

# 信息隐藏基础

---

学期: 大三下

[HIT 信息隐藏基础 算法整理\\_随机翻转嵌入法-CSDN博客](#)

[信息隐藏基础总结 | upupming 的博客](#)

部分借鉴于[信息隐藏基础总结 | upupming 的博客](#)

## 第一章

---

**1.信息加密和信息隐藏是保障信息安全传输的两大手段，其主要区别是什么？**

- 信息加密
  - 现代密码学能保证加密信息计算安全
  - 保护通信信息内容
  - 不保护通信行为
- 信息隐藏
  - 信息是隐蔽的
  - 信息传递(或存在)的行为是隐蔽的
  - 隐藏的信息容量受限

## 2.BMP图像文件的基本结构分为哪几个部分？各部分功能是什么？

|       |
|-------|
| 位图文件头 |
| 位图信息头 |
| 彩色表   |
| 位图数据  |

1. 位图文件头数据结构（14字节），它包含BMP图像文件的类型、文件大小、偏移字节等信息；
2. 位图信息头数据结构（40字节），它包含有BMP图像的长、宽、高，以及颜色位数值等信息；
3. 彩色表，这个部分是可选的，有些位图需要彩色表，有些位图，比如真彩色图（24位的BMP）就不需要调色板；
4. 位图数据，这部分的内容根据BMP位图使用的位数不同而不同，在24位图中直接使用RGB，而其他的小于24位的使用调色板中颜色索引值。

## 3.解释位平面

- 图像的每个像素通道占用8bit，将每个像素通道的特定位抽取出来，形成8个二值平面图像 – 位平面

## 4.LSB隐写方法的规则是什么？

- 方法：将欲嵌入的秘密信息转化为比特流，对比特流进行加密或置乱，逐行或逐列替换载体图像的最低比特位
- 规则：
  - 如果秘密信息与最低比特位相同，则不改动
  - 如果秘密信息与最低比特位不同，则使用秘密信息值代替最低比特位
    - 0嵌入偶数，1嵌入奇数→像素值不改变
    - 0嵌入奇数→像素值减1
    - 1嵌入偶数→像素值加1

## 5.LSB隐写方法的优缺点是什么？

### 优点

1. 隐蔽性好：由于只修改了像素或样本的最低有效位，所以在大多数情况下，这种改变对人的视觉或听觉几乎是不可察觉的，实现了较好的隐蔽性。
2. 简单易实现：LSB隐写技术的算法相对简单，容易理解和实现，不需要复杂的数学知识或高级编程技巧。
3. 嵌入率高：与其他隐写技术相比，LSB可以在不引起注意的前提下嵌入较大量的数据。
4. 适用范围广：可以应用于各种类型的数字文件，如图片、音频和视频文件。
5. 对图像质量的影响小

### 缺点

1. 易被检测：尽管LSB隐写对肉眼来说不易察觉，但使用专门的隐写分析工具可以较容易地检测到其存在。特别是在对图像或音频进行统计分析时，LSB隐写可能会留下可检测的痕迹。
2. 对文件压缩敏感：如果载体文件经过压缩，尤其是有损压缩，那么嵌入的信息可能会被破坏，因为压缩过程可能会改变最低有效位。
3. 抗干扰能力弱：由于信息是存储在最低有效位中，任何形式的编辑或处理（如裁剪、调整亮度、添加滤镜等）都可能破坏隐藏的信息。
4. 安全性有限：如果攻击者知道使用了LSB隐写技术，且能够访问原始载体，那么隐藏的信息可以相对容易地被恢复或破坏。
5. 鲁棒性差：LSB隐写术的鲁棒性相对较差，这意味着一旦载体数据（如图像）经过压缩、格式转换等操作，嵌入的隐藏信息可能会丢失或损坏。

data:image/svg+xml,%3csvg%20xmlns='http://www.w3.org/2000/svg'%20width='12'%20height='12'%20viewBox='0%200%2012%2012'%20fill='none'%3e%3cpath%20d='M1.98484%207.23472L5.73484%203.48472C5.76967%203.44986%205.81103%203.4222%205.85655%203.40333C5.90208%203.38445%205.95088%203.37474%206.00016%203.37474C6.04944%203.37474%206.09824%203.38445%206.14376%203.40333C6.18928%203.4222%206.23064%203.44986%206.26547%203.48472L10.0155%207.23472C10.0858%207.30509%2010.1254%207.40052%2010.1254%207.50004C10.1254%207.59955%2010.0858%207.69498%2010.0155%207.76535C9.9451%207.83571%209.84967%207.87524%209.75016%207.87524C9.65065%207.87524%209.55521%207.83571%209.48485%207.76535L6.00016%204.28019L2.51547%207.76535C2.48063%207.80019%202.43927%207.82783%202.39374%207.84668C2.34822%207.86554%202.29943%207.87524%202.25016%207.87524C2.20088%207.87524%202.15209%207.86554%202.10657%207.84668C2.06105%207.8

2783%202.01969%207.80019%201.98484%207.76535C1.95%207.73051%201.92237%207.68914%201.90351%207.64362C1.88465%207.5981%201.87495%207.54931%201.87495%207.50004C1.87495%207.45076%201.88465%207.40197%201.90351%207.35645C1.92237%207.31093%201.95%207.26956%201.98484%207.23472Z'%20fill='%230057FF'%20stroke='%230057FF'%20stroke-width='0.5'/%3e%3c/svg%3e

6. 安全性较低：由于LSB隐写术嵌入信息的方式较为简单，熟悉隐写术的人员可能相对容易地提取隐藏信息。

data:image/svg+xml,%3csvg%20xmlns='http://www.w3.org/2000/svg'%20width='12'%20height='12'%20viewBox='0%200%2012%2012'%20fill='none'%3e%3cpath%20d='M1.98484%207.23472L5.73484%203.48472C5.76967%203.44986%205.81103%203.4222%205.85655%203.40333C5.90208%203.38445%205.95088%203.37474%206.00016%203.37474C6.04944%203.37474%206.09824%203.38445%206.14376%203.40333C6.18928%203.4222%206.23064%203.44986%206.26547%203.48472L10.0155%207.23472C10.0858%207.30509%2010.1254%207.40052%2010.1254%207.50004C10.1254%207.59955%2010.0858%207.69498%2010.0155%207.76535C9.9451%207.83571%209.84967%207.87524%209.75016%207.87524C9.65065%207.87524%209.55521%207.83571%209.48485%207.76535L6.00016%204.28019L2.51547%207.76535C2.48063%207.80019%202.43927%207.82783%202.39374%207.84668C2.34822%207.86554%202.29943%207.87524%202.25016%207.87524C2.20088%207.87524%202.15209%207.86554%202.10657%207.84668C2.06105%207.82783%202.01969%207.80019%201.98484%207.76535C1.95%207.73051%201.92237%207.68914%201.90351%207.64362C1.88465%207.5981%201.87495%207.54931%201.87495%207.50004C1.87495%207.45076%201.88465%207.40197%201.90351%207.35645C1.92237%207.31093%201.95%207.26956%201.98484%207.23472Z'%20fill='%230057FF'%20stroke='%230057FF'%20stroke-width='0.5'/%3e%3c/svg%3e

7. 容量限制：尽管LSB隐写术可以嵌入大量信息，但仍然受到载体数据容量的限制，无法在不引起明显质量损失的情况下隐藏过多的信息。

data:image/svg+xml,%3csvg%20xmlns='http://www.w3.org/2000/svg'%20width='12'%20height='12'%20viewBox='0%200%2012%2012'%20fill='none'%3e%3cpath%20d='M1.98484%207.23472L5.73484%203.48472C5.76967%203.44986%205.81103%203.4222%205.85655%203.40333C5.90208%203.38445%205.95088%203.37474%206.00016%203.37474C6.04944%203.37474%206.09824%203.38445%206.14376%203.40333C6.18928%203.4222%206.23064%203.44986%206.26547%203.48472L10.0155%207.23472C10.0858%207.30509%2010.1254%207.40052%2010.1254%207.50004C10.1254%207.59955%2010.0858%207.69498%2010.0155%207.76535C9.9451%207.83571%209.84967%207.87524%209.75016%207.87524C9.65065%207.87524%209.55521%207.83571%209.48485%207.76535L6.00016%204.28019L2.51547%207.76535C2.48063%207.80019%202.43927%207.82783%202.39374%207.84668C2.34822%207.86554%202.29943%207.87524%202.25016%207.87524C2.20088%207.87524%202.15209%207.86554%202.10657%207.84668C2.06105%207.82783%202.01969%207.80019%201.98484%207.76535C1.95%207.73051%201.92237%207.68914%201.90351%207.64362C1.88465%207.5981%201.87495%207.54931%201.87495%207.50004C1.87495%207.45076%201.88465%207.40197%201.90351%207.35645C1.92237%207.31093%201.95%207.26956%201.98484%207.23472Z'%20fill='%230057FF'%20stroke='%230057FF'%20stroke-width='0.5'/%3e%3c/svg%3e

