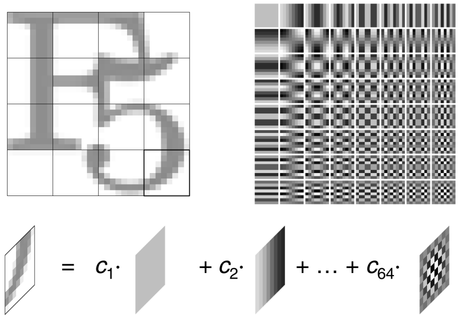
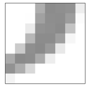
DCT Coefficients

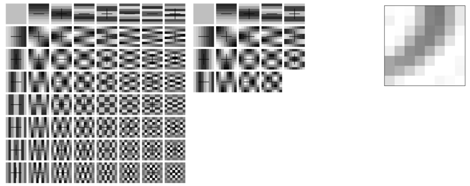


DCT Compression

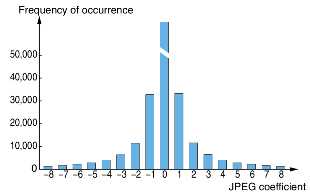


64 brightness values

➽ 19 nonzero JPEG coefficients



DCT Characteristic Properties



P(X=1) > P(X=2) > P(X=3) > P(X=4)

P(X=1)-P(X =2) > P(X =2)-P(X=3) > P(X=3)-P(X=4)

Model-Based Steganography

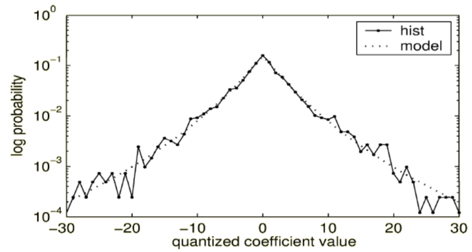
Generalized Cauchy model with probability density function (pdf)

- Generalized Cauchy distribution (GCD):

$$P(x) = \frac{p-1}{2s}\left|\frac{|x|}{s}+1\right|^{-p}.$$

- $p > 1, s > 0$ are the two parameters.

Illustration of GCD



Two-Class Pattern Classification

Two components in a cover work (c_{inv}, c_{emb}):

$$\begin{aligned} p_0 &= P(c_{emb} = 0 | c_{inv} = MSB_7(2i)) \\ &= \frac{T_c[2i]}{T_c[2i] + T_c[2i+1]} \\ &= 1 - P(c_{emb} = 1 | c_{inv} = MSB_7(2i)). \end{aligned}$$

The probability of $2i$ in the bin $(2i, 2i+1)$.

Arithmetic Decompress and Compress

Map a uniformly distributed bitstream to a new bitstream with specific distribution.

Presentation: Arithmetic Coding

- `http://en.wikipedia.org/wiki/Arithmetic_coding`
- `http://www.cs.cmu.edu/~aarti/Class/10704/Intro_Arith_coding.pdf`

Reverse Compression

- In embedding:

  uniformly distributed bitstream

  $\xrightarrow{\text{Decompress}}$

  GCD distributed bitstream

- In detection:

  GCD distributed bitstream

  $\xrightarrow{\text{Compress}}$

  uniformly distributed bitstream

Embedding Efficiency

The average number of embedded bits per unit distortion.

- LSB: $2 = 1/0.5$.
  - 1 bit: for a uniform distribution binary sequence.
  - Change: 50% of chance to change.
  - Efficiency:

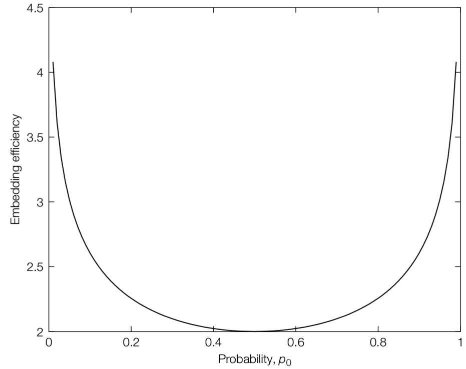
$$\frac{1}{0.5}.$$

Embedding Efficiency

The average number of embedded bits per unit distortion.

