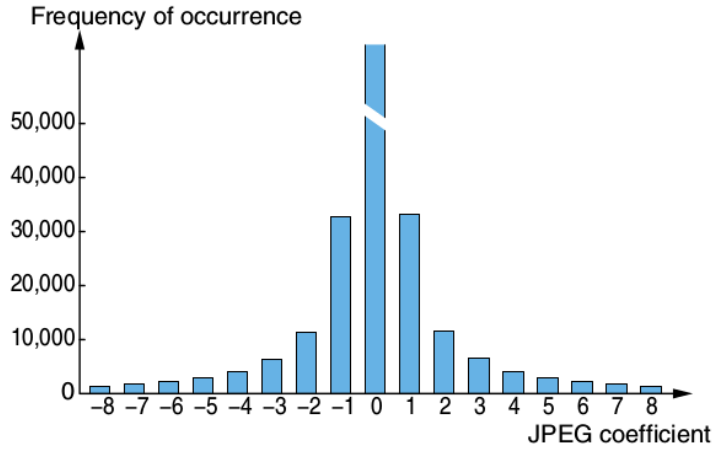
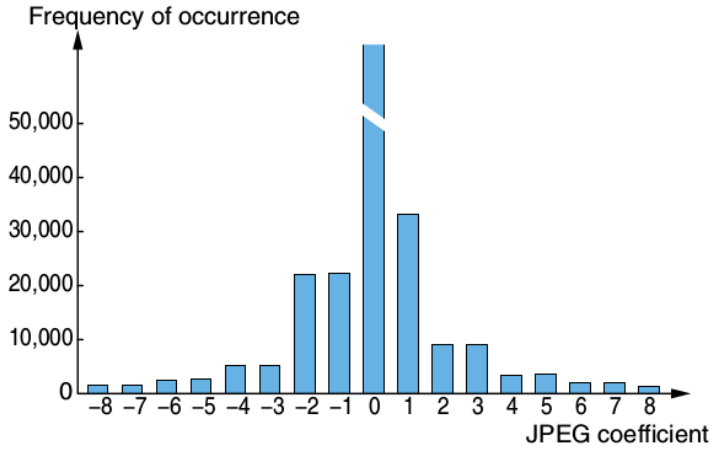
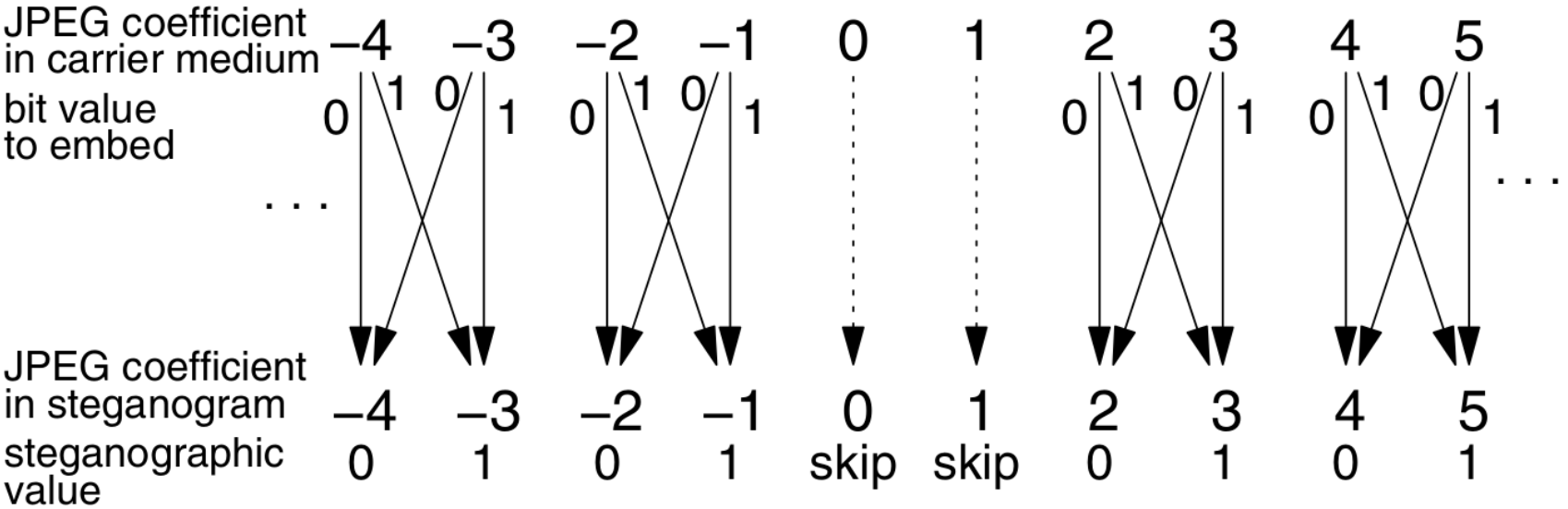
Masking Embedding as Natural Processing

- Preserving statistics
  - Losing capacity.
- Mimicking some natural process
  - F3, F4, F5, ...