2783%202.01969%207.80019%201.98484%207.76535C1.95%207.73051%201.92237%207.68914%201.90351%207.64362C1.88465%207.5981%201.87495%207.54931%201.87495%207.50004C1.87495%207.45076%201.88465%207.40197%201.90351%207.35645C1.92237%207.31093%201.95%207.26956%201.98484%207.23472Z'%20fill='%230057FF'%20stroke='%230057FF'%20stroke-width='0.5'/%3e%3c/svg%3e

8. 易受攻击：LSB隐写术相对容易受到各种攻击手段的破坏，例如通过统计分析或特定算法检测到隐藏信息的存在，从而破坏隐写的效果。

## 6. LSB隐写方法的评价指标是什么？

- 嵌入容量
- 嵌入率
- PSNR（峰值信噪比）
- 安全性（抗隐写分析能力）

# 第二章 面向JPEG图像的隐写方法

---

## 1. JPEG图像与BMP图像相比较的优缺点是什么？

### JPEG图像的优点：

1. 压缩率高，文件小：JPEG使用有损压缩技术，可以大幅度减小文件大小，便于存储和分享，适合网络传输和存储。
2. 广泛兼容：JPEG是最广泛支持的图像格式之一，几乎所有的图像查看软件、网页浏览器和数码相机都支持JPEG格式。
3. 调整压缩质量：JPEG格式允许用户在压缩时选择质量等级，平衡图像质量和文件大小。

### JPEG图像的缺点：

1. 有损压缩：压缩过程会损失一部分图像信息，特别是在重复编辑和保存时，图像质量会逐渐下降。
2. 不支持透明度：JPEG格式不支持透明背景，这限制了其在某些设计工作中的应用。

## BMP图像的优点：

1. 无损质量：BMP格式存储的是未压缩的像素数据，因此图像质量最高，没有任何质量损失。
2. 简单直接：BMP格式结构简单，易于程序处理和编辑，不涉及复杂的压缩算法。
3. 设备无关性：BMP文件是设备无关的，意味着它可以跨设备显示，无需考虑设备的显示能力
4. ~~支持透明度（取决于具体实现）：某些BMP变种格式支持透明通道，适合需要透明背景的图像处理。~~

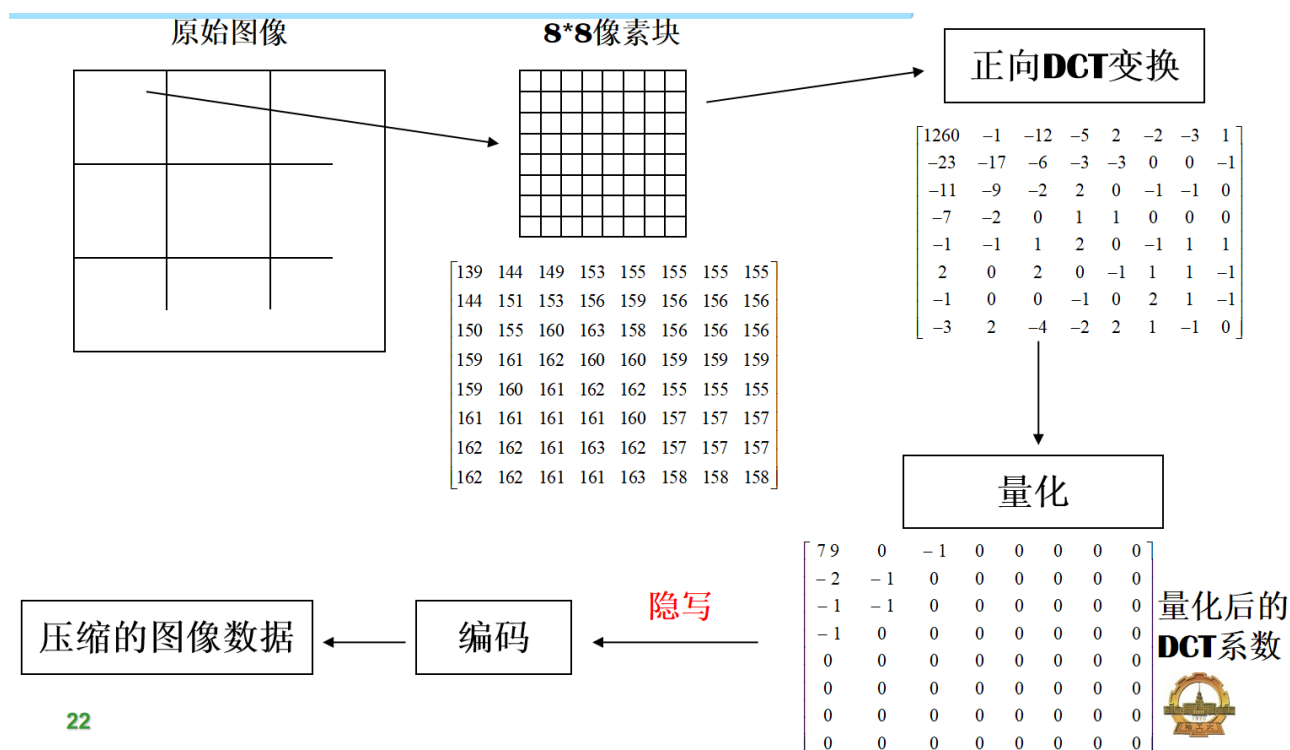
## BMP图像的缺点：

1. 文件大小大：由于不进行压缩，BMP文件的大小远大于同等尺寸的JPEG文件，占用更多的存储空间和带宽。
2. 兼容性和共享性差：相对于JPEG，BMP格式文件因其较大的大小，在网络上传输和分享时不太方便。
3. 效率低：较大的文件大小意味着在加载、编辑和保存时需要更多的时间和资源。

## 2.DCT变换前图像进行哪些预处理，目的是什么？

- DCT目的：压缩
- 图像分块
  - 将图像分割成大小是8\*8的小块，这些小块在整个压缩过程中都是单独被处理的
  - 目的：将大任务分割成小任务，减小计算量，实现并行化处理，加快图像压缩速度
- 将彩色图像转换成为YCbCr模型
  - Y表示亮度(Luminance)，Cb和Cr分别表示绿色和红色的“色差值”
  - 目的：把重要的信息和不重要的信息分开，分离亮度和色彩信息，因为人眼对亮度的变化比对色彩的变化更敏感。通过分离，可以对色度分量应用更高级别的压缩，而不是那么显著地影响到图像的视觉质量。

### 3. JPEG压缩分为哪几个步骤？哪些位置适合隐写，为什么？



- 在量化后和编码前进行信息隐写的原因是因为这一阶段既保证了隐写信息的稳定性又尽量减少了对图像质量的影响。量化步骤通过简化图像数据来减小文件大小，而编码步骤则对这些简化后的数据进行高效压缩。通过在这两个步骤之间插入隐写信息，可以确保：
  - 隐写信息在量化过程中不会丢失。
    - 量化过程和正向DCT变换对图像数据进行了有损压缩，减少了图像的数据量。如果在量化之前添加隐写信息，该信息可能会在量化过程中被修改或丢失。
  - 对图像视觉质量的影响最小化，因为已经考虑了人眼对图像细节的敏感度。
  - 隐写操作的灵活性和效率，利用JPEG压缩算法的特性来更好地隐藏和保留信息。
- 量化和DCT计算不可逆



#### 4. JPEG压缩过程中使用DCT变换的目的是什么？

通过**DCT**，空间表达式可以转换成频谱表达式或频率域，在频域中只用少量的数据就可以表示空间域中大量的数据，从而达到数据压缩的目的。

1. 可以将图像的像素空间转换到频域中，从而用少量的数据表示图像。
2. **DCT**产生的系数容易被量化，能获得较好的压缩快。
3. **DCT**算法的性能好，计算速度快。
4. **DCT**算法是可逆的，所以利用逆**DCT**变换可以解压图像。

#### 5. 相比于Jsteg隐写F3隐写算法进行了哪些方面的改进？

- **Jsteg**隐写是将秘密信息嵌入在量化后的DCT系数的LSB上，但原始值为-1，0，+1的DCT系数除外。
- **F3**隐写

每个**非0的DCT**数据用于隐藏**1比特秘密信息**，为**0的DCT**系数不负载秘密信息。

如果秘密信息与**DCT的LSB**相同，便不作改动；如果不同，将**DCT系数的绝对值减小1**，符号不变。

当**原始值为+1或-1且预嵌入秘密信息为0**时，将这个位置**归0并视为无效**，在下一个**DCT**系数上重新嵌入。

1. 减少可见性：F3在嵌入信息时不增加系数的绝对值。相比之下，Jsteg可能通过增加或减少系数的值来嵌入信息，这在某些情况下可能会引入更明显的视觉变化。F3仅在需要时减小系数值，减少了隐写对图像质量的影响。
2. 提高隐蔽性：由于F3避免了对系数进行增加的操作，它在统计分析下更难被检测。在Jsteg中，频繁的增加操作可能会在统计上引起可检测的异常，而F3通过仅仅减小系数值的方式，减少了这种风险。



## 6. 计算使用Jsteg算法和F3算法嵌入秘密信息后的DCT系数

$$\begin{bmatrix} 97 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ -3 & -2 & 4 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

