



南开大学
Nankai University

南 开 大 学

网 络 空 间 安 全 学 院

信息隐藏技术实验七

奇偶校验位隐藏法实验

穆禹宸 2012026

年级：2020 级

专业：信息安全、法学双学位班

指导教师：李朝晖

2023 年 4 月 11 日

目录

一、 实验要求	1
(一) 实验目的	1
(二) 实验环境	1
(三) 实验要求	1
二、 实验原理	1
(一) 方法一	1
1. 信息嵌入	1
2. 信息提取	2
(二) 方法二	2
1. 信息嵌入	2
2. 信息提取	2
三、 实验步骤	2
(一) 主函数	2
(二) 奇偶校验位函数	3
(三) 加密函数	4
(四) 解密函数	5
四、 实验结果	5
五、 实验心得体会	6

一、实验要求

(一) 实验目的

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。

(二) 实验环境

所需实验环境

主要有如下内容

- (1) 运行系统：Windows11
- (2) 实验工具：Matlab2022a
- (3) 数据：PNG 格式图像

(三) 实验要求

1. 在 Matlab 之中完成
2. 编写实验代码和报告，并给出截图
3. QQ 群提交作业

二、实验原理

(一) 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。

1. 信息嵌入

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 $b_i = m_i$ 。

2. 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。

(二) 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。

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 $b_i (i = 1, 2, \dots, n)$ 。

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

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 $b_i = m_i$ 。

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。

三、 实验步骤

本实验采用了第二种部分翻转法：

实验步骤

主要有如下四个步骤：

- (1) 主函数调用，实现整个实验。
- (2) 奇偶检验位函数。
- (3) 加密函数。
- (4) 解密函数。

(一) 主函数

```
1 function HideAndExtract()  
2     x=imread('Lena.bmp'); % 载体图像  
3     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  
4     y=imbinarize(y);  
5     [m, n]= size(y);
```

```

6
7     subplot(2, 2, 1);
8     imshow(x) ; title('原始图像');
9
10    subplot(2, 2, 2);
11    imshow(y) ; title('水印图像');
12
13    x=Hide(x,m,n,y);
14    subplot(2, 2, 3);
15    imshow(x ,[]) ; title('伪装图像');
16
17    t=Extract();
18    subplot(2,2,4);
19    imshow(t,[]); title(" 提取出的水印图像");
20 end

```

主函数调用方法用于将一个灰度图像嵌入到另一个灰度图像中，并提取出嵌入的图像。函数的具体实现如下：

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。

(二) 奇偶校验位函数

```

1 function out = checksum (x, i, j)
2     % 计算特定一维向量的第 m 个区域的最低位的校验和
3     temp= zeros(1, 4);
4     temp(1) = bitget(x(2*i-1,2*j-1), 1);
5     temp(2) = bitget(x(2*i-1,2*j), 1);
6     temp(3) = bitget(x(2*i, 2*j-1), 1);
7     temp(4) = bitget(x(2*i, 2*j ), 1);
8     out=rem(sum(temp), 2);
9 end

```

用于计算一个特定一维向量的第 `m` 个区域的最低位的校验和。函数的输入参数包括一个一维向量 `x`，以及两个整数 `i` 和 `j`，表示要计算的区域的位置。函数的输出参数是一个整数 `out`，表示计算得到的校验和。

函数的具体实现如下：

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。

总之，这段代码实现了一个简单的校验和计算函数，用于检测一个特定一维向量的第 m 个区域的最低位是否正确。

(三) 加密函数

这里借助随机数对 2*2 方块内的任意一个像素的最低比特进行翻转。

```

1 function result=Hide(x,m,n,y)
2     for i =1:m
3         for j =1:n
4             if checksum(x, i, j) ~= y(i, j) % 需要反转一位
5                 random= int8(rand()*3);
6                 switch random % 任意反转一位
7                     case 0
8                         x(2*i-1,2*j-1)= bitset(x(2*i-1,2*j-1), 1, ~
9                         ⇨ bitget(x(2*i-1,2*j-1), 1));
10                    case 1
11                        x(2*i-1,2*j)= bitset(x(2*i-1,2*j) , 1 , ~
12                        ⇨ bitget(x(2*i-1,2*j), 1));
13                    case 2
14                        x(2*i, 2*j-1)= bitset(x(2*i, 2*j-1) ,1 ,~ bitget(x(2*i ,
15                        ⇨ 2*j-1) , 1));
16                    case 3
17                        x(2*i , 2*j)= bitset(x(2*i , 2*j) , 1 , ~ bitget(x(2*i , 2*j)
18                        ⇨ , 1));
19                end
20            end
21        end
22    end
23    imwrite(x , 'watermarkedImage.bmp');
24    result=x;
25 end

