

第4讲 信息隐藏与数字水印

主要内容

1. 信息隐藏基本理论

2. 空域/变换域信息隐藏技术

3. 数字水印

4. 应用与发展

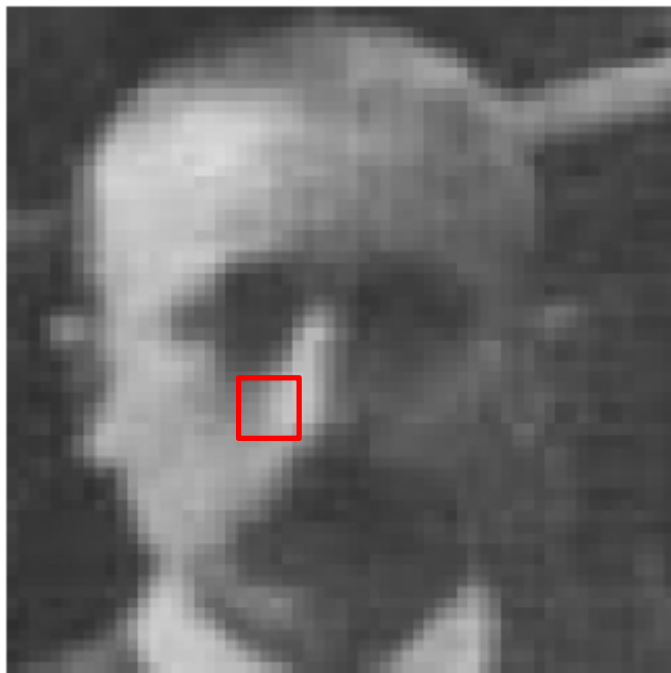
2.1 空域信息隐藏技术

空域隐藏技术是指在图像、视频、音频等载体的空间域上进行信息隐藏。通过直接改变宿主媒体的某些像素值或采样值来嵌入数据。

空域信息隐藏技术无需对原始媒体进行变换，**计算简单，效率较高**，但由于水印要均衡不可感知性和鲁棒性，因而可选择的属性范围较小。此外，难以抵抗常见信号处理的攻击及噪声干扰的影响，**鲁棒性较差**。

Pixel domain

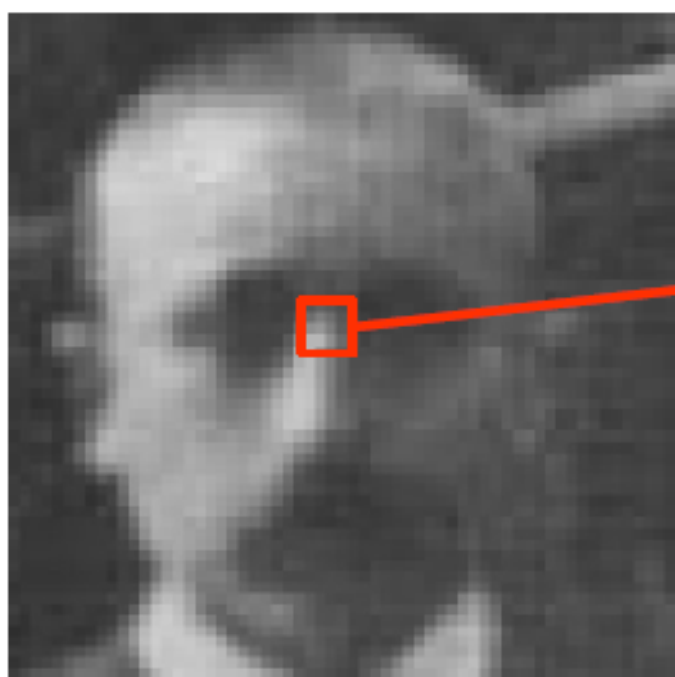
- The stego-message is hidden in the array of integer numbers a digital image consists of



100 102 104 156 157 190 201 201
100 102 130 120 123 191 199 199
103 105 127 118 125 190 190 188
110 112 112 116 123 131 190 189
101 102 106 102 120 130 191 199
101 104 107 109 134 135 199 220

Frequency domain

- In some cases, for instance with JPEG images, the stego message is hidden into the (block) DCT coefficients of the images



-16	90	37	-17	-1	-2	-2	-1
63	10	-46	-14	12	0	0	2
-2	-9	-5	12	4	-5	-2	1
1	-3	-2	0	-3	-1	1	1
0	-2	-1	-1	0	1	1	-1
0	0	0	0	-1	0	0	0
0	-1	0	0	0	0	0	0
0	-1	0	1	0	0	0	0

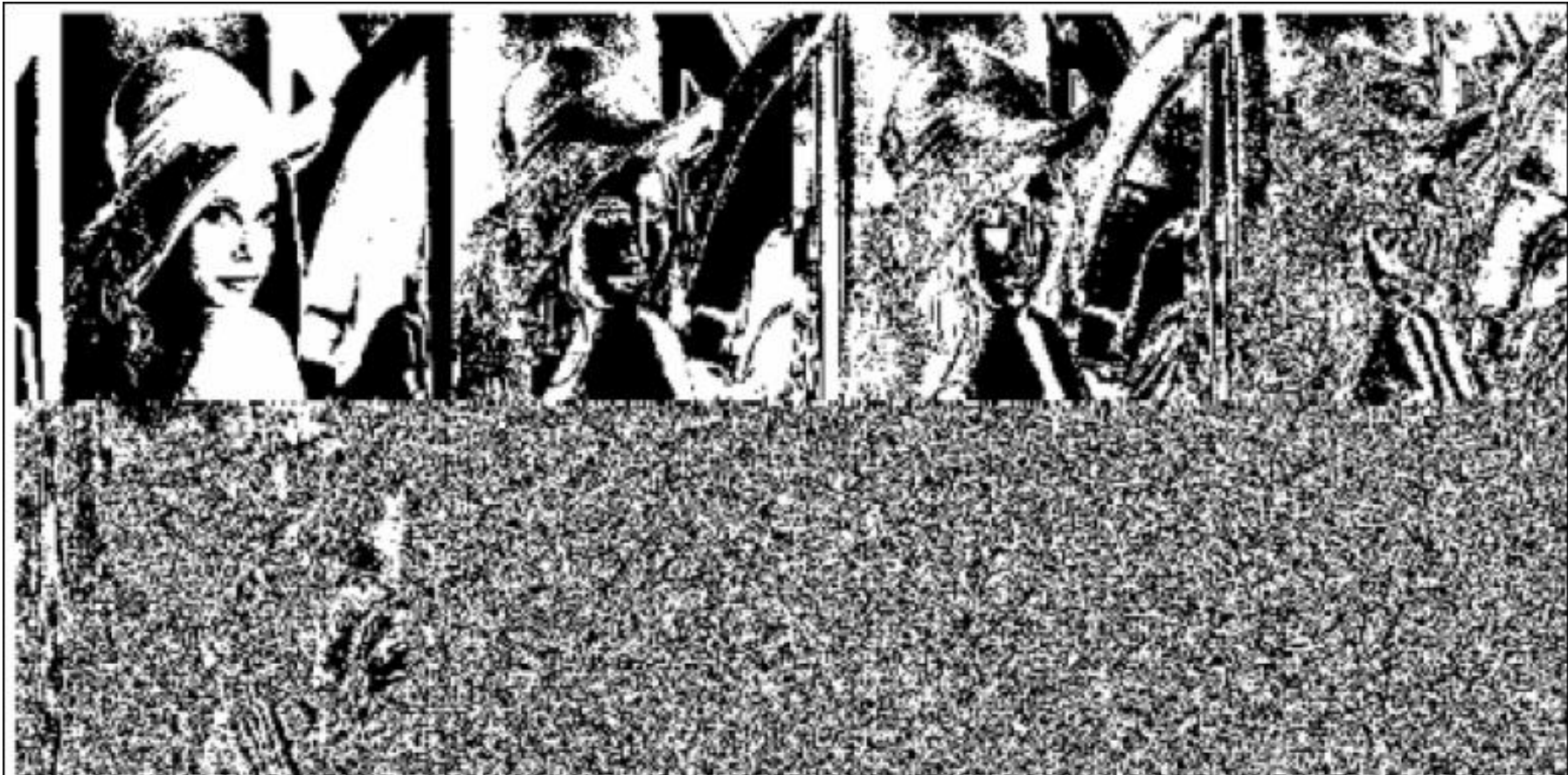
典型算法①

基于替换LSB的空域信息隐藏技术

位平面

◆ 图像的位平面表示

8级灰度图像，取值0~255



LSB 信息隐藏算法

1. 直接替换LSB方法 (Least Significant Bit)

- 核心思想：用秘密信息取代LSB
- 实现方法：
 - If 信息和载体LSB相同，则不作修改；
 - If 信息和载体LSB不同，修改如下：
 - 原始灰度值为奇数，则减1
 - 原始灰度值为偶数，则加1

◆ 基于替换LSB的空域信息隐藏

例4-1: 设待隐藏信息为1001，取灰度图像的4个像素值(0-255整数)的最低位进行隐藏。

隐藏前8位灰度值	二进制表示	隐藏后二进制	隐藏后八位灰度值
34	0010001 0	0010001 1	35
180	1011010 0	1011010 0	180
255	1111111 1	1111111 0	254
2	0000001 0	0000001 1	3

LSB 信息隐藏算法

➤ 嵌入位置的选取

- ◆ 连续嵌入法（载体的所有位置都使用）

 - 秘密信息 + 伪随机序列

 - 重复嵌入

- ◆ 随机间隔法

 - 利用伪随机数发生器

LSB 信息隐藏算法

2. 基于奇偶校验位的LSB方法

- ◆ 将载体分成不重叠的区域，在每个区域中嵌入1比特信息 **区域 I 内含奇数个像素！**
- ◆ 奇偶校验位计算公式：

$$p(I) = \sum_{j \in I} x_j \bmod 2$$

- ◆ If 奇偶校验位与秘密信息不同，则将区域 I 中LSB取值反转

LSB 信息隐藏算法

◇ 例子

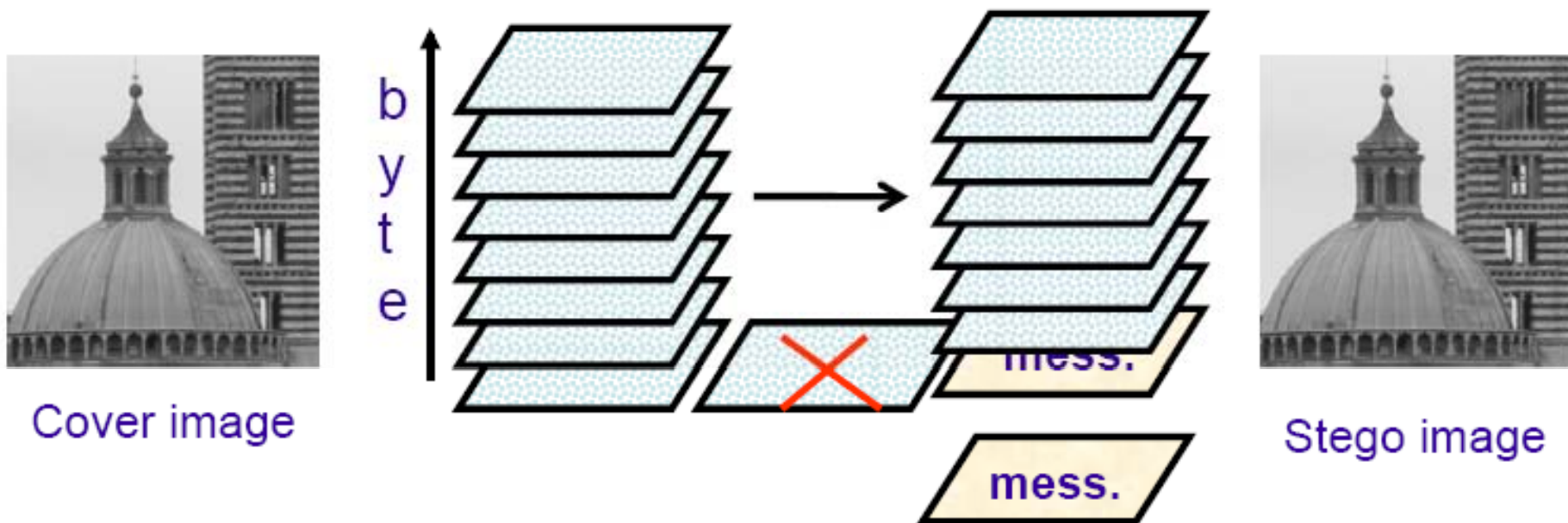
StegoDos, S-Tools, Mandelsteg, EzStego,
Steganos等

◆ 基于替换LSB的空域信息隐藏

LSB (the Least Significant Bits) 即最不重要的比特。改变 LSB 主要的考虑是不重要数据的调整对原始图像的视觉效果影响较小。以图像为例，图像部分像素的最低一个或者多个位平面的值被隐藏数据所替换。

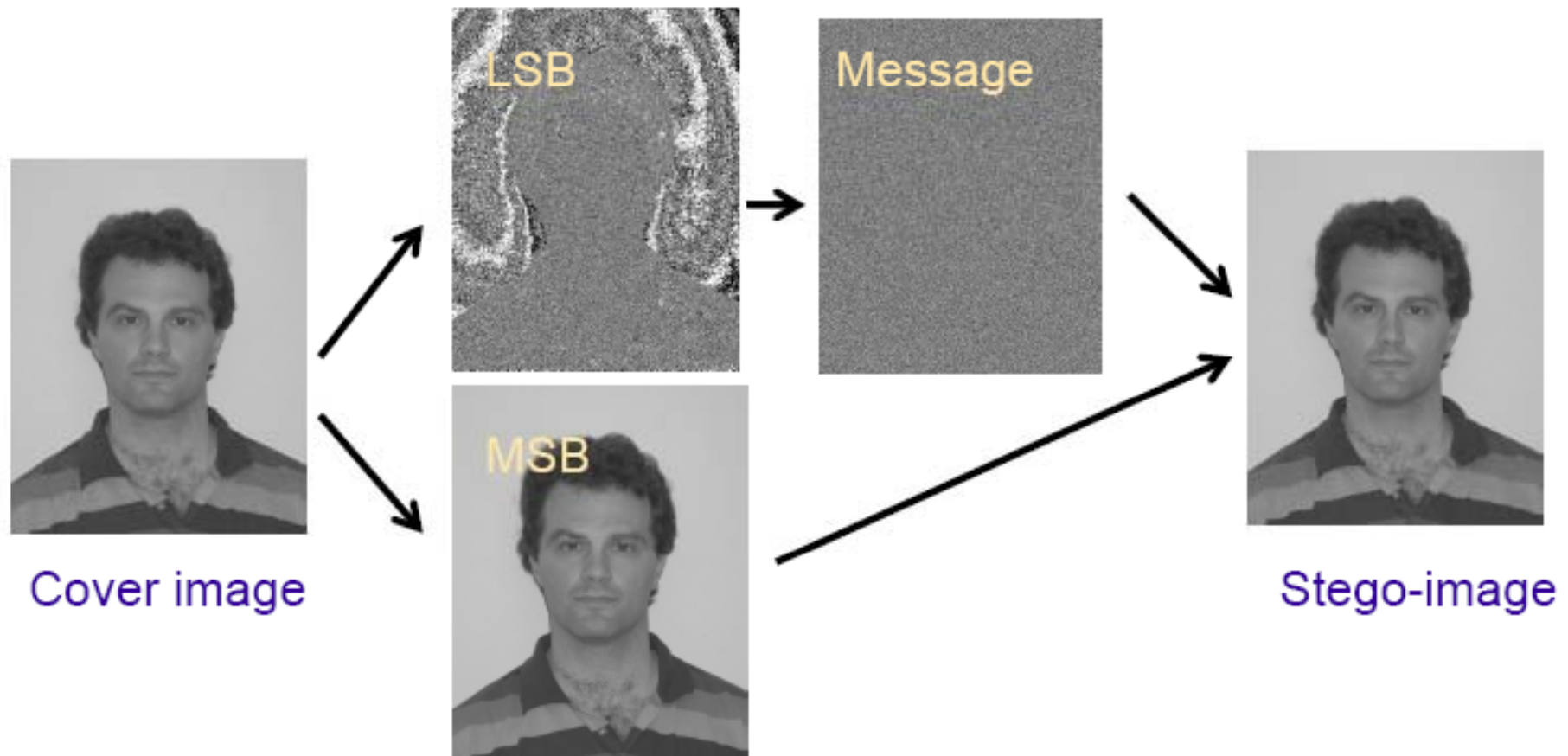
A detailed example: LSB embedding

The LSB's of the pixels of an image (or the DCT coefficients) are substituted with the stego-message (payload = 1bpp or 1bpnzc)



