**信息隐藏第三次实验**

**1410658 杨旭东**

1. **流载体的LSB方法**
2. 嵌入：选择一个载体元素的子集 ，其中共有个元素，用以隐藏秘密信息的个比特。然后在这个子集上执行替换操作，把的最低比特用来替换。
3. 提取：找到嵌入信息的伪装元素的子集，从这些伪装对象中抽出它们的最低比特位，排列之后组成秘密信息。
4. **选择隐藏位置子集的方法**
5. 顺序选取

缺点：已嵌和未嵌部分数据特征不同。

1. 随机间隔法

用伪装密钥k作为伪随机数发生器的种子，生成一个伪随机序列，嵌入位置为



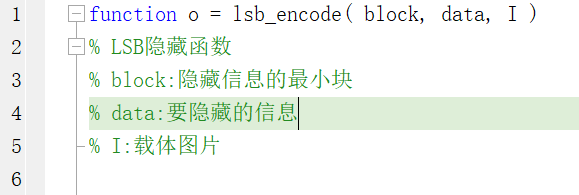
1. **LSB方法的特点**
2. 简单，易实现，容量大。
3. 安全性不高，不能抵抗叠加噪声、有损压缩等破坏。
4. **LSB算法的基本原理**

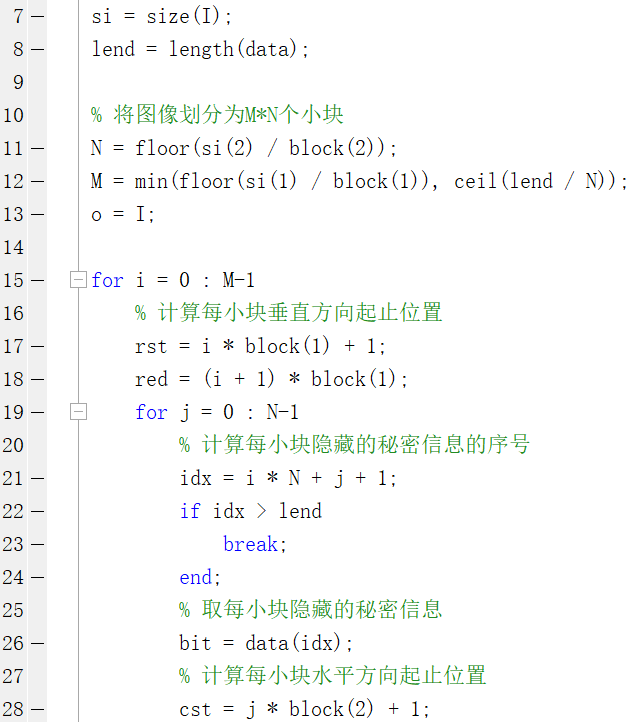
对空域的LSB做替换，用来替换LSB的序列就是需要加入的水印信息、水印的数字摘要或者由水印生成的伪随机序列。由于水印信息嵌入的位置是LSB，为了满足水印的不可见性，允许嵌入的水印强度不可能太高。然而针对空域的各种处理，如游程编码前的预处理，会对不显著分量进行一定的压缩，所以LSB算法对这些操作很敏感。

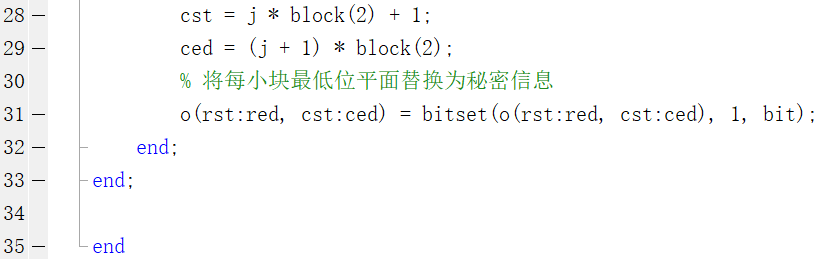
1. **LSB算法基本步骤**
2. 将原始载体图像的空域像素值由十进制转换成二进制；
3. 用二进制秘密信息中的每一比特信息替换与之相对应的载体数据的最低有效位；
4. 将得到的含秘密信息的二进制数据转换为十进制像素值，从而获得含秘密信息的图像。
5. **代码实现与解释**

