图像信息隐藏算法二

钮心忻、杨榆、雷敏
北京邮电大学 计算机学院 信息安全中心
yangyu@bupt.edu.cn

图像信息隐藏技术分类

- 图像隐写算法（隐蔽性）
- 图像鲁棒性水印（能够抵抗各种信号处理、攻击）
- 图像脆弱性水印（完整性验证、篡改定位）

图像水印算法介绍

- 普通图像水印
- 图像鲁棒性水印
- 图像脆弱性水印

鲁棒水印

- 什么是鲁棒水印?
  - Watermarks designed to survive legitimate and everyday usage of content are referred to as ***robust watermarks***.
- 什么是安全水印?
  - Whereas robust watermarks are designed to survive normal processing, secure watermarks are designed to resist any attempt by an adversary to thwart their intended purpose.

鲁棒水印

- 鲁棒水印和安全水印的关系
  - 安全水印必须是鲁棒的，然而，仅具有鲁棒性的水印远未达到安全性的要求。
- 鲁棒水印的性能权衡
  - 设计鲁棒水印前，考察水印可能遭受的“处理”是一个重要的环节。
  - 鲁棒水印需要抵抗的处理包括：有损压缩，数模/模数转换，录音，打印扫描，语音重放，和二次录音，去噪，格式转换等等。

鲁棒水印

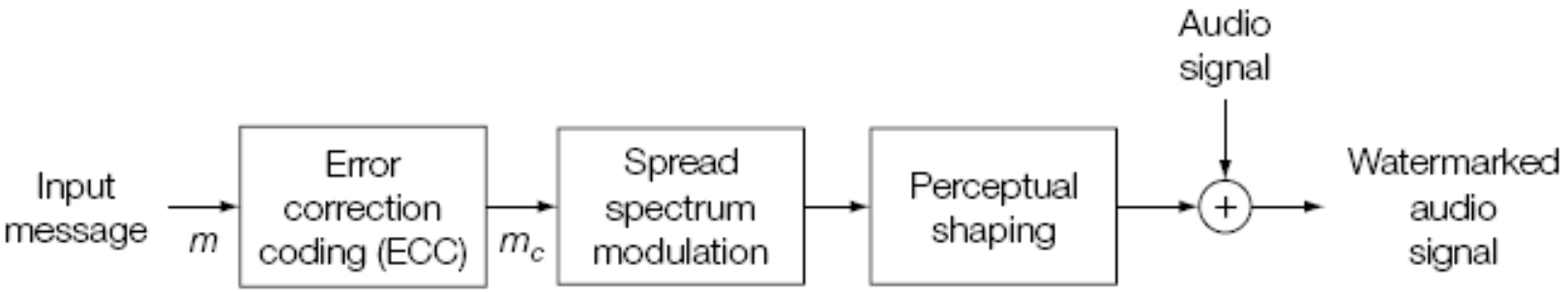
- 鲁棒水印的性能权衡
  - 增强鲁棒性通常会牺牲其他性能，例如计算开销增大，容量降低，透明性下降，甚至牺牲对于其他操作的稳健性。
  - 因此，通常不会在一个算法中抵抗所有处理。
  - 例如：用于监测广告的水印需要抵抗广播过程中的信号处理，包括数模转换，有损压缩等等，但这个过程不会出现旋转或半色调处理。

鲁棒水印通用设计策略

- 冗余嵌入
  - *Redundant embedding* can increase robustness to cropping, filtering, and additive noise.
  - The redundancy can be in the sample domain, the frequency domain, or any other domain in which only part of the signal is distorted by processing.
- 案例
  - 水印信息经过纠错编码后，再嵌入载体。
  - 每一比特水印，反复嵌入N次，提取时取平均。

鲁棒水印通用设计策略

- 扩频水印
  - *Spread spectrum* watermark reference patterns are redundant in the spatial and frequency domains.
  - These provide general robustness against filtering, additive noise, and cropping.

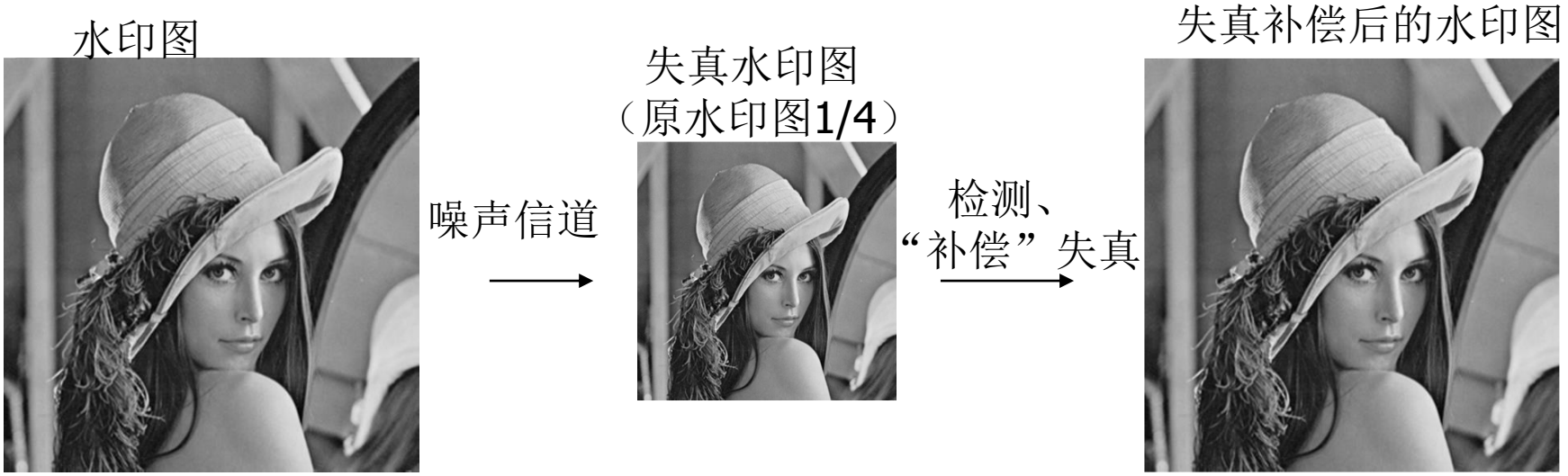


鲁棒水印通用设计策略

- 在重要感知“区域”嵌入水印
  - Embedding watermarks in perceptually significant components of content ensures their robustness against any processing that maintains acceptable fidelity.

鲁棒水印通用设计策略

- 检测时，“补偿”失真
  - If a detector can determine that a specific process has been applied to a Work since the time it was watermarked, the detector might either invert that process, or apply it to the reference mark.



鲁棒水印通用设计策略

- 嵌入时，预补偿“失真”
  - Sometimes we can anticipate that watermarks will be subject to one of a small collection of possible distortions.
  - In these cases, it may be possible to apply the inverse distortion during the embedding process.

预补偿失真（放大为原水印图的4倍）

失真水印图（预补偿水印图的1/4）

噪声信道

检测

水印图（预补偿和失真相抵消，直接提取）

鲁棒水印应对“几何失真”的策略

- 几何失真
  - Geometric distortions, such as temporal delay or spatial scaling, are generally more difficult to handle than valumetric distortions,
  - and robustness against them is a current topic of research.
  - 时域几何失真：$\mathbf{c_n}[t] = \mathbf{c}[st + \delta].$
  - 空域几何失真：$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \end{bmatrix},$$

鲁棒水印应对“几何失真”的策略

- 典型几何失真

鲁棒水印应对“几何失真”的策略

- 穷举法
  - *Exhaustive search* entails inverting a large number of possible distortions,
  - and testing for a watermark after each one.
  - As the number of possible distortions increases, the computational cost and false positive probability using this approach can become unacceptable.

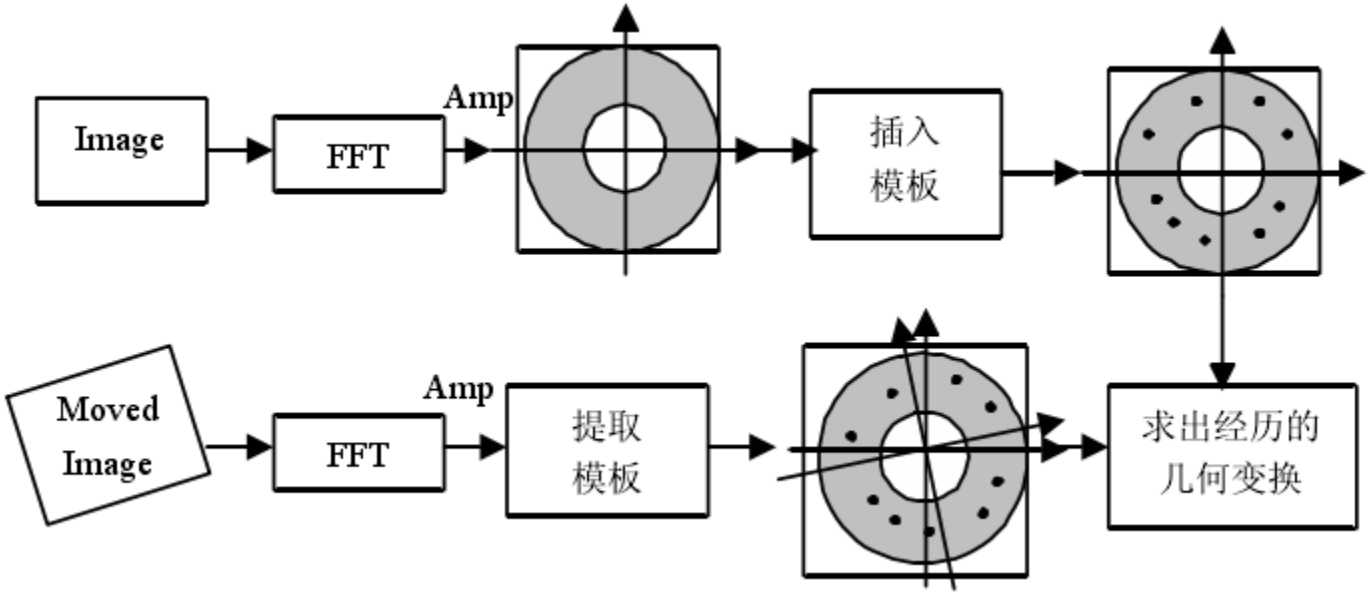
鲁棒水印应对“几何失真”的策略

- 自相关
  - In the autocorrelation approach, we embed a periodic watermark and register based on the peaks in a Work’s autocorrelation pattern.

