coefficients becomes
- Y [k1, k2] = X [k1, k2]/Q [k1,k2].
- The output of the quantization stage is
- $\hat{Y}$ [k1, k2] = round(Y [k1, k2]).
- A Q-table is designed to minimize the mean square error or maximize the visual quality.
- For a specific coding process, the prototype Q-table can be scaled with a factor in (1~100) to achieve different quality and compression ratios.

The JPEG Standard cont.

- The DCT coefficients are weighted by the Q-table.
- Example:

$$\begin{bmatrix} 1946.5 & 4.7 & 3.5 & 5.4 & -12.3 & 6.6 & -2.7 & 0.2 \\ 669.9 & 8.3 & -18.1 & 0.6 & 4.2 & 0.2 & 0.6 & -0.2 \\ -27.7 & -6.9 & 7.3 & 6.5 & -0.3 & -1.8 & 1.3 & -0.9 \\ -127.0 & 2.8 & 4.4 & -2.7 & 1.2 & 0.1 & -0.4 & -0.3 \\ 9.1 & 3.6 & -0.2 & -0.8 & -0.1 & 0 & -0.1 & 0.2 \\ 23.0 & -0.5 & 1.0 & 0.3 & -0.8 & 0.6 & 0.4 & 0 \\ -3.1 & 0.1 & 0.1 & -0.2 & -0.3 & 0.2 & 0.1 & -0.3 \\ -1.2 & 0.2 & 0.1 & -0.4 & 0.1 & 0.1 & -0.4 & -0.1 \end{bmatrix}$$

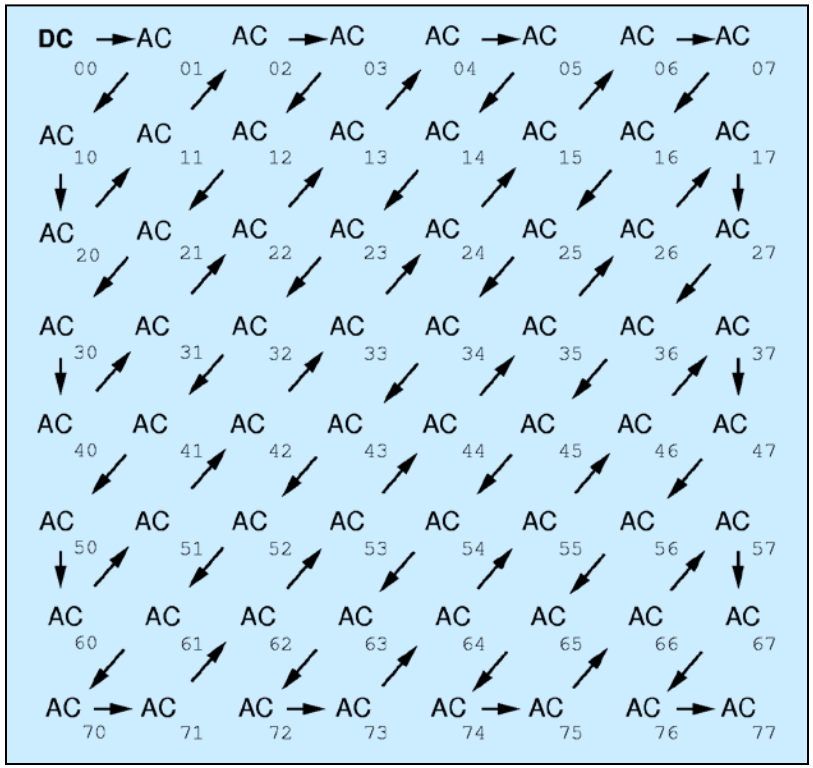
The JPEG Standard cont.

- The weight DCT coefficients are rounded to integers. The resulting block is then coded through the entropy coding.
- Example:

$$\begin{bmatrix} 1947 & 5 & 4 & 5 & -12 & 7 & -3 & 0 \\ 670 & 8 & -18 & 1 & 4 & 0 & 1 & 0 \\ -28 & -7 & 7 & 7 & 0 & -2 & 1 & -1 \\ -127 & 3 & 4 & -3 & 1 & 0 & 0 & 0 \\ 9 & 4 & 0 & -1 & 0 & 0 & 0 & 0 \\ 23 & 0 & 1 & 0 & -1 & 1 & 0 & 0 \\ -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