## 原始DCT系数

**0 0 1 1 0**

秘密信息

$$\text{Jstep: } \begin{bmatrix} 96 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ -2 & -3 & 5 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . \end{bmatrix}$$

$$F_3 : \begin{bmatrix} 96 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 53 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

# 第三章 感知影响的估计以及感知模型

---

## 1. 质量和保真度的相同点和不同点是什么？

- 保真度：是处理前后信号相似度的一种度量。
- 质量：是指对吸引力的绝对度量
- 相同点
  - 衡量标准：质量和保真度都是用来评估信息在传输或处理过程中的保持程度，无论是图像、视频还是音频。
  - 都是对图像感知可见性的估计与度量
- 不同点
  - 在一些场合，保真度是评价含隐藏信息图像好坏的重要指标，目的是尽量保持原始图像与处理后的图像的一致性
  - 在另一些场合下，图像的质量是重要的评价指标，比如，一个现代艺术品图像的保护等。
  - 评价标准不同

## 2. 主观测试的具体步骤是什么？

- 准备、训练、哑元测试、正式测试、评分
- 准备
  - 在正式测试之前，评价员将被告知测试的目的和方法，并被要求依照图像检索的场景来熟悉和理解前表规定的评分等级。
- 训练
  - 示例程序被用来让评价员熟悉测试程序的流程
  - 在示例程序中，5个不同的参考图和5个不同强度攻击的目标图被按照正式测试的逻辑呈现，并要求评价员按照正式测试来给出评分
  - 评价员在示例程序中的评分不计入最终的测试结果，但可以用来确认他对试验的目的和方法有了正确的认识
  - 在示例程序中出现的参考图不会在正式测试中出现。
- 哑元测试
  - 在训练结束之后，评价员被告知进入正式测试
  - 但事实上，最初几个测试结果作为哑元测试，其结果并不被程序记录，只是为了让评价员适应测试的环境。

- 正式测试

- 由多个循环的展示（**Presentation**）构成，每个展示随机的从测试数据库中选择一个参考图，并相应的随机选择由某个攻击方式和某个攻击强度产生目标图，得到一个评分记录。
- 在随机选择中，相同的参考图和相同的攻击方式都不会连续出现两次一个展示包括四个时间固定的阶段
- 评分阶段的时延为5秒，在这5秒之内，评价员必须给出目标图质量/失真的打分
- 如果由于评价员的犹豫错过了这个时间，测试程序将自动进行下一次循环，并放弃这一次打分的记录。

- 评分

### 3. 视觉测量的最主要特性是什么？

- 频率灵敏度

- 人类听视觉系统对于某一输入信号的响应依赖于信号的频率。对于听觉来说，不同频率的信号会被感知成不同的音调。对于视觉来说，存在三种形式的频率响应。
  - a. 空间频率（图案或者纹理）
  - b. 谱频率（色彩）
  - c. 时间频率（运动或者闪烁）

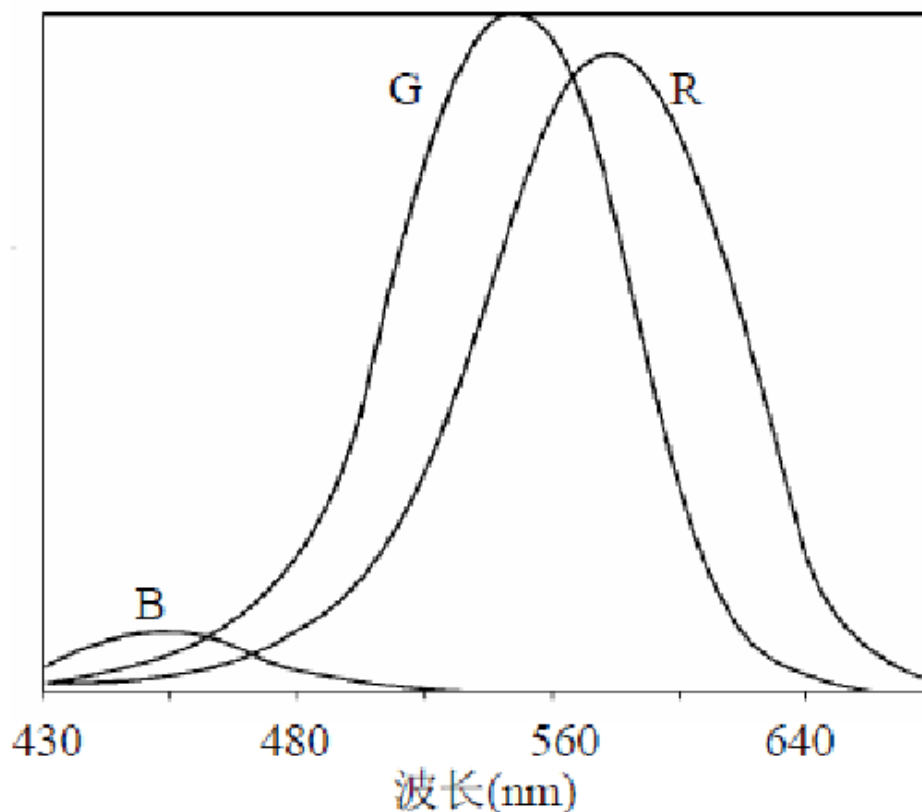
- 亮度灵敏度

- 人眼对于亮度越高的信号越不敏感。
- 亮度灵敏度是非线性的，并且已经通过对数关系、立方根关系以及一些更为复杂的模型对此进行了建模。

### 4. 人们对红绿蓝三种颜色中的哪种颜色最敏感？

- 绿色

相对灵敏度



## 5. 什么是掩蔽效应？

- 掩蔽效应是一种测度，它用来测量观察者在“掩蔽”信号存在的情况下对某一激励的反应。
  - 背景环境会影响人类的感知效果，一个信号的存在能够隐藏或掩蔽另一个信号的存在。
  - 例如，在一幅图像加入均匀噪声后，虽然噪声是均匀的，但在这幅图像中噪声的可见性却非常不均匀，它明显地依赖于局部图像结构。在相对平滑的区域，噪声明显可见。相比之下，在纹理复杂部分，噪声几乎完全不可见。

## 6. 建立Watson模型的目的什么？

- 利用灵敏度、掩蔽以及综合的思想，现在来描述Watson提出的一个用来测量视觉逼真度的模型。
- 用来测量视觉逼真度，评价含噪声图像的感知效果，远远优于 PSNR。
- Watson模型是一种基于视觉感知理论的图像质量评估模型。它的目的是模拟人类视觉系统的感知特性，以更准确地评估图像质量和视觉失真。这种模型特别强调在图像处理和压缩中造成的视觉失真评估，尤其是在有损压缩算法中，如何在减少数据量的同时尽可能保留图像的视觉质量。

- **提高压缩效率**：通过理解人类视觉系统对图像细节的感知，可以更有效地设计压缩算法，去除人眼难以察觉的信息，从而在不显著降低感知质量的情况下达到更高的压缩比。
- **优化图像处理**：在图像编辑、去噪、增强等处理过程中，利用Watson模型可以指导处理算法保留或强调视觉上重要的特征，改善图像的视觉效果。
- **客观评估图像质量**：提供一种基于视觉感知的方法，用于客观评价图像的视觉质量，这对于图像处理技术的研究和开发非常重要。

## 第四章 基于视觉特性的隐写技术

---

- 位平面复杂度分割(bit-plane complexity segmentation, BPCS)隐写技术
- 像素值差异(pixel-value differencing, PVD)隐写技术

### 1.基于视觉特性的隐写技术优点是什么？

- 提高了信息隐藏的隐蔽性
- 增加信息嵌入率
- 利用视觉特性进行更为隐蔽的秘密信息嵌入
  - 在视觉不敏感的区域嵌入较多的秘密信息
  - 在视觉较敏感的区域嵌入较少的秘密信息

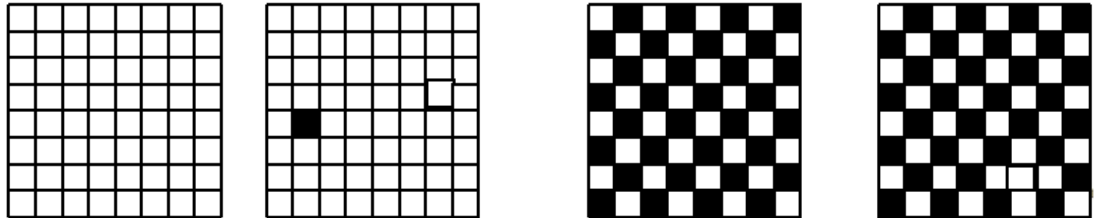
### 2.如何评价图像块的复杂度？

- C等于所有相邻像素对中取值不同的像素对数目，越大越复杂
- $C_{max} = 7 \text{（每行7对）} * 8 \text{（共8行）} * 2 \text{（行、列均考虑）} = 112$

## ● BPCS隐写的主要思想：

- 将载体数据的多个位平面都分成固定大小的小块，由于人的视觉对变化剧烈、复杂度较高的位平面小块不敏感，所以用这些位平面小块来负载秘密信息
- 考虑到人的视觉特性，有较好的隐蔽性
- 信息嵌在多个位平面上，可能有更大的容量
- 可以应用于空间域，也可以用于小波压缩域等变换域