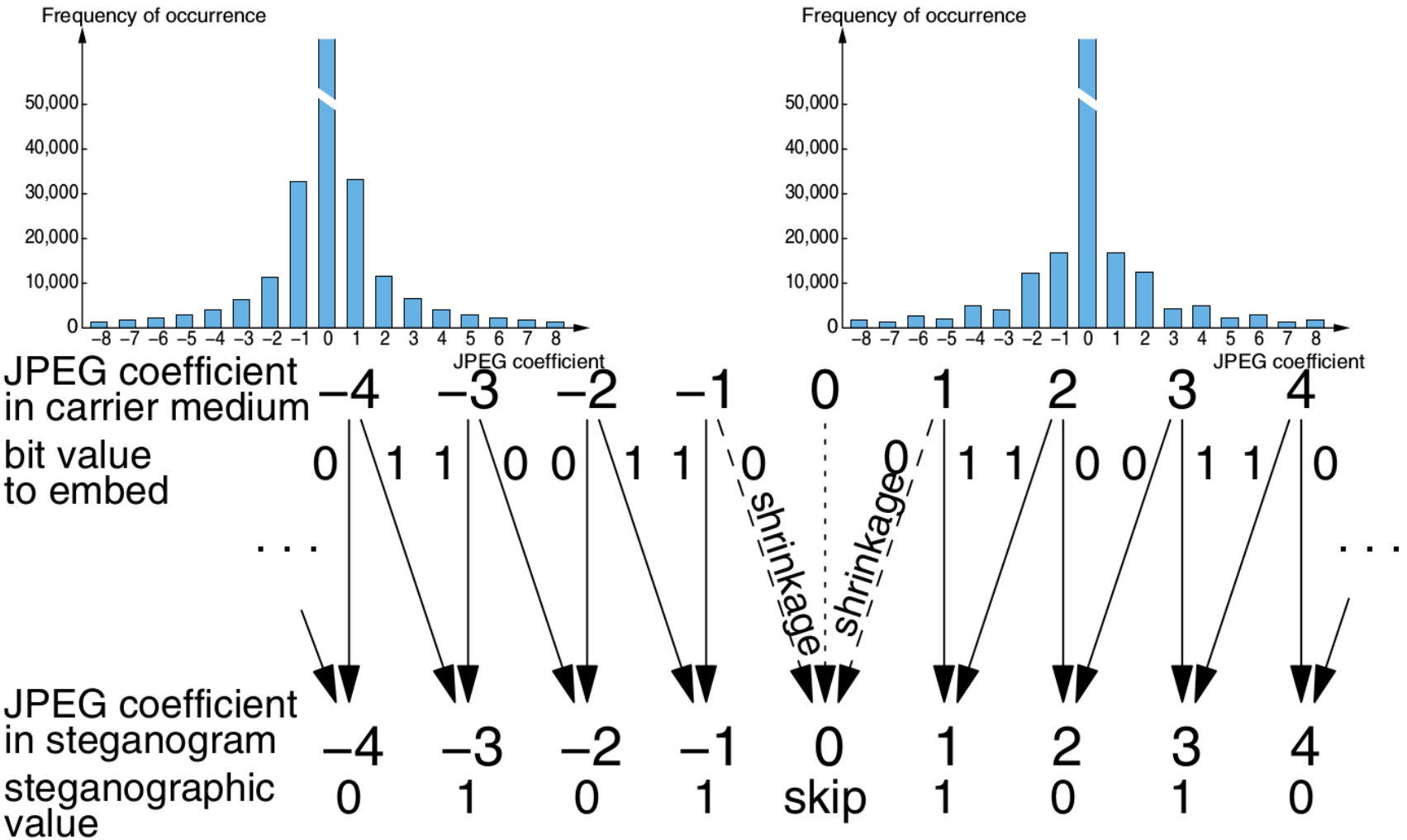
Jsteg


Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient

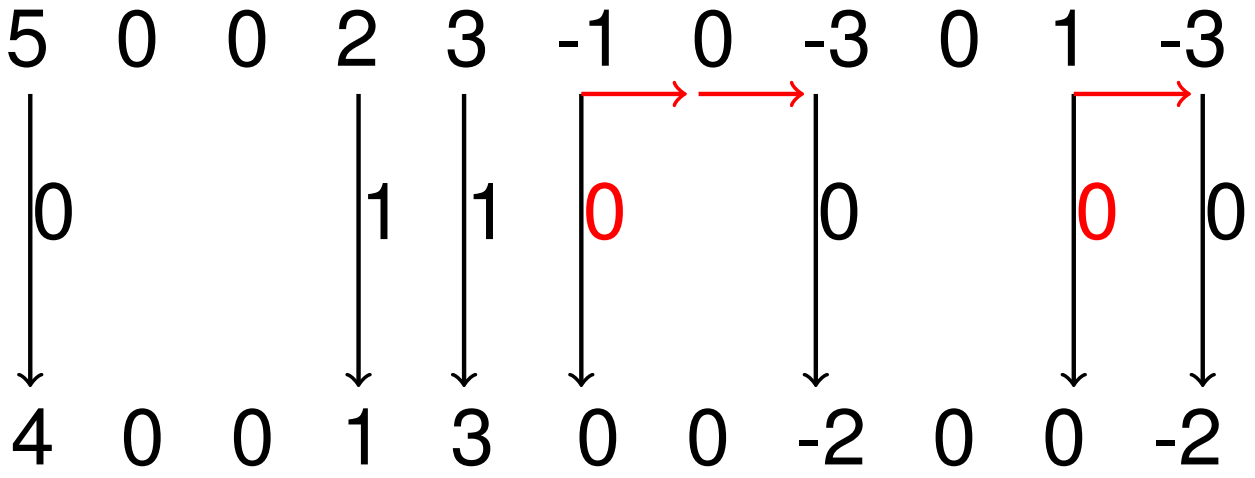

Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient


JPEG coefficient in carrier medium
bit value to embed
JPEG coefficient in steganogram
steganographic value
−4 −3 −2 −1 0 1 2 3 4 5
0 1 0 1
. . .
−4 −3 −2 −1 0 1 2 3 4 5
0 1 0 1 skip skip 0 1 0 1

F3


Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
