



第三章 信息隐藏



大纲

- 概述
- 图像信息隐藏
- 视频信息隐藏



图像信息隐藏

空域隐写

- 非自适应隐密
 - LSB替换 (最低比特位, Least Significant Bit)
 - LSB匹配
- 自适应隐密
 - BPCS : 基于视觉特性

位平面

$(90)_{10} = (0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0)_2$

$(128)_{10} = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)_2$

$(193)_{10} = (1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)_2$

$(225)_{10} = (1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)_2$

90	128
193	225

0	1
1	1

1	0
1	1

0	0
0	1

1	0
0	0

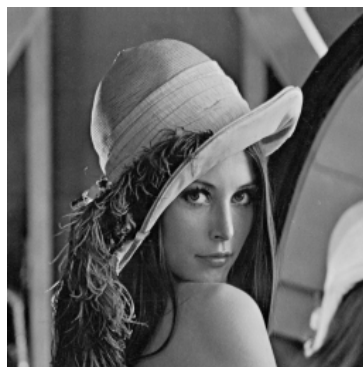
1	0
0	0

0	0
0	0

1	0
0	0

0	0
1	1

LSB



标准测试图像Lena



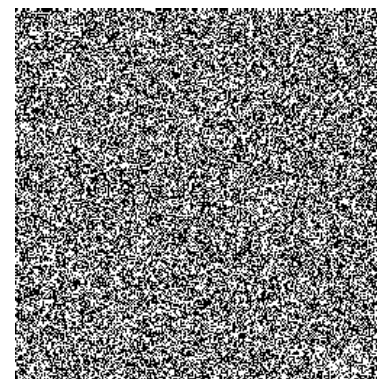
a) 第1 (最高 MSB) 位平面



b) 第3位平面



(c) 第5位平面



d) 第8 (最低 LSB) 位平面

LSB替换



Information

$H = 1101011110101101$

1. 将待嵌入的信息转换成加密的二进制比特流，可用一个收发双方都知道的**密钥**对该秘密信息比特流进行**加密**或者**置乱**处理以增加安全性。
2. 选取用于嵌入秘密信息的载体图像的像素，该像素可以**顺序选取**或者**随机选取**。顺序选取对于图像的改变相对有规律，随机选取将秘密信息嵌入后的对载体图像的像素值的改变散布在整幅图内，提高了安全性。
3. 将选取像素的像素值的**最低有效位**与对应的秘密信息比特进行比较，若相同，则不需改变载体图像的灰度值；若不同，将载体图像灰度值的最低有效位进行0\1翻转，也即0变成1，1变成0。将所有秘密信息都嵌入完成后，即可获得LSB替换隐密后的**隐密图像**。

LSB替换



Information

$H = 1101011110101101$

5	11	10	13	
20	31	41	51	
27	10	17	44	
37	85	14	35	

Stego Image

Least Significant Bit (LSB)

0000010 1	0000101 0	0000101 0	0000110 0
0001010 0	0001111 1	0010100 0	0011001 1
0001101 1	0000101 0	0001000 1	0010110 0
0010010 1	0101010 1	0000111 0	0010001 0

LSB替换

5	10	11	12	
20	31	40	51	
27	10	17	45	
37	85	15	34	

Image

5	11	10	13	
20	31	41	51	
27	10	17	44	
37	85	14	35	

Stego-Image

$$MSE = \frac{1}{16} \left[(5 - 5)^2 + (10 - 11)^2 + \dots + (34 - 35)^2 \right] = 0.4375$$

$$PSNR \text{ (Peak Signal to Noise Ratio)} = 10 \times \log_{10} \left[\frac{255^2}{MSE} \right] = 51.72$$

$$\text{Capacity} = 1 \text{ (bit per pixel, bpp)}$$

LSB替换



a) 嵌入前的Lena原图



b) 嵌入后的Lena隐密图

LSB替换

5	10	11	12	
20	31	40	51	
27	10	17	45	
37	85	15	34	

Image

5	11	10	13	
20	31	41	51	
27	10	17	44	
37	85	14	35	

Stego-Image

- 1.接收者可以根据事先约定的密钥对图像进行解密或逆置乱的操作，然后确定隐藏信息嵌入的位置。
- 2.如果该位置的像素值 $\text{mod } 2$ 的值为1，则此处隐藏的信息为1；
- 3.如果该位置的像素值 $\text{mod } 2$ 的值为0，则此处隐藏的信息为0。
- 4.这样即使隐蔽通信的第三方能得到隐密图像的最低位平面信息，在不知道密钥的情况下，无法知道隐密信息藏的位置，获得秘密消息。

LSB替换

000001 01	000010 11	000010 00	000011 11
000101 10	000111 01	001010 01	001100 01
000110 01	000010 10	000100 11	001011 10
001001 01	010101 11	000011 00	001000 01

PSNR = 43.27

Capacity= 2 bpp

Trade-off: Image quality and Payload

LSB替换-再思考

- 当秘密信息比特序列嵌入到载体图像最低位平面时，将原来LSB 位的“0”，“1”置换成秘密信息的“0”或者“1”。如果秘密信息比特和载体图像灰度值的LSB一致，则不修改原始载体像素；否则，修改像素的像素值，若原始值 $2i$ ，则变为 $2i+1$ ，若为 $2i+1$ ，则修改为 $2i$ 。
- 灰度值是属于 $0\sim 255$ 之间的，根据LSB的原理，像素值嵌入秘密信息前后的变化为F：
 $0\leftrightarrow 1, 2\leftrightarrow 3, \dots, 252\leftrightarrow 253, 254\leftrightarrow 255$ ，从不进行 $2i$ 到 $2i-1$ 的变换，或者反之。