5



## 3.为什么要将秘密信息进行共轭处理？

- 将复杂度小的秘密信息复杂度提高，避免直接替换时对图像影响过大
- 可逆
- 共轭处理即将秘密信息构成的块与棋盘状小块做异或，这样可以增加复杂度。

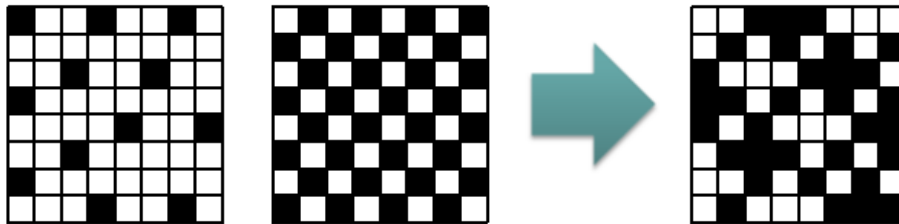
取  $a < 0.5$ ，如果原来的复杂度小于  $aC_{\max}$ ，经过共轭处理之后复杂度变为  $(1 - a)C_{\max}$ ，满足大于  $aC_{\max}$  的条件，适合进行隐写。

## 4.阐述BPCS隐写算法过程

### ● BPCS隐写的嵌入步骤：

1. 将图像的所有位平面分成相同大小的块(如8x8)
2. 计算每个小块的复杂度  $C_{\max}$ 
  - $C_{\max}$  等于所有相邻像素对中取值不同的像素对数目
  - 8x8的块， $C_{\max}$  的取值范围为0-112

3. 将复杂度大于 $aC_{\max}$ 的位平面块用于负载秘密信息，其中 $a$ 是参数，其值必须小于0.5
4. 将秘密信息构成同样大小的块：
  - 若其复杂度大于 $aC_{\max}$ ，则直接替换原位平面
  - 若其复杂度小于 $aC_{\max}$ ，则做共轭处理后再替换
    - 共轭处理即将秘密信息构成的块与棋盘状小块做异或



5. 记录下经过共轭处理的小块信息(位置、位平面)，将其也作为秘密信息，嵌入载体中
  - 可以预先划定一部分区域用来记录该信息
  - 可以在嵌入主要信息之后依次嵌入共轭处理块的信息

## 5.使用PVD方法实现隐写

秘密信息：1110101

像素值：(105, 113, 118, 101)



- PVD隐写的嵌入步骤：

- 记小块中像素灰度差值为  $d = p_{i+1} - p_i$ 。


其中， $|d|$  的取值范围为  $[0, 255]$

- 将整个亮度范围分成K个区域 $R_k(k=0,1,...,K-1)$ ，每个区域的宽度都是2的整数幂

- 例如：  $[0, 7], [8, 15], [16, 31], [32, 63], [64, 127], [128, 255]$

- 记每个区域的下界、上界和宽度为： $l_k, u_k, w_k$

- 如果 $|d|$  落在区域 $R_k$ 中，则这个小块中嵌入  $\log_2 w_k$  个秘密信息比特

**[0,7],[8,15],[16,31],[32,63],[64,127],[128,255]** 

**12      3      3      4      5      6      7**

- 将  $\log_2 w_k$  个秘密信息比特转化为十进制值  $b$ ，并计算：

$$d' = \begin{cases} l_k + b, & d \geq 0 \\ -(l_k + b), & d < 0 \end{cases}$$

- 嵌入过程如下：

$$(p'_i, p'_{i+1}) = f[(p_i, p_{i+1}), d'] = \begin{cases} (p_i - r_c, p_{i+1} + r_f), & d' \bmod 2 = 1 \\ (p_i - r_f, p_{i+1} + r_c), & d' \bmod 2 = 0 \end{cases}$$

- 其中： $r_c = \left\lceil \frac{d' - d}{2} \right\rceil, r_f = \left\lfloor \frac{d' - d}{2} \right\rfloor$



(105, 113)

$$d_1 = p_{i+1} - p_i = 113 - 105 = 8 \quad [8, 15] \quad 3 \quad 111 \quad b=7$$

$$\begin{array}{ccc} l_k & u_k & w_k \\ 8 & 15 & 8 \end{array} \quad d'_1 = l_k + b = 8 + 7 = 15 \quad d'_1 \geq 0$$

$$r_c = \left\lceil \frac{d'_1 - d_1}{2} \right\rceil = \left\lceil \frac{15 - 8}{2} \right\rceil = 4$$

$$r_f = \left\lfloor \frac{d'_1 - d_1}{2} \right\rfloor = \left\lfloor \frac{15 - 8}{2} \right\rfloor = 3$$

$$d_1 \bmod 2 = 0 \quad (p'_i, p'_{i+1}) = (105 - 3, 113 + 4) \\ = (102, 117)$$

(118, 101)

$$d_2 = p_{i+1} - p_i = 101 - 118 = -17 \quad [16, 31] \quad 4 \quad 0101 \quad b=5$$

$$\begin{array}{ccc} l_k & u_k & w_k \\ 16 & 31 & 16 \end{array} \quad d'_2 = -(l_k + b) = -(16 + 5) = -21$$

$$r_c = \left\lceil \frac{d'_2 - d_2}{2} \right\rceil = \left\lceil \frac{-21 + 17}{2} \right\rceil = -2$$

$$r_f = \left\lfloor \frac{d'_2 - d_2}{2} \right\rfloor = \left\lfloor \frac{-21 + 17}{2} \right\rfloor = -2$$

$$d_2 \bmod 2 = 1 \quad (p'_i, p'_{i+1}) = (118 + 2, 101 - 2) \\ = (120, 99)$$

(118, 101, 120, 99)

## 第五章 调色板图像的隐写方法

### 1. 描述调色板图像的结构及应用领域

- 1 不超过 256 种颜色的调色板，定义每种颜色的 RGB 各颜色分量值。
- 2 图像内容，其中每个像素是一个不超过 8 位的索引值，指向调色板中的对应颜色。

格式一般为 GIF 和 PNG。应用领域为测绘卫星、气象卫星，因为这些图比较小。

# 调色板图像的结构

## 1. 调色板（颜色表）：

- 调色板是一个有限的颜色集合，每个颜色在调色板中都有一个唯一的索引号。调色板的大小可以根据需要而定，常见的有256色（8位）、16色（4位）等。
- 每个颜色通常由红、绿、蓝（RGB）三个颜色分量组成，每个分量的取值范围通常是0到255。

## 2. 像素数据：

- 图像的每个像素不直接存储颜色值，而是存储一个指向调色板中某个颜色的索引。通过这种方式，图像数据可以被高效地压缩，尤其是对于包有限颜色的图像。

## 3. 图像头信息：

- 图像文件通常包含头信息，描述图像的宽度、高度、调色板大小、颜色深度等基本信息，以便正确地解析和显示图像。

# 调色板图像的应用领域

## 1. Web图像：

- 在Web早期，带宽较低，为了减少图像文件的大小，经常使用调色板图像。GIF和部分PNG格式的图像就是基于调色板的，它们在保证足够图像质量的同时，能够显著减少文件大小。

## 2. 数字艺术和游戏设计：

- 在数字艺术和早期游戏设计中，调色板图像因其文件大小小和易于操作的特点而广泛使用。设计师可以通过修改调色板中的颜色来快速改变图像或游戏场景的整体色调，这在计算和存储资源有限的时代尤为重要。