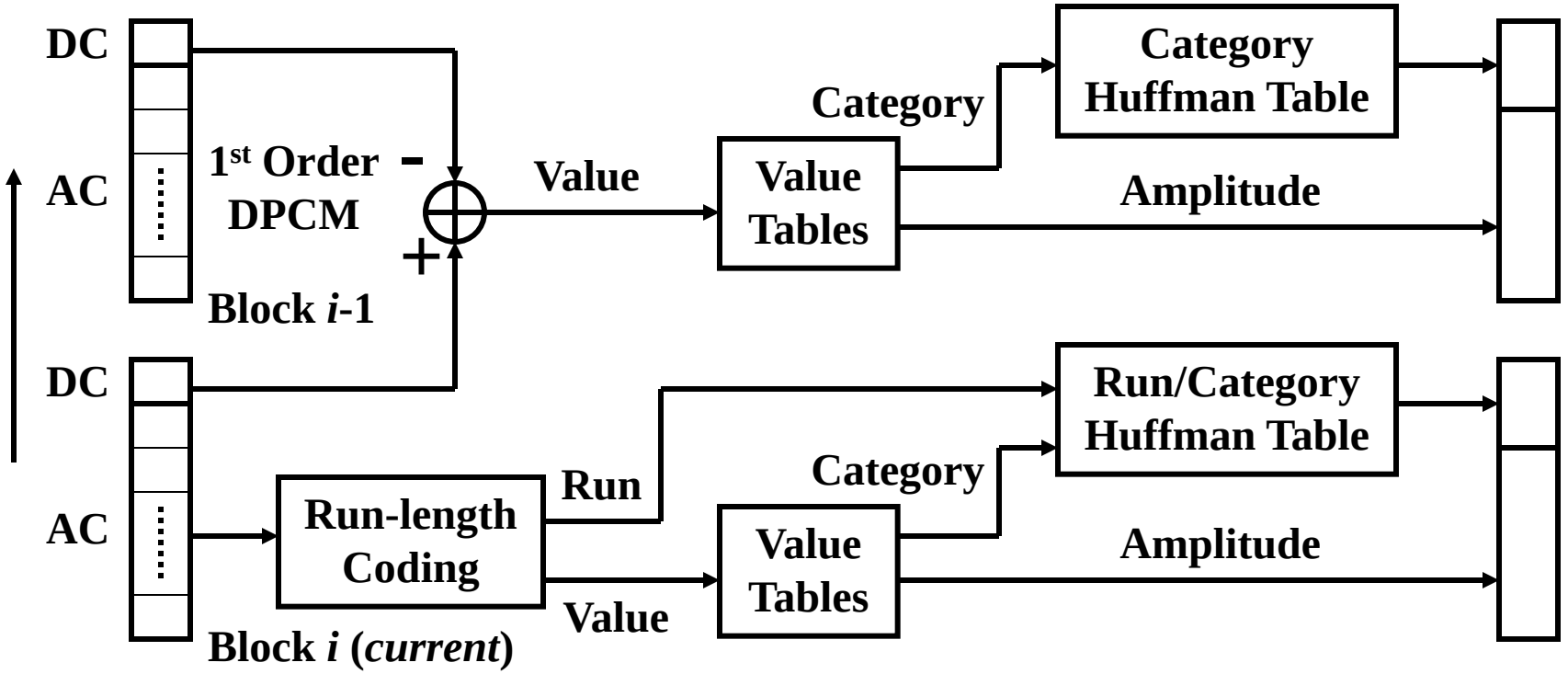
The JPEG Standard cont.

- Each quantized block is scanned through a zig-zag pattern and converted into 1-D data streams.