Visual imperceptibility

- LSB replacement looks perfect (but is not): the LSB plane of an image is very similar to noise



◆ 基于替换LSB的空域信息隐藏

本方法特点:

- (1) 具有较大的信息隐藏容量
- (2) 计算简单
- (3) 掩密图像失真小
- (4) 隐藏数据的鲁棒性较差

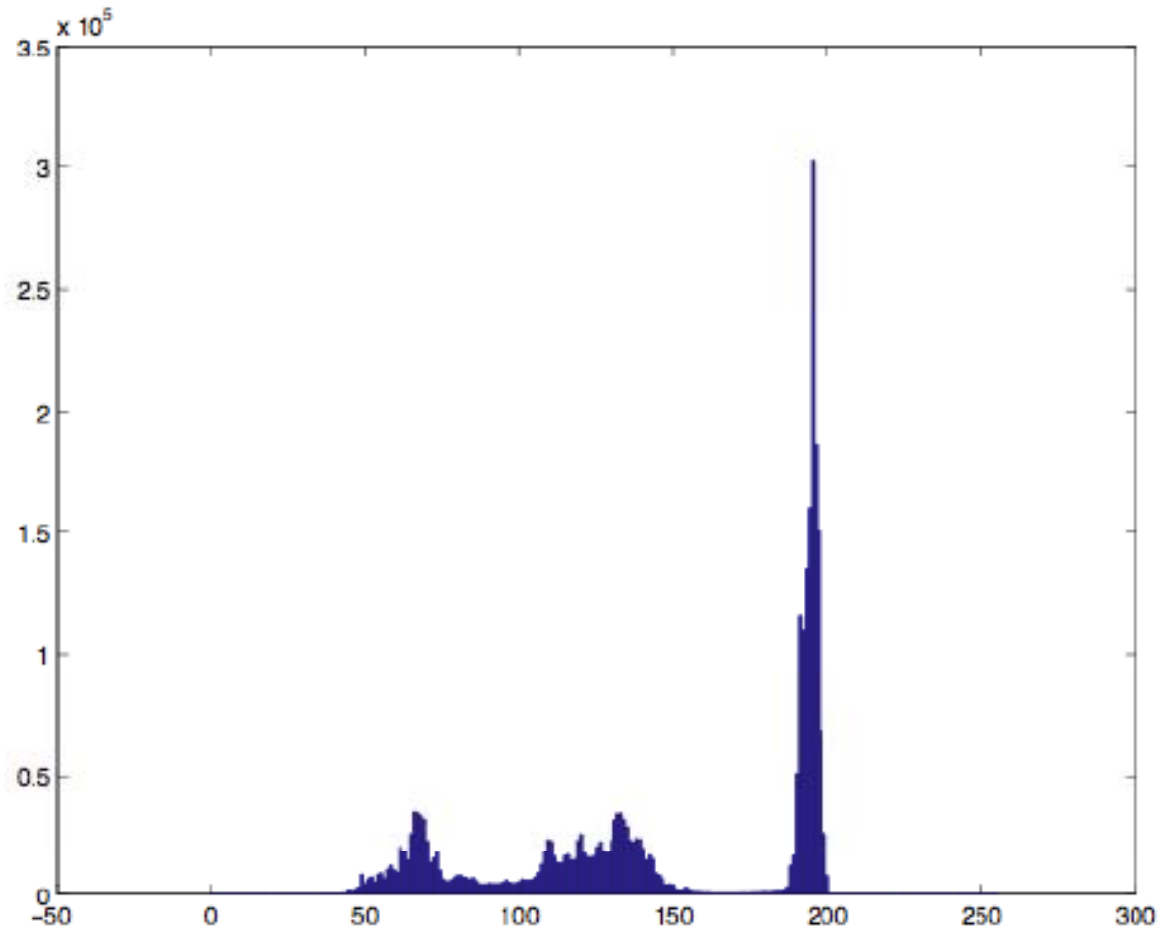
基于替换LSB的空域信息隐藏（隐写）技术

不可检测性（安全性）？

- 隐写分析

Attacking LSB replacement

As a matter of fact, steganalysis LSB replacement steganography is quite easy (at least for high payload)



Attacking LSB replacement

- If $x(i)$ is even we have 01100000**0** which remains as is or is increased by 1 \rightarrow 01100000**1**
- If $x(i)$ is odd we have 01100000**1** which remains as is or is decreased by 1 \rightarrow 01100000**0**
- Consider the pair (0,1): (000000000, 000000001)
- Half of the pixels equal to 0 pass to 1 and half of the pixels equal to 1 pass to 0
- At the end we have about the same number of pixels = 0 e pixels = 1, that is $h_{\text{stego}}(0) = h_{\text{stego}}(1)$

◆ LSB替换隐写前后的图像灰度直方图

$$h'(2k) \approx \frac{1}{2} \cdot h(2k) + \frac{1}{2} \cdot h(2k+1)$$

$$h'(2k+1) \approx \frac{1}{2} \cdot h(2k) + \frac{1}{2} \cdot h(2k+1)$$

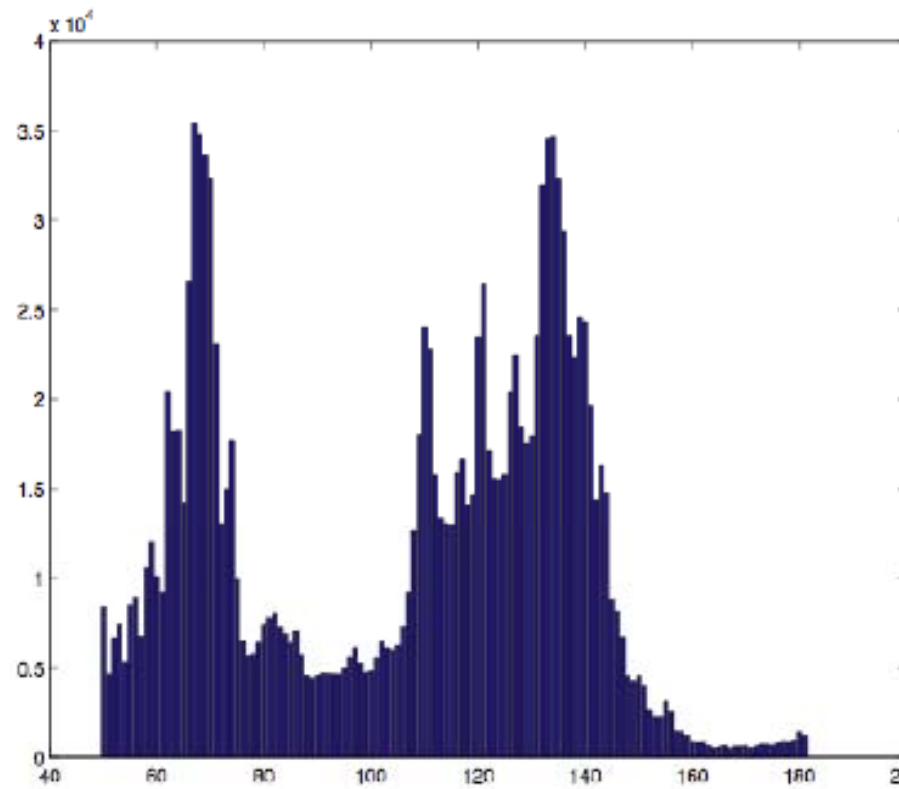
◆ 隐写前

$$\frac{h(2k) - h(2k+1)}{\sqrt{2} \cdot \sqrt{h(2k) + h(2k+1)}} \sim N(0, 1)$$

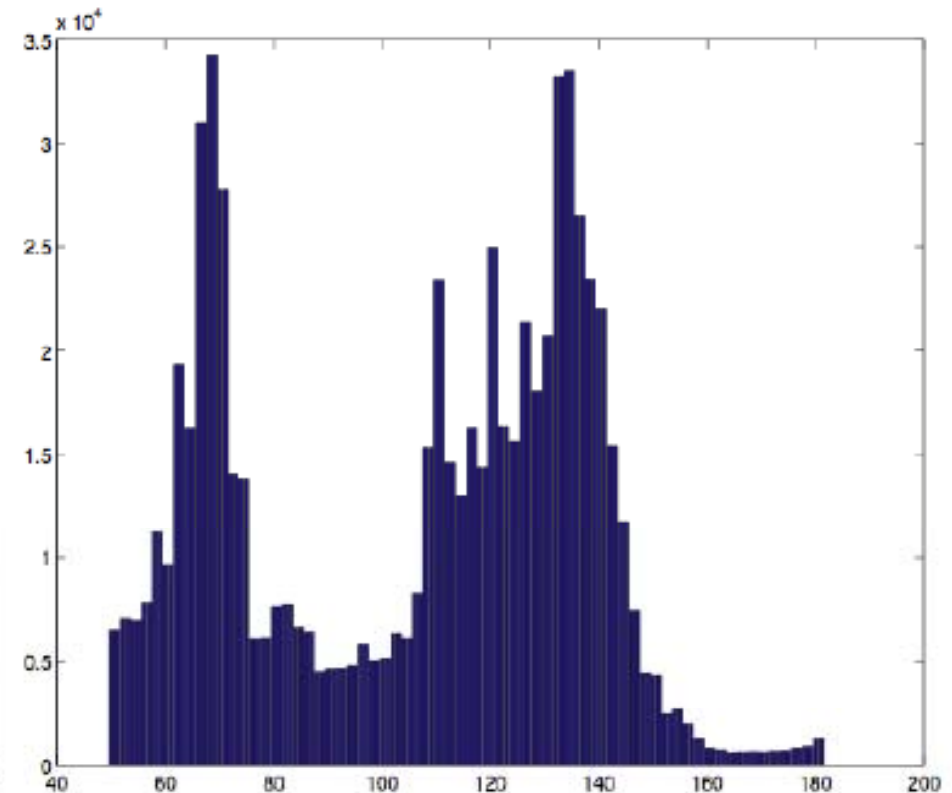
◆ 隐写后 $h'(2k) \approx h'(2k+1)$

Attacking LSB replacement

The histogram of stego-images has a very characteristic behaviour



Original histogram



Histogram of stego-image

Countermeasures

- Perfect steganography requires that **all** image statistics are preserved, however
 - It is impossible to derive adequate statistical models of images (slightly better in the DCt domain)
 - It would be too complicated
- Four empirical approaches are used in practice
 - Model-preserving steganography
 - Stochastic modulation
 - Steganalysis-aware steganography
 - Distortion minimization

◆ **MLSB (Multiple LSB) 替换信息隐藏**

- 方法一：随机选取最低多个位平面的比特位进行替换
- 方法二：随机选取像素，同时替换其多个最低有效位

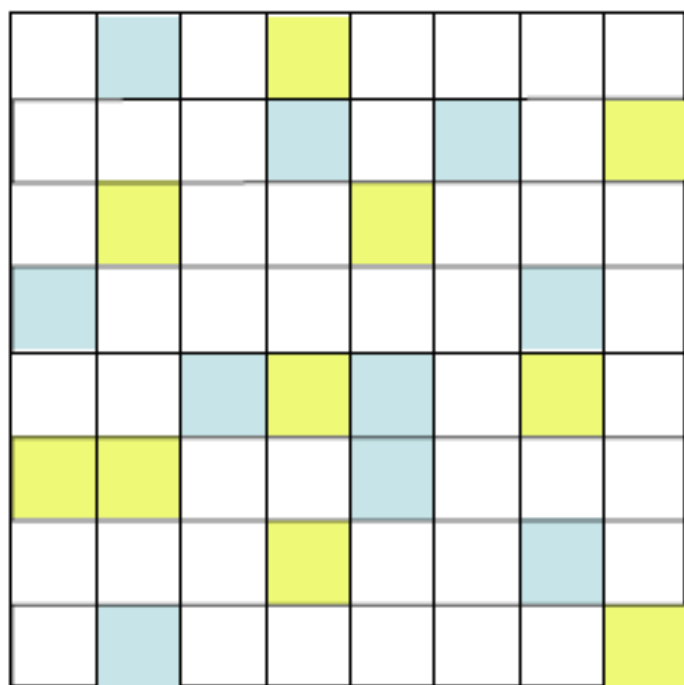
◆ ± 1 隐写（“LSB匹配隐写”）


- 基本思想： 当待嵌入的秘密信息比特与像素值的LSB不同时，对像素值进行随机加1或减1操作，使所得像素值的LSB等于秘密信息比特


典型算法 ②

基于Patchwork的信息隐藏技术

Examples: patchwork



 $A = \{a_i\}_{i=1,n}$

 $B = \{b_i\}_{i=1,n}$

- By letting $a'_i = a_i + d$,
 $b'_i = b_i - d$,
the watermark is inserted in the image
- Detection is achieved by computing the quantity

$$S_n = \frac{1}{n} \sum_{i=1}^n (a_i - b_i)$$

- A typical value for n is about 10.000

◆ Patchwork 空域信息隐藏

Patchwork 算法（拼凑方法）只是试图回答是否有水印存在，因而实际隐藏的只是 1 bit 信息。（单比特水印技术）

Patchwork 算法的一般步骤如下：

- （1）用一个密钥初始化一个伪随机数发生器；
- （2）根据伪随机数发生器的输出，随机选择 n 个像素对，其灰度值为 (a_i, b_i) ；

◆ Patchwork空域信息隐藏

(3) 令 $a_i^w = a_i + 1$, $b_i^w = b_i - 1$, 完成信息的嵌入。这样

整个图像的平均亮度保持不变。检测时, 令

$$s = \sum_{i=0}^{N-1} (a_i^w - b_i^w)$$

if $s \approx 2N$, 则存在隐藏信息

otherwise, 不存在隐藏信息

2.2 变换域信息隐藏技术

- 信息隐藏过程是在变换域中进行的。借助信号进行正交变换后能量重新分布的特点，进行信息隐藏，可较好的解决不可感知性和稳健性的矛盾。
- 信息隐藏中常用的变换有：
 - 离散傅立叶变换 (**Discrete Fourier Transform, DFT**)
 - 离散余弦变换 (**Discrete Cosine Transform, DCT**)
 - 离散小波变换 (**Discrete Wavelet Transform, DWT**)
 - 此外，近年出现 **Curvelet**变换等。

◆ 离散余弦变换 (DCT)

DCT变换是一种通常使用的图像压缩标准，JPEG、MPEG-2等标准，采用的变换都是DCT。

基于DCT的数字水印算法首先从载体中获取特征进行二维离散余弦变换，然后选择适当的系数将水印嵌入，最后进行二维离散余弦反变换得到加入水印的图像。

◆ 离散余弦变换 (DCT)

选择什么样频段的系数是一个很有争议的问题。将水印加入高频段，这样不至于使原始图像失真；将水印加入到图像的低频段可以增强水印的鲁棒性。现在更为统一的意见是将水加入到原始图像中频段以在信噪比和鲁棒性之间平衡。

◆ 离散小波变换 (DWT)