LSB替换

- “值对”现象

- 当秘密信息比特和图像的选定像素点的灰度值的最低有效位不同时，将会改变原始图像的该像素点灰度值的最低有效位，最低有效位从0变成1或者从1变成0，即将灰度值 $2i$ 改为 $2i+1$ 或将灰度值 $2i+1$ 改为 $2i$ 。
- 秘密信息加密后的信息若看成为转换的二进制比特序列是0和1出现**概率相同**的随机序列，利用LSB替换算法最终会导致**最低有效位为0和1的个数趋于相等**。
- 对于一个整幅图像来说，这会使像素点灰度值前七位相同而最低有效位分别为0和为1的两个灰度值构成一个“值对”，对于每一对“值对”中的灰度值出现的概率也趋于相同。
- 利用“值对”现象可对LSB替换进行**检测**，从而发现有隐密信息存在。

LSB替换-值对现象

- “Pov” (Pairs of Values)

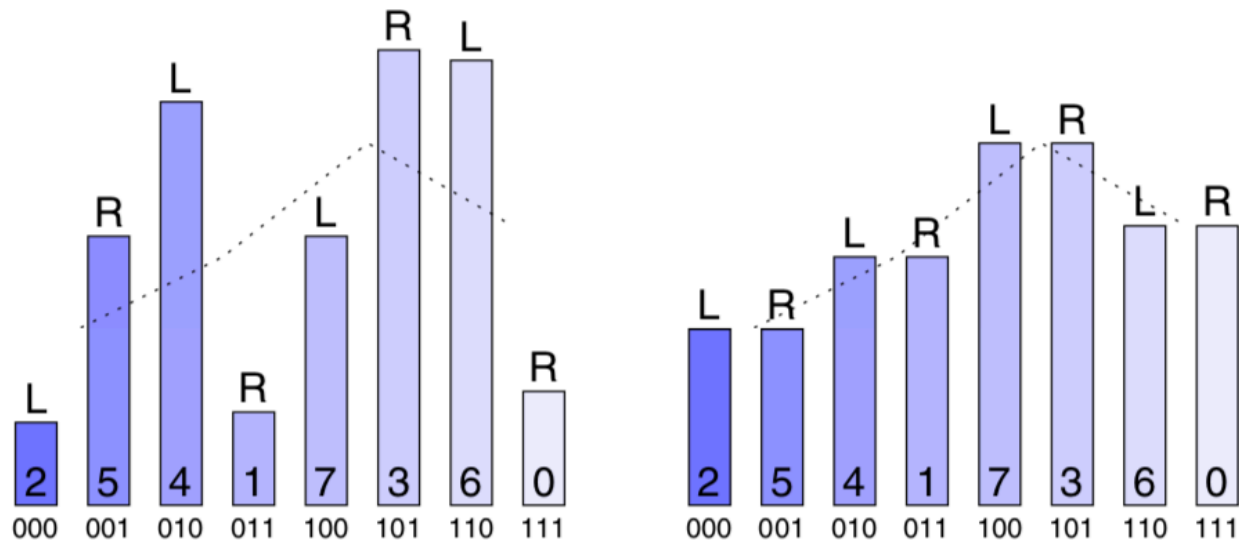


Fig. 14. Histogram of colours before and after embedding a message with EzStego

F: $0 \leftrightarrow 1, 2 \leftrightarrow 3, \dots, 253 \leftrightarrow 254, 254 \leftrightarrow 255$

$2i \leftrightarrow 2i+1$

Never, $2i \leftrightarrow 2i-1$

怎么检测隐写?
隐写分析

χ^2 隐写分析

- 定义待检测图像中灰度值为 i ($i \in \{0, 1, 2, \dots, 255\}$) 的个数为 N_i 。对于自然图像来讲，如果嵌入了大量的隐密信息，像素值为奇数和偶数出现的频率在理论上应该是一样的，因此有 N_{2i} 与 N_{2i+1} 较为接近，而没有嵌入信息则相距较远，这也就产生了“值对”现象。
- 以 512×512 的标准图像 Crowd 为载体嵌入隐密信息观察直方图。Crowd 嵌入秘密信息前后的部分灰度直方图。可以看出，当嵌入隐密信息后，灰度值对趋于均匀化， χ^2 检测方法就是利用这种统计特性差异进行隐密分析的。

χ^2 隐写分析



(a) 标准测试图像 Crowd

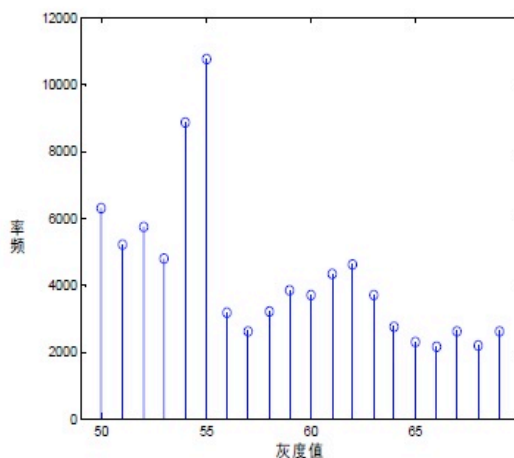
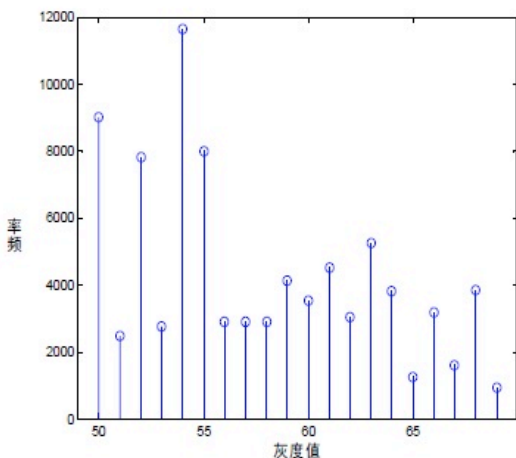
隐写分析的假设:

1 嵌入的0, 1比特概率各为50%

2 改动的规则: $2i \leftrightarrow 2i+1$, 不会有 $2i \leftrightarrow 2i-1$

3 如果秘密信息完全替代了最低位平面, 则 h_{2i} , h_{2i+1} 的值会比较接近

4 如果图像未经隐写, 则 h_{2i} , h_{2i+1} 的值会相差较远



χ^2 隐写分析

- 卡方检验是一种用途很广的计数资料的假设检验方法。它属于非参数检验的范畴。
- 统计样本的实际观测值与理论推断值之间的偏离程度，实际观测值与理论推断值之间的偏离程度就决定卡方值的大小，如果卡方值越大，二者偏差程度越大；反之，二者偏差越小；若两个值完全相等时，卡方值就为0，表明理论值完全符合。

χ^2 隐写分析

■ 细节

设 $N'_{2i} = (N_{2i} + N_{2i+1}) / 2$ ，因为灰度总在 $2i$ 和 $2i+1$ 之间转换，不会改变 N_{2i} 与 N_{2i+1} 的总量。所以可构造：

$$\chi^2_{k-1} = \sum_{i=1}^k \frac{(N_{2i} - N'_{2i})^2}{N'_{2i}}$$

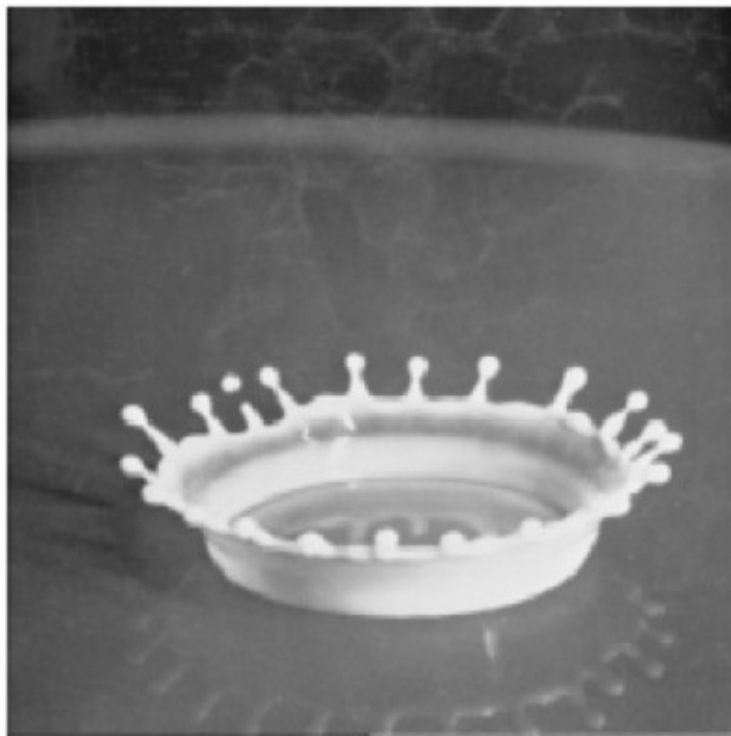
服从 χ^2 分布，卡方值越小表示含有隐密信息的可能性越大。

- 什么是卡方分布？
- 若 n 个相互独立的随机变量 $\xi_1, \xi_2, \dots, \xi_n$ ，均服从标准正态分布（也称独立同分布于标准正态分布），则这 n 个服从标准正态分布的随机变量的平方和构成一新的随机变量，其分布规律称为卡方分布。

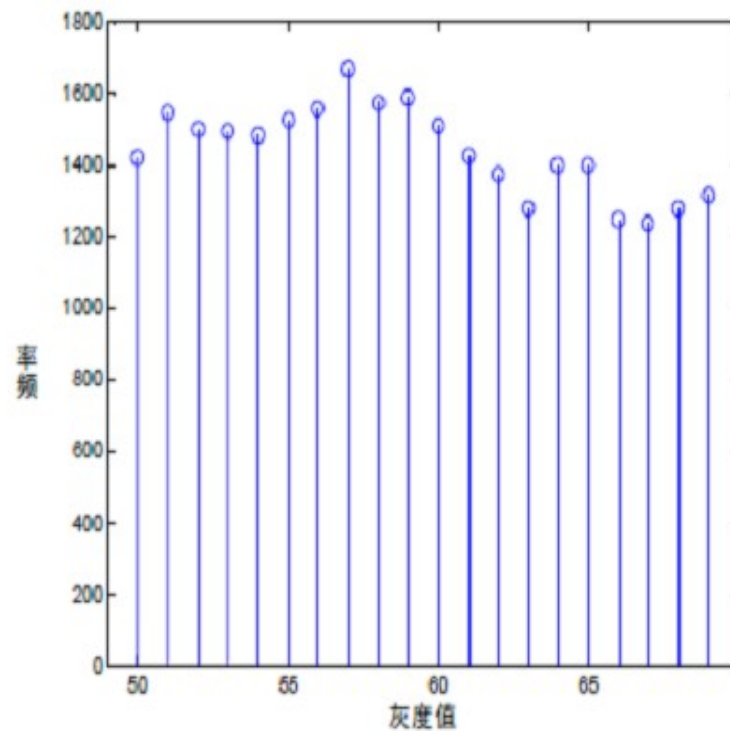
χ^2 隐写分析

- 定义待检测图像中灰度值为 i ($i \in \{0, 1, 2, \dots, 255\}$) 的个数为 N_i 。对于自然图像来讲，如果嵌入了大量的隐密信息，像素值为奇数和偶数出现的频率在理论上应该是一样的，因此有 N_{2i} 与 N_{2i+1} 较为接近，而没有嵌入信息则相距较远，这也就产生了“值对”现象。
- 以 512×512 的标准图像 Crowd 为载体嵌入隐密信息观察直方图。Crowd 嵌入秘密信息前后的部分灰度直方图。可以看出，当嵌入隐密信息后，灰度值对趋于均匀化， χ^2 检测方法就是利用这种统计特性差异进行隐密分析的。
- 然而 χ^2 检测方法并不适用灰度级比较丰富的图像，选取比较平滑的大小为 512×512 的标准图片 Milkdrop，原始图像灰度值对已经相当均匀。