JPEG coefficient in carrier medium
−4 −3 −2 −1 0 1 2 3 4
bit value to embed
0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
. . .
. . .
shrinkage
shrinkage
JPEG coefficient in steganogram
−4 −3 −2 −1 0 1 2 3 4
steganographic value
0 1 0 1 skip 1 0 1 0

F3 Algorithm



Embedding 01100.

What Is the Problem in F3?

In normal work

- Decreasing

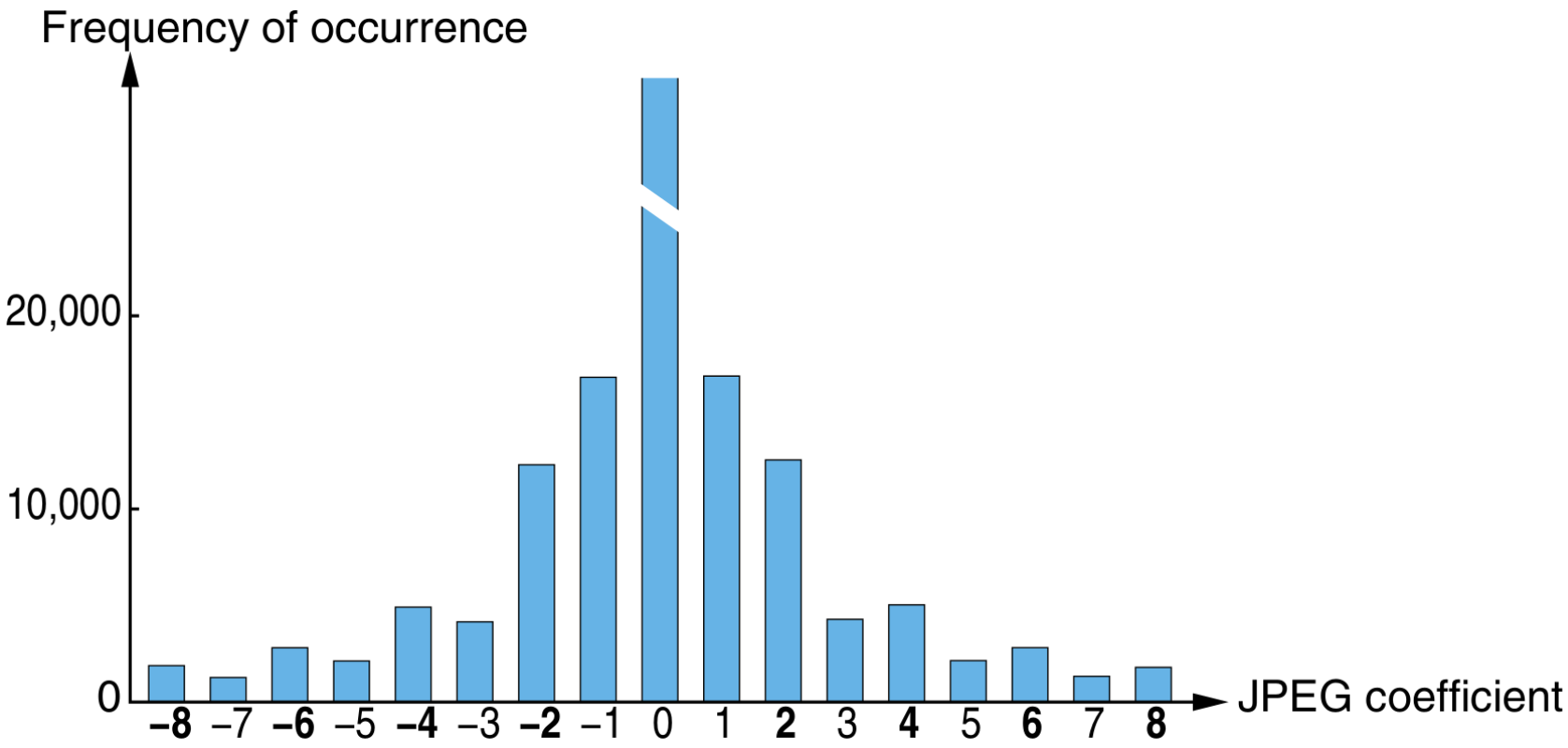
$$P(2i-1) > P(2i).$$

In Steganographic work

- More on even.