## 3. 数据可视化：

- 在数据可视化领域，调色板用于映射或表示不同的数据值，通过颜色的变化直观地显示数据的分布、密度或变化趋势。

## 4. 隐写术：

- 如前所述，调色板图像也可以用于隐写术，即在图像中隐藏信息。通过微妙地调整调色板中颜色的顺序或选择，可以在不影响图像视觉效果的前提下，隐藏大量信息。

## 5. 图像处理和压缩：

- 在图像处理领域，调色板图像常用于特定的压缩算法中，通过优化调色板来减少图像文件的大小，同时尽可能保持图像质量。

2. 对于颜色数量为N的调色板图像Gifshuffle算法可容纳多少信息?  
bit

$$\lfloor \log_2(N!) \rfloor$$

### 3.描述Gifshuffle算法的过程

1. 计算可容纳的最大嵌入量

$$\lfloor \log_2(N!) \rfloor$$

2. 将二进制数转换为十进制

3. 根据密码本，改变对应索引值

1 将真实颜色进行排序

2 秘密信息变为十进制:  $1010 \rightarrow 10$

3 查询密码本对应的序列: 10 对应的序列为 1, 3, 2, 4

4 改变图像内容中索引顺序为 1, 3, 2, 4

### 4. 使用EZStego算法为什么要将亮度依次排序?

相邻两个亮度之间互相交换时，图片变化不明显

## 5. 描述EZStego算法过程

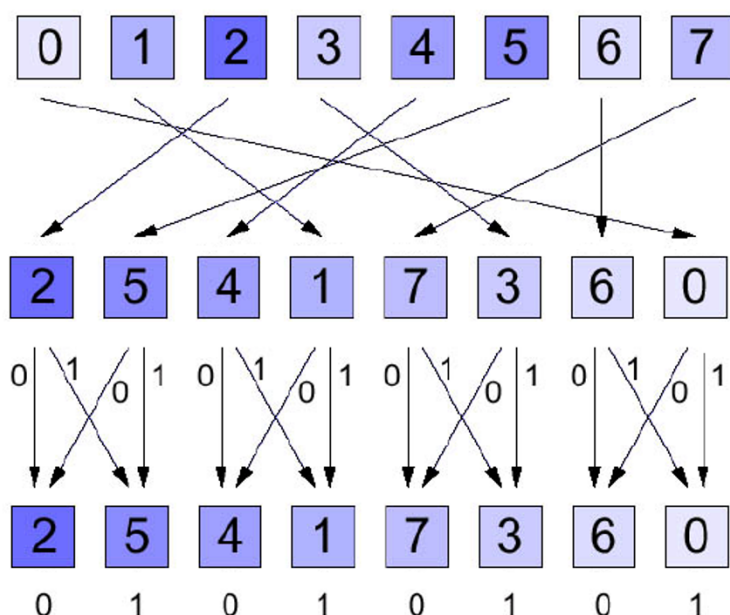
1. EZStego隐写是将调色板的颜色亮度依次排序，颜色的亮度是由不同颜色分量线性叠加而成。

其表达式为：

$$Y = 0.299R + 0.587G + 0.114B$$

2. 为每个颜色分配一个亮度序号，然后用奇数序号表示嵌入秘密比特1，用偶数序号表示嵌入秘密比特0。

3. 将调色板图像像素内容（索引值）使用LSB隐写代替，并将图像像素索引值改为新的亮度序号所对应的索引值。



1. 将调色板的颜色亮度依次排序
2. 为每个颜色亮度分配一个序号
3. 将调色板图像像素内容（索引值）使用LSB隐写代替，并将图像像素索引值改为新的亮度序号所对应的索引值。
4. 然后用奇数序号表示嵌入秘密比特1，用偶数序号表示嵌入秘密比特0。

两两一对，进行嵌入

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 139 | 144 | 149 | 153 | 155 | 155 | 155 | 155 |
| 144 | 151 | 153 | 156 | 159 | 156 | 156 | 156 |
| 150 | 155 | 160 | 163 | 158 | 156 | 156 | 156 |
| 159 | 161 | 162 | 160 | 160 | 159 | 159 | 159 |
| 159 | 160 | 161 | 162 | 162 | 155 | 155 | 155 |
| 161 | 161 | 161 | 161 | 160 | 157 | 157 | 157 |
| 162 | 162 | 161 | 163 | 162 | 157 | 157 | 157 |
| 162 | 162 | 161 | 161 | 163 | 158 | 158 | 158 |

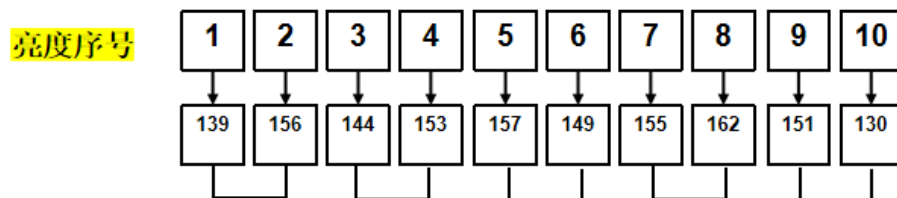
1 0 1 1 1



|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 139 | 153 | 157 | 144 | 155 | 155 | 155 | 155 |
| 144 | 151 | 153 | 156 | 159 | 156 | 156 | 156 |
| 150 | 155 | 160 | 163 | 158 | 156 | 156 | 156 |
| 159 | 161 | 162 | 160 | 160 | 159 | 159 | 159 |
| 159 | 160 | 161 | 162 | 162 | 155 | 155 | 155 |
| 161 | 161 | 161 | 161 | 160 | 157 | 157 | 157 |
| 162 | 162 | 161 | 163 | 162 | 157 | 157 | 157 |
| 162 | 162 | 161 | 161 | 163 | 158 | 158 | 158 |

原始图像索引值

隐写后图像索引值



15



1. 139 的亮度序号为1，是奇数，要嵌入的是1，不需改变
2. 144 的亮度序号为3，是奇数，要嵌入的是0，需要改变，与亮度序号为4的153交换
3. 149 的亮度序号为6，是偶数，要嵌入的是1，需要改变，与亮度序号为5的157交换
4. 153 的亮度序号为4，是偶数，要嵌入的是1，需要改变，与亮度序号为3的144交换
5. 155 的亮度序号为7，是奇数，要嵌入的是1，不需改变


根据调色板所能表示的颜色，在嵌入信息时，将该颜色更换为其中能表示的最相近的颜色，来保证图像变化最小

## 6.描述调色板图像隐写分析原理

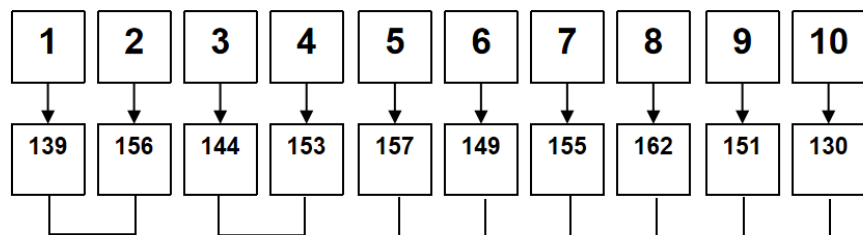
根据调色板所能表示的颜色，在嵌入信息时，将该颜色更换为能表示的最相近的颜色，来保证图像变化最小

7.

使用EZStego算法完成以下的隐写

|     |     |     |     |     |     |     |     |                                                                                                                           |
|-----|-----|-----|-----|-----|-----|-----|-----|---------------------------------------------------------------------------------------------------------------------------|
| 139 | 144 | 149 | 153 | 155 | 155 | 155 | 155 | <b>0 1 0 0 0</b><br><b>1 0 1 1 1</b><br> |
| 144 | 151 | 153 | 156 | 159 | 156 | 156 | 156 |                                                                                                                           |
| 150 | 155 | 160 | 163 | 158 | 156 | 156 | 156 |                                                                                                                           |
| 159 | 161 | 162 | 160 | 160 | 159 | 159 | 159 |                                                                                                                           |
| 159 | 160 | 161 | 162 | 162 | 155 | 155 | 155 |                                                                                                                           |
| 161 | 161 | 161 | 161 | 160 | 157 | 157 | 157 |                                                                                                                           |
| 162 | 162 | 161 | 163 | 162 | 157 | 157 | 157 |                                                                                                                           |
| 162 | 162 | 161 | 161 | 163 | 158 | 158 | 158 |                                                                                                                           |

原始图像索引值





|   |     |     |   |     |
|---|-----|-----|---|-----|
| 0 | 139 | 1 奇 | → | 156 |
| 1 | 144 | 3 奇 | ✓ | 144 |
| 0 | 149 | 6 偶 | ✓ | 149 |
| 0 | 153 | 4 偶 | ✓ | 153 |
| 0 | 155 | 7 奇 | → | 162 |
| 1 | 155 | 7 奇 | ✓ | 155 |
| 0 | 155 | 7 奇 | → | 162 |
| 1 | 155 | 7 奇 | ✓ | 155 |
| 1 | 144 | 3 奇 | ✓ | 144 |
| 1 | 151 | 9 奇 | ✓ | 151 |

## 第六章 隐写分析技术

### 1. 隐写分析的目的是什么？

- 判断一张图像是否含有隐蔽信息
- 是否有，长度，内容

## 2. 使用卡方检验进行隐写分析的原理是什么？

### ● $\chi^2$ 分析方法的原理

- 设图像中灰度值为 $j$ 的像素数为 $h_j$ ,  $j \in [0, 255]$
  - 嵌入的0, 1比特的概率各为50%
  - 改动规则:
    - $2i \rightarrow 2i+1$  或  $2i+1 \rightarrow 2i$
    - 不会有:  $2i \rightarrow 2i-1$  或  $2i+1 \rightarrow 2i+2$
  - 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。
  - 如果图像未经密写,  $h_{2i}, h_{2i+1}$  的值会相差较远。
- 先计算  $(2i+2i+1)/2$ , 隐写后的值的直方图, 算这个的分布
  - 然后计算要检测的图像的直方图是否与刚刚的分布拟合
  - 若拟合了, 就是隐写了

## 3. 什么是正向翻转, 负向翻转和0翻转？

正向翻转 $F_1$ , 负向翻转 $F_{-1}$ 和0翻转 $F_0$ 。

$F_1$ 为 $2i$ 与 $2i+1$ 的相互变化关系, 即0-1, 2-3, ..., 254-255;

$F_{-1}$ 为 $2i-1$ 与 $2i$ 的相互变化关系, 即-1-0, 1-2, ..., 255-256;