鲁棒水印应对“几何失真”的策略

- 同步法
  - Synchronization/registration patterns can be embedded in content to simplify the search.
  - These prevent an increase in the false alarm rate
  - and are usually more computationally efficient than an exhaustive search.
  - However, they introduce two failure modes: failure to correctly detect the registration pattern and failure to detect the watermark after registration.

鲁棒水印应对“几何失真”的策略



- 模板是通过增加所选择系数的幅值产生一个人为的局部峰值点产生的。通过模板点和检测到的极值点的匹配，确定水印图像经历的几何变换，
- 一旦经历的几何变换确定，再对水印图像进行逆变换，就可在Fourier变换域内检测到水印

鲁棒水印应对“几何失真”的策略

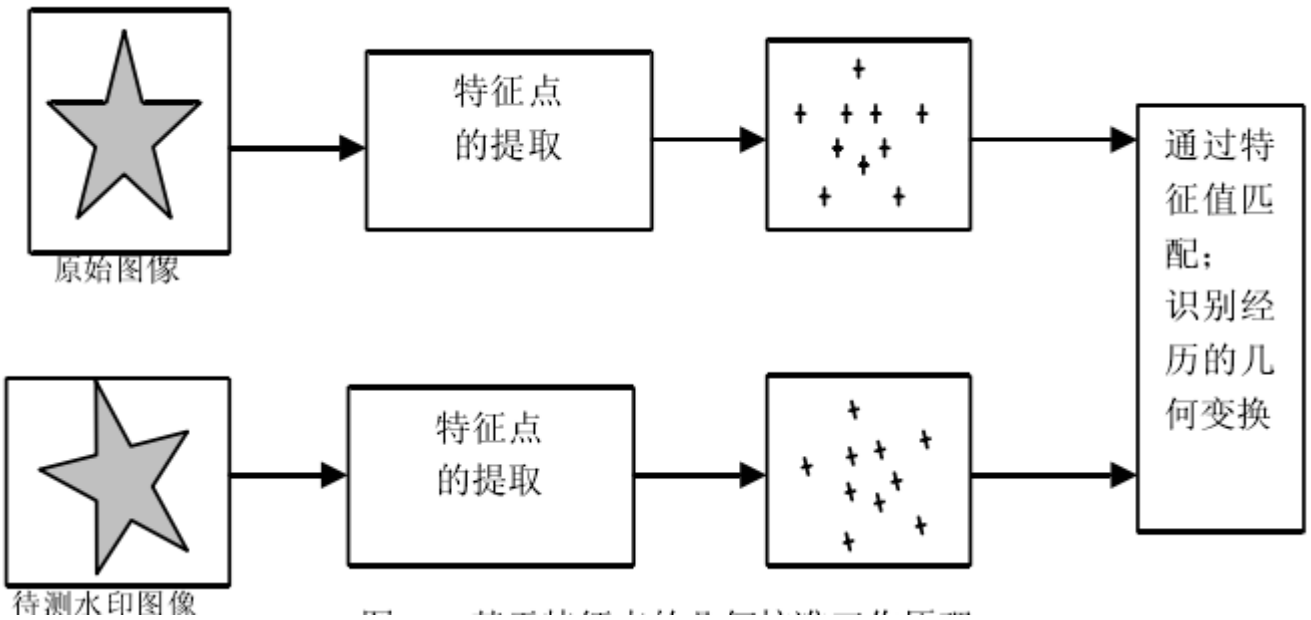
- 隐式同步
  - In *implicit synchronization*, we register according to feature points found in the original, unwatermarked Work. This depends on development of a reliable feature-extraction method.
  - 例如
  - 有算法策略为：检测到导频信号后嵌入水印，导频信号定义为那些快速爬升到峰值的样点。这样的导频信号对于延时失真具有鲁棒性

鲁棒水印应对“几何失真”的策略

- 隐式同步案例
  - 我们通常在时域（例：像素）或变换域（例：DWT）中嵌入水印，
  - Kutter等提出了“第二代数字水印”概念。
  - 利用图像特征进行水印的嵌入和提取。
    - 方法一：以图像特征为参考，嵌入和提取水印。
    - 方法二：直接嵌入水印到特征中。

鲁棒水印应对“几何失真”的策略

- 隐式同步案例



鲁棒水印应对“几何失真”的策略

- 不变水印
  - *Invariant watermarks* can be constructed using such techniques as log-polar Fourier transforms.
  - These remain unchanged under certain geometric distortions, thereby eliminating the need to identify the specific distortions that have occurred.

强鲁棒性水印

- 抗打印扫描水印
- 打印扫描特点
  - 印刷品的灰度与数字图像的灰度表示方法完全不同。
  - 在激光打印、各种印刷中，要用二值输出表示图像的灰度层次，因此采用半色调技术。
  - 半色调技术的实质是：在一个小点阵的网格内用黑点的多少表现灰度层次。

强鲁棒性水印

- 打印扫描特点
  - 由半色调复合点的形状、激光束的扩散、纸张的吸水特性和光滑度等因素造成的半色调复合点变化，也常会导致输出图像变得模糊不清。
  - 在使用扫描仪进行扫描的过程中也可能造成图像畸变。

强鲁棒性水印

- 打印扫描过程的失真
  - 像素失真
    - 主要源于打印的D/A过程的半色调处理，
    - 以及扫描的A/D过程的噪声和量化影响。
- 几何失真
    - 主要由扫描过程引起的旋转和缩放失真 。

强鲁棒性水印

- 几何失真矫正
  - 提取图像边缘，并利用RADON投影变换来检测图像的倾角 。完成后，逆向旋转图像。
  - 图像的RADON变换是将原始图像变换为它在各个角度的投影表示。图像 f(x,y) 在任意角度 θ 上的RADON投影定义为

$$R_\theta(x') = \int_{-\infty}^{\infty} f(x'\cos\theta - y'\sin\theta, x'\sin\theta - y'\cos\theta)dy'$$

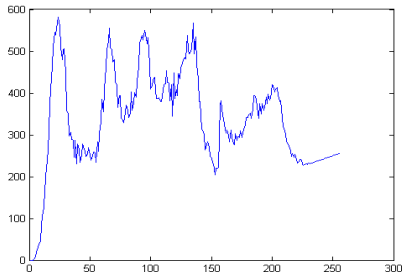
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

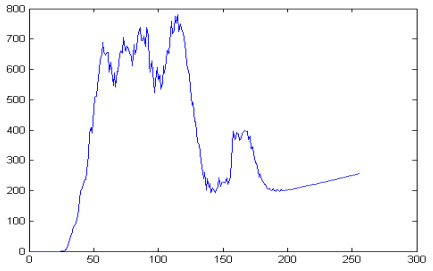
打印扫描对图像的影响

- 考察哪些数据在图像经过打印和扫描后没有改变或者改变较小
- 做一些统计分析

打印扫描对图像灰度值的影响

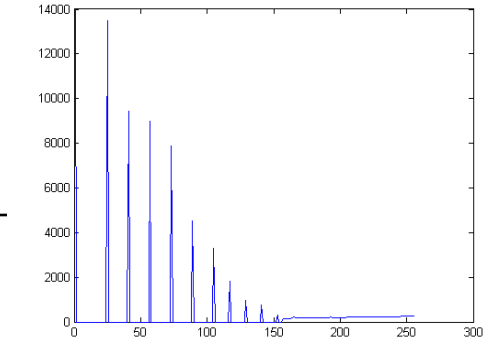
- 统计256x256的LENA灰度图像的像素值中每个灰度值出现的个数，研究其像素值的分布情况