$$P(2i-1) < P(2i).$$

Defects of F3

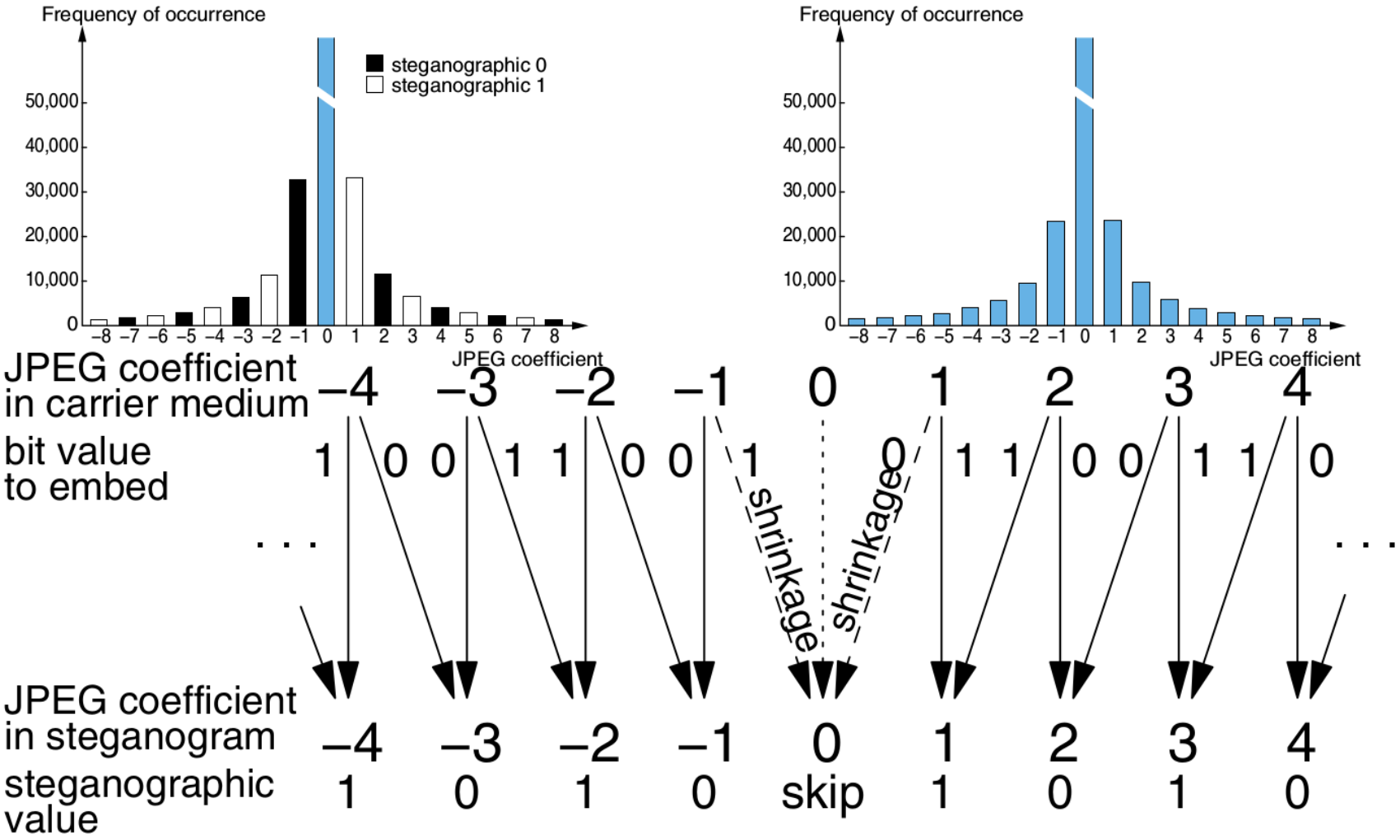


Reason

Repeated embedding after shrinkage.