- LSB: $2 = 1/0.5$.
- Model Based:
  - Information:
    $$H(p_0) = -p_0 \log_2 p_0 - (1 - p_0) \log(1 - p_0).$$
  - Change:
    $$p_0(1 - p_0) + (1 - p_0)p_0 = 2p_0(1 - p_0).$$
  - Efficiency:
    $$\frac{-p_0 \log_2 p_0 - (1 - p_0) \log(1 - p_0)}{2p_0(1 - p_0)}.$$

Illustration

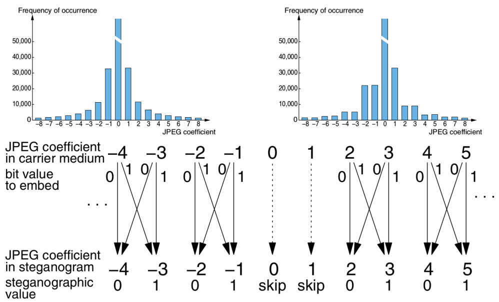


The Cost of Correction

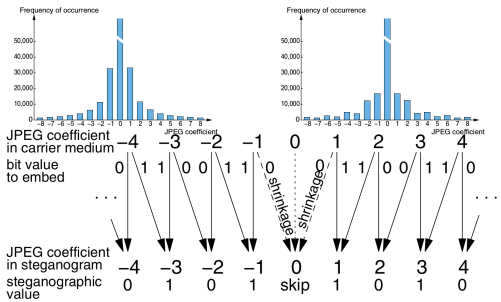
Losing capacity.

- F3, F4, F5, ...

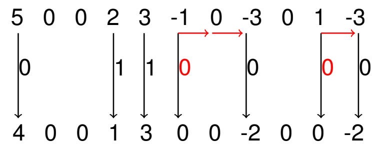
Jsteg



F3



F3 Algorithm



Embedding 01100.

What Is the Problem in F3?

In normal work

- Decreasing

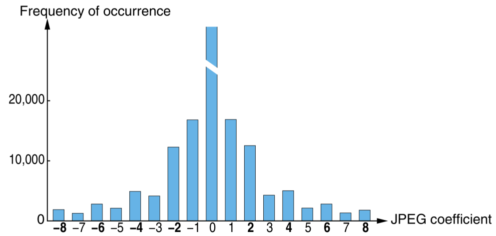
$$P(2i - 1) > P(2i).$$

In Steganographic work

- More on even.

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

Defects of F3

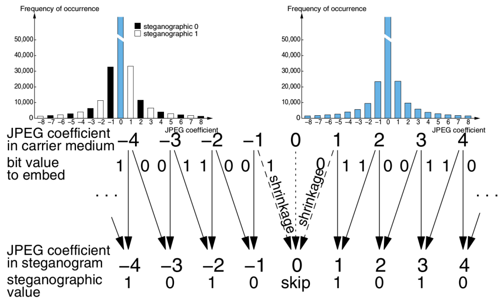


Reason

Repeated embedding after shrinkage.

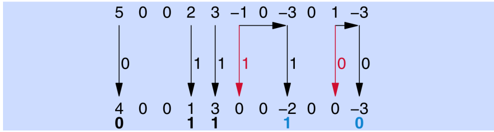
- Happens for embedding 0 only.
- Equivalent to add more 0 into the message code.

F4

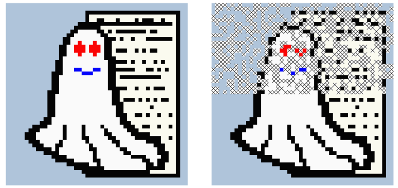


F4 Algorithm

- 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

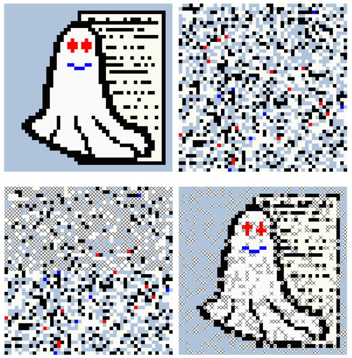


F4 Defects



Compare similar blocks or reverse fitting GCD.

Random Walk



More Payload?

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

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

Presentation: Matrix Embedding

- The idea of parity matrix.
- Efficiency.