使用MATLAB实现LSB算法，用lsb\_decode函数实现秘密信息嵌入，用lsb\_encode函数实现秘密信息提取，之后img\_lsb.m主程序对这两个函数进行调用测试，最后对提取信息进行误码率检测。

1. lsb\_encode

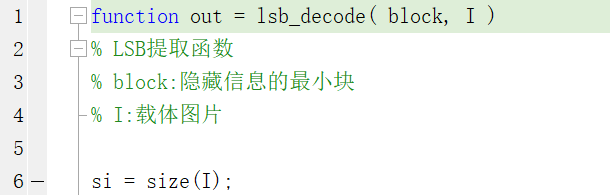


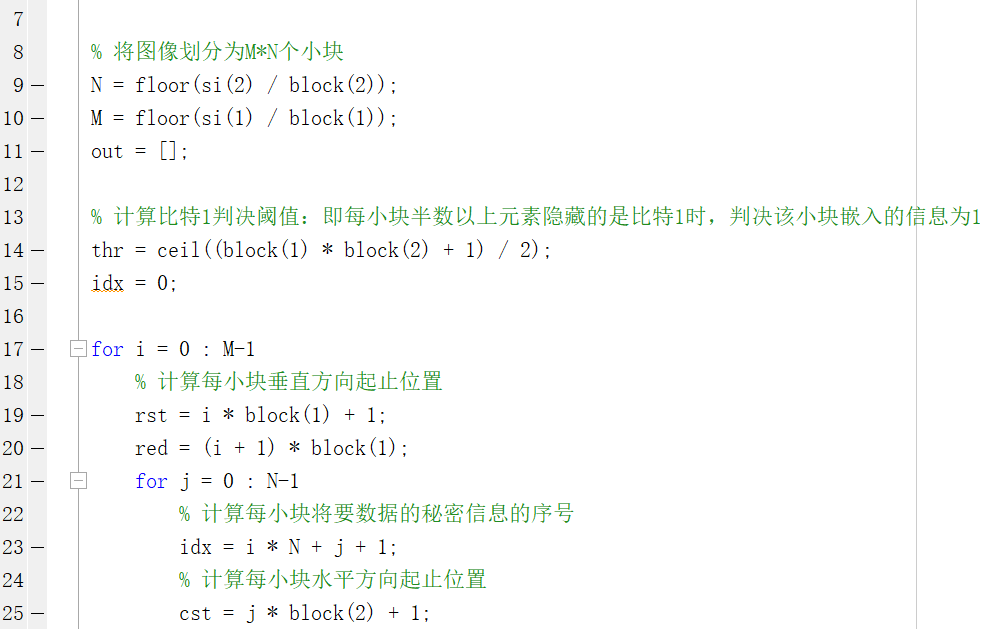


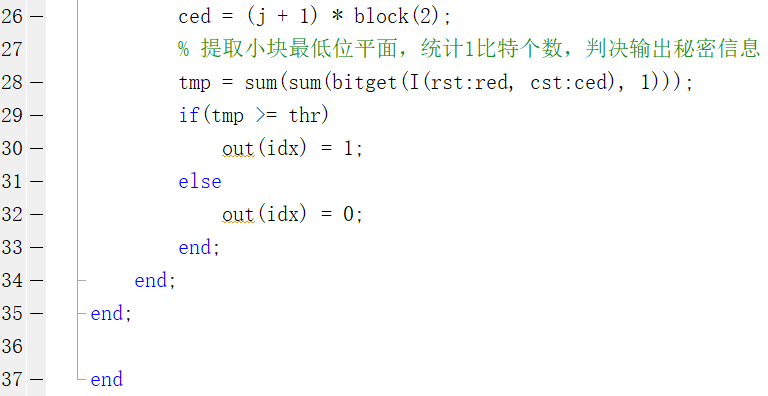


如果只把信息保存在一个像素中很容易丢失，所以根据信息长度与载体图片的长度划分小块，以一个小块存储一位信息，能大大增加信息隐藏的健壮性。

1. lsb\_decode函数



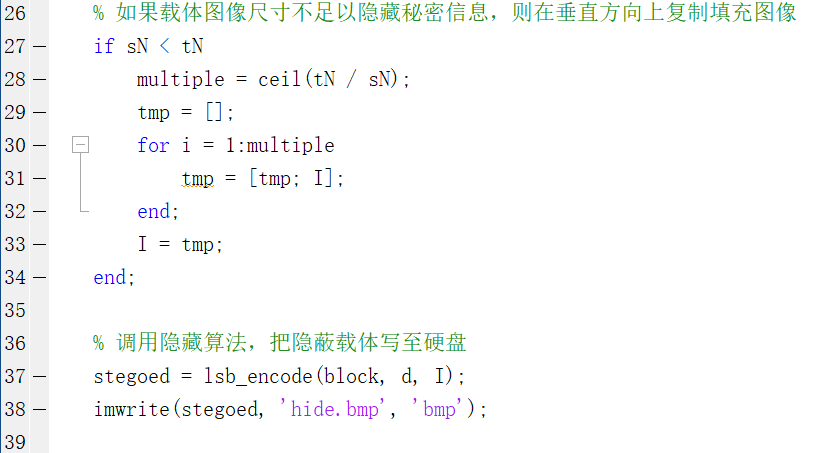