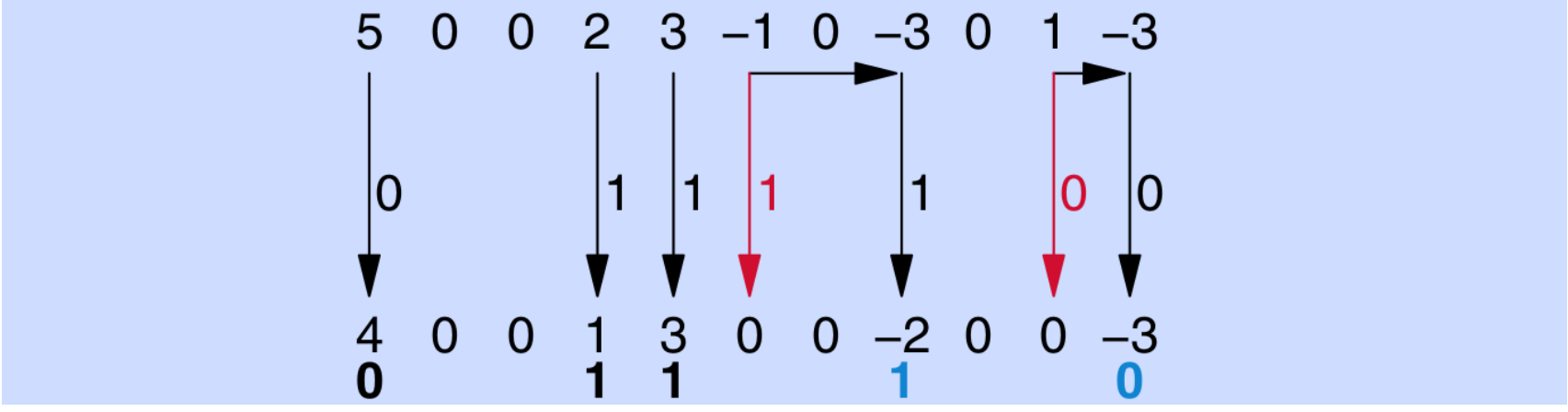
- Happens for embedding 0 only.
- Equivalent to add more 0 into the message code.

F4

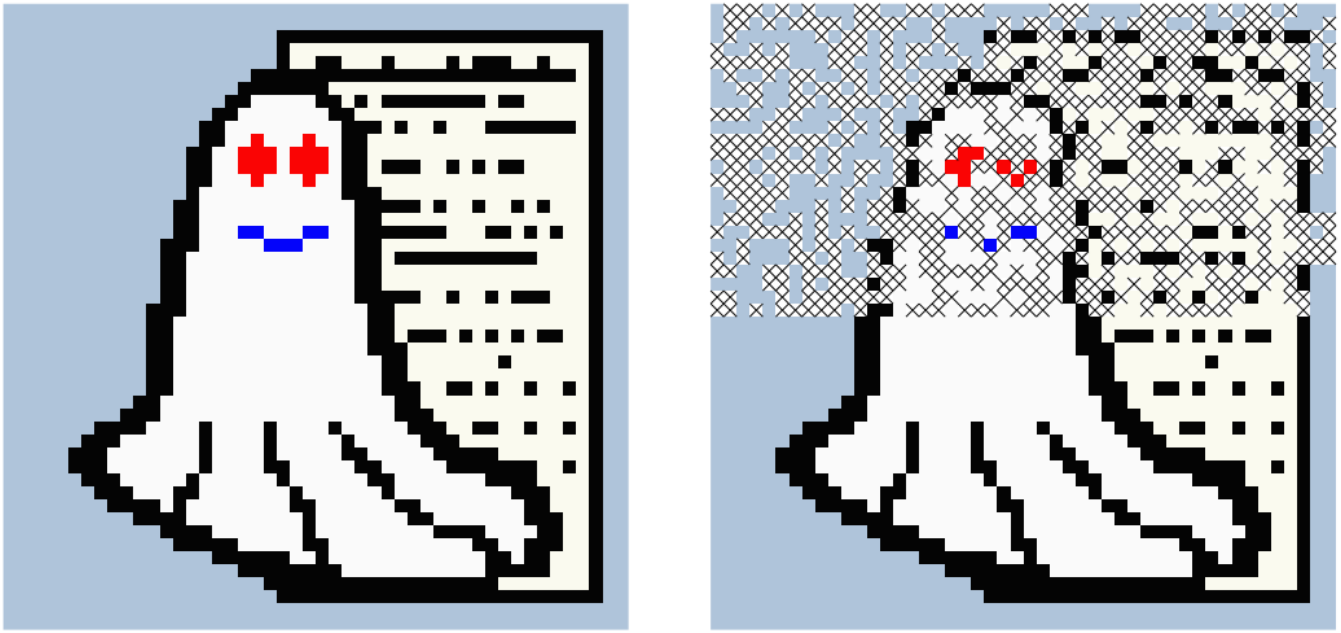

Frequency of occurrence
steganographic 0
steganographic 1
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
Frequency of occurrence
50,000
40,000
30,000
20,000
10,000
0
−8 −7 −6 −5 −4 −3 −2 −1 0 1 2 3 4 5 6 7 8
JPEG coefficient
JPEG coefficient in carrier medium
−4 −3 −2 −1 0 1 2 3 4
bit value to embed
1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0
shrinkage
shrinkage
JPEG coefficient in steganogram
−4 −3 −2 −1 0 1 2 3 4
steganographic value
1 0 1 0 skip 1 0 1 0

F4 Algorithm

- Steganographic interpretation
  - Positive coefficients: LSB
  - Negative coefficients: inverted LSB
- Skip 0, adjust coefficients to message bit
  - Decrement positive coefficients
  - Increment negative coefficients
  - Repeat if shrinkage occurs