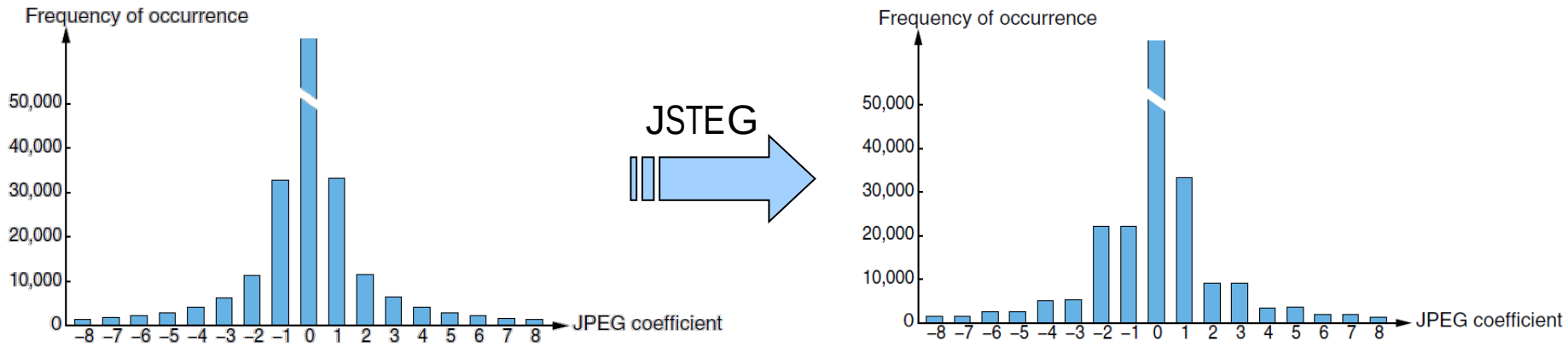
The JPEG Standard-Summary

- The DC and AC bit streams are coded through Huffman coding together with DPCM and run-length coding.



JSTEG Algorithm

- After quantization, Jsteg replaces (over writes) the least significant bits (LSB) of the frequency coefficient by the secret message. The embedding mechanism skips all coefficient with the value 0 and 1.
- Resistant against the visual attacks and good capacity with 12.8 % of the steganogram's size, but the secret message is easily detected by statistical attacks. (chi-square test)
- Jsteg influences pairs of the coefficients frequency of occurrence !!!

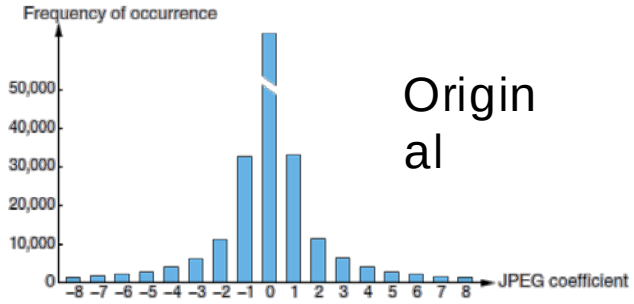


F3 Algorithm

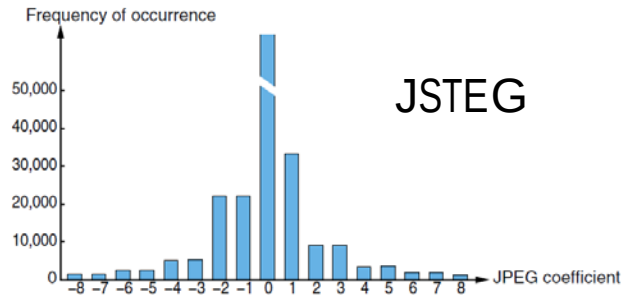
1. Does not overwrite bits -- Decrement the non-zero coefficient's absolute value only if the LSB does not match. Zero coefficients are skipped.
2. The LSB of a non-zero coefficient will match the secret message after embedding.

Advantage: statistical attack (chi-square test) will not be successful

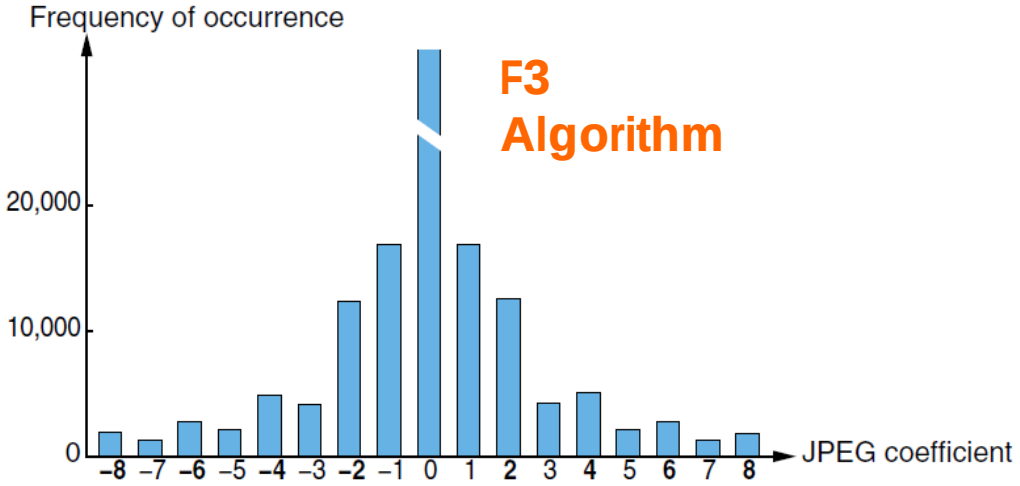
Disadvantage: Less capacity and surplus of even coefficients caused by shrinking.
The surplus of even coefficients can be detected by statistical means.



