实验七 奇偶校验位隐藏法实验

学号：2110688 姓名：史文天 专业：信息安全

实验要求

1、隐藏：利用奇偶校验位隐藏法，实现将秘密图像嵌入到位图中；

2、提取：将秘密图像提取出来。

实验原理

1. 信息嵌入
   选择L(m) 个不相重叠区域，计算出每一区域I 的所有最低比特的奇偶校验位bi(i = 1, 2,..., n)。

$$b_i = \sum_{j \in I} LSB(c_j) mod 2$$

区域I 隐藏一个信息比特。若bi 与mi 不同，则将该区域中某个像素的最低比特位进行翻转，使得奇偶校验位与mi 相同，即bi = mi。

例如一个区域内所有像素的最低比特位有偶数个1，计算得奇偶校验位bi = 0。如果要嵌入的秘密信息比特为1，即mi = 1，要想满足bi = mi 则需要翻转某个像素的最低比特位，使得该区域的最低有效位有奇数个1，即bi = 1，从而满足bi = mi。

2. 信息提取

用同样的方法划分载体区域，计算出奇偶校验位，即可构成秘密信息。

原图像展示

实验步骤

主函数

```
function HideAndExtract()
    x=imread ('Lena.bmp'); %载体图像
    y=imread ('lion.bmp'); %秘密信息图像   是灰度图像，长宽均为载体图像的一半
    y=imbinarize(y);
    [m, n]= size(y);
```

```
    subplot(2, 2, 1);
    imshow(x) ; title('原始图像');

    subplot(2, 2, 2);
    imshow(y) ; title('水印图像');

    x=Hide(x,m,n,y);
    subplot(2, 2, 3);
    imshow(x ,[]) ; title('伪装图像');

    t=Extract();
    subplot(2,2,4);
    imshow(t,[]); title("提取出的水印图像");
end
```

- 首先，函数使用imread 函数从文件中读取两个灰度图像，分别作为载体图像和秘密信息图像。其中，秘密信息图像的大小应该是载体图像大小的一半。
- 然后，函数使用imbinarize 函数将秘密信息图像二值化，将其转换为一个二值图像。
- 接下来，函数使用size 函数获取秘密信息图像的大小，并将其存储在变量m 和n 中。
- 函数使用subplot 函数创建一个2x2 的图像窗口，并在第一个子图中显示载体图像，第二个子图中显示秘密信息图像。
- 函数调用Hide 函数，将秘密信息图像嵌入到载体图像中，并将结果存储在变量x 中。
- 函数使用subplot 函数在第三个子图中显示嵌入了秘密信息的伪装图像。
- 函数调用Extract 函数，从伪装图像中提取出嵌入的秘密信息，并将结果存储在变量t 中。
- 最后，函数使用subplot 函数在第四个子图中显示提取出的秘密信息图像。

奇偶校验函数

```
function out = checksum (x, i, j)
   %计 算 特 定 一 维 向 量 的 第m个 区 域 的 最 低 位 的 校 验 和
   temp= zeros(1, 4);
   temp(1) = bitget(x(2*i-1,2*j-1), 1);
   temp(2) = bitget(x(2*i-1,2*j), 1);
   temp(3) = bitget(x(2*i, 2*j-1), 1);
   temp(4) = bitget(x(2*i, 2*j ), 1);
   out=rem(sum(temp), 2);
end
```

- 首先，函数创建一个长度为4 的零向量temp，用于存储从x 中提取的四个比特位的值。
- 然后，函数使用bitget 函数从x 中提取四个比特位的值，并将它们存储在temp 向量中。具体来说，temp(1) 存储x(2i-1,2j-1) 的最低位，temp(2) 存储x(2i-1,2j) 的最低位，temp(3) 存储x(2i,2j-1) 的最低位，temp(4) 存储x(2i,2j) 的最低位。
- 接下来，函数使用sum 函数计算temp 向量中所有元素的和，并使用rem 函数计算这个和的模2 值，即为最终的校验和。

加密函数

```
function result=Hide(x,m,n,y)
    for i =1:m
        for j =1:n
            if checksum(x, i, j) ~= y(i, j) %需要反转一位
                random= int8(rand()*3);
                switch random  %任意反转一位
                    case 0
                        x(2*i-1,2*j-1)= bitset(x(2*i-1,2*j-1), 1, ~ bitget(x(2*i-
1,2*j-1), 1));
                    case 1
                        x(2*i-1,2*j)= bitset(x(2*i-1,2*j) , 1 , ~ bitget(x(2*i-
1,2*j), 1));
                    case 2
                        x(2*i, 2*j-1)= bitset(x(2*i, 2*j-1) ,1 ,~ bitget(x(2*i ,
2*j-1) , 1));
                    case 3
                        x(2*i , 2*j)= bitset(x(2*i , 2*j) , 1 , ~ bitget(x(2*i ,
2*j) , 1));
                end
            end
        end
    end
    imwrite(x , 'watermarkedImage.bmp');
    result=x;
end
```