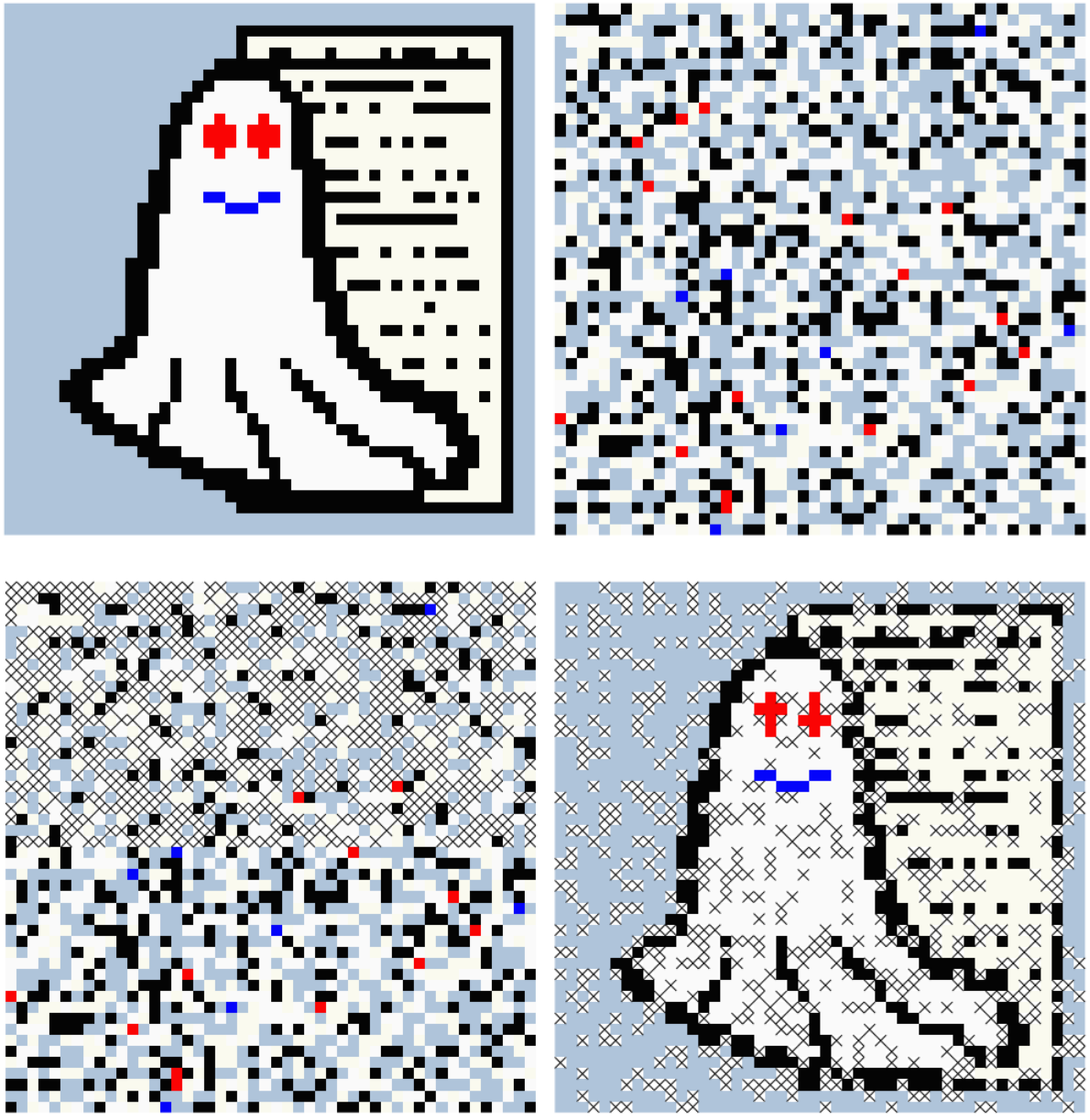


F4 Defects



Compare similar blocks or reverse fitting GCD.

Random Walk



More Efficiency?

Example: Embedding 1736 bits

- F4: 1157 changes.
- F5: 459 changes by matrix encoding.
  - Embedding efficiency: 3.8 bits per change.

Matrix Encoding

Embedding b_1, b_2 to x_1, x_2, x_3 with at most 1 change.

$$b_1 = LSB(x_1) \text{ XOR } LSB(x_2)$$
$$b_2 = LSB(x_2) \text{ XOR } LSB(x_3)$$

- Four equal probability cases.
- Change x_i accordingly.

Example

$$b_1 = LSB(x_1) \text{ XOR } LSB(x_2)$$
$$b_2 = LSB(x_2) \text{ XOR } LSB(x_3)$$

0,0	1,0	0,1	1,1
/	$\bar{x}_1$	$\bar{x}_3$	$\bar{x}_2$

Efficiency:

$$2/(3/4) = 8/3 > 2.$$

A Hamming Code

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

Parity Check Matrix H

Using coset to indicate message

- Encode 2^p messages in a $2^{(2^p-1)}$ code space.
- Giving a (2^p-1)-bit code c, we extract the p-bit message by

$$m = Hc \tag{1}$$

  where $H \in \{0,1\}^{p\times(2^p-1)}$.
- A message m can be represented by different c in a coset structure.