小波变换具有多分辨分析特点，能充分反映人类的视觉特性，特别是新的图像压缩标准，如 JPEG 2000等都采用了基于小波变换的方法，因而在小波变换域研究水印是极为重要的。

基于 DWT的数字水印算法和基于 DCT的数字水印算法的基本思想是基本一样的。但是由于基于 DWT的数字水印算法具有多分辨特性，水印的嵌入变得更为灵活。

图像的常用变换

◆ DFT

◆ DCT

◆ DWT

DFT

- ◆ 离散傅立叶变换

- ◆ 复数结果，有实部和虚部

- ◆ 幅值和相位

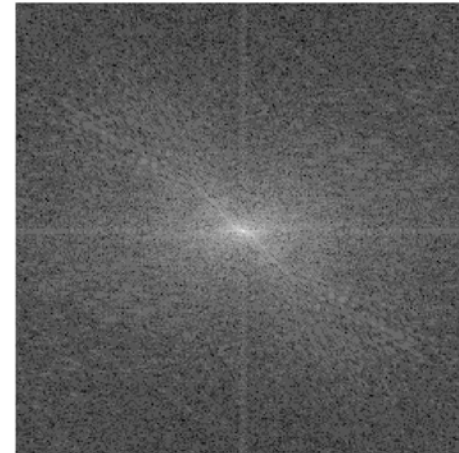
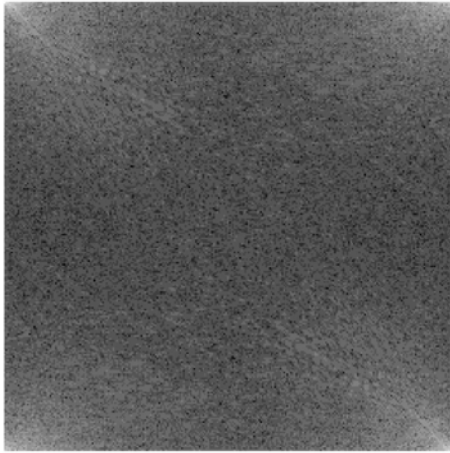
- ◆ 程序说明

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

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

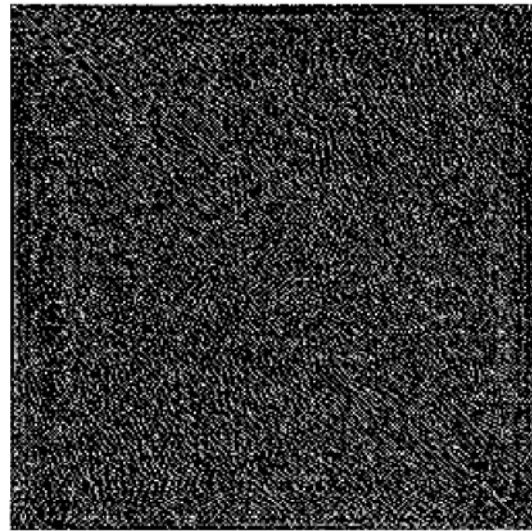
DFT

◆ 幅値



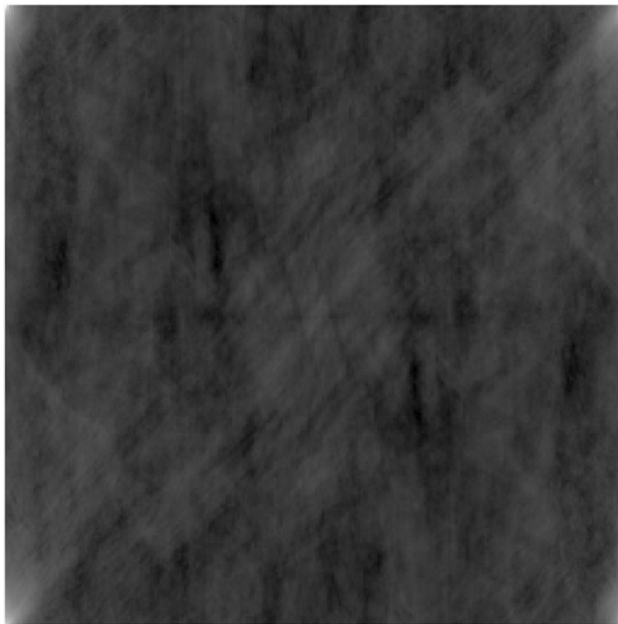
DFT

◈ 相位



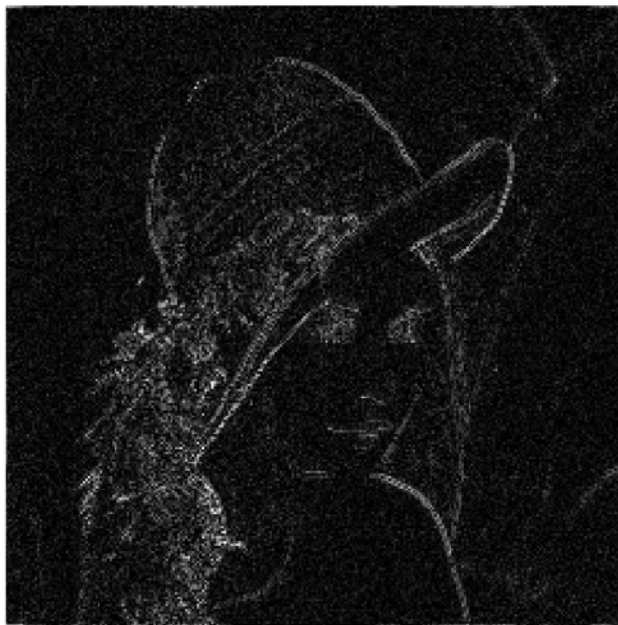
DFT

◆ 幅值对图像的贡献



DFT

◆ 相位对图像的贡献



DCT

◆ 离散余弦变换

◆ 实数到实数

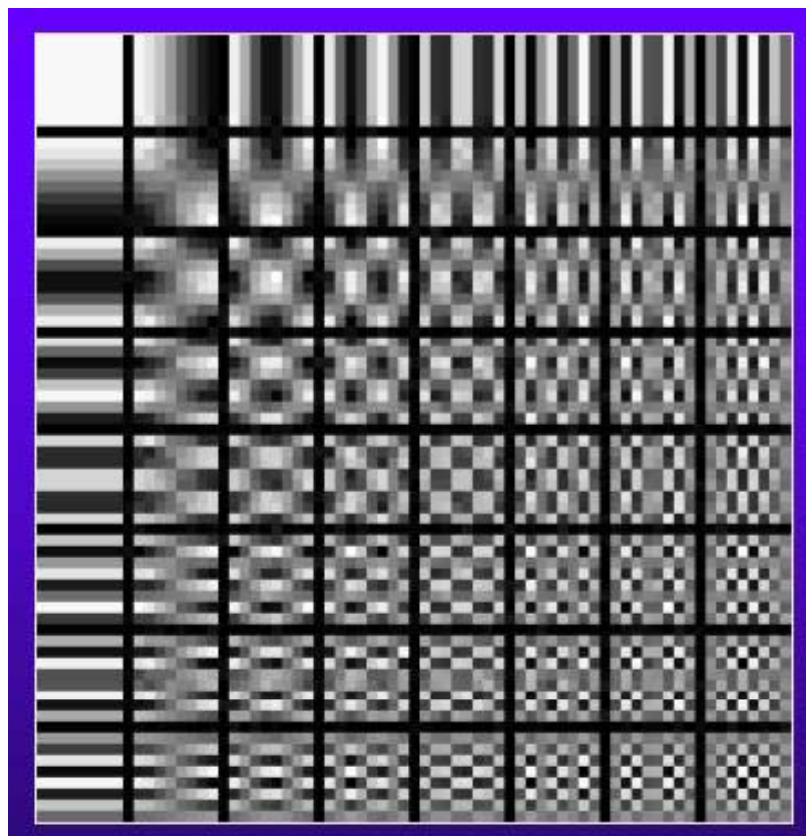
◆ 酉变换；对于实数就是正交变换

$$F(u, v) = c(u)c(v) \sum_{i=0}^N \sum_{j=0}^N f(i, j) \cos \left[\frac{(2i+1)u\pi}{2N} \right] \cos \left[\frac{(2j+1)v\pi}{2N} \right]$$

$$\text{where } c(u) = \begin{cases} \sqrt{\frac{1}{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & u \neq 0 \end{cases}$$