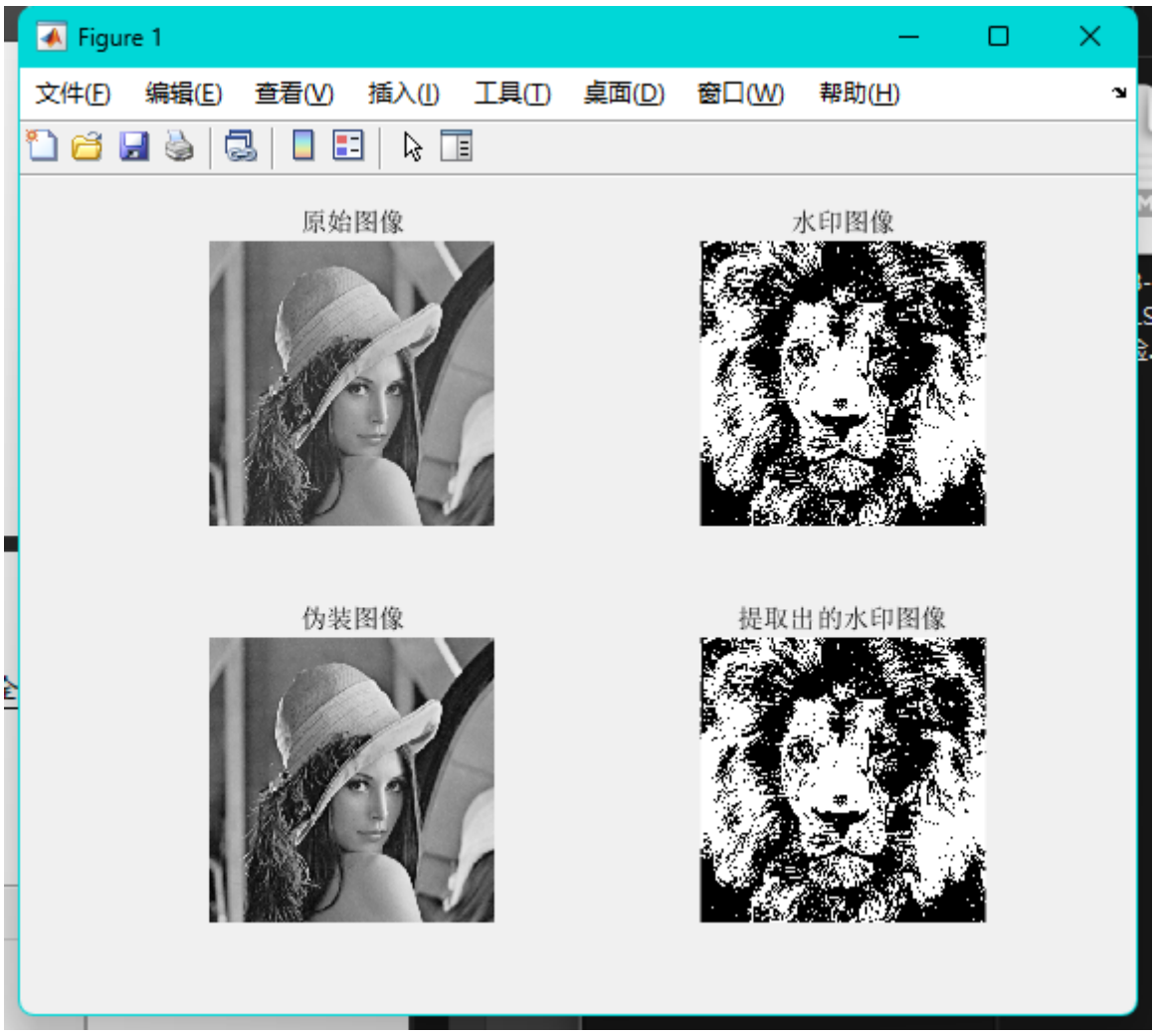
- 首先，函数使用两个整数m 和n 表示秘密信息图像的大小，并使用x 表示载体图像。
- 函数使用一个双重循环遍历秘密信息图像的每个像素，并使用checksum 函数计算载体图像中对应区域的校验和。
- 如果计算得到的校验和与秘密信息图像中对应像素的值不相等，说明需要在载体图像中反转一位来嵌入秘密信息。
- 函数使用rand 函数生成一个随机整数random，用于指定要反转的比特位。
- 函数使用switch 语句根据random 的值来反转载体图像中的一个比特位。具体来说，如果random 的值为0，则反转x(2i-1,2j-1) 的最低位；如果random 的值为1，则反转x(2i-1,2j) 的最低位；如果random 的值为2，则反转x(2i,2j-1) 的最低位；如果random 的值为3，则反转x(2i,2j) 的最低位。
- 函数使用imwrite 函数将嵌入了秘密信息的载体图像保存到文件中，并将结果存储在变量result 中。

解密函数

```
function out=Extract()
    c=imread('watermarkedImage.bmp');
    [m, n]= size(c);
    secret = zeros(m/2 , n/2);
    for i =1:m/2
        for j =1: n/2
            secret(i, j)= checksum(c, i, j);
        end
    end
    out=secret;
end
```

- 首先，函数使用imread 函数从文件中读取嵌入了秘密信息的灰度图像，并将其存储在变量c 中。
- 函数使用size 函数获取灰度图像的大小，并将其存储在变量m 和n 中。
- 函数创建一个大小为m/2 x n/2 的零矩阵secret，用于存储提取出的秘密信息。
- 函数使用一个双重循环遍历secret 矩阵的每个元素，并使用checksum 函数计算灰度图像中对应区域的校验和，并将其存储在secret 矩阵中。
- 函数将提取出的秘密信息矩阵存储在变量out 中，并将其作为函数的输出参数返回。

实验结果



实验总结

通过这次实验，我验证了奇偶校验位隐藏法的有效性，通过将秘密信息嵌入到数据流中并利用奇偶校验位的特性进行隐藏，实现了在传输过程中不引起明显变化。然而，在实际应用中需谨慎考虑安全性和攻击可能性，未来的研究可以进一步探索其在隐秘通信和数字水印等领域的潜在应用。