Original



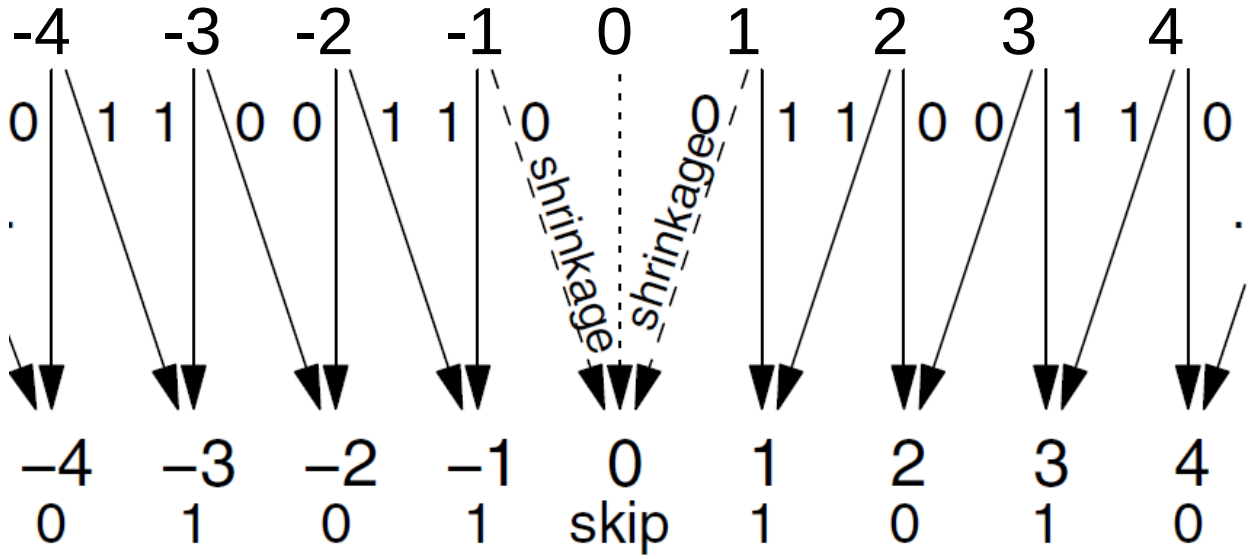
JSTEG



F3 Algorithm

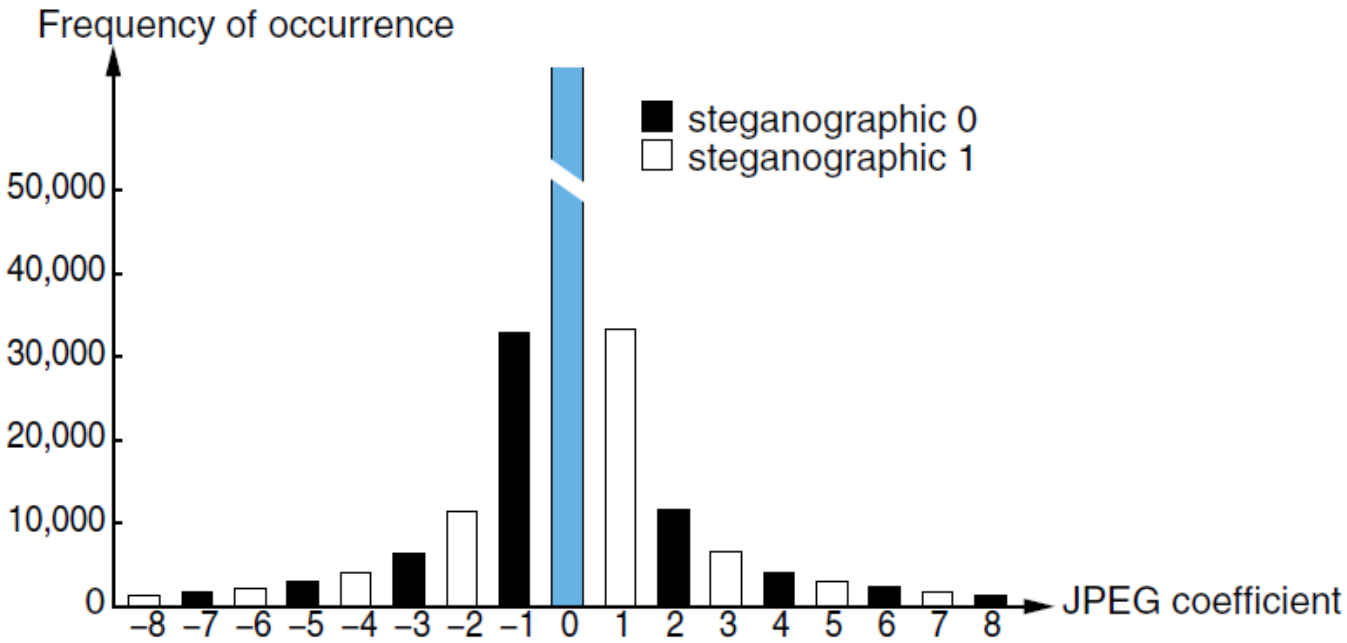
F3 Shrinkage

- Shrinkage of coefficients causes a decrease in embedding capacity.
- Since the receiver cannot tell between a skipped zero and a zero that was generated due to shrinkage, repetitive embedding is necessary.



F4 Algorithm

- Mapping negative coefficients to the inverted steganographic value.
- Even negative coefficients and odd positive coefficients represent a steganographic one.
- Even positive coefficients and odd negative coefficients represent a steganographic zero.