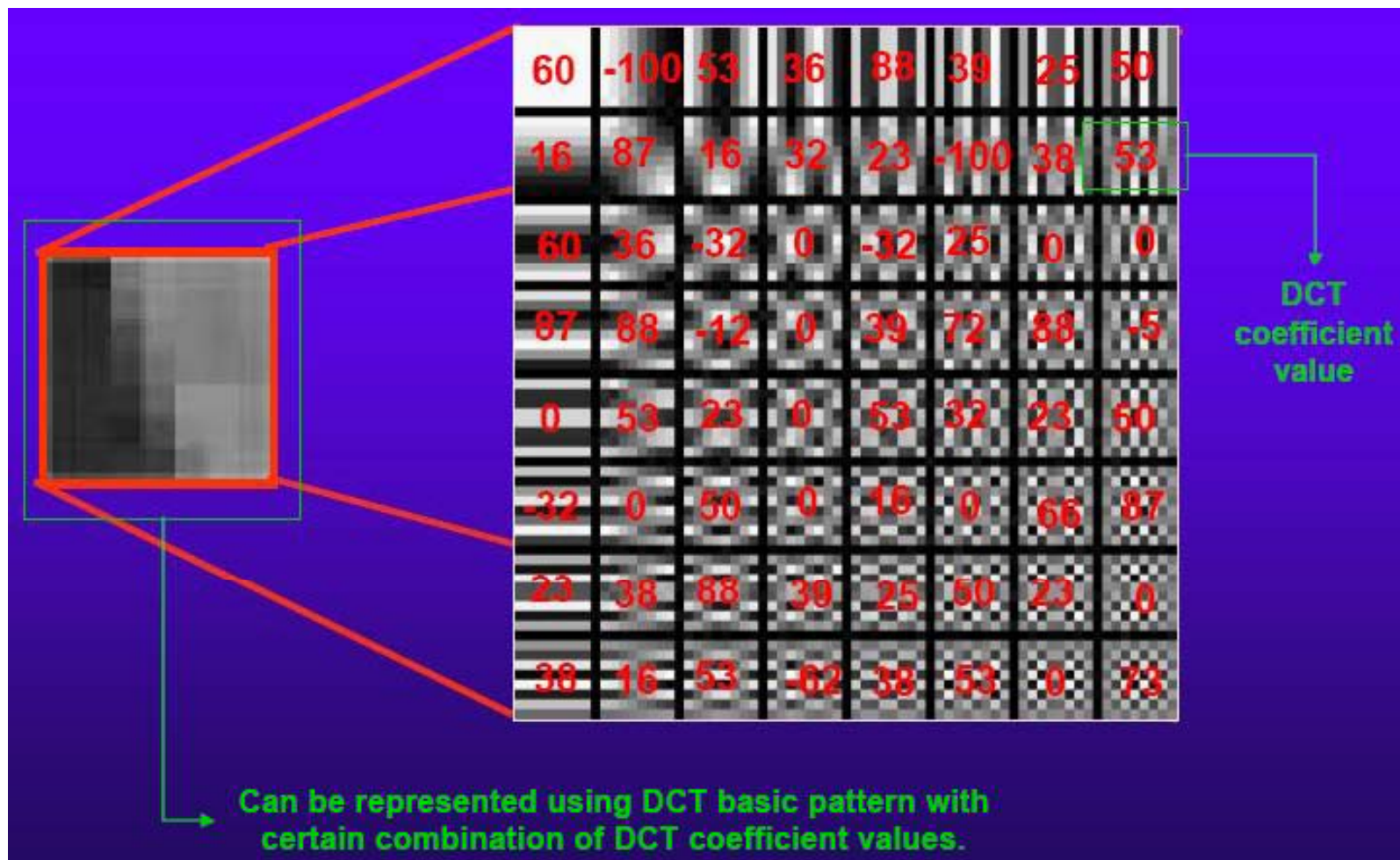
DCT

- ◆ 8×8 分解时的基图像
 - ◆ 基图像：两个向量的外积



DCT

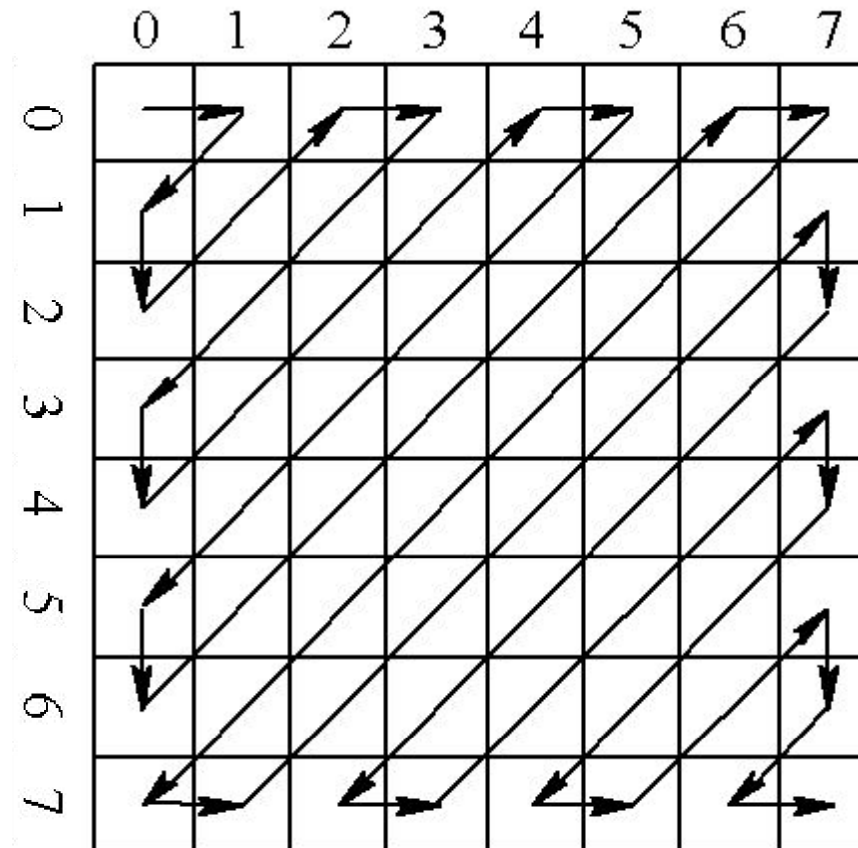
- ◆ 用基图像表示图像（体会图像和系数之间联系）



DCT

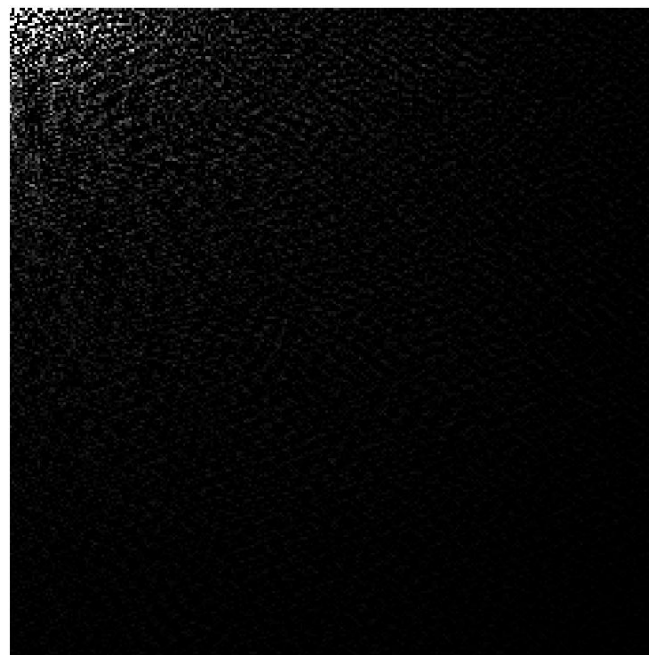
◆ Zigzag Scanning

- Transform 2D DCT coefficients to 1D ones



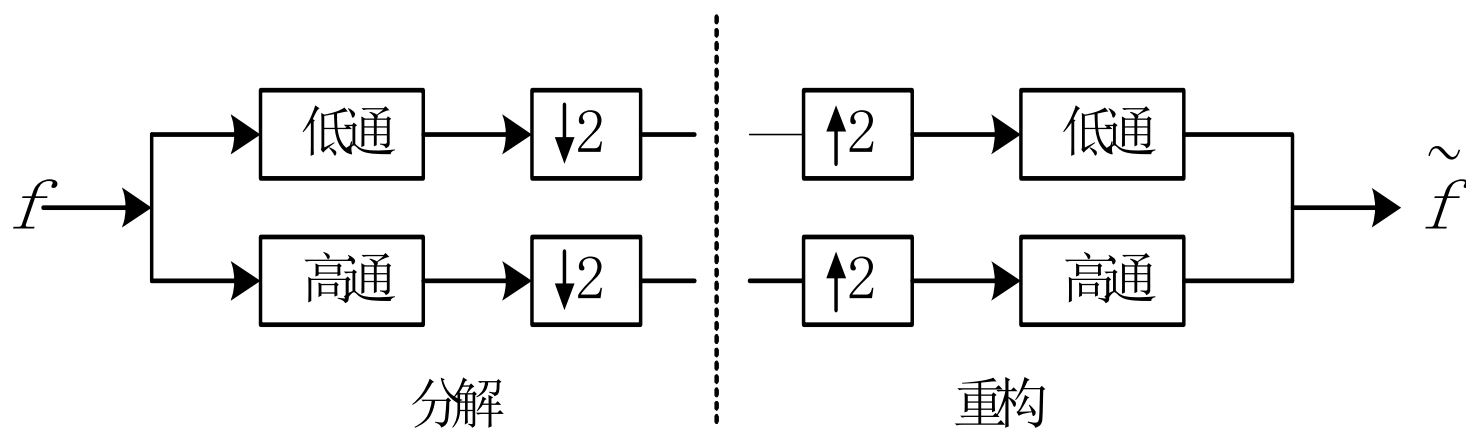
DCT

- ◆ 整幅图像做DCT变换



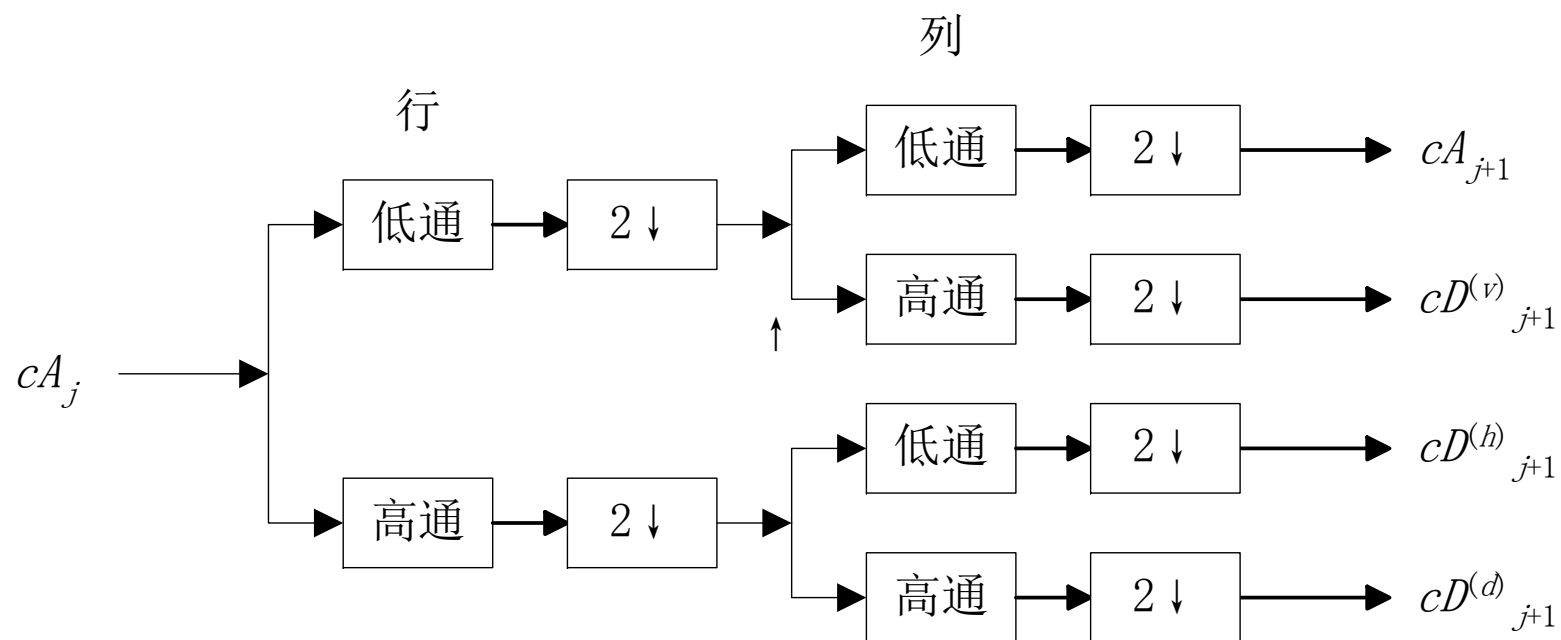
DWT

- ◆ 小波变换实质上是一种滤波运算



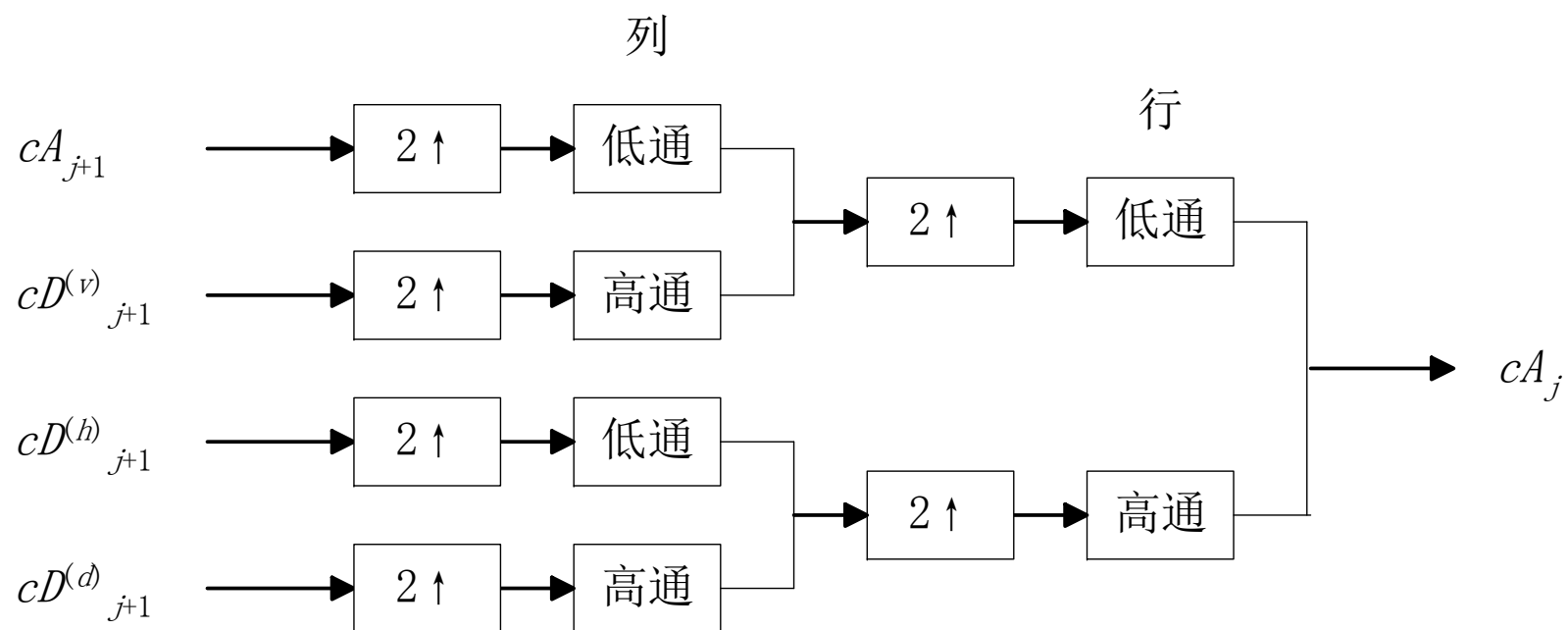
DWT

◇ 二维分解



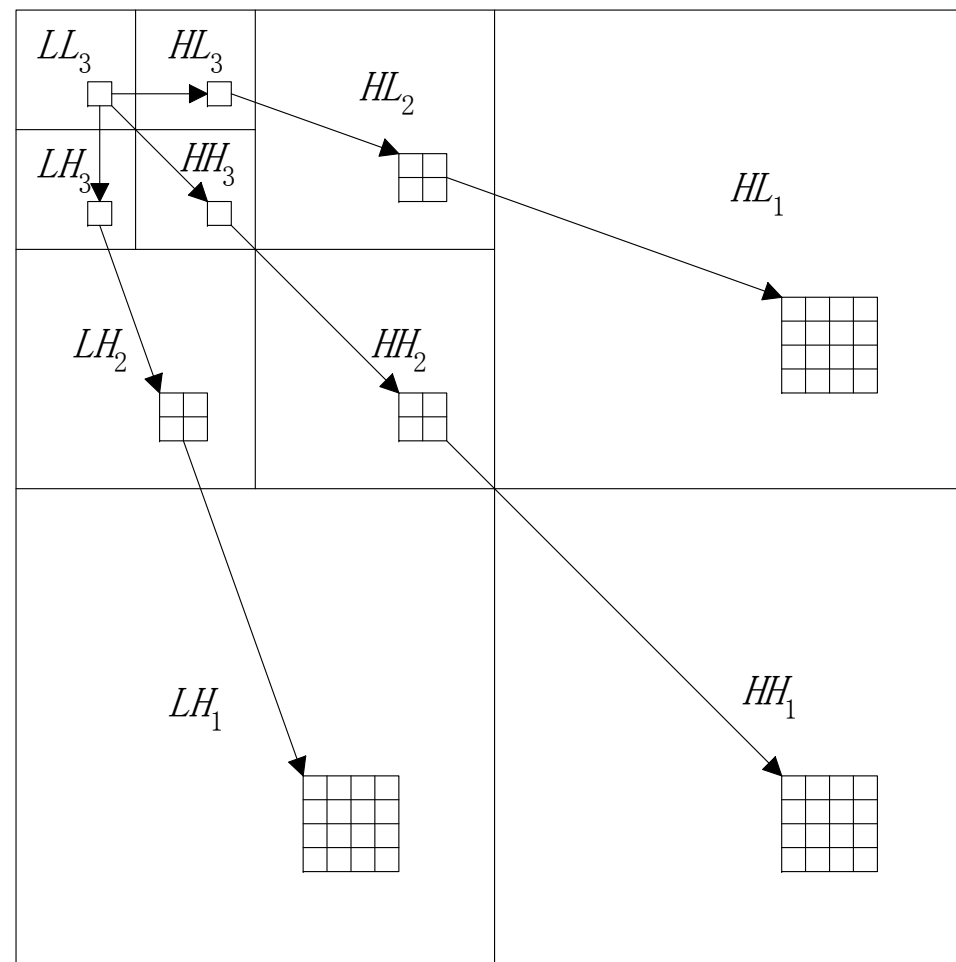
DWT

◆ 二维重构



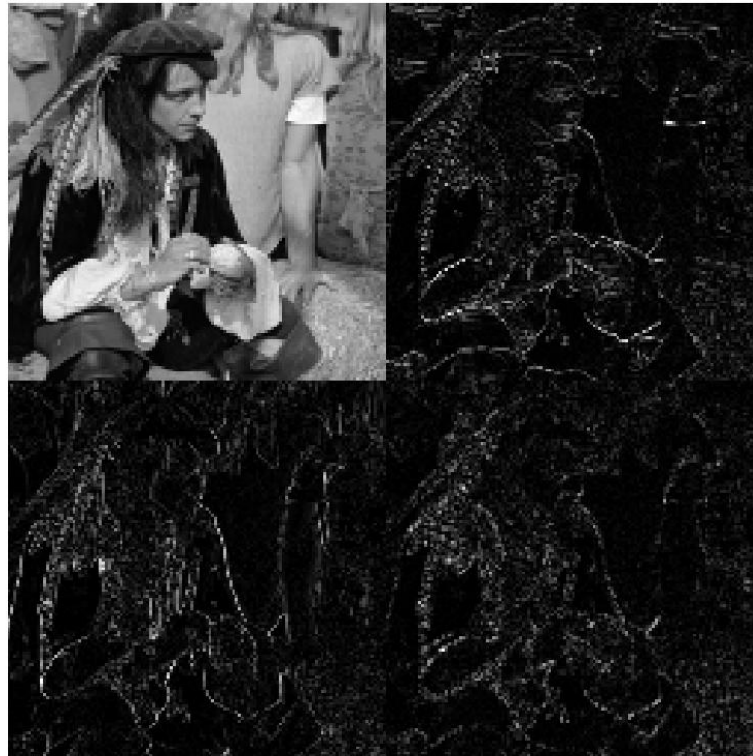
DWT

◆ 父子关系



DWT

◆ 一层分解

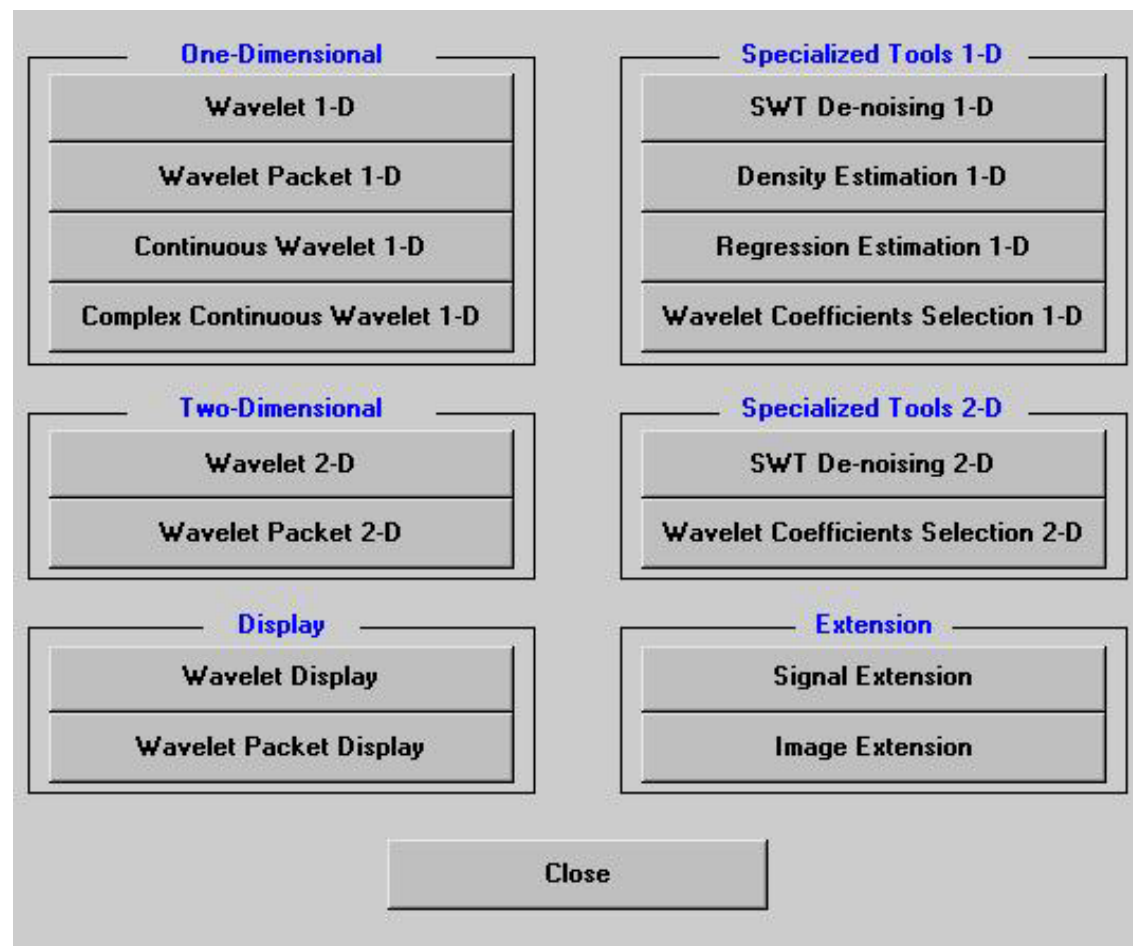


DWT


- ◆ 常用函数
 - ◆ dwt2, idwt2
 - ◆ wavedec2, waverec2
 - ◆ detcoef2
 - ◆ appcoef2

DWT

◆ wavemenu




Original Image




50
100
150
200
250

50 100 150 200 250

Approximation coef. at level 2



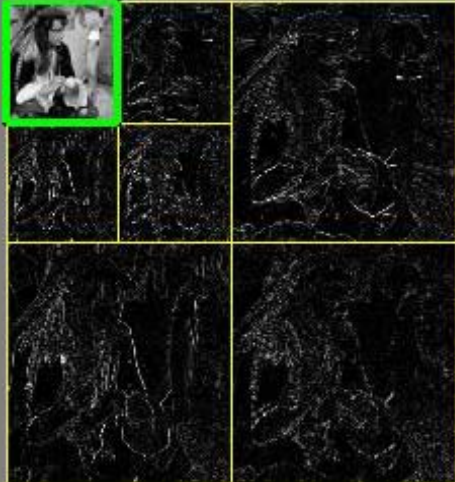
Synthesized Image



idwt

dwt

Image Selection



Decomposition at level 2

Data (Size)man256.bmp

Waveletdb1

Level2

Analyze

StatisticsCompress

HistogramsDe-noise

Decomposition at level :2

View mode : Square

Full Size

1	3
2	4

Operations on selected image :

Visualize
Full Size
Reconstruct

Colormapgray

Nb. Colors256

Brightness-+

Close

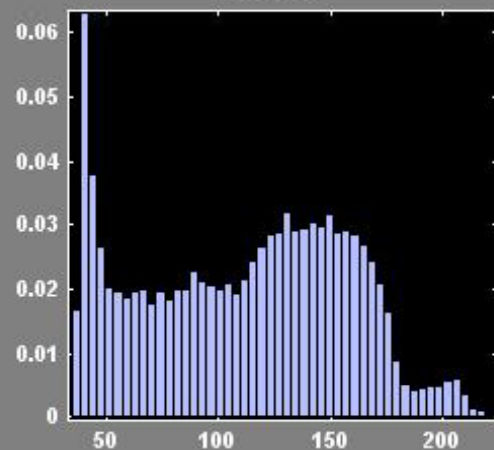
X+	Y+	XY+	Center On	X	Y	Info	X =	History	<-	>	View Axes
X-	Y-	XY-					Y =		<<	>>	

man256.bmp [256 x 256] analyzed at level 2 with db1 ---> X = 1 : 256 Y = 1 : 256

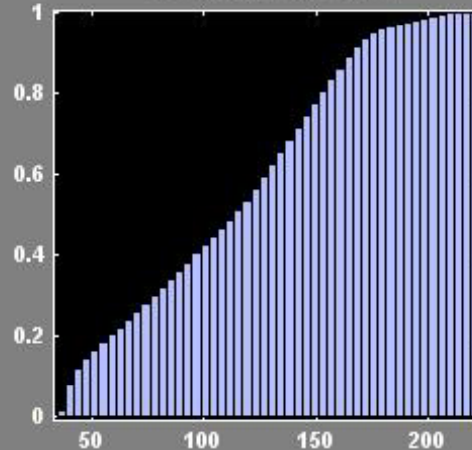
Reconstructed approximation at level 1



Histogram



Cumulative histogram



Mean	111	Maximum	222.5	Standard deviation	45.64
Median	116	Minimum	34.75	Median absolute deviation	37.25
Mode	40.38	Range	187.8	Mean absolute deviation	39.37

Data (Size) man256.bmp

Wavelet db 1

Level 2

- ☐ Original image
- ☐ Synthesized image
- ☒ Approximation
- ☐ Detail

Approximation at

Level 1

- ☐ Coefficients
- ☒ Reconstructed

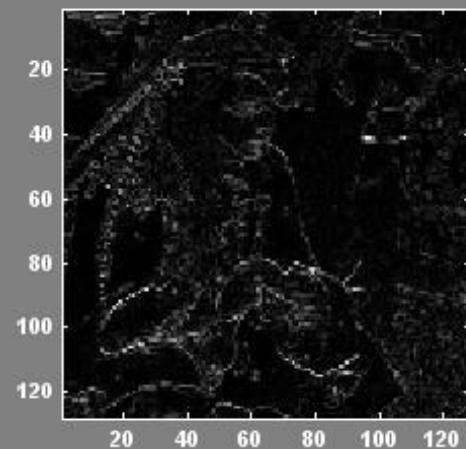
Number of bins 50

Show statistics

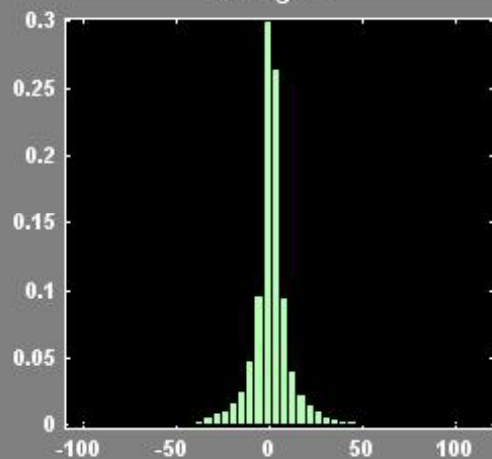
Close

man256.bmp (256 x 256) analyzed at level 2 with db1 ---> X = 1 : 256 Y = 1 : 256

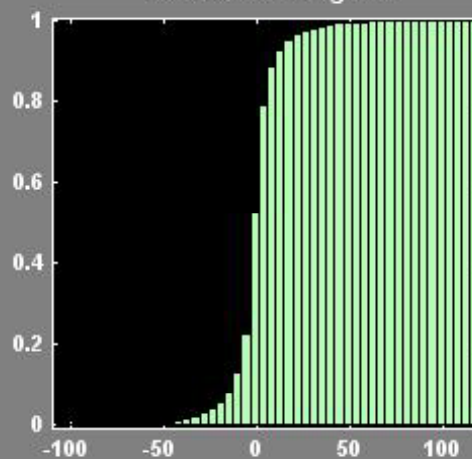
Coefficients of detail at level horizontal 1



Histogram



Cumulative histogram



Mean	0.09402	Maximum	118.5	Standard deviation	13
Median	0	Minimum	-108.5	Median absolute deviation	3.5
Mode	-1.81	Range	227	Mean absolute deviation	7.444

Data (Size) man256.bmp

Wavelet db 1

Level 2

- ☐ Original image
- ☐ Synthesized image
- ☐ Approximation
- ☒ Detail

Detail at

Level 1

Horizontal

- ☒ Coefficients
- ☐ Reconstructed

Number of bins 50

Show statistics

Close

2.2 变换域信息隐藏技术

◆ 变换域信息隐藏方法的主要步骤:

- (1) 应用DCT、DFT、DWT等方法将原始宿主信号变换到频域空间;
- (2) 在变换域选择 n 个系数以隐藏信息;
- (3) 根据一定的规则或公式修改选择的 n 个变换系数;
- (4) 进行反变换以得到掩密载体。

◆ 变换域信息隐藏的优点

(1) 变换域中嵌入的信号能量可以较均匀的分布到空域的所有像素上，有利于保证不可见性。

(2) 在变换域，HVS (Human Visual System) /HAS (Human Auditory System) 的某些特性可以更方便地结合到嵌入过程中，有利于不可感知性和稳健性能的提高。