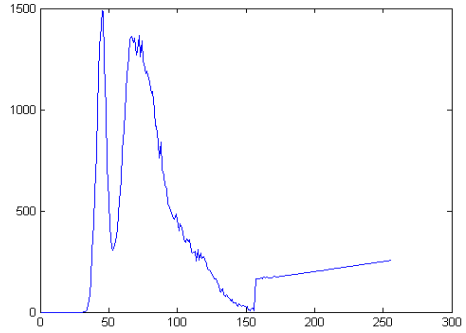




(a) 原始图像　　(b) 打印扫描后

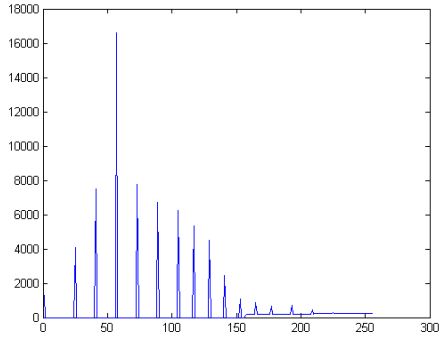
LENA图像的像素值分布的直方图

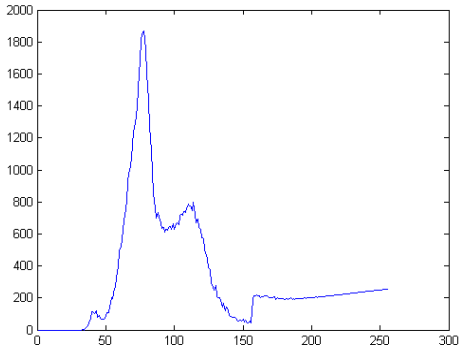




(a) 原始图像　　(b) 打印扫描后

COUPLE图像的像素值分布的直方图





(a) 原始图像　　(b) 打印扫描后

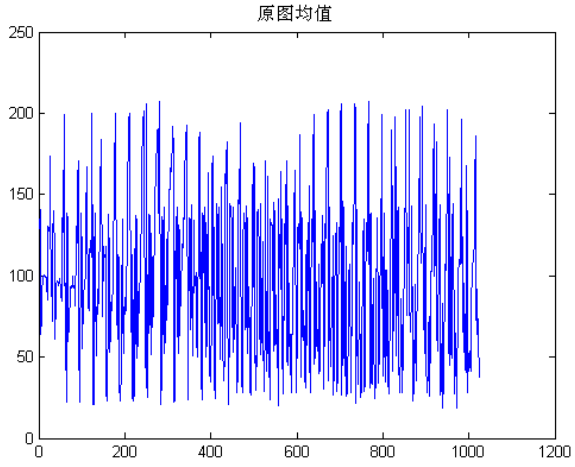
GIRL图像的像素值分布的直方图

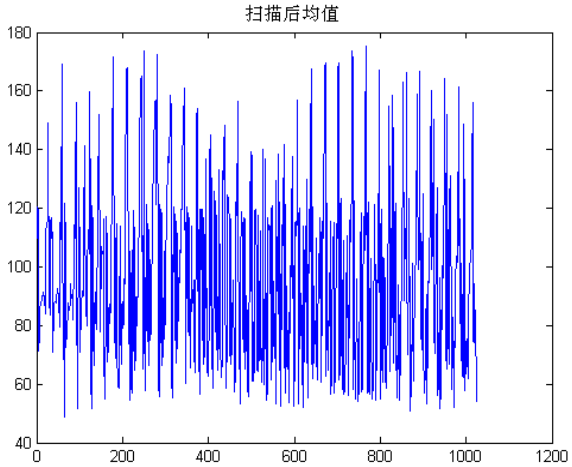
打印扫描对图像灰度值的影响

- 从像素值分布的图形上，可以看出打印扫描前后的差别比较明显，这种较大的差异将导致在空间域水印检测非常困难

打印扫描对图像均值的影响

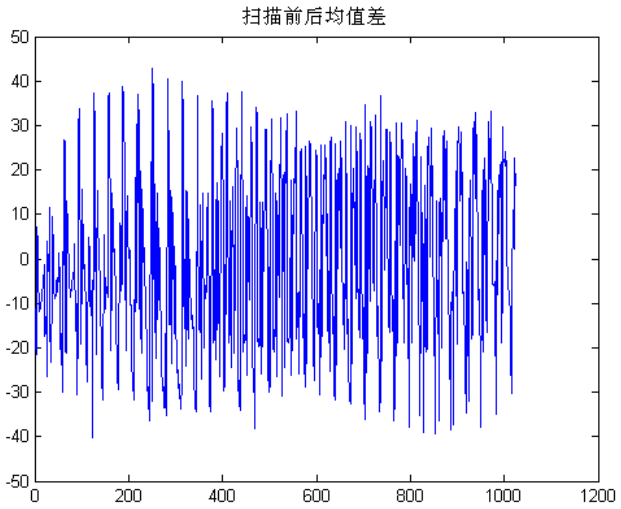
- 将256x256的LENA灰度图像按照8x8进行分块，得到1024个小块，计算每个小块所有像素的平均值
- 比较原始图像的均值和打印扫描后的均值曲线





(a) 原始图像　　(b) 打印扫描后

LENA图像的均值分布图



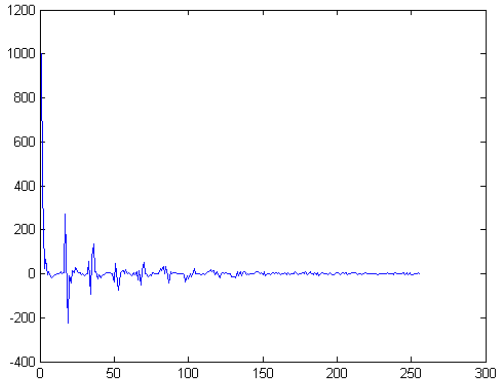
两曲线之差

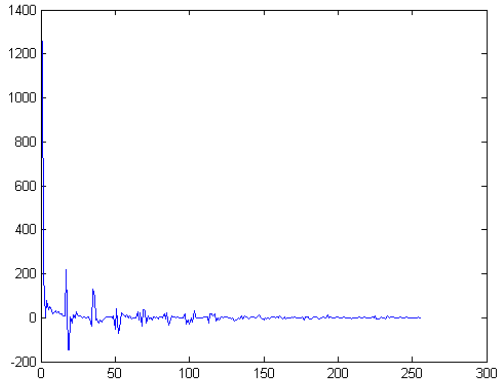
小结

- 原始图像在打印扫描后均值的变化并无规律性
- 通过图像的灰度值、均值构造能够抵抗打印扫描的数字水印算法是困难的
- 结论：在空间域难以建立对打印扫描鲁棒的数字水印算法

打印扫描对图像变换域的影响

- 将图像按照16x16分块的形式，打印扫描前后的LENA灰度图像的对应块DCT系数变化的情况





(a) 原始图像　　(b) 打印扫描后

LENA图像中的一个16x16分块的DCT系数变化图

打印扫描对图像变换域的影响

- 对256x256的LENA灰度图像按照整幅图像在打印前后进行对比
- 按照8x8分块对LENA灰度图像在打印扫描前后的DCT系数的变化情况逐块进行比较
- 结论：从图像在打印扫描前后的变化趋势来看，打印扫描对图像DCT系数影响较小

基于人类视觉特性的图像块分类

- 为了保证水印的不可见性，抗打印和扫描的水印嵌入容量和强度不能太大
- 但为了保证水印经过打印扫描后仍能被检测出来，又要求嵌入足够量和足够强度的水印
- 为此，需要结合人类视觉特性，选择适当区域和水印嵌入强度，在满足不可见的前提下，最大限度的提高水印嵌入强度

人类视觉系统（HVS）特点

- 人类视觉系统（HVS）对于一幅图像的每个区域的敏感度是不一样的
- HVS对于亮度变化大的区域的敏感度要大于亮度变化小的区域
  - 高信息量区域：亮度变化大的区域
  - 低信息量区域：亮度变化小的区域
- 在高信息量区域中，HVS对亮度突然变化的区域最敏感
  - 关键区域：亮度突然变化的区域，一般是图像中包含信息量最大，对人们的理解最为重要的部分
  - 随机纹理区域：具有规则变化的区域（如窗帘、头发等），人眼会产生一定的适应性，以至于很容易在人的意识中遗忘，这些区域包含的内容意义并不大，对图像理解不起决定性作用

结合HVS对图像分类

- 将图像块划分为三类
  - 低信息量区域
  - 随机纹理区域
  - 关键区域
- HVS对前两类图像块不敏感，所以叠加的水印分量的强度可较强；HVS对关键区域对最敏感，因此叠加的水印强度应较弱

基于亮度变化率的分类

- 一幅图像中某一个子块，平均亮度

$$L_m = \sum_{i=1}^{n} l_i / n$$

- l_i：每个像素的亮度分量值
- n：表示子块中像素的个数

基于亮度变化率的分类

- 亮度变化：2个相邻像素亮度分量之差的绝对值称为两个像素间的亮度变化

$$D_{i,j} = \left| l_i - l_j \right|$$

- 亮度变化率：所有相邻像素亮度变化的总和与像素总数的比

$$C_m = \sum_{i=1}^{n} \sum_{j=1}^{n} D_{i,j} / n$$

区分高、低信息量区域

- 亮度变化率反映了图像子块内部亮度变化的大小和快慢
- 根据选定的阈值将图像划分成“低信息量区域”和“高信息量区域”
  - 低信息量区域：亮度变化率小于阈值
  - 高信息量区域：亮度变化率大于阈值