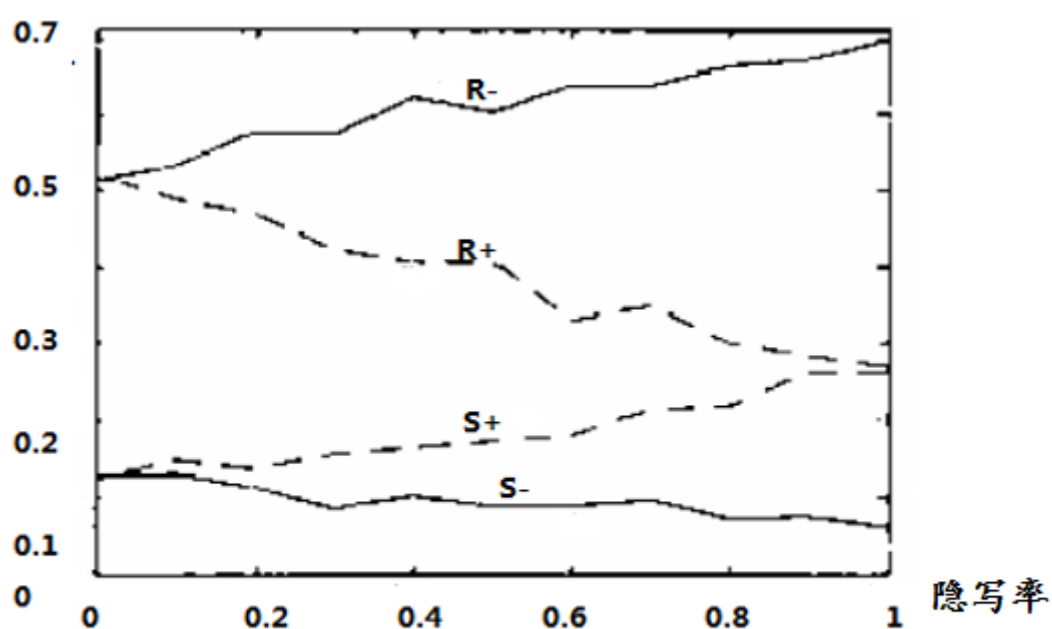
0翻转  $\rightarrow$  不变

## 4. 描述RS隐写分析过程, 阐述其原理

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。

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的比例记为  $R_{-m}$ ，像素相关性减小的图像块比例记为  $S_{-m}$ 。

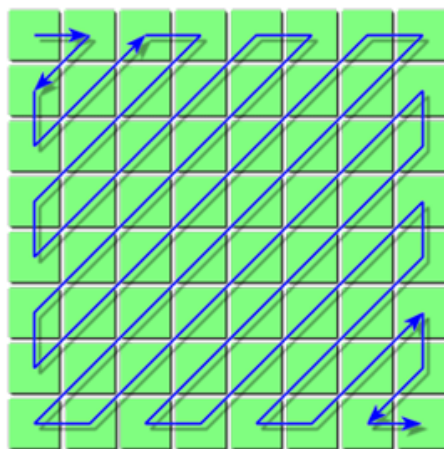
如果待检测图像没有经过LSB密写，那么无论是经过非正翻转还是非负翻转，会等同的增加图像块的混乱度，都应该满足以下规律： $R_m \approx R_{-m}$ ， $S_m \approx S_{-m}$ ，并且  $R_m > S_m$ ， $R_{-m} > S_{-m}$ 。



不同隐写率下的  $R, S$

## 像素相关性计算

对图像块进行之子排序：

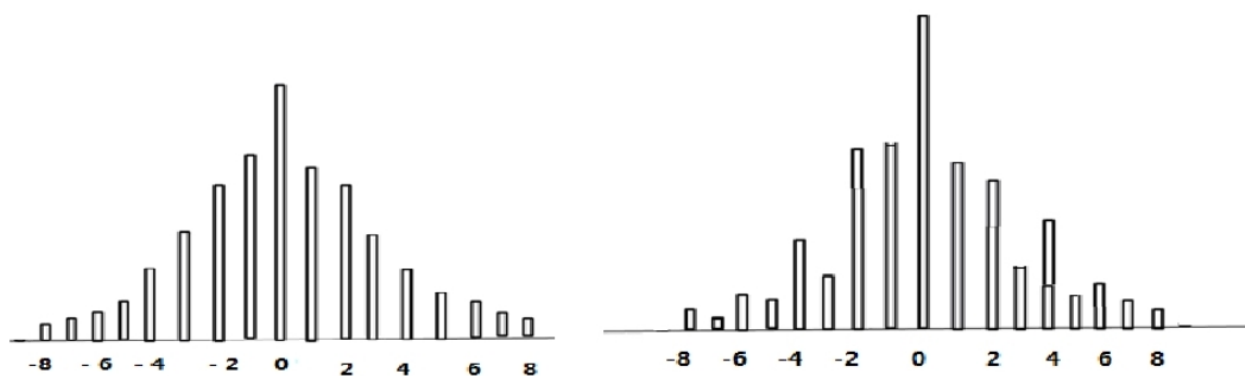


$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} |x_i - x_{i+1}|$$

$x_i$ ：像素值       $n$ ：像素个数

## 5. 怎样判断DCT系数直方图是否改变？

- 从系数直方图服从拉普拉斯分布变成无序分布



拉普拉斯分布

- 1 未隐写时，DCT 系数服从 Laplace 分布
- 2 隐写之后，DCT 系数不再服从 Laplace 分布，利用分布拟合，给定置信区间，判断是否符合 Laplace 分布。

# 第七章 抗隐写分析的隐写

## 1. 直方图补偿法原理

- 通过在隐写后的图像中进行额外的操作，将隐写引起的直方图失真失真补偿回来。
- 主要补偿方式是通过调整未嵌入秘密信息的像素值。

假设原始图像中  $f_{2i} > f_{2i+1}$ ，要求如下：

用  $f_{2i+1}$  中未嵌入的像素来补偿  $f_{2i}$  中所有嵌入成  $f_{2i+1}$  的像素：

$$(1 - \alpha)f_{2i+1} \geq \frac{\alpha}{2}(f_{2i} - f_{2i+1})$$

步骤如下：

- 1 统计原始图像的灰度直方图，确定  $f_{2i}, f_{2i+1}$
- 2 计算秘密信息隐藏率  $\alpha$
- 3 将所有的像素值排成一个向量，长度为  $l$
- 4 秘密信息嵌入到  $\lfloor \alpha \times l \rfloor$  为止
- 5 如果  $f_{2i}$  变大，将  $\lfloor \alpha \times l \rfloor + 1$  之后的像素中的值为  $2i$  的像素变为  $2i+1$ ，尽可能保持直方图不发生变化；反之类似。

## 2. 直方图补偿法中怎样计算秘密信息嵌入率

$$(1 - \alpha)f_{2i+1} \geq \frac{\alpha}{2}(f_{2i} - f_{2i+1})$$
$$\Rightarrow \alpha \leq \frac{2f_{2i+1}}{f_{2i} + f_{2i+1}}$$

$f_i \dots$  原始图像像素数目  
 $\alpha \dots$  秘密信息嵌入率

## 3. 随机翻转嵌入法的原理

设秘密信息为  $w$ ，对应隐藏该位的像素灰度值为  $x(i, j)$ 。

对 LSB 方法进行修改，在嵌入数据时不但可将  $2i \leftrightarrow 2i + 1$ ，还可以将  $2i \leftrightarrow 2i - 1$ 。

秘密信息  $w$  与  $x(i, j)$  最后一位比较，相同无需改变，不相同，做如下改变：

$$x(i, j) = \begin{cases} x(i, j) - 1, & T \leq 0, & 0 < x(i, j) < 255 \\ x(i, j) + 1, & T > 0, & 0 < x(i, j) < 255 \\ x(i, j) - 1, & & x(i, j) = 255 \\ x(i, j) + 1, & & x(i, j) = 0 \end{cases}$$

$$T = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} x(u, v) - 9x(i, j)$$

20190622140448

$T$  在  $x(i, j)$  处于边界时无法计算，这是可以采用其他方法给边界插值，特殊情况，对总体的影响不大。

这在进行 RS 隐写分析时，正负翻转都会增加混乱程度。

#### 4. 随机翻转嵌入法怎样处理边界溢出问题

- 255 翻转只能变成 254
- 0 翻转只能变成 1

#### 5. F5隐写算法使用了哪两种重要方法，其目的是什么？

- 混洗技术：使秘密信息分布在整幅图像中
- 矩阵编码：是为了提高嵌入效率。
- F5的主要特点是在  $2^k - 1$  个载体数据中嵌入  $k$  bit 的秘密信息，而最多只需要修改一个载体数据。