(3) 变换域的方法可与国际数据压缩标准兼容，从而便于实现在压缩域内的信息隐藏算法。

◆ 变换域信息隐藏的缺点

- (1) 隐藏信息量比空域方法低；而计算量大于空域算法；
- (2) 在正变换和反变换计算过程中，由于数据格式的转换，通常会造成信息的丢失，这等效于一次轻微的攻击，对于大量数据的隐藏时很不利的。

典型算法③

基于扩频的信息隐藏技术

➤ 扩频技术

扩频技术的一个重要优点是具有很强的抗干扰性。利用扩频技术的原理，将水印分布在许多数据频域系数中，加入每个频域系数的信号能量很小而且不可随意检测。然而，水印检测过程知道水印的位置和内容，它能够将许多微弱的信号集中起来形成具有较高信噪比的输出值，要破坏水印需要强噪声加入所有的频域系数中，但是破坏水印的同时也造成原始数据质量严重下降。

➤ 扩频水印算法

- **基本原理：**基于扩频通信原理，扩频水印技术采用叠加伪随机序列的方式嵌入水印。具体地，在原始数据的空域或变换域上叠加经水印信息调制后的伪随机序列，检测时计算待检信号与伪随机序列之间的互相关系数，然后与阈值比较提取隐藏信息。

- **数学表示：**载体信号 $X = \{x(0), x(1), x(2), \dots, x(N-1)\}$

待嵌入的水印序列 $W = \{w_0, w_1, w_2, \dots, w_{M-1}\} \quad w_j = \{+1, -1\}$

M 个长度为 N 的一维伪随机序列

$$S_j = \{s_j(0), s_j(1), s_j(2), \dots, s_j(N-1)\}, j = 0, 1, 2, \dots, M-1 \quad s_j(i) = \{+1, -1\}$$

➤ 扩频水印算法

➤ 数学表示

嵌入水印后的载体信号 $X' = \{x'(0), x'(1), x'(2), \dots, x'(N-1)\}$

➤ 水印嵌入算法

在扩频水印方法中嵌入水印的基本操作是，在原始载体上以加性或乘性的方式叠加一个经水印信息调制后的扩频序列。因此，扩频水印方法的嵌入规则可统一地公式化描述为

$$x'(i) = x(i) + \alpha \cdot \sum_j^{M-1} S_j(i) \cdot w_j$$

其中， $i = 0, 1, 2, \dots, N-1$ 。 α 是嵌入强度。

➤ 扩频水印算法

➤ 水印嵌入算法（续）

上式也可简写成
$$X' = X + \alpha \cdot \sum_j^{M-1} S_j \cdot w_j$$

如果嵌入水印时进一步使用感知模型来控制水印强度，设各位置上的载体元素所需嵌入强度对应为 $H=\{h_0, h_1, h_2, \dots, h_{N-1}\}$ ，则扩频水印算法的嵌入规则

变为

$$x'(i) = x(i) + \alpha \cdot h_i \cdot \sum_j^{M-1} S_j(i) \cdot w_j$$

即

$$X' = X + \alpha \cdot H \circ \sum_j^{M-1} S_j \cdot w_j$$

➤ 扩频水印算法

➤ 水印提取算法

提取水印采用基于相关性检测的水印判决方法。含水印载体在传输过程中可能受到各种干扰引起失真,记水印检测端收到的载体信号为 $X'' = \{x''(0), x''(1), x''(2), \dots, x''(N-1)\}$

(1) 计算相关系数

$$\rho_j(X'', S_j) = \frac{1}{N} \sum_{i=0}^{N-1} \left[\left(x''(i) - \overline{X''} \right) \cdot S_j(i) \right]$$

其中, $j = 0, 1, 2, \dots, M-1$ 。 $\overline{X''}$ 表示 X'' 中全部元素的平均值。

➤ 扩频水印算法

➤ 水印提取算法（续）

(2) 利用阈值化方法判决水印信息，判决规则为

$$w'_j = \begin{cases} +1, & \rho_j(X'', S_j) > 0 \\ -1, & \rho_j(X'', S_j) < 0 \end{cases}$$

其中， $j = 0, 1, 2, \dots, M-1$ 。所提取水印为 $W' = \{w'_0, w'_1, w'_2, \dots, w'_{M-1}\}$

通过统计比较原始水印与检出水印的对应元素，可得出水印检测算法的正检率和误检率，分别是正确检出和错误检出的水印比特数所占比率。

The spread spectrum paradigm

- Let us consider first the case of 1-bit watermarking
- A watermarking signal \mathbf{U} is generated by starting from the secret key (\mathbf{U} plays the role of the secret key)
- For instance, the sequence \mathbf{U} may be an i.i.d. sequence having a fixed pdf (e.g. $N(0,1)$)
- The marked signal \mathbf{Y} is formed by adding (Add-SS) a scaled version of \mathbf{U} to \mathbf{X}

$$\mathbf{Y} = \mathbf{X} + \gamma \mathbf{U}$$

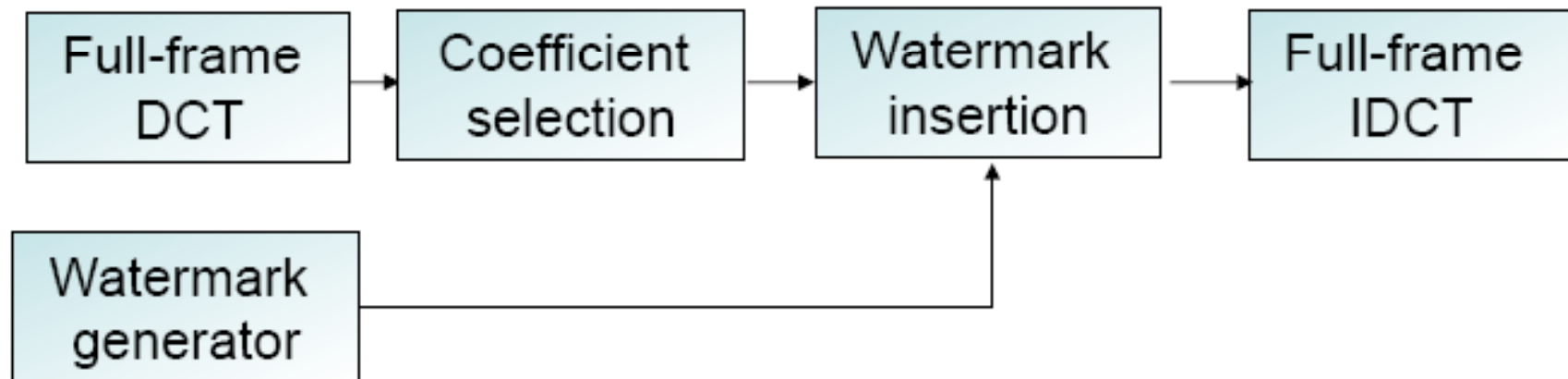
- A multiplicative version of SS also exists for which (Mul-SS)

$$\mathbf{Y} = \mathbf{X}(1 + \gamma \mathbf{U}) \quad \text{or} \quad \mathbf{Y} = \mathbf{X}(1 + \gamma |\mathbf{U}|)$$

$$\rho_n = \frac{\mathbf{y} \cdot \mathbf{w}}{\|\mathbf{y}\| \cdot \|\mathbf{w}\|}$$

$$\rho = \frac{1}{n} \sum_{i=1}^n y_i w_i \quad \begin{cases} \rho > T_\rho & H_1 \\ \rho < T_\rho & H_2 \end{cases}$$