高信息量区域的进一步划分

- 亮度平均变化率：某一个子块与其所有相邻子块（8个）的亮度变化率之差的均方根，称为该子块的亮度平均变化率

$$R_m = \sqrt{\sum_{k=1}^{8}(C_k - C_i)^2 / 8}$$

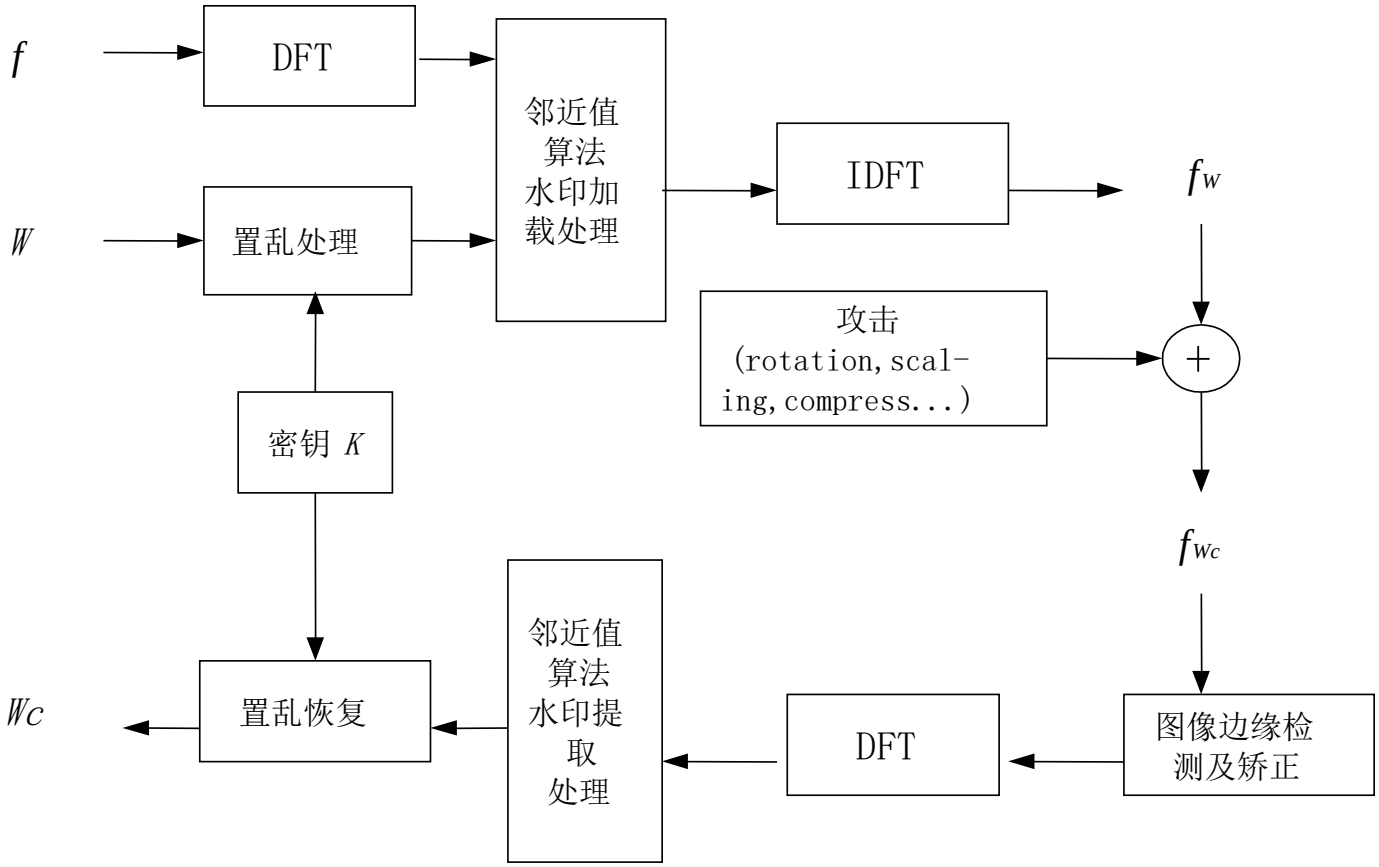
- 亮度相对变化率：子块的亮度平均变化率与该区域的平均亮度的比值，称为该子块的亮度相对变化率

$$V_m = R_m / L_m$$

高信息量区域的进一步划分

- 根据设定的阈值可将“高信息量区域”进一步划分成“随机纹理区域”和“关键区域”
  - 随机纹理区域：亮度相对变化率小于阈值
  - 关键区域：亮度相对变化率大于阈值

抗打印扫描水印算法（一）——基于量化的方法



抗打印扫描水印算法（一）——基于量化的方法

- 嵌入算法
  - 对载体图像作DFT变换
  - 以密钥K为种子对水印图像随机置乱
  - 根据水印数据（0或1），利用邻近值算法，对载体图像的DFT中低频系数的幅度进行修改，嵌入水印信息，DFT系数的相位保持不变
  - 对修改后的DFT变换域系数，作IDFT，得到嵌入水印的图像

抗打印扫描水印算法（一）——基于量化的方法

- 提取算法
  - 对受到攻击后的含水印图像进行边缘提取，和图像的尺寸提取（size）
  - 对边缘图像进行RADON变换，计算出图像的旋转角度θ
  - 对图像反向旋转θ角，作size大小的DFT变换
  - 利用邻近值算法，从DFT的中低频系数中提取出置乱后的水印信息
  - 以密钥*K*为种子，对数据进行置乱恢复，提取嵌入的水印