#### 6. F5隐写后图像的哪种统计特性发生了变化？

- 块边界的不连续性增加， $k$  越小，现象越明显

## 7. 使用F5隐写算法进行秘密信息嵌入，计算嵌入信息后的像素值

$$x_1, x_2: 0 \ 1$$

$$x_1, x_2, x_3: 1 \ 0 \ 1$$

$$a_1, a_2, a_3: 123 \ 124 \ 125$$

$$a_1, \dots, a_7: 13 \ 17 \ 16 \ 16 \ 19 \ 21 \ 20$$

- 第一个

$$a_1, a_2, a_3 = 1, 0, 1$$

$$x_1 = a_1 \oplus a_2 = 1 \oplus 0 = 1$$

$$x_2 = a_2 \oplus a_3 = 0 \oplus 1 = 1$$

只需要修改 $x_1$ 而不影响 $x_2$ ，修改 $a_1$ 为122

- 第二个

关键是构造等式使得  $x_1, x_2, x_3$ （待嵌入秘密信息）相互之间各有一个相同的载体数据（ $a$ ），可以采用这样的构造方法（需要  $2^k - 1 = 7$  个载体像素）：

$$x_1 = a_1 \oplus a_4 \oplus a_5 \oplus a_7$$

$$x_2 = a_2 \oplus a_4 \oplus a_6 \oplus a_7$$

$$x_3 = a_3 \oplus a_5 \oplus a_6 \oplus a_7$$

现在：

$$x_1, x_2, x_3 = 1, 0, 1$$

$$a_1, \dots, a_7 = 13, 17, 16, 16, 19, 21, 20 = 1, 1, 0, 0, 1, 1, 0$$

验证等式分别是：

$$1 \neq 1 \oplus 0 \oplus 1 \oplus 0$$

$$0 = 1 \oplus 0 \oplus 1 \oplus 0$$

$$1 \neq 0 \oplus 1 \oplus 1 \oplus 0$$

所以只需要修改影响  $x_1, x_3$  而不影响  $x_2$  的像素值的像素值，就是修改  $a_5$  为 18。

## 第八章 数字水印技术

---

### 1. 数字水印主要应用在哪些领域，其目的是什么？

- 版权保护
  - 所有者鉴别
  - 所有权验证



- 操作跟踪
- 拷贝控制
- 内容认证
  - 完整性认证
  - 篡改检测

## 2. 水印扩频的目的是什么？扩频有哪几种方式？

- 目的：提高水印的鲁棒性（随机性、不含纹理、类似噪声）。
- 方式：
  - 基于片率
    - 按位扩展：将每一个水印比特或符号重复多次的方法
    - 延拓扩展：将整个原始信息重复多次的方法
    - 随机扩展：将水印信号与一个随机序列相结合，通过随机选择宿主信号中的位置来嵌入水印比特
  - 基于伪随机序列
    - 单伪随机序列
    - 双伪随机序列

### 3. 描述水印嵌入过程，并给出各项名称

$$X' = X + (\alpha b - \lambda s')w$$

The diagram illustrates the components of the watermark embedding formula  $X' = X + (\alpha b - \lambda s')w$ . Arrows point from each variable to its corresponding term:

- $X'$  points to 含水印图像 (Image with watermark)
- $X$  points to 原始图像 (Original image)
- $\alpha$  points to 嵌入强度 (Embedding strength)
- $b$  points to 水印信息 (Watermark information)
- $\lambda$  points to 原始图像与扩频序列的相关性 (Correlation between original image and spread spectrum sequence)
- $s'$  points to 扩频序列 (Spread spectrum sequence)
- $w$  points to 扩频序列 (Spread spectrum sequence)

Additionally, the value  $=1$  is shown next to the  $\lambda$  term, indicating a specific correlation value.

$$s' = \frac{1}{N} \sum_{i=1}^N X_i w_i$$

- N 表示扩频序列长度
- 以上公式保证提取水印信息时四舍五入后得到的值为  $\alpha b$

### 4. 水印提取的原理是什么？

$$z = \frac{1}{N} \sum_{i=1}^N X_i' w_i$$

提取的原理是根据下面的等式：

$$\text{cor}(X', \omega) = \alpha b$$

还是上面的例子，计算：

$$\begin{aligned}\alpha b &= \text{cor}(X', \omega) \\ &= \frac{1}{4}(143, 140, 145, 157) \cdot (-1, 1, 1, -1) \\ &= \frac{1}{4} \times (-15) \\ &= -4\end{aligned}$$

因为  $\alpha = 4$ ，所以水印信息为  $-4/4 = -1$ 。

## 5. 计算含水印的像素值

$$X' = X + (\alpha b - \lambda s')w$$

$$s' = \frac{1}{N} \sum_{i=1}^N X_i w_i$$

水印信息：**-1 1 -1 1**

$$\alpha = 4$$

扩频序列：**1 → 1 -1 -1 1**

**-1 → 1 1 1 -1**

扩频水印：**1 -1 -1 1 1 -1 -1 1 -1 1 1 -1 1 -1 -1 1**

|     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 139 | 144 | 149 | 153 | 155 | 155 | 155 | 155 |
| 144 | 151 | 153 | 156 | 159 | 156 | 156 | 156 |
| 150 | 155 | 160 | 163 | 158 | 156 | 156 | 156 |
| 159 | 161 | 162 | 160 | 160 | 159 | 159 | 159 |
| 159 | 160 | 161 | 162 | 162 | 155 | 155 | 155 |
| 161 | 161 | 161 | 161 | 160 | 157 | 157 | 157 |
| 162 | 162 | 161 | 163 | 162 | 157 | 157 | 157 |
| 162 | 162 | 161 | 161 | 163 | 158 | 158 | 158 |

计算含水印图像像素值



来看一下第一位水印  $-1$ ，扩频水印为  $-1, 1, 1, -1$ ，首先计算相关度：

$$\begin{aligned}s' &= \frac{1}{4}(-1 \times 139 + 1 \times 144 + 1 \times 149 + (-1) \times 153) \\ &= \frac{1}{4}\end{aligned}$$

然后计算嵌入水印后的图像像素：

$$\begin{aligned}X' &= X + (\alpha b - \lambda s')\omega \\ &= (139, 144, 149, 153) + (-1, 1, 1, -1) \times (4 \times (-1) - 1 \times \frac{1}{4}) \\ &= (139, 144, 149, 153) + (4.25, -4.25, -4.25, 4.25) \\ &= (143.25, 139.75, 144.75, 157.25) \\ &= (143, 140, 145, 157)\end{aligned}$$

- 最后的结果要四舍五入

## 第九章 空域数字图像鲁棒水印

---

### 1. 空域数字图像鲁棒水印技术主要基于图像的哪种不变特征进行的？

- 不同灰度级的像素个数多少的比例不变性。（整体上）

### 2. 算法中使用低通滤波的目的是什么？

- 图像的低频部分更具鲁棒性，载体图像将通过高斯滤波去掉高频部分，得到宿主图像的低频部分。

### 3. 算法中怎样保证水印的安全性？

- $I_{low}$ ：宿主图像的低频部分
- $H$ ：不同灰度级下的像素个数

增加安全性：仅选择  $M$  个灰度级的像素来嵌入水印，其中  $M < 256$ 。

### 4. 描述空域数字图像鲁棒水印技术过程

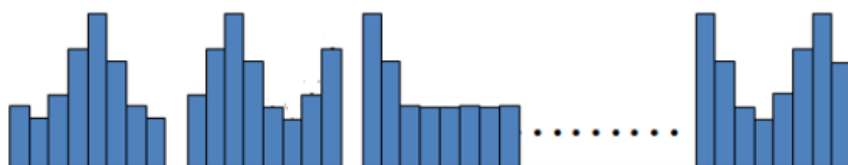
- 1 高斯滤波
- 2 抽取直方图，如上一题所述。
- 3 像素分组，将  $L_G$  个相邻灰度级下的像素分到同一个分组中，这样我们就会得到  $M_G = \lfloor M/L_G \rfloor$  个像素分组。第  $i$  个分组中有  $h_G(i)$  个像素。每组分  $B_1$  和  $B_2$  两部分。
- 4 水印嵌入，修改  $B_1$ ， $B_2$  中的像素，将  $B_1$  中的像素落入  $B_2$  中的灰度值，或者  $B_2$  中的像素落入到  $B_1$  中的灰度值，从而使得  $\frac{h'_{B1}(i)}{h_{B2}(i)} = \lambda$ （嵌入 1 的话是  $\lambda$ ，0 是  $\frac{1}{\lambda}$ ）

## 低通滤波

图像的低频部分更具鲁棒性，载体图像将通过高斯滤波去掉高频部分  $I_{high}$  得到宿主图像的低频部分  $I_{low}$ 。

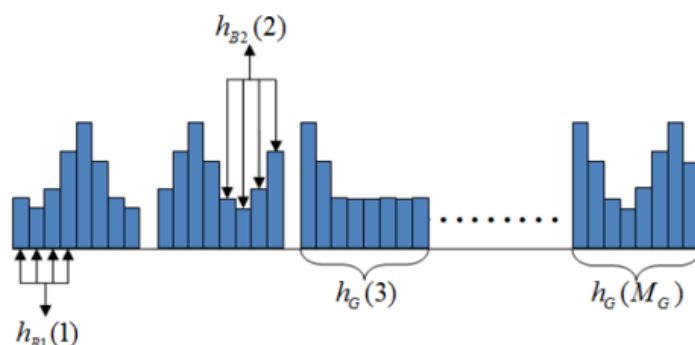
## 像素直方图提取

假设  $I_{low}$  表示宿主图像的低频部分， $H$  表示不同灰度级下的像素个数。为了增加水印算法的安全性，仅选择  $M$  个灰度级的像素来嵌入水印，其中  $M < 256$ 。