Embedding Efficiency of Hamming Code

Giving a p-bit message

- With probability $1/2^p$, $m = Hc$, and no change at all.
- With probability $1 - 1/2^p$, $m = H(c + e_i)$
  - e_i has only a 1, i.e. only change c at a bit.
- The average change is:

$$0 \cdot 1/2^p + 1 \cdot (1 - 1/2^p) = 1 - 1/2^p.$$

Efficiency is: $p/(1 - 1/2^p)$

Upper Bound on Embedding Efficiency

For a message set $\mathcal{M}$, in a n-pixel image, what is the minimal number of change R (in the sense of expectation).

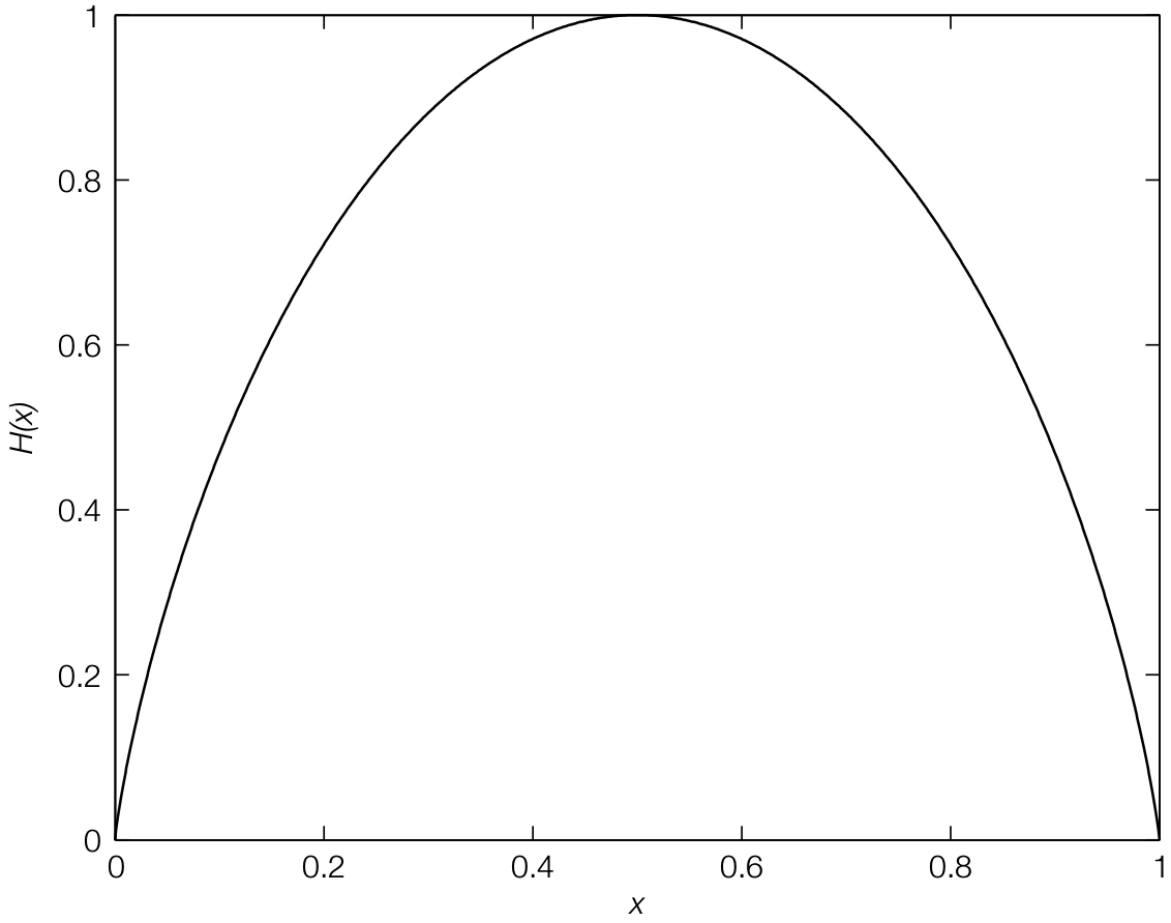
- The bound of $\frac{\log_2 |\mathcal{M}|}{R}$:
  - Larger means better efficiency.
  - The upper bound indicates the optimal situation.

Just Some Math

$$\log_2 |\mathcal{M}| \leq \log_2 \binom{n}{R} 2^R$$
$$\leq nH(R/n) \quad \text{information theory}$$

