

信息隐藏技术实验七

李潇逸 2111454

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

方法一

该方法特点是翻转最低位，影响不大。把载体划分成几个不相重叠的区域，在一个载体区域中存储一比特信息。

信息隐藏

选择 $L(m)$ 个不相重叠区域，计算出每一区域所有像素的最低比特位的奇偶校验位（即“1”的个数的奇偶性），记为 $b_i (i = 1, 2, \dots, n)$ 。

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

嵌入信息时，在对应区域的奇偶校验位上嵌入信息比特 m_i （奇数个 1 代表 1，偶数个 1 代表 0），如果奇偶校验位 b_i 与 m_i 不匹配，则将该区域中所有元素的最低比特位进行翻转，使得奇偶校验位与 m_i 相同，即 $b_i = m_i$ 。

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

信息提取

在接收端，收方与发方拥有共同的伪装密钥作为种子，可以伪随机地构造载体区域。收方从载体区域中计算出奇偶校验位，排列起来就可以重构秘密信息。

方法二

该方法特点是翻转像素少。

把载体划分成几个不相重叠的区域，在一个载体区域中存储一比特信息。

信息隐藏

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 $b_i (i = 1, 2, \dots, n)$ 。

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

区域 I 隐藏一个信息比特。若 b_i 与 m_i 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 m_i 相同，即 $b_i = m_i$ 。

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 $b_i = 0$ 。如果要嵌入的秘密信息比特为 1，即 $m_i = 1$ ，要想满足 $b_i = m_i$ 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 $b_i = 1$ ，从而满足 $b_i = m_i$

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。

实验步骤

1. 读入载体图像和秘密图像
2. 获取秘密图像的长和宽（为载体图像的一半）
3. 依据上述长和宽进行循环，也就是将载体图像分为许多 $2 * 2$ 的小块
4. 循环过程中计算校验和
5. 将得到的校验和与秘密图像的值进行比较，相同不变，不同则随机将载体图像4位中的一位进行翻转
6. 提取时也进行循环，每 $2 * 2$ 的区域进行一次校验和计算，就可以得到最终的图像