χ^2 隐写分析



(a) 标准测试图像Mikedrop



(b) 原始图像直方图（局部）

思考

- χ^2 隐写分析适用条件？
- 抗 χ^2 隐写？

思考

- χ^2 隐写分析适用条件？
 - 隐写分析的假设：
 - 1 嵌入的0, 1比特概率各为50%
 - 2 改动的规则： $2i \leftrightarrow 2i+1$, 不会有 $2i \leftrightarrow 2i-1$
 - 3 如果秘密信息完全替代了最低位平面, 则 h_{2i} , h_{2i+1} 的值会比较接近
 - 4 如果图像未经隐写, 则 h_{2i} , h_{2i+1} 的值会相差较远
 - 由此可总结出卡方隐写分析方法的特点：
 - 1 原理简单清晰
 - 2 易于实现, 检测效果较好
 - 3 只能针对加密后密文的0, 1概率各为50%的情况。
 - 4 只能针对特定的LSB替换嵌入策略
 - 5 只能针对高容量嵌入的情况
 - 6 只能针对特定载体图像, 即原始图像中 h_{2i} , h_{2i+1} 值即相差较远的
 - 为了抵抗卡方隐写分析：可以针对这些局限性做文章。。。。。
 - 如：直方图补偿, 降低嵌入容量等。
 - 降低嵌入容量, 可以抵抗 χ^2 分析, 不能抵抗RS分析

RS隐写分析

- 针对随机嵌入的LSB替换隐写算法
- 利用翻转不对称性，考察空间相关性的偏离程度，也可以估计信息嵌入率。

RS隐写分析

1. 将待检测图像分为大小相等的小图像块。
2. 对每个小图像块分别进行非负和非正翻转。
3. 计算每块图像像素相关性是否增加。

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|$$

规则组R：变换后f值增大；

异常组S：变换后f值减小；

无用组U：变换后f值不变。

Regular groups: $G \in R \Leftrightarrow f(F(G)) > f(G)$

Singular groups: $G \in S \Leftrightarrow f(F(G)) < f(G)$

Unusable groups: $G \in U \Leftrightarrow f(F(G)) = f(G),$

RS隐写分析

4. 将非负翻转后像素相关性增加的图像块的比例记为 R_m ，像素相关性减小的图像块比例记为 S_m 。
5. 将非正翻转后像素相关性增加的图像块比例记为 $R-m$ ，像素相关性减小的图像块比例记为 $S-m$ 。

规则组R：变换后f值增大；

异常组S：变换后f值减小；

无用组U：变换后f值不变。

Regular groups: $G \in R \Leftrightarrow f(F(G)) > f(G)$

Singular groups: $G \in S \Leftrightarrow f(F(G)) < f(G)$

Unusable groups: $G \in U \Leftrightarrow f(F(G)) = f(G)$,

RS隐写分析

- 如果待检测图像没有经过LSB隐写，那么无论是经过非正翻转还是非负翻转，会等同的增加图像块的混乱度，都应该满足以下规律： $R_m \approx R-m$ ， $S_m \approx S-m$ ，并且 $R_m > S_m$ ， $R-m > S-m$ 。
- 如果是经过LSB隐写的（部分像素用过 F_1 正向变换），用 F_1 和 F_{-1} 就会明显不同。

RS隐写分析

- 对隐写图像分小块进行 F_1 翻转和 R_m 和 S_m 计算时，所有像素可以分三类：
 - 1 没有经过翻转的：灰度值不变
 - 2 经过一次翻转的：灰度值变换幅度为1
 - 3 经过两次翻转的：两次翻转又回到了原始值。
- 对隐写图像用 F_{-1} 翻转时，也有一些像素经过两次翻转，但是经历的是一次 F_1 和 F_{-1} ，这样会与原始值相差更大。

RS隐写分析



256x256 uint8							
	1	2	3	4	5	6	7
1	162	162	160	162	163	160	159
2	162	162	160	162	163	160	159
3	163	160	160	160	161	158	157
4	160	158	159	158	160	159	155
5	155	157	157	157	159	158	157
6	156	158	156	153	157	157	156
7	157	157	157	156	157	156	155
8	158	157	157	157	156	156	157
9	157	157	156	154	157	156	157
10	156	156	158	155	157	156	157
11	156	156	156	156	156	157	157
12	158	155	153	157	157	157	158

$$F_1 : 0 \leftrightarrow 1, 2 \leftrightarrow 3, \dots, 254 \leftrightarrow 255$$

$$F_{-1} : -1 \leftrightarrow 0, 1 \leftrightarrow 2, \dots, 255 \leftrightarrow 256$$

$$F_0 : F_0(x) = x$$

• 原矩阵：

- 非负变换
- 非正变换

• 嵌入信息矩阵

- 非负变换
- 非正变换

RS隐写分析

- 原矩阵

$$\begin{bmatrix} 162 & 162 & 160 & 162 \\ 162 & 162 & 160 & 162 \\ 163 & 160 & 160 & 160 \\ 160 & 158 & 159 & 158 \end{bmatrix} f=18$$