F4 embedding example

- Embed the code “01110”
- If LSB and message does not match,
  - Increment negative Coefficients
  - Decrement positive coefficients

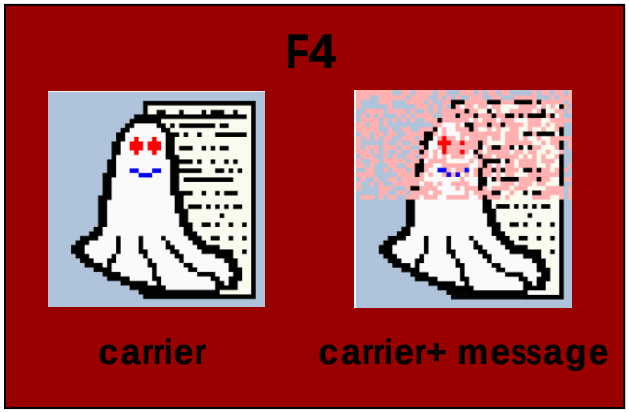
Original Coefficients	5	0	0	2	3	-1	0	-3	0	1	-3
Binary of Coefficients	0101	0000	0000	0010	0011	1111	0000	1101	0000	0001	1101
Inverted LSB Binary (Negative Coefficients)						1110		1100			1100
Message	0	skip	skip	1	1	1	skip	1	skip	0	0
Operation	5-1	nothing	nothing	2-1	same	-1+1	nothing	-3+1	nothing	1-1	same
Final Result	**4**	**0**	**0**	**1**	**3**	**0**	**0**	**-2**	**0**	**0**	**-3**
						shrinkage				**shrinkage**	