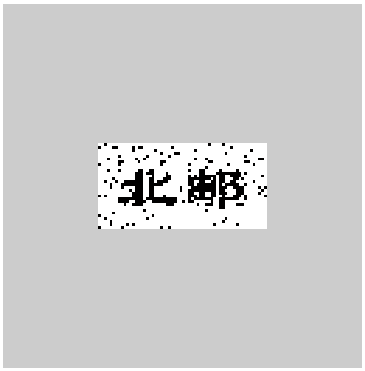
抗打印扫描水印算法（一）
——基于量化的方法之仿真结果

原始图像
（256×256）

含水印图像
（256×256）

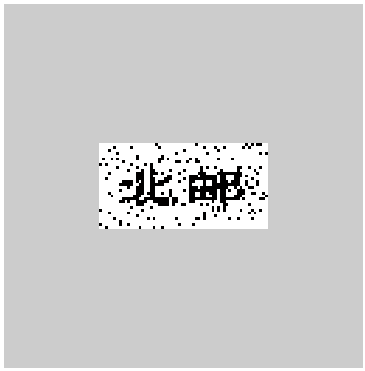
原始水印
（60×30）

抗打印扫描水印算法（一）
——对缩放、压缩、加噪攻击的稳健性



从缩小到200×200的
含水印图像中提取的水印

从JEPG压缩后的
含水印图像中提取
的水印
（quality=75）



从添加高斯噪声后的
含水印图像中提取的水印
（噪声方差=0.0005）

抗打印扫描水印算法（一）
——对缩放旋转攻击的稳健性

放大和旋转后的含水印图像
（size=280×280, θ =2度）

提取的水印

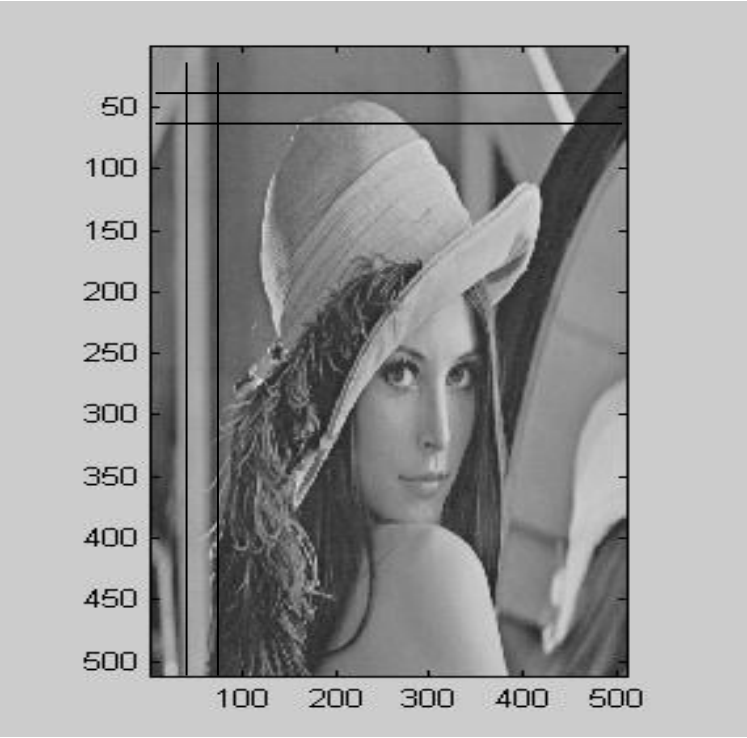
抗打印扫描水印算法（二）——基于系数比较的方法

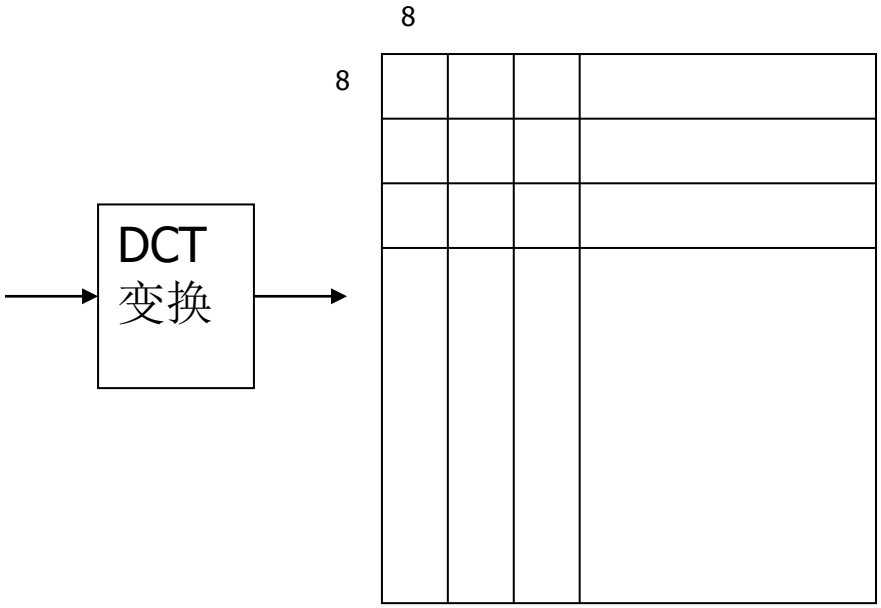
- 虽然打印扫描对图像的DCT系数影响较小，但不同的打印机和扫描仪对图像的影响不同，因而很难找到打印扫描前后DCT系数之间的数量关系
- 由于图像在打印扫描前后的DCT系数的相对关系应该是基本一致的，因而可以通过改变DCT中频系数的相对关系来嵌入数字水印

抗打印扫描水印算法（二）——基于系数比较的方法

- 二维DCT变换
  - 图像压缩标准（JPEG）的核心: DCT变换
  - M × N维的图像，进行二维DCT变换，得到M×N的DCT系数
  - 系数按照Zig-Zag次序排列，左上角为直流系数，其余为交流系数

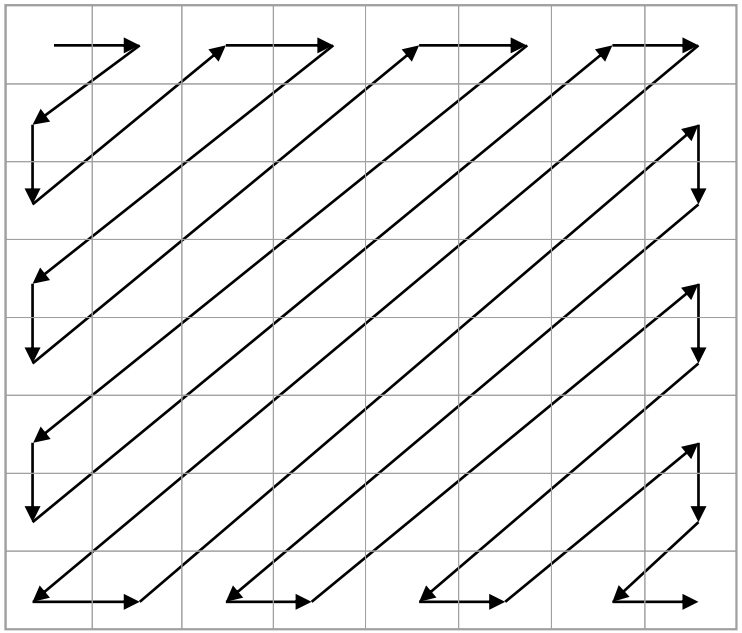
抗打印扫描水印算法（二）——基于系数比较的方法





抗打印扫描水印算法（二）——基于系数比较的方法

- 8×8分块，DCT变换，ZigZag 扫描



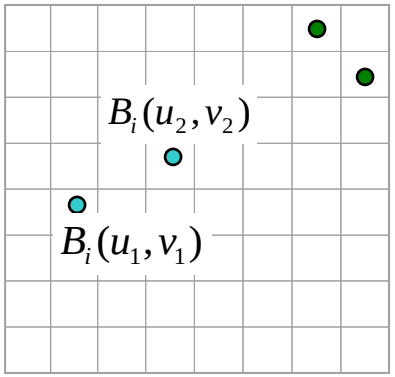
ZigZag扫描方式

- 若用$x[i,j]$表示8×8分块第i行第j列像素，则ZigZag扫描后，像素矩阵排列为：
- $x[0,0], x[0,1], x[1,0], x[2,0], x[1,1], x[0,2] \cdots x[0,7], x[1,6], x[2,5], x[3,4] x[4,3], x[5,2], x[6,1], x[7,0] \cdots x[6,7], x[7,6], x[7,7]$

抗打印扫描水印算法（二）——基于系数比较的方法

- DCT系数特点
  - 左上角部分为直流和低频，右下角部分为高频，中间区域为中频。
  - 直流分量和低频系数值较大，代表了图像的大部分能量，对它们做修改会影响图像的视觉效果。
  - 高频系数值很小，去掉它们基本不引起察觉
  - 最好的水印嵌入区域就是在中频部分

抗打印扫描水印算法（二）——基于系数比较的方法

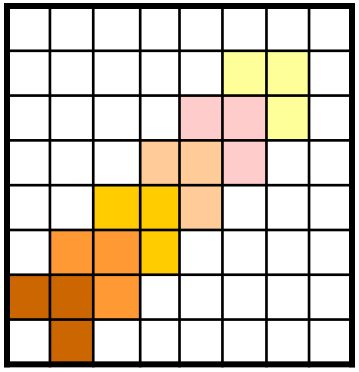


- 原理
  - 利用载体中两个特定系数的相对大小来代表隐藏的信息
- 嵌入
  - 划分图像为若干8×8小块
  - 对各个小块分别做二维DCT变换
  - 选择其中的两个位置，若用 (u_1,v_1) 和 (u_2,v_2) 代表所选定的两个系数的坐标
  - 隐藏1：调整系数使其满足 $B_i(u_1,v_1) > B_i(u_2,v_2)$
  - 隐藏0：调整系数使其满足 $B_i(u_1,v_1) < B_i(u_2,v_2)$

抗打印扫描水印算法（二）——基于系数比较的方法

- 提取
  - 划分图像为若干8×8小块
  - 对各个小块分别做二维DCT变换
  - 比较每一块中约定位置的DCT系数值，根据其相对大小，得到隐藏信息的比特串，从而恢复出秘密信息
- 特点
  - 不需原始图像
- 注意
  - 如果选定位置的两个系数相差太大，则对图像影响较大
  - 增大差距，代表无效
  - 应选择相近的值，如中频系数

抗打印扫描水印算法（二）——基于系数比较的方法



- 扩展
  - 利用DCT中频系数中的三个系数之间的相对关系隐藏信息
- 嵌入
  - 选择三个位置 (u_1,v_1) (u_2,v_2) (u_3,v_3)
  - 嵌入1：令 $B_i(u_1,v_1) > B_i(u_3,v_3)+D$ $B_i(u_2,v_2) > B_i(u_3,v_3)+D$
  - 嵌入0：令 $B_i(u_1,v_1) < B_i(u_3,v_3)-D$ $B_i(u_2,v_2) < B_i(u_3,v_3)-D$
  - 如果数据不符，则修改这三个系数值，使得它们满足上述关系
  - 选择参数D要权衡算法的稳健性和透明性
  - D越大，隐藏算法对于图像处理就越健壮，但是对图像的改动就越大，越容易引起察觉

抗打印扫描水印算法（二）——基于系数比较的方法

- 如果需要做的修改太大，则放弃该块，将其标识为“无效”
- “无效”：对这三个系数做小量的修改使得它们满足下面条件之一：

$$B_i(u_1,v_1) \le B_i(u_3,v_3) \le B_i(u_2,v_2)$$

或

$$B_i(u_2,v_2) \le B_i(u_3,v_3) \le B_i(u_1,v_1)$$

抗打印扫描水印算法（二）——基于系数比较的方法

- 实例
  - 某算法策略为，选D=0.5，系数调整为均值和均值±D。即，令$B_i(u_j, v_j)$为嵌入水印前系数，$B_i'(u_j, v_j)$为嵌入水印后系数。
    - $m = (B_i(u_1, v_1) + B_i(u_2, v_2) + B_i(u_3, v_3))/3$
  - 则若嵌1，调整为：
    - $B_i'(u_1, v_1) = B_i'(u_3, v_3) = m + D; B_i'(u_2, v_2) = m$
  - 若嵌入0，则反之：
    - $B_i'(u_1, v_1) = B_i'(u_3, v_3) = m - D; B_i'(u_2, v_2) = m$