- 非负变换

$$\begin{bmatrix} 163 & 162 & 161 & 163 \\ 162 & 162 & 160 & 162 \\ 163 & 160 & 161 & 160 \\ 161 & 159 & 159 & 158 \end{bmatrix} f=19$$

- 非正变换

$$\begin{bmatrix} 161 & 162 & 159 & 162 \\ 161 & 162 & 160 & 161 \\ 164 & 160 & 159 & 160 \\ 160 & 158 & 159 & 157 \end{bmatrix} f=24$$

- 隐密矩阵 0110101011010011

$$\begin{bmatrix} 162 & 163 & 161 & 162 \\ 163 & 162 & 161 & 162 \\ 163 & 161 & 160 & 161 \\ 160 & 158 & 159 & 159 \end{bmatrix} f=15$$

- 非负变换

$$\begin{bmatrix} 162 & 162 & 161 & 162 \\ 162 & 163 & 161 & 162 \\ 163 & 160 & 161 & 160 \\ 160 & 159 & 159 & 159 \end{bmatrix} f=13$$

- 非正变换

$$\begin{bmatrix} 162 & 164 & 162 & 162 \\ 163 & 161 & 162 & 161 \\ 164 & 161 & 160 & 162 \\ 159 & 157 & 160 & 159 \end{bmatrix} f=21$$

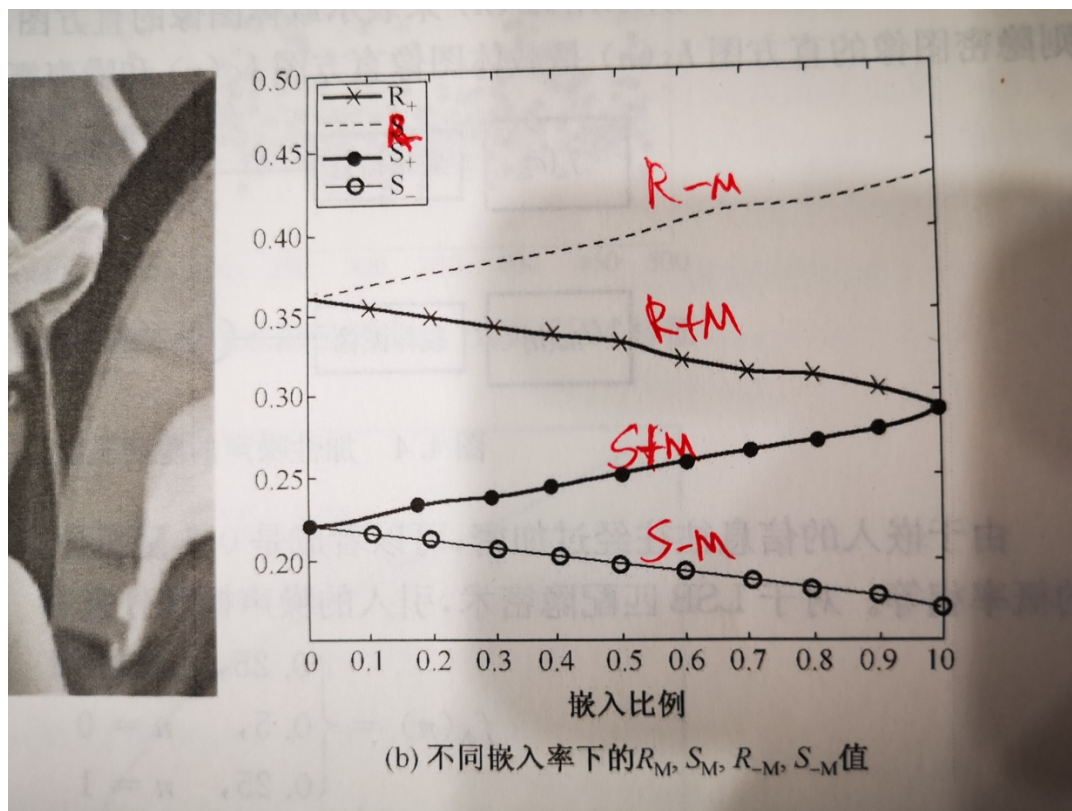
练习

- 原矩阵

$$\begin{bmatrix} 155 & 157 & 157 & 157 \\ 156 & 158 & 156 & 153 \\ 157 & 157 & 157 & 156 \\ 158 & 157 & 157 & 157 \end{bmatrix}$$

- LSB替换嵌入信息：0010010110111010

思考



R+m: 非负翻转后像素相关性减少的图像块的比例

S+m: 非负翻转后像素相关性增加的图像块比例

R-m: 非正翻转后像素相关性减少的图像块比例

S-m: 非正翻转后像素相关性增加的图像块比例

RS分析能不能抵抗：对加密方法的修正，低容量嵌入，特定载体图像等卡方分析方法的局限？

RS分析不能抵抗对嵌入规则的修改？

LSB匹配

$$P_s = \begin{cases} 1 & b \neq LSB(P_c) \& P_c = 0 \\ P_c \pm 1 & b \neq LSB(P_c) \& 0 < P_c < 255 \\ P_c & b = LSB(P_c) \\ 254 & b \neq LSB(P_c) \& P_c = 255 \end{cases}$$

P_c 和 P_s 分别表示选定的载体图像和其对应的隐密图像的像素值， b 为秘密信息比特。

- 将选取像素的像素值的最低有效位与对应的秘密信息比特进行比较，若相同，则不需改变载体图像的灰度值；若不同，对于边界灰度值（即灰度值为0或者255），灰度值只能从0改为1或者从255改为254；对于其他的载体图像灰度值，像素值**随机的加1或者减1，加1和减1的概率相同**。
- 在嵌入数据时不仅可将图像的灰度值从 $2i$ 改为 $2i+1$ 或将 $2i+1$ 改为 $2i$ ，还可以将 $2i$ 改为 $2i-1$ 或将 $2i+1$ 改为 $2i+2$ ，杜绝“值对”现象的产生。