Example: Cox's algorithm



- The watermark is a sequence of i.i.d. Gaussian random variables
- It is inserted in the n (1000) most significant (largest magnitude) DCT coefficients (non-blind, robustness)
- Insertion is accomplished according to a multiplicative rule

$$f_{w,i} = f_i + \gamma f_i w_i$$

The spread spectrum paradigm

- The watermarking signal **U** may assume different forms
 - Pseudo-random noise following a Gaussian or a uniform distribution
 - Binary pseudo-random noise: m-sequences
 - Chaotic signals
 - Periodic self-synchronizing watermarks

Multibit spread spectrum

- For the multibit case the spreading sequence \mathbf{U} is modulated by the to-be-hidden bit M

$$\mathbf{Y} = \mathbf{X} + \gamma \mathbf{U}(-1)^M \quad M \in \{0,1\}$$

- To embed more than a single bit, the sequence \mathbf{U} is split into N_c parts, each of which is modulated by a different bit
- Alternatively, N_c spreading sequences may be generated and superimposed to the host signal all together to embed all the bits at once

$$\mathbf{Y} = \mathbf{X} + \frac{\gamma}{\sqrt{N_c}} \sum_{i=1}^{N_c} \mathbf{U}_i (-1)^{M_i} \quad M_i \in \{0,1\}$$

典型算法④

QIM水印算法

➤ 二值量化索引调制(QIM)

- **基本原理**：二值量化索引调制(QIM)是一种常用的数字水印算法，其基本原理是按照待嵌入的二进制水印比特 $b \in \{+1, -1\}$ ，选择使用两个均匀量化器中的一个对载体信号 x 进行量化处理。
- **抖动(量化)调制**：通常采用抖动(量化)调制(DM)方法，设均匀量化器 $Q_{-1}(\cdot)$ 和 $Q_{+1}(\cdot)$ 的质心分别为

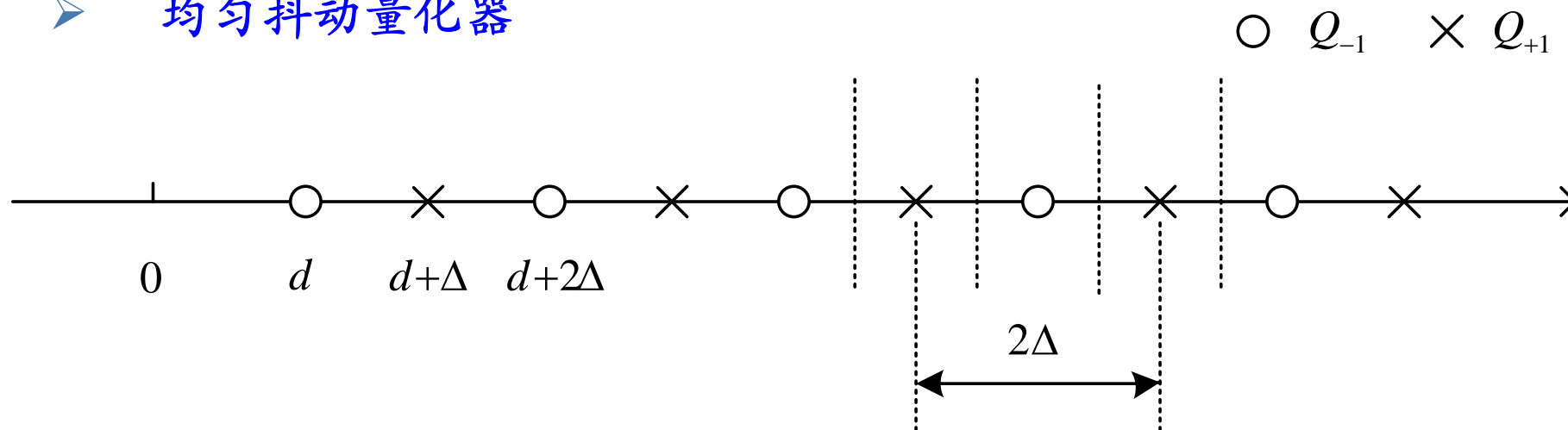
$$\Lambda_{-1} = 2\Delta \cdot Z + d$$

$$\Lambda_{+1} = 2\Delta \cdot Z + d + \Delta$$

其中， Z 为整数， d 是依赖于密钥的任意值。

➤ 二值量化索引调制(QIM)

➤ 均匀抖动量化器



➤ 水印嵌入

$$Q_b(x) = \begin{cases} \left\lceil \frac{x-d}{2\Delta} \right\rceil \cdot 2\Delta + d, & b = -1 \\ \left\lceil \frac{x-d-\Delta}{2\Delta} \right\rceil \cdot 2\Delta + d + \Delta, & b = +1 \end{cases}$$

- 欲嵌入秘密信息比特-1和+1，需分别采用对应的量化器 Q_{-1} 和 Q_{+1} 对载体信号进行量化

➤ 二值量化索引调制(QIM)

➤ 水印提取

利用最小欧氏距离解码器实现，即

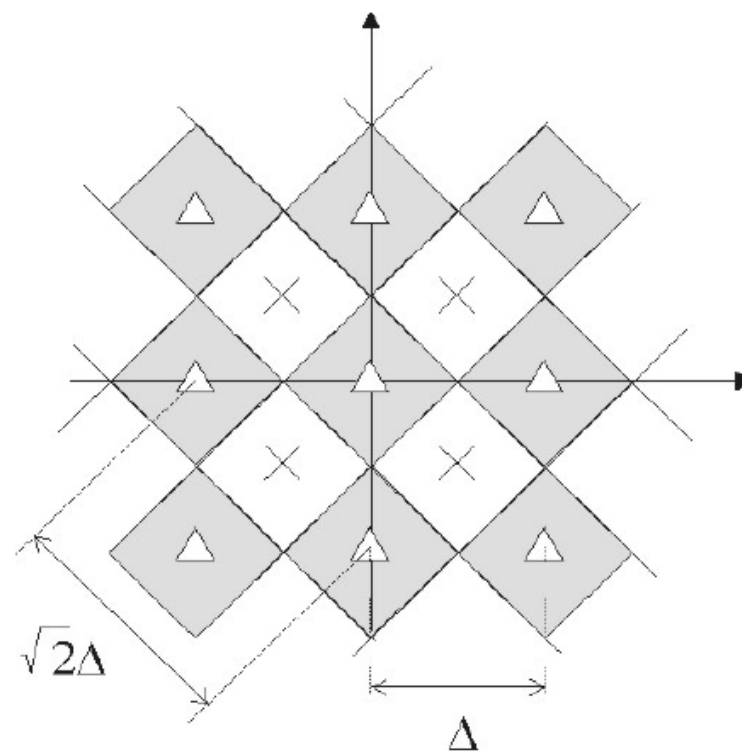
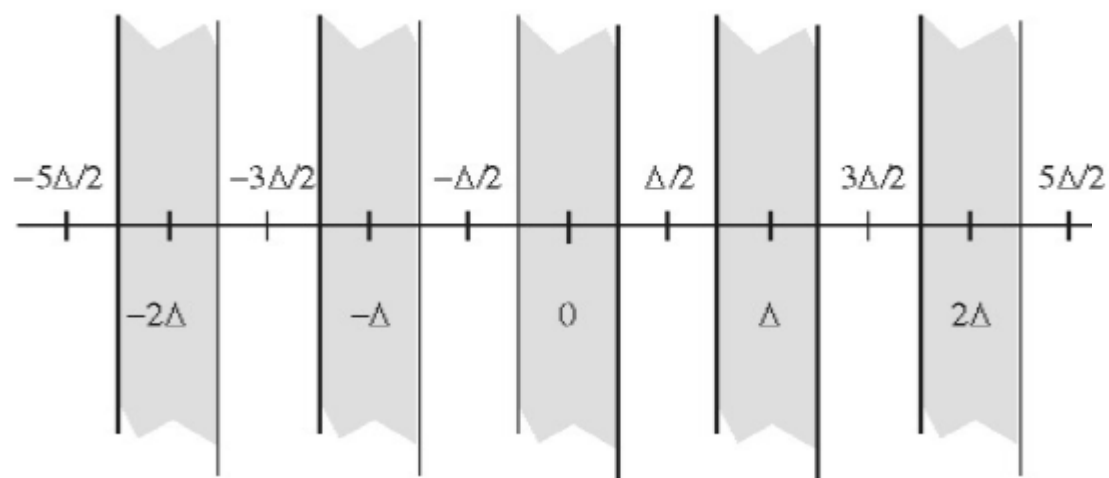
$$\hat{b} = \arg \min_{-1, +1} \|y' - Q_b(y')\|$$

其中， y' 表示解码端接收到的可能经历过某些干扰的含水印信号， \hat{b} 表示提取出的水印比特， $\|\cdot\|$ 表示欧氏距离。

不难理解，当信号失真满足 $\|y' - y\| < \Delta/2$ 时即能提取出正确的水印比特信息，因而QIM水印方法具有较强的鲁棒性。

QIM（量化索引调制）水印

➤ DM 抖动调制



典型算法⑤

Jsteg, F3, F4, F5

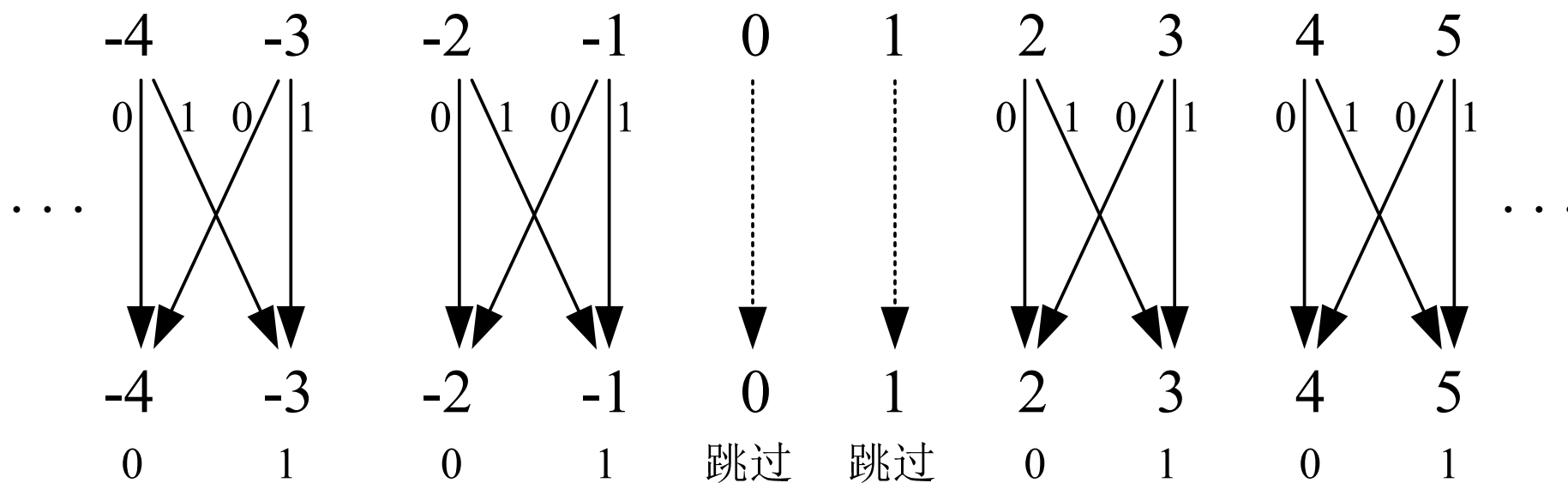
➤ 概述

在DCT域隐藏信息是常见的数字隐写方式之一。JSteg是最早的JPEG图像隐写方法之一，但由于其导致量化后DCT系数直方图出现明显变化，随后相继出现了F3、F4、F5、Outguess和MB等数字图像隐写方法。

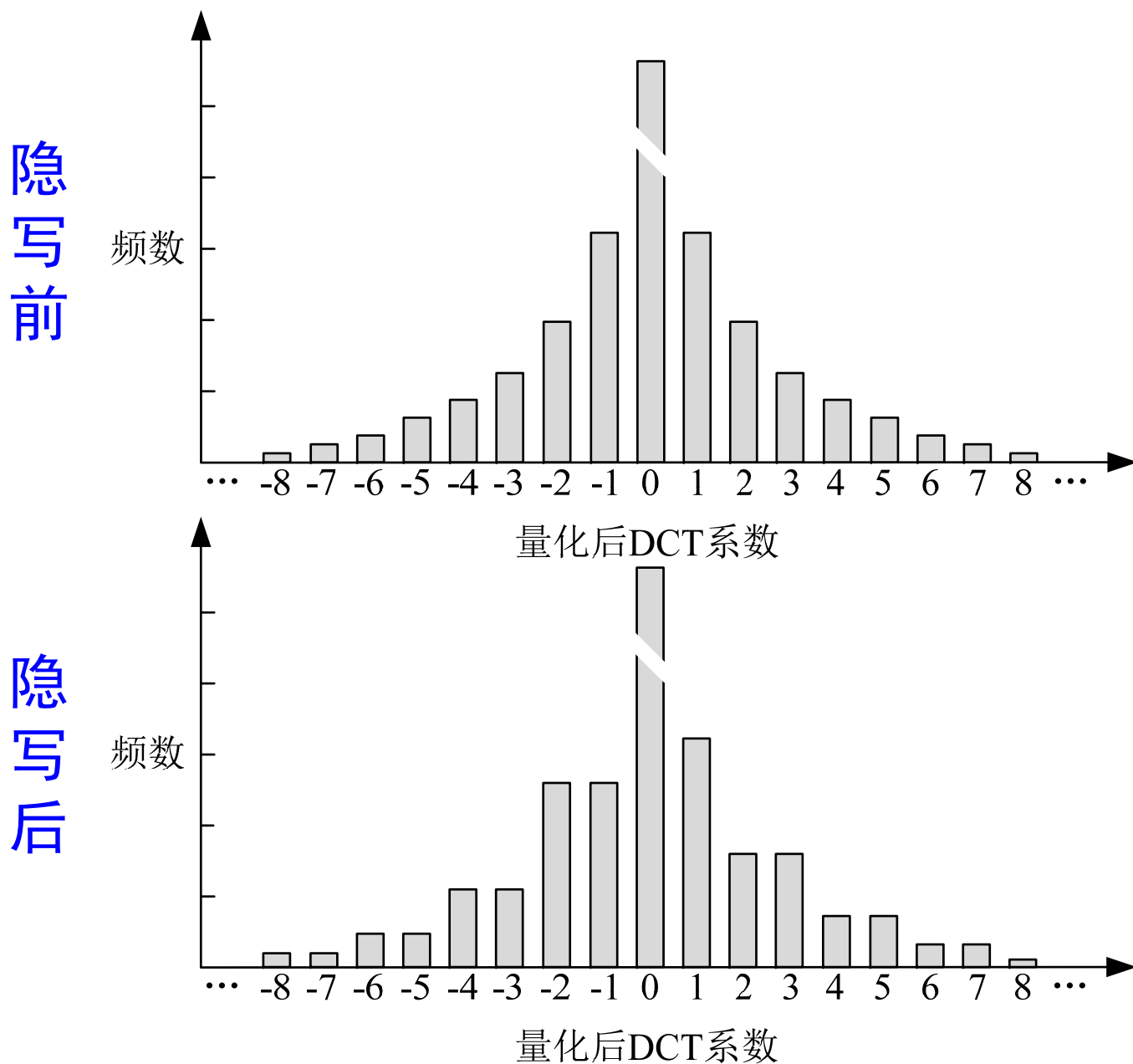
➤ JSteg

- 一种JPEG图像隐写软件
- 策略1：将LSB替换应用到JPEG图像中量化后DCT系数
- 策略2：绝对值为0或1的量化后DCT系数不嵌入任何信息
- 不足：不能抵抗卡方分析