抗打印扫描水印算法（二）——基于系数比较的方法

- 实例
  - 则根据该算法策略，下面几组系数，嵌入水印1，0，1后，变为什么？(D=0.5)
    - (1.3, 1.7, 1.5),(1.8,1.9,1.4),(1.8,2.3,2.2)
  - 解：
    - 第一组均值$m = (1.3 + 1.7 + 1.5)/3 = 1.5$
    - $B_i'(u_1, v_1) = B_i'(u_3, v_3) =$
    - $m + D = 1.5 + 0.5 = 2.0$
    - $B_i'(u_2, v_2) = m = 1.5$
    - 所以，第一组系数调整为(2.0,1.5,2.0)

抗打印扫描水印算法（二）——基于系数比较的方法

- 实例
  - 则根据该算法策略，下面几组系数，嵌入水印1，0，1后，变为什么？(D=0.5)
    - (1.3, 1.7, 1.5),(1.8,1.9,1.4),(1.8,2.3,2.2)
  - 类似地：
    - 第二组均值$m = (1.8 + 1.9 + 1.4)/3 = 1.7$
    - $B_i'(u_1, v_1) = B_i'(u_3, v_3) =$
    - $m - D = 1.7 - 0.5 = 1.2$
    - $B_i'(u_2, v_2) = m = 1.7$
    - 所以，第二组系数调整为(1.2,1.7,1.2)

抗打印扫描水印算法（二）——基于系数比较的方法

- 实例
  - 则根据该算法策略，下面几组系数，嵌入水印1，0，1后，变为什么？(D=0.5)
    - (1.3, 1.7, 1.5),(1.8,1.9,1.4),(1.8,2.3,2.2)
  - 类似地：
    - 第三组均值$m = (1.8 + 2.3 + 2.2)/3 = 2.1$
    - $B_i'(u_1, v_1) = B_i'(u_3, v_3) =$
    - $m + D = 2.1 + 0.5 = 2.6$
    - $B_i'(u_2, v_2) = m = 2.1$
    - 所以，第三组系数调整为(2.6,2.1,2.6)

抗打印扫描水印算法（二）——基于系数比较的方法

- 提取
  - 对图像进行DCT变换，比较每一块相应三个位置的系数，从它们之间的关系，可以判断隐藏的是信息“1”、“0”还是“无效”块，这样就可以恢复秘密信息

抗打印扫描水印算法（二）——基于系数比较的方法

- 实例：
  - 现有一幅采用系数比较法嵌入水印的图像，已知其系数为：
    - (1.7, 1.0, 1.8)，(2.7, 2.2, 2.7)，(1.7, 2.5, 1.8)，(1.7, 1.8, 1.9)
  - 则可从中提取的信息为?
  - 解：
    - 由1.7>1.0,1.8>1.0可知，这组系数嵌入的信息是1;
    - 由2.7>2.2,2.7>2.2可知，这组系数嵌入的信息是1;
    - 由1.7<2.5,1.8<2.5可知，这组系数嵌入的信息是0;
    - 由1.7<1.8<1.9可知，这组系数无效，没有嵌入;

抗打印扫描水印算法（二）——基于系数比较的方法之仿真结果

- 原始图像采用的Lena灰度图像，水印信息是7个英文字母YinCang，换算成ASCII码，用二进制表示就是56个比特

原始图像

嵌入水印后的图像

抗打印扫描水印算法（二）
——基于系数比较的方法之仿真结果

HP4VC激光打印机打印输出
紫光B6210扫描仪扫描的图像

HP6L激光打印机打印输出
紫光B6210扫描仪扫描的图像

抗打印扫描水印算法（二）——基于系数比较的方法

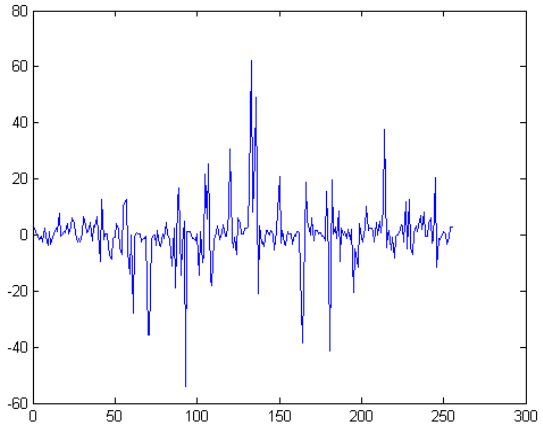
- 水印重复嵌入17次，并作两次周期扩展，提高其冗余度
- 实验结果：水印可以正确提取YinCang
- 结论：相对于打印，扫描对半色调图像中水印信息提取的影响更大。扫描仪的好坏，将直接决定印刷水印提取的成功与否

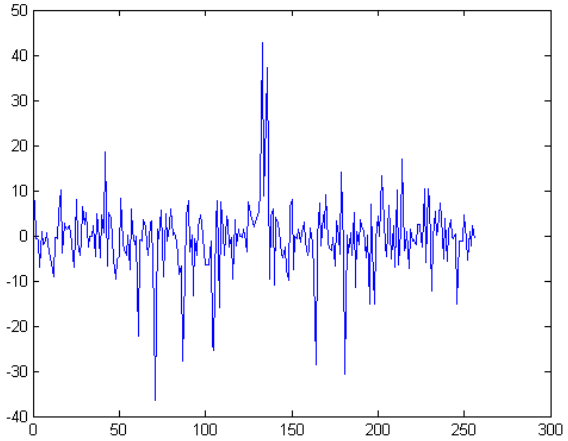
抗打印扫描水印算法（三）——基于系数分类的方法

- 为了寻找打印扫描前后DCT系数的特征, 对图像DCT系数进行分类
- 对于大小为256×256的LENA灰度图像，按16x16分块，可以得到256个小块，对每个小块作DCT变换，每个小块得到一个DC系数和255个AC系数$F_i(u,v)$
- i：第i个小块，u,v=0,1,…,15

抗打印扫描水印算法（三）——基于系数分类的方法

- 图像DCT系数共分为256类，分析每一类数据在打印扫描前后的变化





原始图像　　打印扫描后

LENA图像DCT系数F（8，7）的变化

抗打印扫描水印算法（三）——基于系数分类的方法

- 为了在 F(u,v) 中嵌入水印，需要改变F(u,v)中数据，使其具有某种数字特征；
- 不同的打印机和扫描仪对图像的影响不同，因而很难找到打印扫描前后DCT系数之间的数量关系；
- 通过改变 F(u,v) 中数据的正负号的数量来表达水印信息。

抗打印扫描水印算法（三）——基于系数分类的方法

- 选取水印的嵌入位置，即取定u,v，令
  - $n^{+}(u,v)$ 集合中正数的个数
  - $n^{-}(u,v)$ 集合中负数的个数
- 在 F(u,v) 中嵌入0时
  - 调整F(u,v) 中绝对值较小的数的正负号
  - 使F(u,v) 中正数的个数减负数的个数大于d
- 在 F(u,v) 中嵌入1时
  - 调整F(u,v) 中绝对值较小的数的正负号
  - 使F(u,v) 中正数的个数减负数的个数小于d