LSB匹配

$$H = 1101011110101101$$

5	10	11	12	
20	31	40	51	
27	10	17	45	
37	85	15	34	

Image

5	11	10	11	
20	31	39	51	
27	10	17	44	
37	85	14	33	

Stego-Image

思考

- 为什么LSB匹配算法可以抵抗RS隐写分析？

练习

- 原矩阵

$$\begin{bmatrix} 155 & 157 & 157 & 157 \\ 156 & 158 & 156 & 153 \\ 157 & 157 & 157 & 156 \\ 158 & 157 & 157 & 157 \end{bmatrix}$$

- 嵌入信息：0010010110111010

LSB替换？

LSB匹配？

自适应图像隐写

- 根据载体图像的局部特性自动地调节嵌入秘密数据的多寡。
 - 保证主观视觉质量
 - 局部修改，增强对以图像的全局特征为作为隐密分析特征的隐密分析算法的安全性。

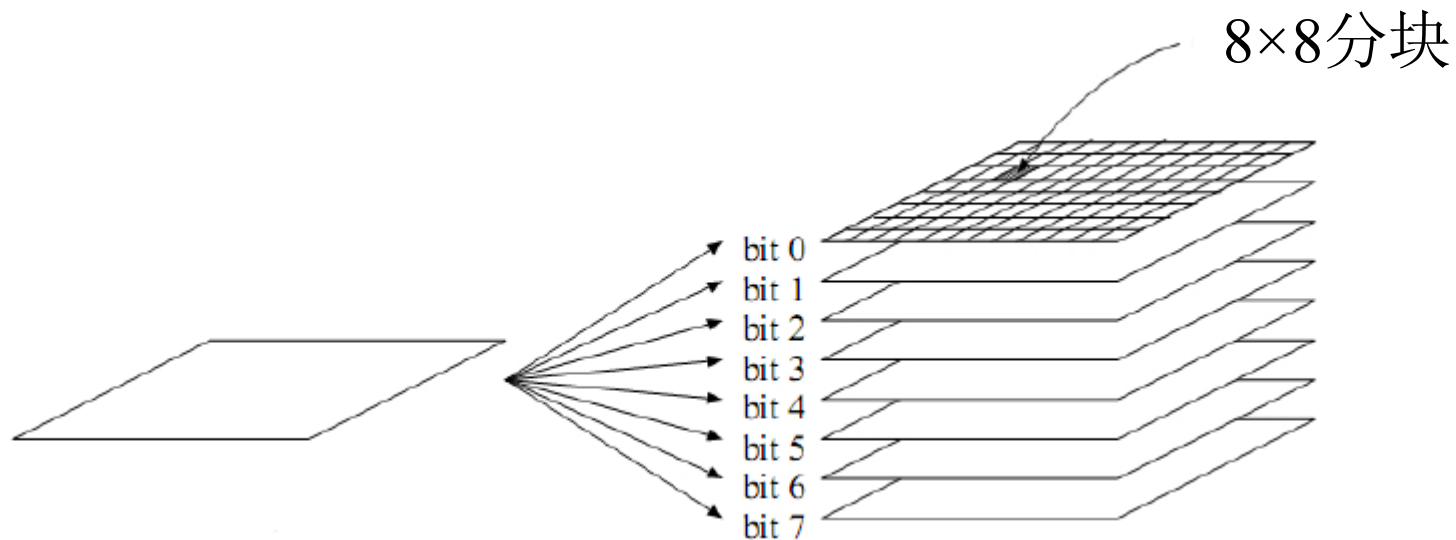
BPCS自适应

- 位平面复杂度分割(bit-plane complexity segmentation , BPCS)
- 根据载体图像的局部特性自动地调节嵌入秘密数据的多寡。
- 将载体数据的多个位平面都分成固定大小的小块。
- 由于人的感觉器官对那些变化剧烈、复杂度较高的位面小块比较不敏感，所以用这些位面小块来隐藏秘密信息更安全。

BPCS-嵌入

- 1.将图像的所有位平面图像分成互不重叠的 8×8 分块；
- 2.利用计算复杂度公式计算每个位平面每个分块的复杂度；
- 3.设定参数 t 值，筛选出复杂度大于 t 的分块作为信息隐藏分块；
- 4.将隐秘信息也分成 8×8 分块，并分别计算其复杂度，复杂度大于 t ，则直接替换图像的对应分块；若隐秘信息复杂度低于 t ，则做异或(XOR)计算后，再替换图像的对应分块。

BPCS-嵌入



8位灰度图像

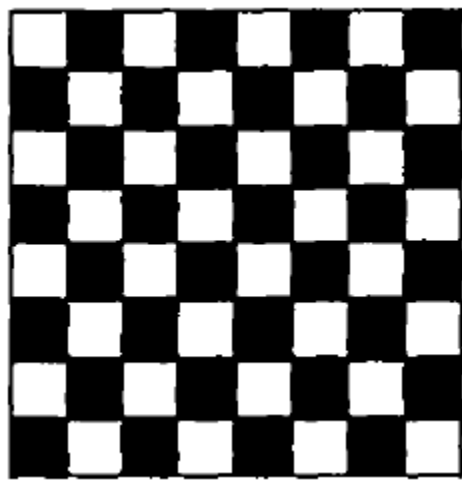
二值位平面图像

BPCS-嵌入

- 1.将图像的所有位平面图像分成互不重叠的 8×8 分块；
- 2.利用计算复杂度公式计算每个位平面每个分块的复杂度；
- 3.设定参数 t 值，筛选出复杂度大于 t 的分块作为信息隐藏分块；
- 4.将隐秘信息也分成 8×8 分块，并分别计算其复杂度，复杂度大于 t ，则直接替换图像的对应分块；若隐秘信息复杂度低于 t ，则做异或(XOR)计算后，再替换图像的对应分块。