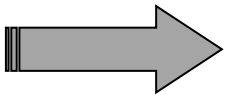
F5 Algorithm

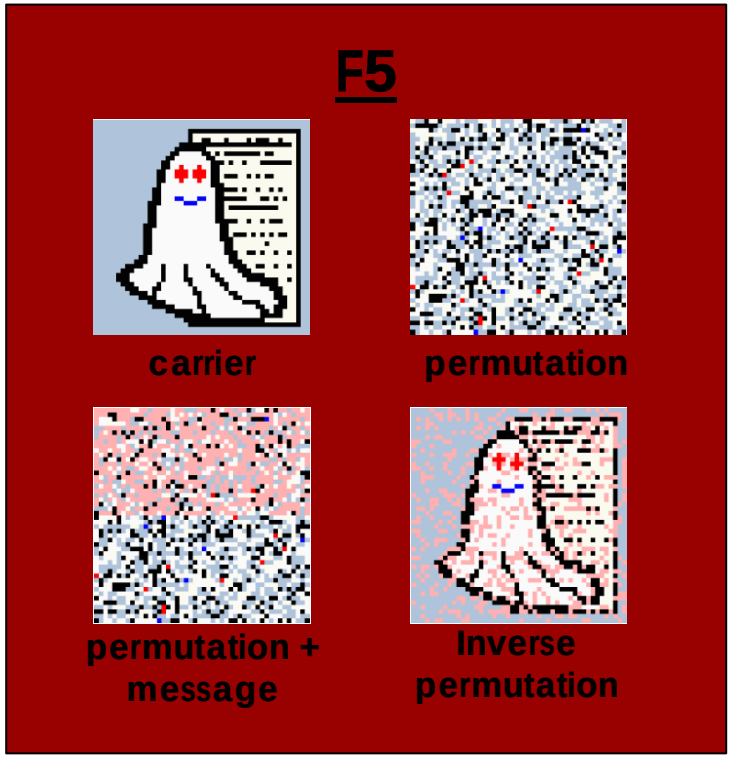
- Overall algorithm the same as F4.
- Extends F4 by adding two distinct features:
  - Permutative straddling
  - Matrix encoding

Permutative Straddling

- F4 embeds the data into the next available non-zero coefficient.
- F5 will scatter the entire message throughout the carrier.
- Uses permutation to equalize the spread of embedded data.







Note: treat each pixel as if it was a JPEG coefficient.

Matrix Encoding

Improves the embedding efficiency from 1.5 bit to 3.8 bit per change.

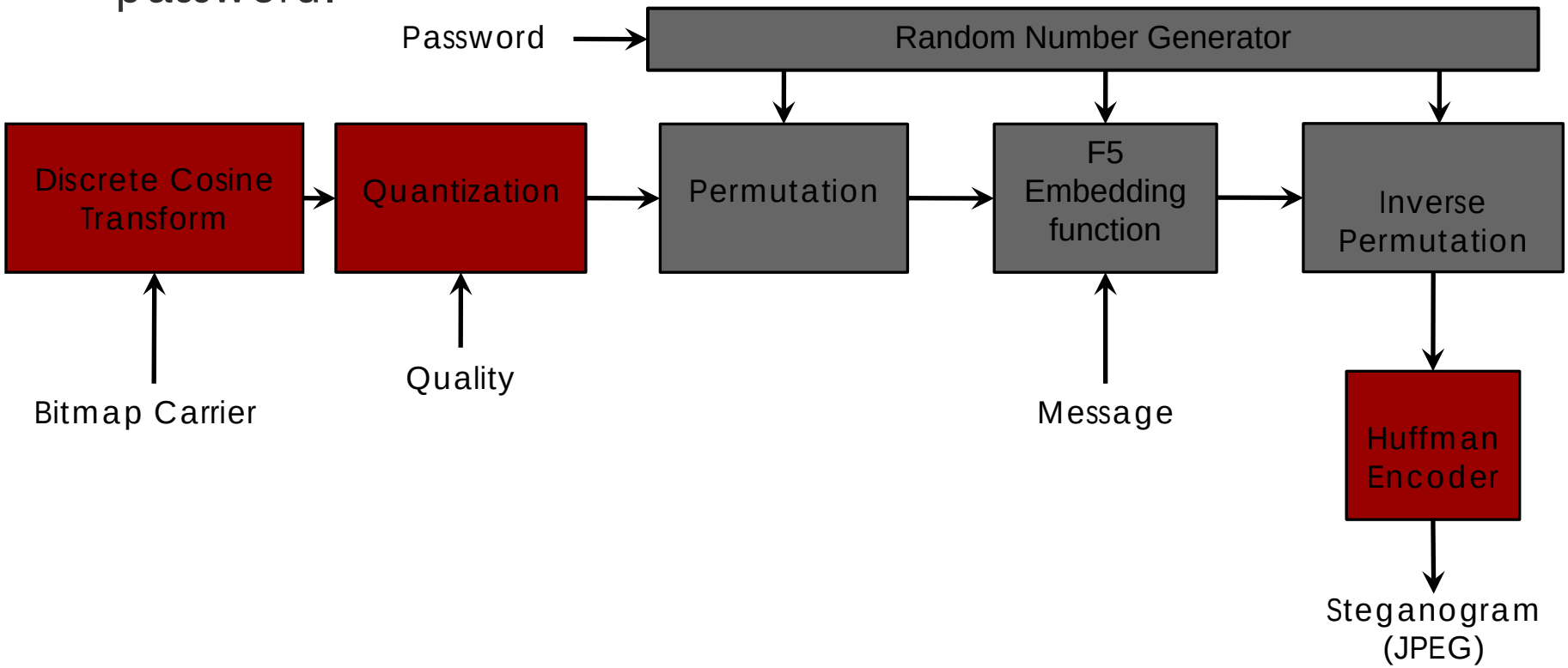
How does it work?