JSteg隐写示意图



JSteg隐写前后图像的灰度直方图



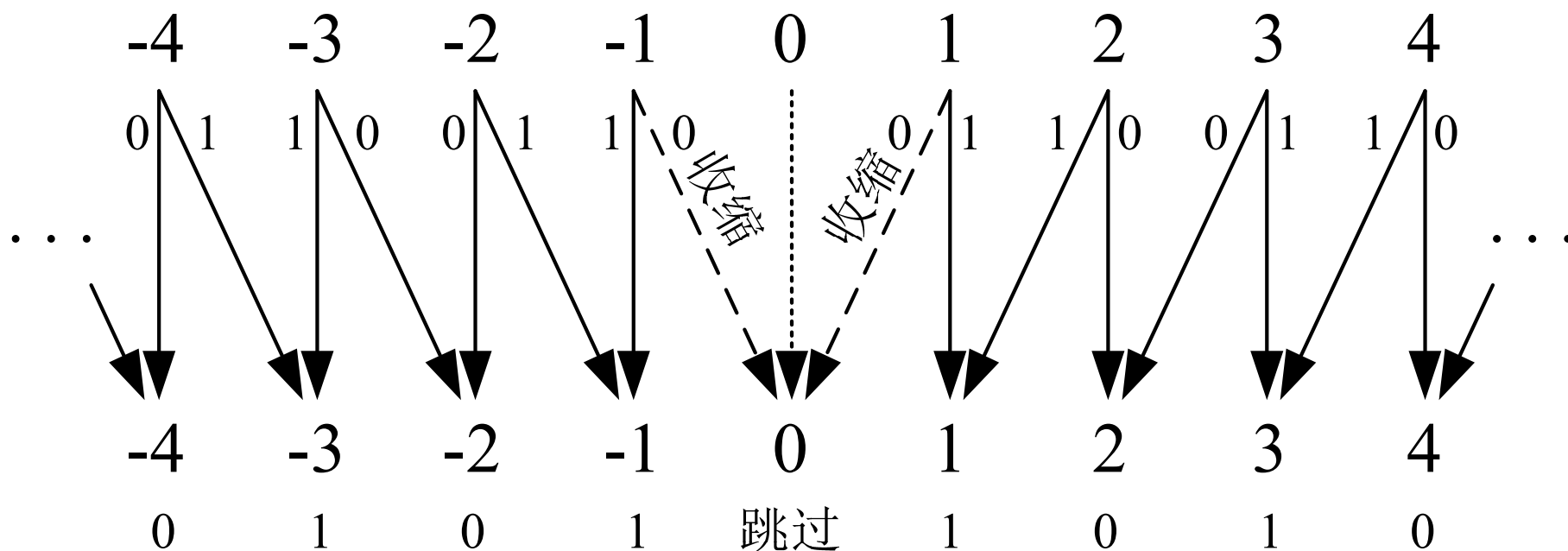
JSteg引起直方图异常

- 统计表明普通JPEG图像的量化后DCT系数直方图一般具有如下特性：
 - ◆ 绝对值越大的DCT系数出现的频数越低，对应直方图单元的值越小；
 - ◆ 随着DCT系数绝对值的增加，其频数下降的幅度减小。
- 不难看出，JSteg算法具有与空域LSB替换隐写算法一样的安全性缺陷，会引起量化后DCT系数直方图出现值对趋于相等的现象。因此，JSteg隐写也无法抵抗卡方检验分析。

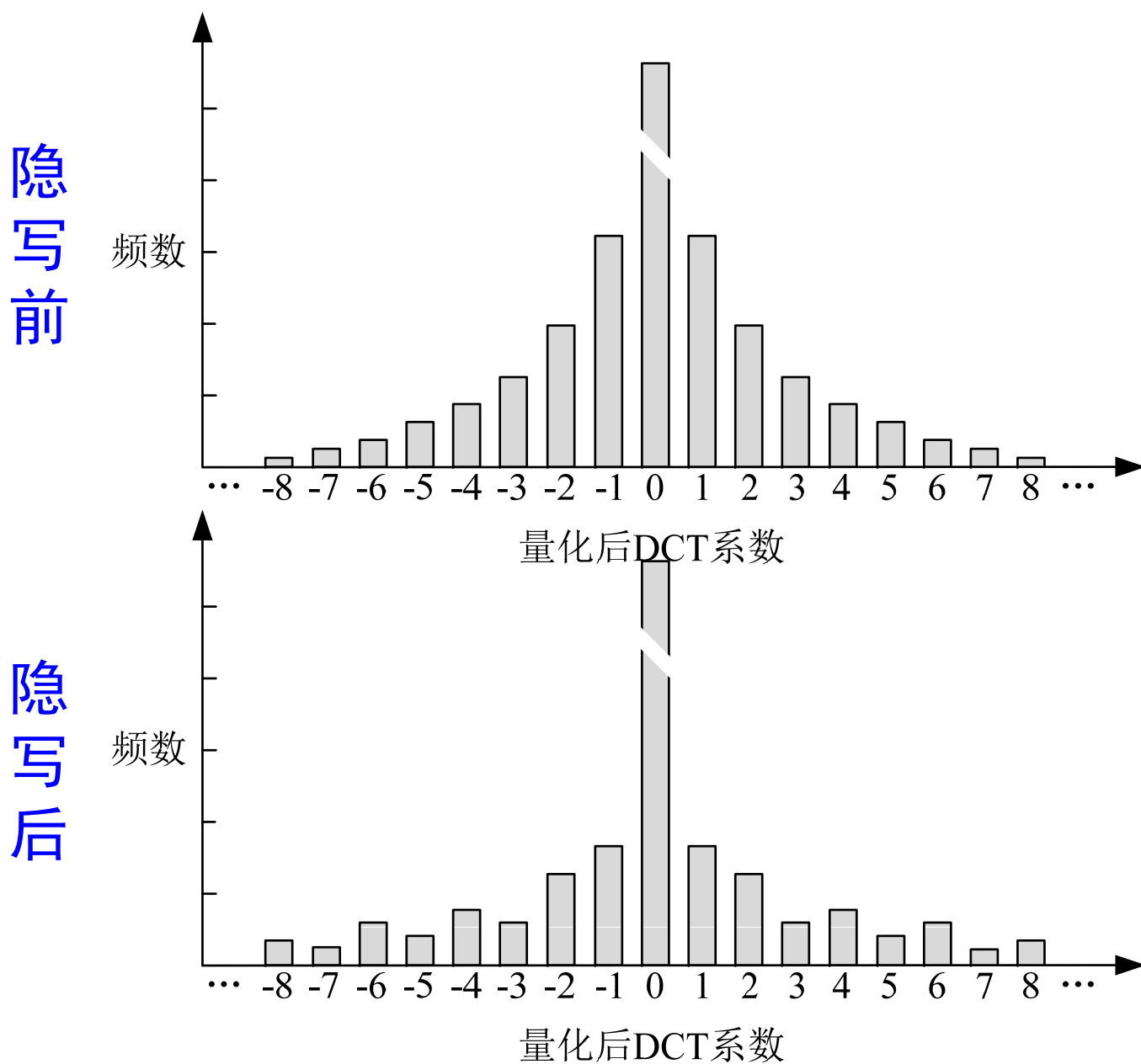
➤ F3

- 改进JSteg算法
- 策略1: 信息嵌入时, 若DCT系数的LSB与待嵌入的秘密信息比特相同, 则不做改动; 否则, 将该DCT系数的绝对值减1
- 策略2: 值为0的量化DCT系数不嵌入任何信息
- 不足: DCT系数直方图异常 (偶数比奇数bin高)

F3 隐写示意图



F3 隐写前后图像的灰度直方图



F3 引起直方图异常

- 在DCT系数直方图中不会出现值对趋于相等的现象，因而F3隐写可以抵抗卡方分析，在一定程度上弥补了JSteg隐写的不足。
- F3隐写却引起了新的直方图异常，即偶数值处单元的高度可能低于比其绝对值小1的奇数单元。
- 由于在值为+1、-1的量化后DCT系数上嵌入秘密信息比特0时，会形成无效的嵌入，继而将这些待嵌入的秘密信息比特0分摊到其它系数上，且最终由修改后为偶数的DCT系数来表征。

➤ F4

➤ 改进F3算法

➤ 策略1：利用正奇系数和负偶系数代表1，

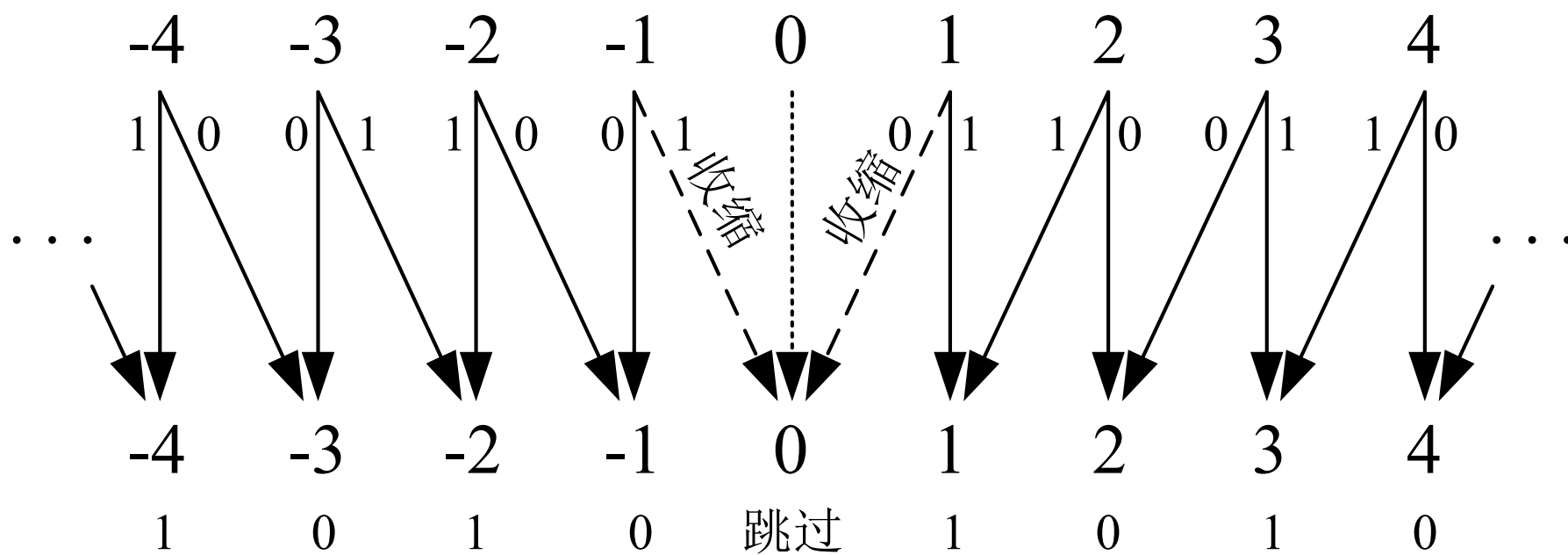
利用正偶系数和负奇系数代表0；

绝对值减1操作（继承F3算法）

➤ 策略2：值为0的量化DCT系数不嵌入任何信息

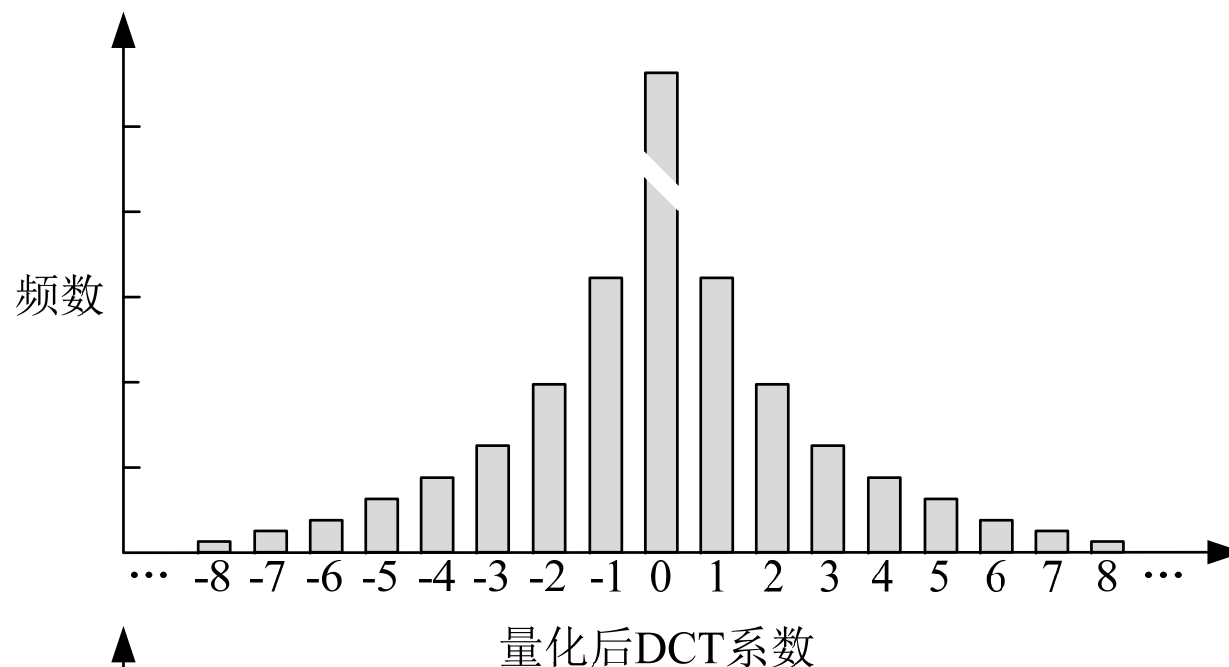
➤ 不足：直方图特性得到保持，但嵌入效率较低

F4 隐写示意图

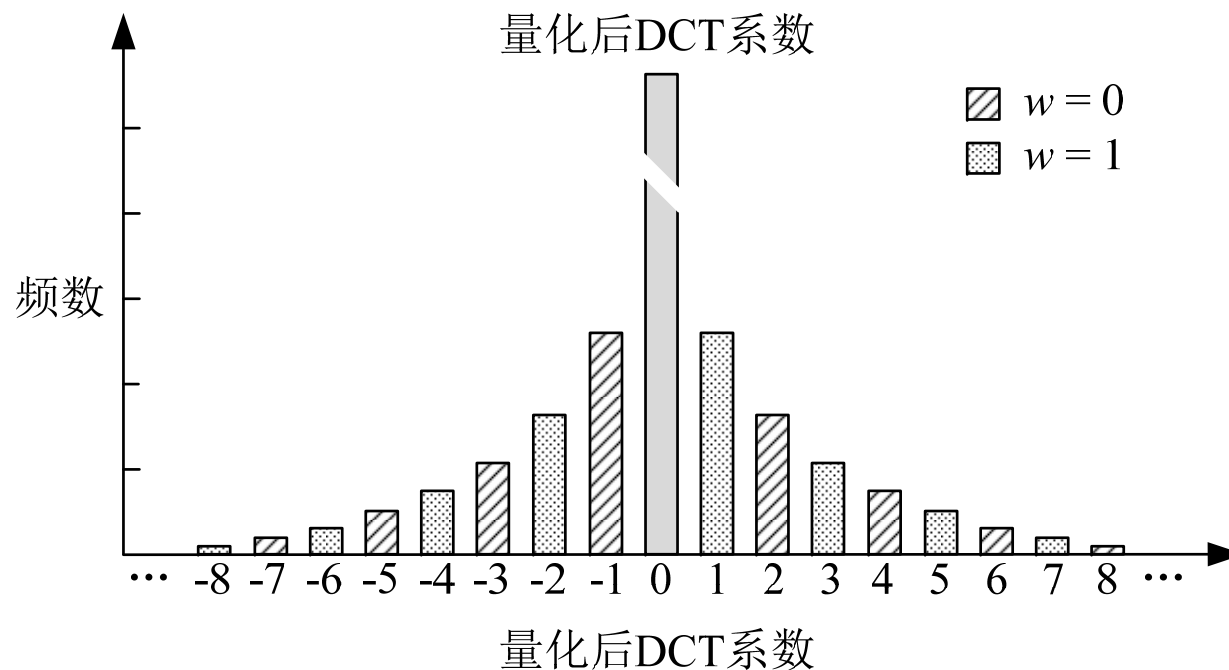


F4 隐写前后图像的灰度直方图

隐
写
前



隐
写
后



F5 隐写

- 改进F4算法
- 策略1：混洗（置乱），使隐密图像的质量较均衡
- 策略2：矩阵编码 $(1, n, k)$ ，提高了嵌入效率（平均每个改动嵌入的比特数）
- 不足：容量够高吗？抗隐写分析能力足够强吗？