代码

```
function HideAndExtract()
    x=imread ("C:\Users\19309\Desktop\document\信息隐藏技术\第七次实验\picture\Xiangling.png");
    y=imread ("C:\Users\19309\Desktop\document\信息隐藏技术\第七次实验\picture\lion.bmp"); %秘密
    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
```

```
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
```

```
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(
                    case 1
                        x(2*i-1,2*j)= bitset(x(2*i-1,2*j) , 1 , ~ bitget(x(
                    case 2
                        x(2*i, 2*j-1)= bitset(x(2*i, 2*j-1) ,1 , ~ bitget(x(
```

```

        case 3
            x(2*i , 2*j)= bitset(x(2*i , 2*j) , 1 , ~ bitget(x(
        end
    end
end
end
imwrite(x , 'watermarkedImage.bmp');
result=x;
end

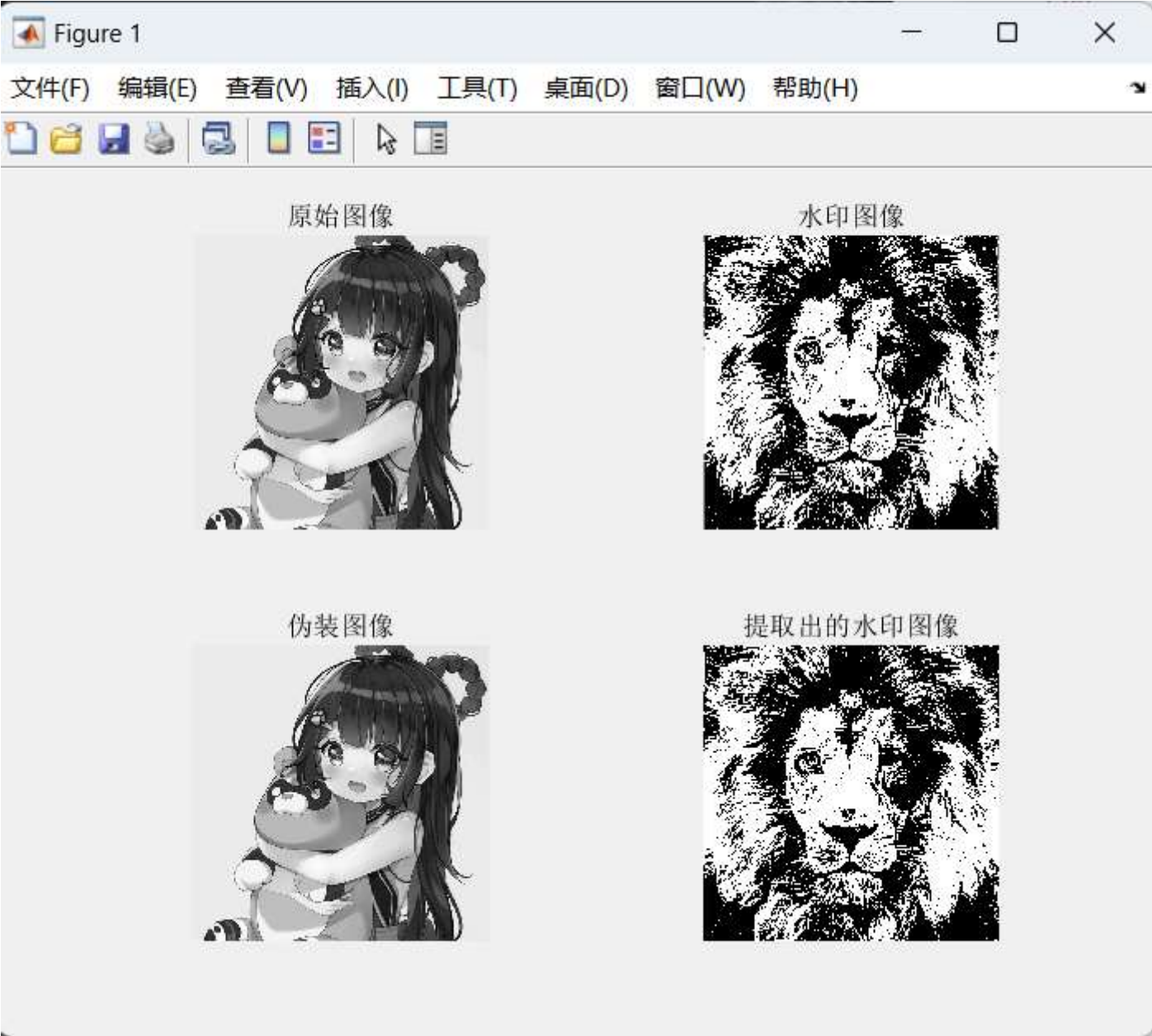
```

```

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

```

实验结果



心得体会

本次实验围绕图像水印的嵌入与提取展开，旨在通过将二进制形式的秘密信息图像融入载体图像之中，强化图像的版权保护与安全防护。实验经历不仅让我掌握了图像水印技术的基础理论与实践技巧，还在MATLAB平台上深化了对图像处理操作及关键函数运用的理解，同时，它强化了我对于信息安全在当前数字时代不可或缺性的认识。

实验伊始，我系统性地探索了图像水印的基本理念与分类体系。图像水印技术，作为一种隐匿或公开于图像内部的特定信息嵌入手段，扮演着守护图像版权与安全的关键角色。两类主要水印形式——可见水印，诸如公司标志、版权声明，直接肉眼可观；与不可见水印，如数字签名、加密代码等，需借助专门工具方可显现——为不同安全需求提供了灵活解决方案。

随后的实践环节，我详细学习并实践了图像水印嵌入与提取的标准化流程及核心技术。这一过程涵盖了从读取原始图像与秘密信息图像，到信息处理、校验和计算、像素位操作直至最终图像输出的一系列步

骤。对应地，提取过程则逆向操作，确保隐藏信息的准确复现。在此期间，我熟练掌握了MATLAB图像处理的关键指令，诸如 `imread` 用于图像读取，`imwrite` 完成图像保存，`imshow` 实现图像显示，以及 `imbinarize`、`bitset`、`bitget` 等用于二值图像处理与位操作的函数，还包括了随机数生成函数 `rand` 和逻辑判断工具 `switch`，这些技能为实现图像水印技术提供了坚实的工具支持。

我深刻体会到，在这个信息即是资产的时代，确保信息安全的重要性不言而喻。随着数字化进程加速，信息保护面临的挑战日益严峻，这促使我们不断探索和采用更为先进、有效的安全技术，如图像水印，来捍卫数据的完整性和私密性。通过本次实验，我不仅增长了技术知识，更加强化的对信息安全领域紧迫性和重要性的认知。