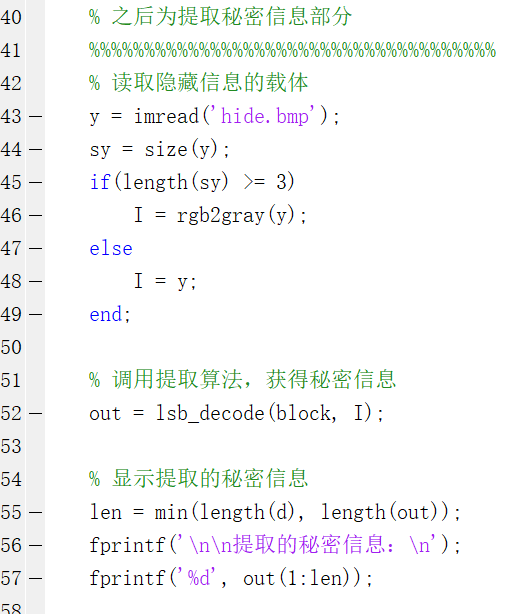
由于嵌入信息时是一小块为一个整体，所以提取时也同样要这样做，由于图片在传输或者保存过程中可能会有一些改变，所以计算每个小块内隐藏信息的阈值，如果一半以上为1则这个小块隐藏的信息为1否则为0。

1. 主程序img\_lsb.m

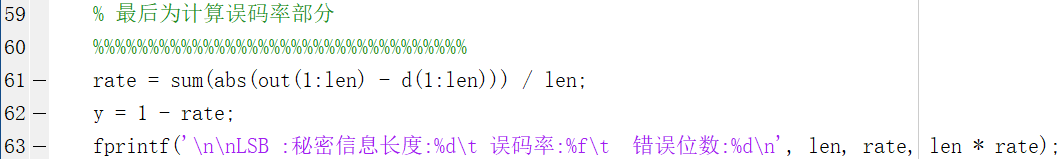




在调用嵌入函数之前，需要保证载体图片为灰度图片，如果载体图片不足以隐藏秘密信息，则人为把载体图片扩充以隐藏，以保证信息的正常嵌入。



调用提取函数，获得秘密信息。



最后计算一下提取的信息与隐藏信息的误码率，用于查看实验效果。

1. **实验结果展示**

载体原图：lena\_g.bmp 利用LSB嵌入秘密信息的图片：hide.bmp

运行结果：