BPCS-嵌入

- 复杂度：
 - C为所有相邻像素对中取值不同(即一个为0, 另一个为1)的像素对数目总和, C_{max} 是复杂度的最大可能值。
 - $0 \leq C \leq 112$ (8×8 小块)。

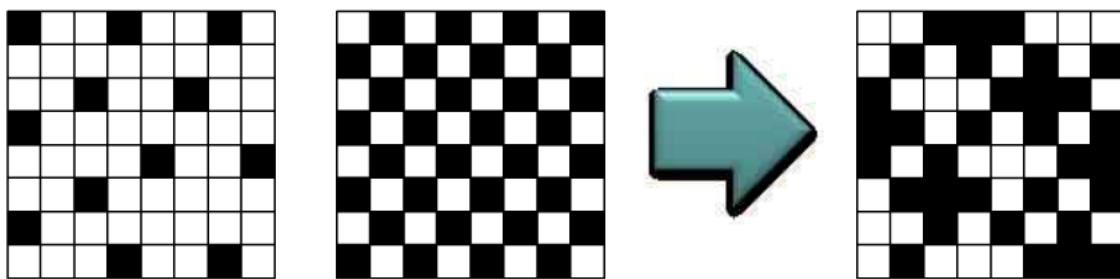


此时C最大： $C_{max} = 112$ 。为什么？

8×8 棋盘小块, 0-Black, 1-White

BPCS-嵌入

- 3.筛选复杂度大于 aC_{max} 的位平面分块作为信息隐藏分块； a 是参数，值必须小于0.5。^{为什么？}
- 4.将秘密信息也分成 8×8 位平面小块，并分别计算其复杂度
 - 复杂度大于 aC_{max} ，直接替换图像的对应分块；
 - 复杂度低于 aC_{max} ，则共轭处理（异或棋盘小块）后再替换。



BPCS-嵌入

- 5.记录哪些小块是经过异或处理的，这部分信息要嵌入到载体数据中，且不能影响已经嵌入的秘密信息，并且要能够正确提取。
 - 例如，可以用LSB方法将这些额外的信息隐藏在预先划定的区域。

BPCS-提取

- 取出图像中复杂度大于 aC_{max} 的位平面分块；
- 处理经过共轭处理的分块，再进行共轭处理，恢复原始信息；
- 提取隐藏的秘密消息。

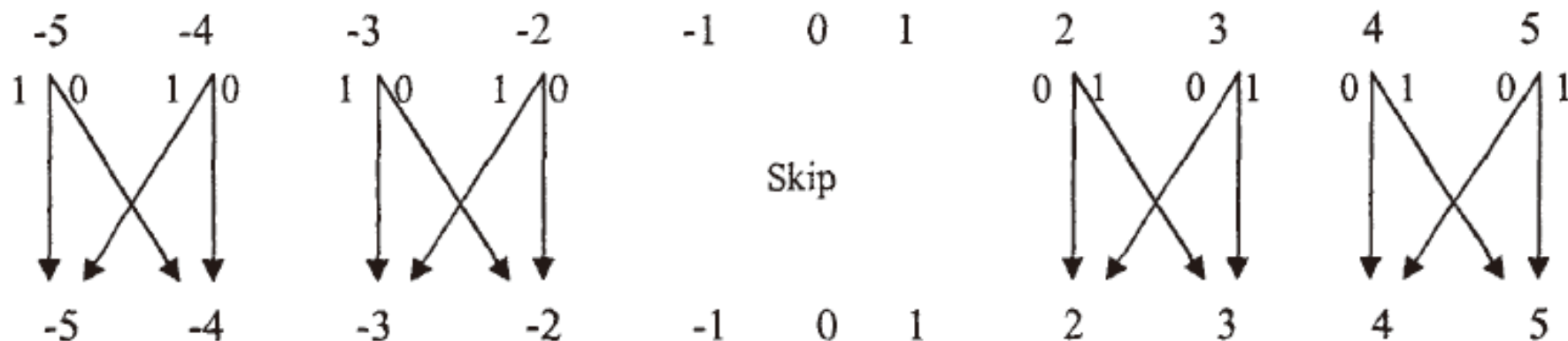
变换域隐写

- 在变换域进行隐写可以将对图像修改的能量分散，一方面可以结合人类视觉特性增强载体的不可察觉性，另一方面也可以提高一定的鲁棒性。
- 变换域隐藏的总体思想，就是将秘密信息隐藏在载体的最重要部位
- DCT变换等，都是能量守恒变换，在变换域中将能量集中，隐藏时将秘密信息与载体的视觉重要部分紧密联系在一起

■ JSTEG

■ F5

JSTEG



- JSteg是最早以JPEG图像为载体的隐密算法，主要是利用了LSB替换思想在DCT域实现。
- 其主要思路是：将一个二进制位的隐密信息嵌入到量化后的DCT系数的LSB上，但对原始值为-1、0、1的DCT系数例外。
- 提取隐密信息时，只需将载密图像中不等于-1、0、1的量化DCT系数的LSB逐一取出即可。
- 顺序或随机选择元素。