Consider we want to embed x_1 and x_2 in LSB locations a_1, a_2, and a_3.

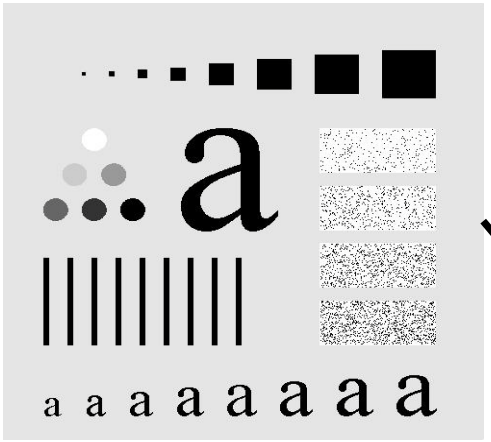
$$x_1 = a_1 \oplus a_3,\ x_2 = a_2 \oplus a_3 \Rightarrow \text{change nothing}$$
$$x_1 \neq a_1 \oplus a_3,\ x_2 = a_2 \oplus a_3 \Rightarrow \text{change } a_1$$
$$x_1 = a_1 \oplus a_3,\ x_2 \neq a_2 \oplus a_3 \Rightarrow \text{change } a_2$$
$$x_1 \neq a_1 \oplus a_3,\ x_2 \neq a_2 \oplus a_3 \Rightarrow \text{change } a_3.$$

F5 encoding process

- Permutation is generated using user-defined password.



F5 example

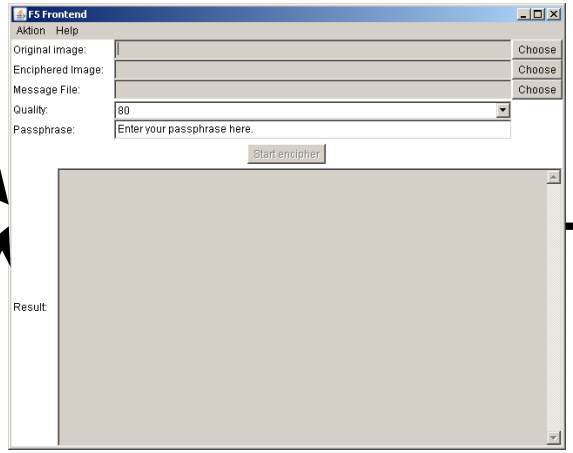


Carrier Image

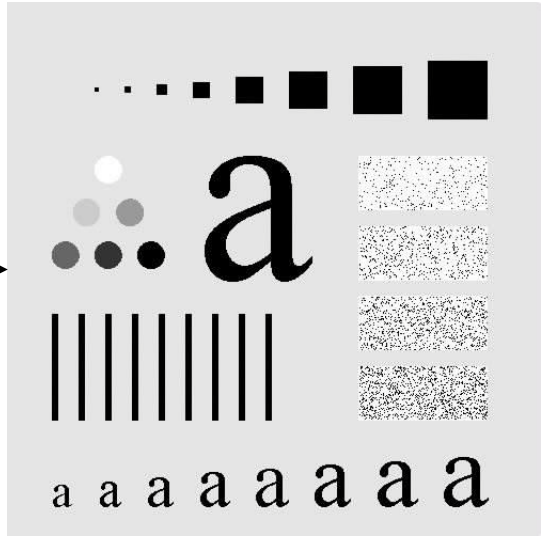
ECE -643-001
Dr. Yun Shi
Digital Image Processing

F5 Presentation
Davang Patel
Thomas Schulze

Secret Message



F5 Encrypt/Decrypt Program



Resultant Steganogr am