## 像素组选择

提取了图像低频部分的直方图 $H_M$ 后，我们将 $L_G$ 个相邻灰度级下的像素分到同一个分组中，这样我们就会得到 $M_G = \lfloor M/L_G \rfloor$ 个像素分组。第 $i$ 个分组中有 $h_G(i)$ 个像素。每组分为 $B_1$ 和 $B_2$ 两部分。



9

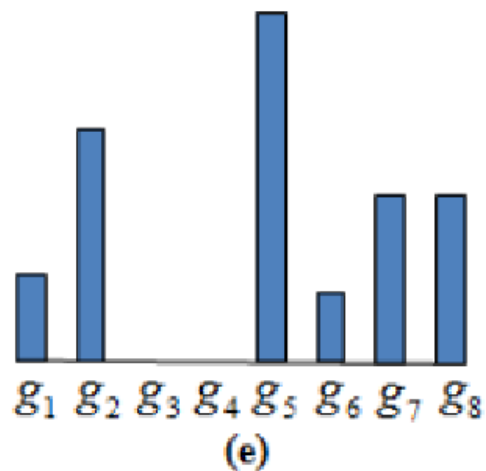
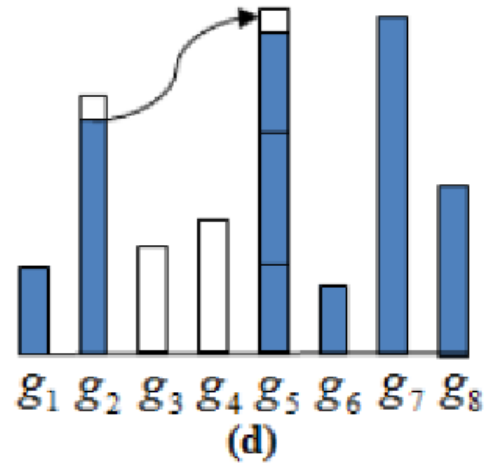
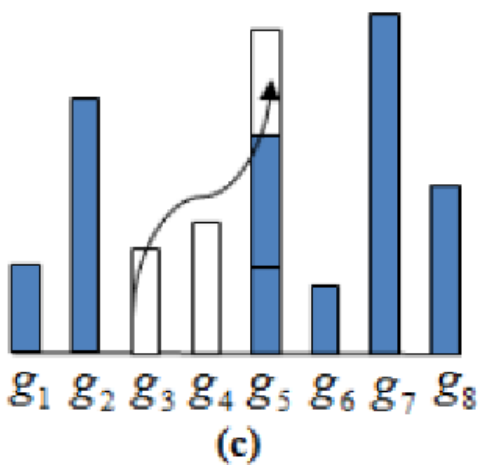
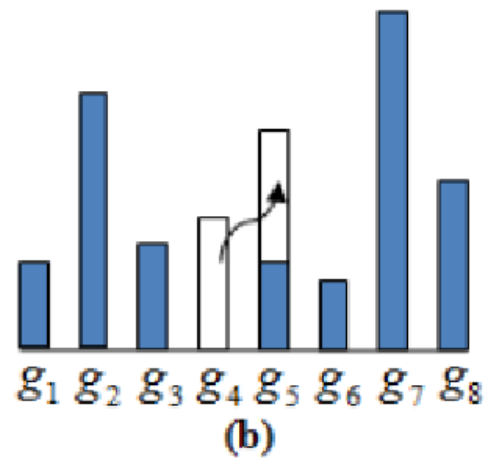
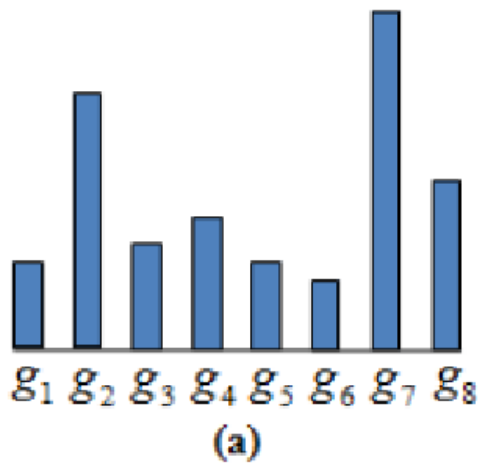


## 水印嵌入

$$\begin{cases} \frac{h'_{B1}(i)}{h'_{B2}(i)} = \lambda, & \text{if } w_i = 1 \text{ and } h_{B1}(i) < \lambda h_{B2}(i) \\ \frac{h'_{B1}(i)}{h'_{B2}(i)} = 1/\lambda, & \text{if } w_i = 0 \text{ and } h_{B2}(i) < \lambda h_{B1}(i) \end{cases}$$

假设 $W = (w_1, w_2, w_3, \dots, w_{L_{\max}})$ 为要嵌入的水印信息。每个像素分组 $h_{G(i)}$ 由Bin1与Bin2组成， $h_{B1}(i)$ 与 $h_{B2}(i)$ 则分别表示每个Bin中的像素数。

当 $w_i = 1$ 且 $h_{B1}(i) < \lambda h_{B2}(i)$ ，我们需要将Bin2中一定数目( $N_1$ )像素的灰度值改变使其落入到Bin1中从而使得 $\frac{h'_{B1}(i)}{h'_{B2}(i)} = \lambda$ 。同理，当 $w_i = 0$ 且 $h_{B2}(i) < \lambda h_{B1}(i)$ 时，我们需要将Bin1中一定数目( $N_0$ )的像素的灰度值改变使其落入到Bin2中，从而使得 $\frac{h'_{B1}(i)}{h'_{B2}(i)} = \frac{1}{\lambda}$ 。



## 5. 数字图像可逆水印的目的是什么？

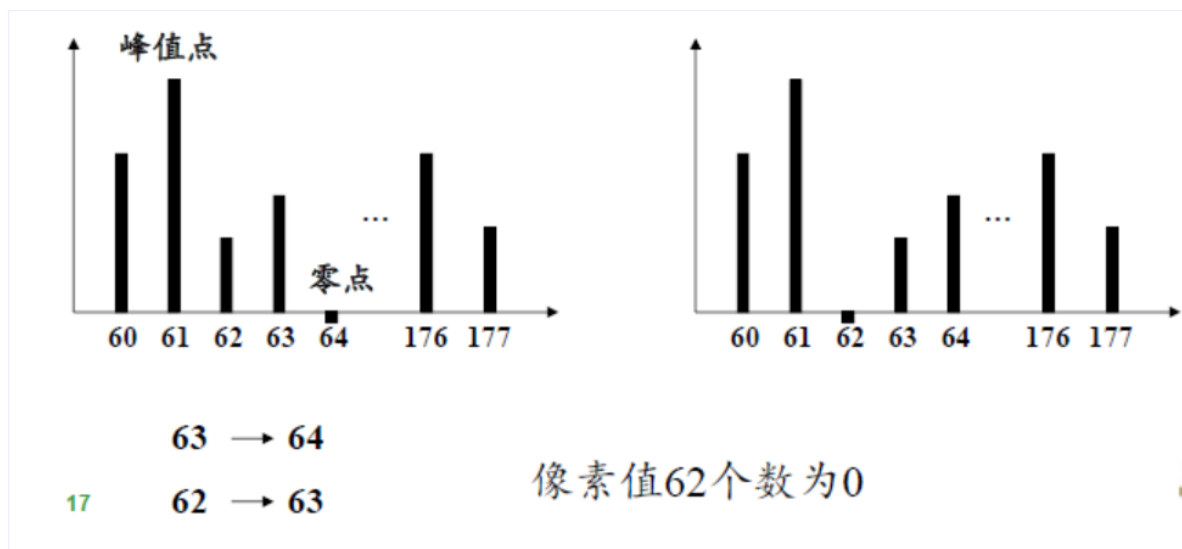
- 提取水印之后，原始图像还能够被恢复。军事、司法和医学用图像要求在加入水印后能够被无损恢复。
- 可逆水印是指嵌入原始媒体中的水印可被完全清除，原始媒体可完全恢复的一种水印，可逆水印又被称作无损水印或可逆信息隐藏



## 6. 描述基于直方图平移的数字图像可逆水印过程

嵌入：

- 1 判断像素值的峰值点 (61) 和零点 (64, 必须有零点, 不然无法弄)
- 2 将峰值点与零点的像素平移成相邻位置 ( $63 \rightarrow 64$ ,  $62 \rightarrow 63$ )
- 3 向峰值像素值中嵌入水印信息, 1 不变, 0 变为零点值 ( $61 \rightarrow 62$ )。



20190622225204

提取：

- 1 从像素值为 61 和 62 的像素值之中提取水印, 看到 61 提取水印 1, 看到 62 提取水印 0, 同时将 62 变为 61
- 2 平移修改的像素  $63 \rightarrow 62$ ,  $64 \rightarrow 63$