嵌入实例

- 嵌入信息：0010100110111010...
- 63个AC系数：57, 45, 0, 0, 0, 0, 23, 0, -30, -16, 0, 0, 1, 0, 0, 0, 0, 0, 0, ..., 0
- JSTEG
 - 56, 44, 0, 0, 0, 0, 23, 0, -31, -16, 0, 0, 1, 0, 0, 0, 0, 0, 0, ..., 0.
提取信息：00101

category表格里：-30 00001 -15:0000 -16:01111 -17:01110

练习



9. Huffman Coding

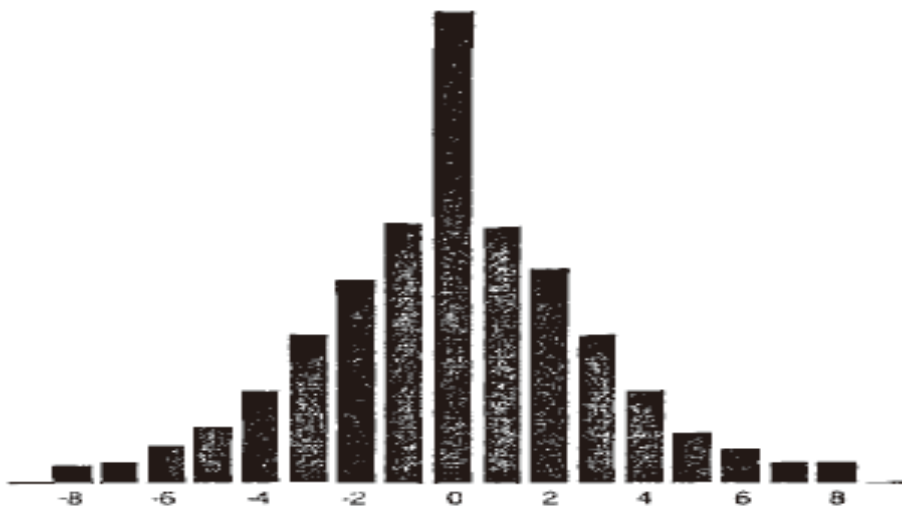
Category	Values	Bits for the value
1	-1,1	0,1
2	-3,-2,2,3	00,01,10,11
3	-7,-6,-5,-4,4,5,6,7	000,001,010,011,100,101,110,111
4	-15,...,-8,8,...,15	0000,...,0111,1000,...,1111
5	-31,...,-16,16,...,31	00000,...,01111,10000,...,11111
6	-63,...,-32,32,...,63	000000,...,011111,100000,...,111111
7	-127,...,-64,64,...,127	0000000,...,0111111,1000000,...,1111111
8	-255,...,-128,128,...,255	...
9	-511,...,-256,256,...,511	...
10	-1023,...,-512,512,...,1023	...
11	-2047,...,-1024,1024,...,2047	...

Figure 6. Table of values and bits for the value

JSTEG

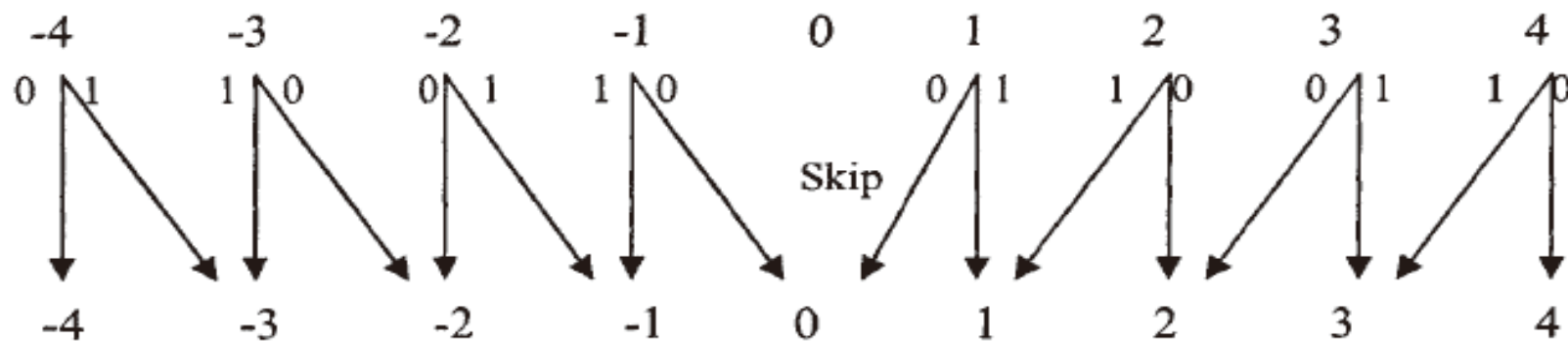
■ JPEG图像的DCT系数通常满足以下三个特性

- (1) DCT系数的绝对值越大，其对应直方图中的值出现频率就越小。
- (2) 随着DCT系数绝对值的增大，其出现频率下降的幅度减小。
- (3) 各系数出现的频率关于0对称。



JSTEG: DCT系数“成对效应”

F3



■ 嵌入:

- 只修改不为0的DCT系数
- 若不匹配则绝对值减1，符号不变，避免“成对效应”
- 若原DCT系数为+1或-1，而待嵌入秘密比特位为0，则原系数会变为0，本次嵌入操作无效，重新选择嵌入位。

■ 提取：将图像中不为0的DCT系数最低比特位按序取出。

嵌入实例

- 嵌入信息：0010100110111010...
- 63个AC系数：57, 45, 0, 0, 0, 0, 23, 0, -30, -16, 0, 0, 1, 0, 0, 0, 0, 0, 0, ..., 0
- JSTEG
 - 56, 44, 0, 0, 0, 0, 23, 0, -31, -16, 0, 0, 1, 0, 0, 0, 0, 0, 0, ..., 0.
提取信息：00101
- F3
 - 56, 44, 0, 0, 0, 0, 23, 0, -29, -16, 0, 0, 0, 0, 0, 0, 0, 0, 0, ..., 0.
提取信息：00101

category表格里：-30 00001 -15:0000 -16:01111 -17:01110

练习