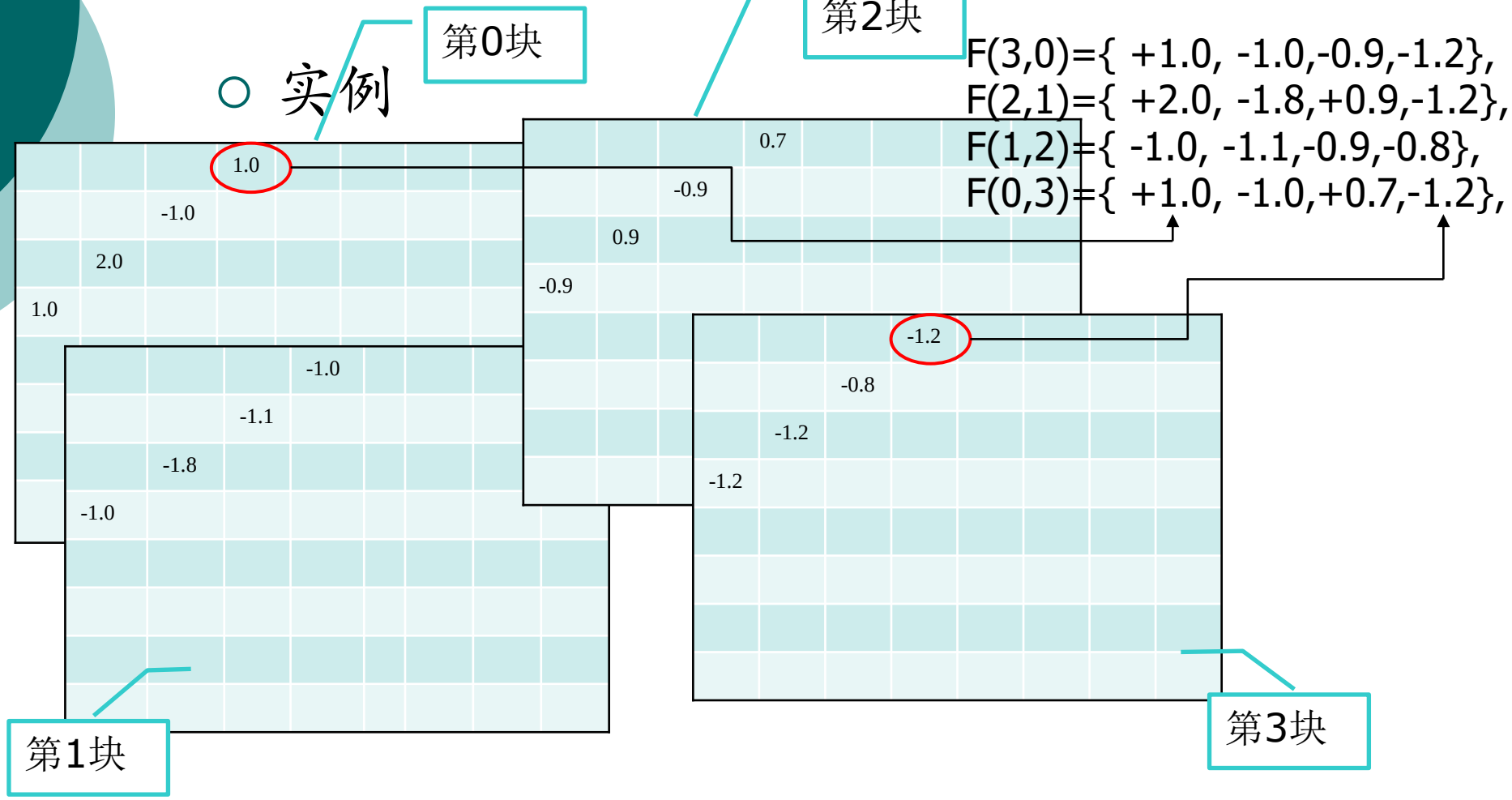
抗打印扫描水印算法（三）——基于系数分类的方法

- 实例
  - 设图像大小为16*16，按照8*8大小分块，并做DCT变换。选取下标(起始下标为0)满足u+v=3的系数嵌入水印。

第0块	第1块
第2块	第3块

抗打印扫描水印算法（三）——基于系数分类的方法

- 实例



抗打印扫描水印算法（三）——基于系数分类的方法

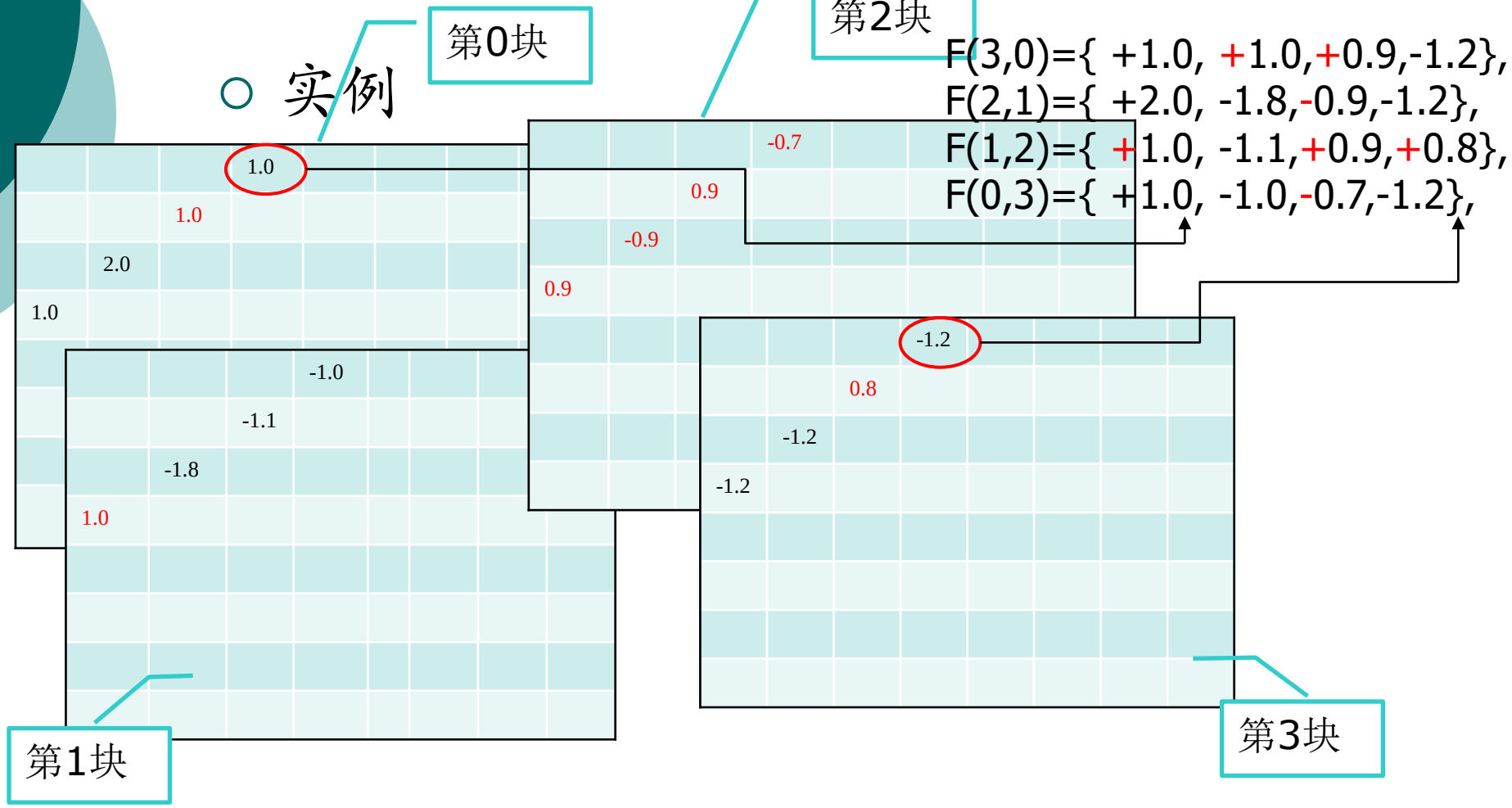
- 实例
  - 已知4个集合分别为:
    - F(3,0)={ +1.0, -1.0,-0.9,-1.2},
    - F(2,1)={ +2.0, -1.8,+0.9,-1.2},
    - F(1,2)={ -1.0, -1.1,-0.9,-0.8},
    - F(0,3)={ +1.0, -1.0,+0.7,-1.2},
  - 若鲁棒性参数d为2，嵌入0时，要求$n_+ \geq n_- + d$；嵌入1时，要求$n_- \geq n_+ + d$。则嵌入0，1，0，1后，系数调整为什么？

抗打印扫描水印算法（三）——基于系数分类的方法

- 实例
  - 解：
    - 在F(3,0)={ +1.0, -1.0,-0.9,-1.2}中嵌入0，应选2个绝对值较小的系数，翻转其符号。则，-1.0和-0.9的符号被翻转，所以嵌入后，系数变为：F(3,0)={ +1.0, +1.0,+0.9,-1.2}，
    - 同理，在F(2,1)={ +2.0, -1.8,+0.9,-1.2}嵌入1，系数变为： F(2,1)={ +2.0, -1.8,-0.9,-1.2}
    - 在F(1,2)={ -1.0, -1.1,-0.9,-0.8}嵌入0，系数变为：F(1,2)={ +1.0, -1.1,+0.9,+0.8}
    - 在F(0,3)={ +1.0, -1.0,+0.7,-1.2}嵌入1，系数变为： F(0,3)={ +1.0, -1.0,-0.7,-1.2}

抗打印扫描水印算法（三）——基于系数分类的方法

- 实例



抗打印扫描水印算法（三）——基于系数分类的方法

- 实例
  - 依次将变更后的系数矩阵做DCT逆变换，得到4个8*8的图像子块。
  - 按顺序重组图像子块，得到完整的、包含水印信息的图像。

抗打印扫描水印算法（三）——基于系数分类的方法

- 对打印的图像进行扫描，重新得到数字图像，将图像按照16×16进行分块，对每一块进行DCT变换，得到DCT系数
- 对于嵌入水印的位置，计算F(u,v) 正数和负数的个数
  - 正数个数大于负数个数：0
  - 正数个数小于负数个数：1
- 对所有的嵌入位置进行提取，得到所嵌入的水印信息