F5 隐写

表 不同 k 值下的矩阵编码性能

k	n	嵌入效率 E	数据利用率 R (%)
1	1	2.00	100.00
2	3	2.67	66.67
3	7	3.43	42.86
4	15	4.27	26.67
5	31	5.16	16.13
6	63	6.09	9.52
7	127	7.06	5.51
8	255	8.03	3.14
9	511	9.02	1.76
10	1023	10.01	0.98

F5 隐写

F5 隐写的秘密信息嵌入主要包括以下步骤：

- ① 读取 JPEG 图像的量化后 DCT 系数，对这些 DCT 系数进行混洗。
- ② 统计可用的 DCT 系数，依据待嵌入的秘密信息长度计算矩阵编码三元组 $(1, n, k)$ 。
- ③ 依序取 n 个非零的 AC 系数和 k 个秘密信息比特，利用矩阵编码进行嵌入。计算判断载体系数是否需要改动：若不需要，则继续下一组的嵌入；若需要，则改变相应系数的 LSB。若改动后有新的载体系数变为 0，则此次嵌入无效，需重新确定 n 个可用系数进行矩阵编码嵌入。
- ④ 重复执行③，直到秘密信息嵌入完毕。
- ⑤ 对修改后 DCT 系数进行逆混洗，恢复 DCT 系数原始排列顺序，保存隐写后 JPEG 图像。

2.1 空域信息隐藏技术

2.2 变换域信息隐藏技术

2.3 其它信息隐藏技术

(1) 人的生理模型技术

(2) 网格水印算法

(3) 无载体信息隐藏技术

.....

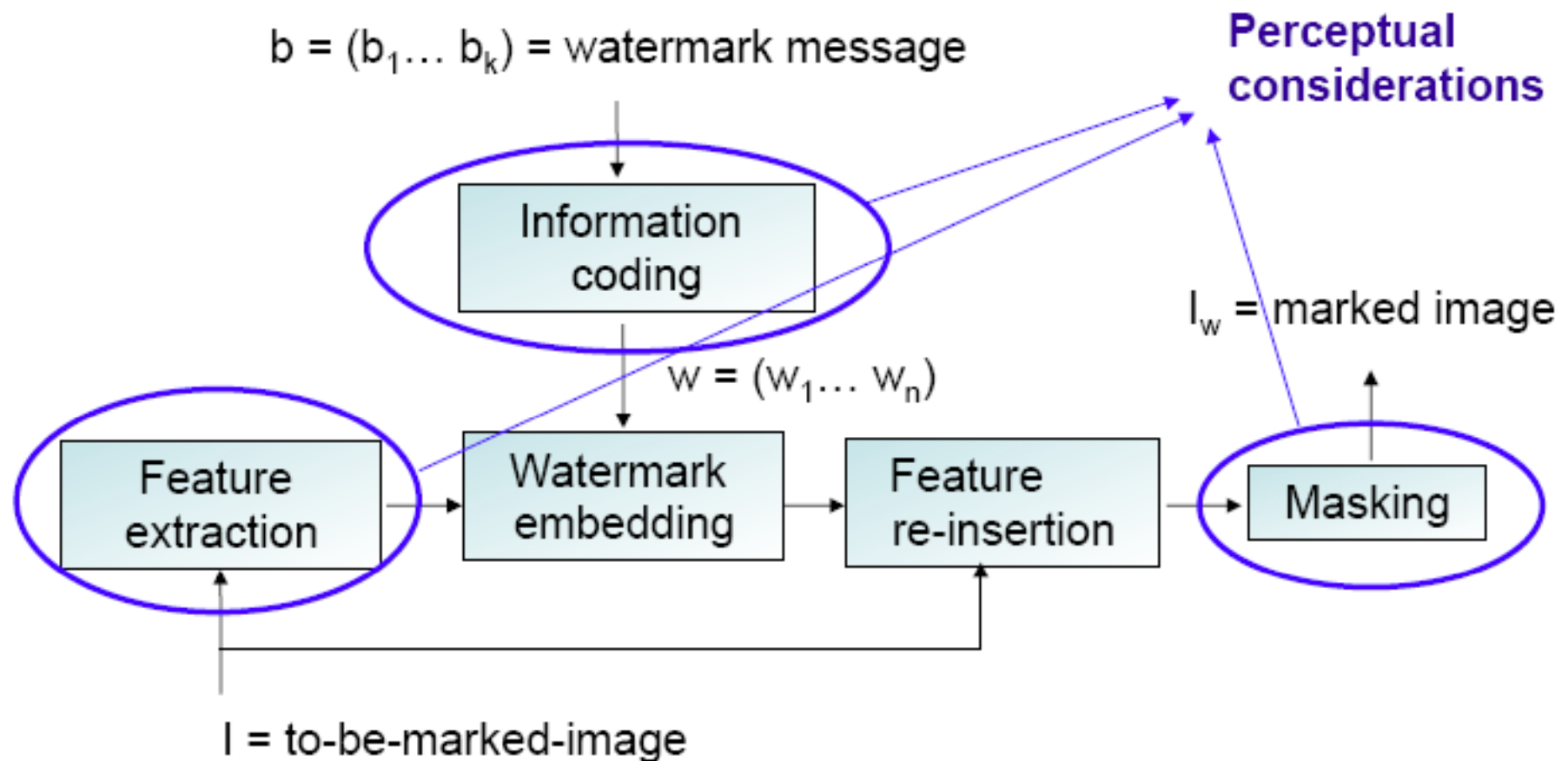
基于HVS的信息隐藏技术

➤ 人的生理模型技术

人的生理模型包括人类视觉系统 **HVS** 和人类听系统 **HAS**。该模型不仅被多媒体数据压缩系统利用，同样可以供信息隐藏技术利用。它的基本思想是利用从模型中导出的 **JND** (**Just Noticeable Difference**) 描述来定媒体（图像、声音、视频）的各个部分所能容忍的隐秘信号的最大强度，从而能够避免破坏视觉（听觉）质量，因而这一方法同时具有好的透明性和稳健性。

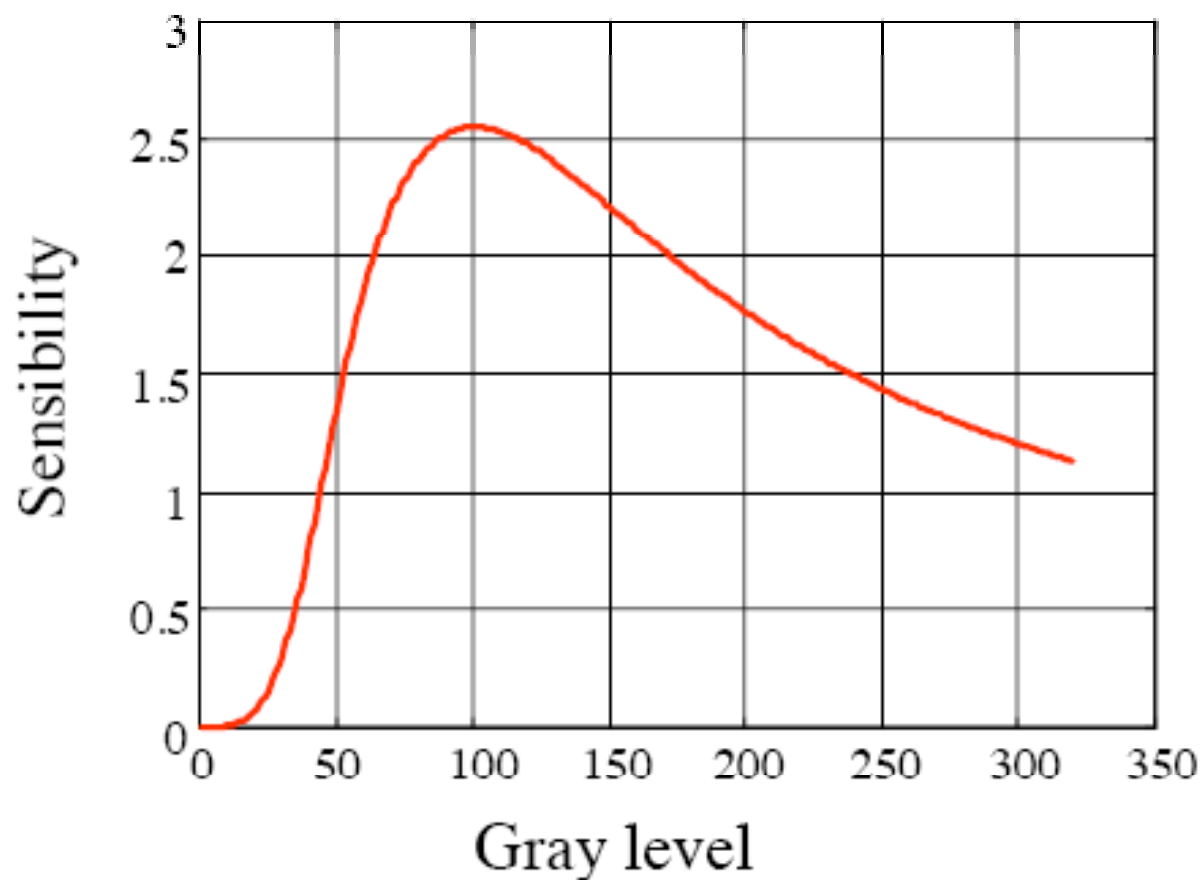
Blind watermark embedding

- Embedding goes through the definition of a watermark signal (\mathbf{w}) to be mixed with the host features (\mathbf{f})



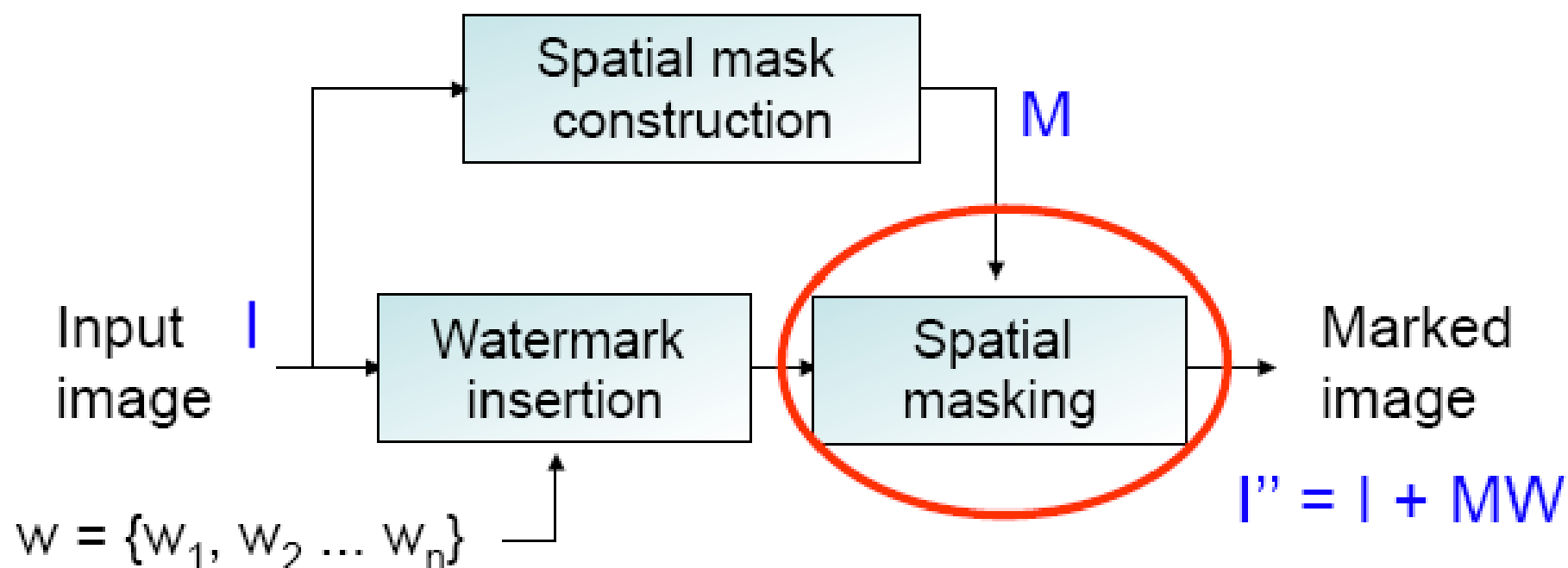
HVS (Human Visual System)

➤ 人眼视觉敏感度 ~ 背景信号灰度级



HVS (Human Visual System)

- 基于HSV的水印嵌 (自适应地控制水印嵌入强度)



HVS (Human Visual System)

➤ Spatial Mask



Lenna image



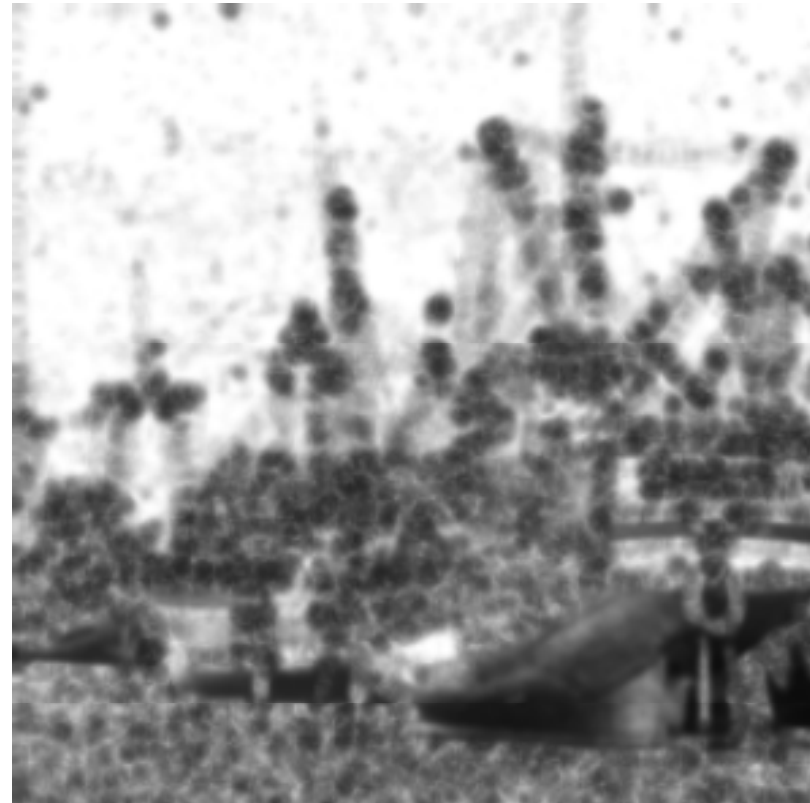
Watermark sensibility map for
Lenna image (1-M)

HVS (Human Visual System)

➤ Spatial Mask



Boat image



Watermark sensibility map for
Boat image (1-M)