Binary entropy function

$$H(x) = -x \log_2 x - (1-x) \log_2 (1-x).$$



Continue the Math

$$\alpha = \frac{\log_2 |\mathcal{M}|}{n} \leq H(R/n)$$

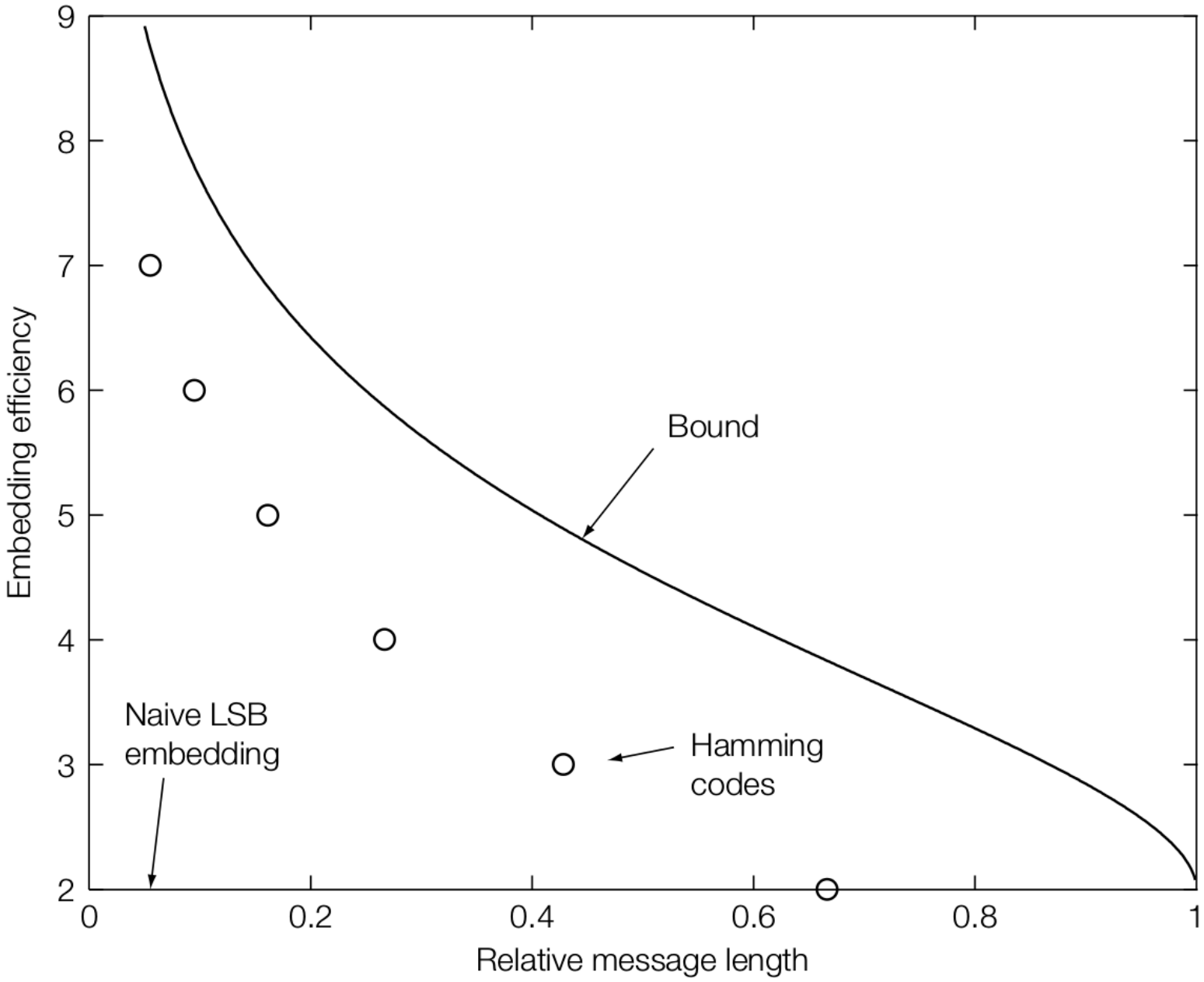
$$\frac{n}{R} \leq 1/H^{-1}(\alpha), \quad H^{-1} \in [0, 0.5]$$

$$\frac{\log_2 |\mathcal{M}|}{R} \frac{n}{\log_2 |\mathcal{M}|} \leq 1/H^{-1}(\alpha)$$

$$e = \frac{\log_2 |\mathcal{M}|}{R} \leq \frac{\alpha}{H^{-1}(\alpha)}.$$

- α: relative message length.
- e embedding efficiency.

Illustration



Selection Rule

Choose the parts/locations to change.

- Known for both side: shared.
- Only known for sender: nonshared.

Nonshared Selection Rule

Motivation:

- In JPEG compress:
  - DCT: float value.
  - Round into integer.
- To minimize the change:
  - Choose values have largest rounding error to change, e.g. 5.47:
    - to embed 0: $5.47 \to 6, +0.53$.
    - to embed 1: $5.47 \to 5, -0.47$.
- More like normal compress procedure, but
  - How recipient detect the message?

Other Cases

- Adaptive steganography
  - If the neighborhood has certain property ...
  - But embedding may change the property.
- Eg. using the pixels with largest neighbor variance.

Presentation: Project 1

- E_BLIND
- D_LC

The key points

- The tips
  - Scaling and shifting of the reference mark etc.
  - Noise from value clipping
    - Use low/high contrast image
- Performance: the plot of detection value
  - Using different reference mark
  - Using different cover work

Project: F3+F4

Steganography system: F3+F4

- Explain the procedure of F3 and F4.
- Show the DCT coefficients histogram before and after modification.
- Difference between Digital watermarking and Steganography.

Project: F5

Steganography system: F5

- At least two different matrix encoding.
- Analysis of their embedding efficiency.
- Impact of random walk.