抗打印扫描水印算法（三）——基于系数分类的方法之仿真结果

- 强度d=230，嵌入16比特，嵌入位置k=u+v=15
- 检测正确率100%

原始图像

嵌入水印后图像

打印扫描后的图像

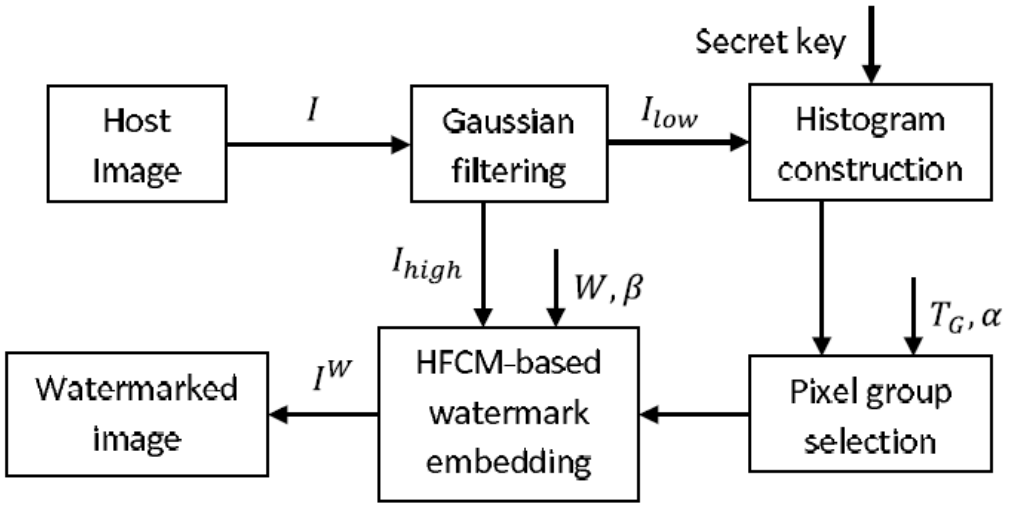
抗打印扫描水印算法（三）——基于系数分类的方法和之仿真结果

- k越小，鲁棒性越好
- 但k越小，DCT系数就越接近低频，对图像的质量影响就越大
- 为了扩大水印的容量，可在中低频嵌入水印，为保证图像的质量，可针对不同的k选取不同的嵌入强度d

抗剪切攻击水印算法
——基于直方图形状调整的稳健算法

- 剪切操作会导致水印图像丢失大量像素，并引发失同步问题，是一种成本低、强度高的攻击方法，一般算法难以抵御该攻击。
- 本算法将水印信息嵌入直方图形状中，能够耐受包括剪切、随机抖动和平滑滤波在内的多种攻击。

抗剪切攻击水印算法
——基于直方图形状调整的稳健算法



算法主要流程如图所示：

a. **获取低频信号：**载体图像I经过低通滤波，分为低频信号I_{low}和高频信号I_{high}，$I = I_{low} + I_{high}$

b. **构造直方图：**在秘钥控制下，随机挑选若干个灰度级，构造低频信号I_{low}的直方图，

c. **像素分组：**根据参数组重量T_G和安全带α，筛选得出像素组

d. **嵌入水印：**调整像素组直方图从而嵌入水印W，按参数补偿因子β调整高频信号I_{high}，并与含水印的低频信号叠加得到含水印图像I^W

抗剪切攻击水印算法
——基于直方图形状调整的稳健算法

- 获取低频信号

$$F(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$I_{\text{low}}(x, y) = F(x, y, \sigma) * I(x, y)$$

$$I_{\text{high}}(x, y) = I(x, y) - I_{\text{low}}(x, y)$$

使用2-D高斯低通滤波器。 x和y为像素坐标 ， σ=1为标准差，模块F形状为$(2k\sigma + 1) \times (2k\sigma + 1)$，算法取k=3，因此F形状为$7 \times 7$

抗剪切攻击水印算法
——基于直方图形状调整的稳健算法

○ 构造直方图

$$H_S = \{h_S(K_i) | i = 1, 2, \ldots, S\}$$

- 从256个灰度级中，随机挑选S个灰度级，$K_i = pn(i)$
- 计算I_{low}中灰度级为K_i的像素个数，记为$h_S(K_i)$

○ 直方图分箱

$$M_B = \left\lfloor \frac{S}{L_B} \right\rfloor \qquad h_B(i) = h_S\left(K_{(i-1)\cdot L_B+1}\right) + h_S\left(K_{(i-1)\cdot L_B+2}\right) + \cdots + h_S\left(K_{i\cdot L_B}\right)$$

每L_B个频数$h_S(K_i)$分为一箱，记为$h_B(i)$，共可分M_B箱

抗剪切攻击水印算法
——基于直方图形状调整的稳健算法

○ 像素分组

$$h_G(i) = h_B(2i-1) + h_B(2i), \quad i = 1, 2, \ldots, \left\lfloor \frac{M_B}{2} \right\rfloor.$$

- 每两箱构成一个分组，记为 $h_G(i)$，共有$\left\lfloor \frac{M_B}{2} \right\rfloor$个分组

$$N_S = \sum_{i=1}^{S} h_S(K_i). \qquad g(i) = \frac{h_G(i)}{N_S}$$

- 挑选像素量足够多的分组用于嵌入水印，$g(i) \geq T_G$ 的分组被选中。

抗剪切攻击水印算法
——基于直方图形状调整的稳健算法

○ 像素分组

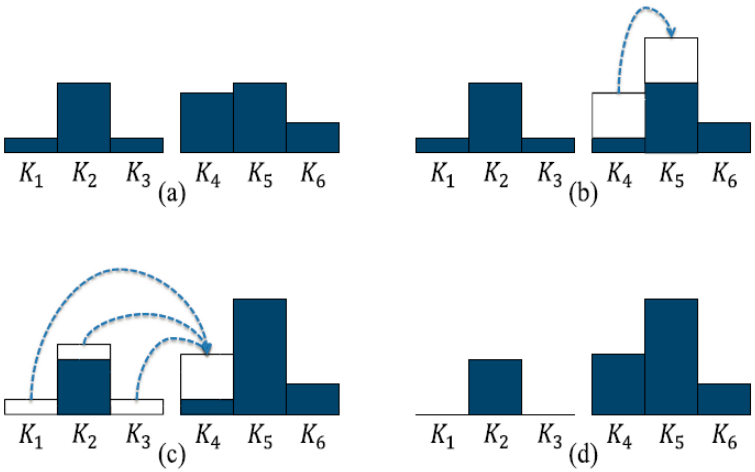
$$(1-\alpha)\cdot g_{\min} < g(i) < g_{\min}$$

- 提取水印时， 同样根据$g(i) \geq T_G$来筛选分组。考虑到剪切等攻击会改变分组像素量$g(i)$，引入“安全带”概念。
- 记被选中的分组中，最小分组像素数量为$g_{\min}$。修改未被选中的分组中，像素量过于接近的那些分组。从它们中随机选择$[g(i)-(1-\alpha)\,.\,g_{\min}].N_s$个像素，“移至”最临近的选中分组。

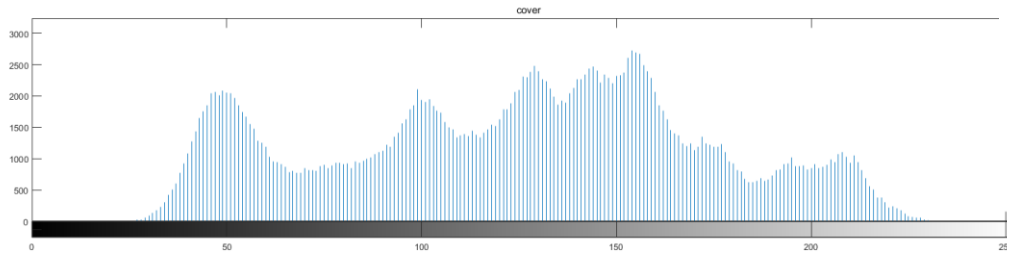
$$\begin{cases} \dfrac{h_B(2i-1)}{h_B(2i)} \geq 2, & \text{if } w_i = 1 \\ \dfrac{h_B(2i-1)}{h_B(2i)} \leq \frac{1}{2}, & \text{if } w_i = 0. \end{cases} \qquad \begin{cases} N_0 = \dfrac{2h_B(2i-1)-h_B(2i)}{3} \\ N_1 = \dfrac{2h_B(2i)-h_B(2i-1)}{3}. \end{cases}$$

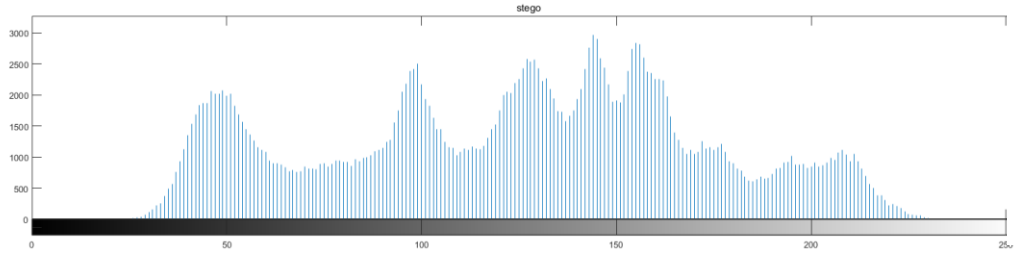
- 调整相邻分组像素量以嵌入水印信息

抗剪切攻击水印算法
——基于直方图形状调整的稳健算法



调整策略：随机、就近、按比例
算法耐受剪裁、缩放、旋转等多种攻击。





小结

- 算法设计
  - 强稳健水印算法往往牺牲计算复杂度、透明性、容量等性能指标，来获取稳健性。
  - 强稳健水印算法通常针对使用场景设计，不要求一个算法能够抵抗所有攻击。
- 典型攻击
  - 压缩编码、几何攻击、数模/模数变换等。
- 强稳健算法典型策略
  - 冗余嵌入、扩频、重要感知区域嵌入等。

小结

- 典型算法
  - 基于DFT系数法量化的方法
  - 基于DCT系数相对关系的方法
  - 基于DCT系数分类的方法
  - 这些方法可以应用到其他变换域，例如，基于DCT系数的分类方法可以应用于DWT系数。