```

用于将一个二值秘密信息图像嵌入到一个灰度载体图像中。函数的具体实现如下：

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。

(四) 解密函数

```
1 function out=Extract()
2     c=imread('watermarkedImage.bmp');
3     [m, n]= size(c);
4     secret = zeros(m/2 , n/2);
5     for i =1:m/2
6         for j =1: n/2
7             secret(i, j)= checksum(c, i, j);
8         end
9     end
10    out=secret;
11 end
```

用于从一个嵌入了二值秘密信息的灰度图像中提取出秘密信息。函数的具体实现如下:

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 $m/2 \times n/2$ 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。

四、 实验结果

实验结果如下所示:

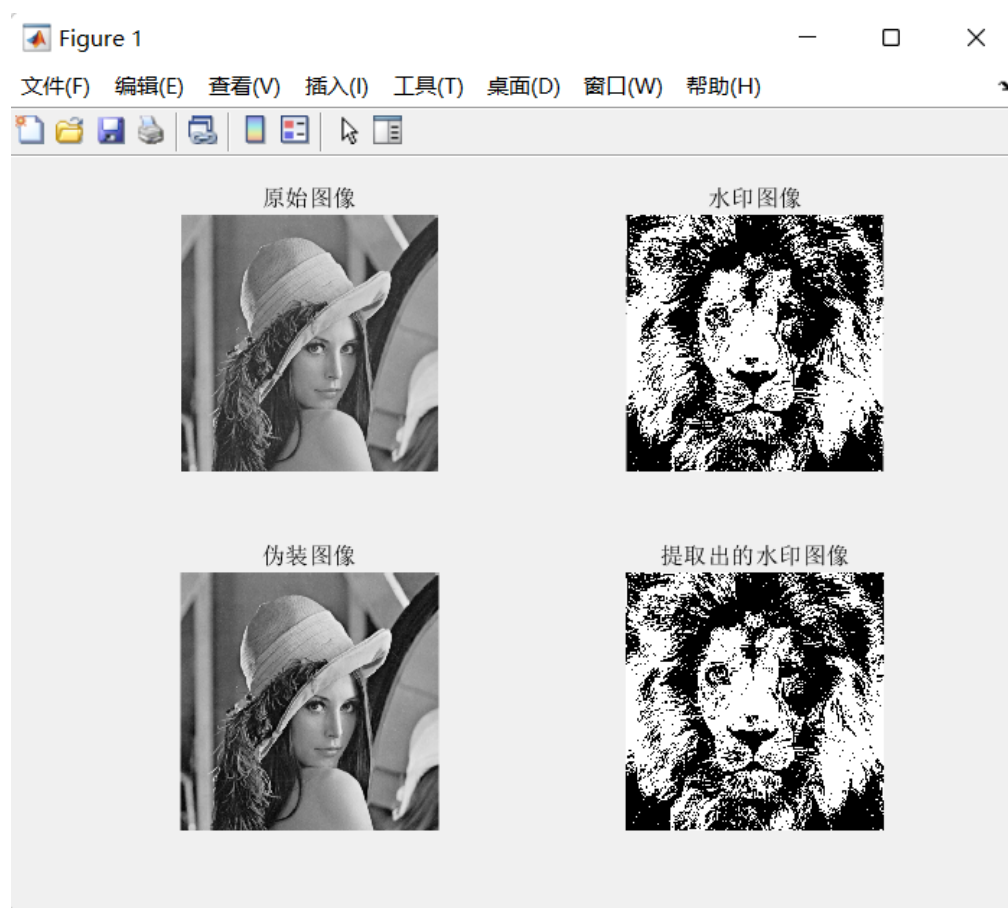


图 1: 实验结果

可见实验取得圆满成功!

五、 实验心得体会

本次实验是关于图像水印的嵌入和提取的实验,通过在载体图像中嵌入二值秘密信息图像,来保护图像的版权和安全。在实验中,我学习了图像水印的基本概念和实现方法,掌握了 MATLAB 中图像处理的基本操作和函数的使用,同时也深刻认识到信息安全的重要性。

首先,我了解了图像水印的概念和分类。图像水印是指在数字图像中嵌入一些特定的信息,以保护图像的版权和安全。根据嵌入的信息类型,图像水印可以分为可见水印和不可见水印。可见水印是指直接嵌入到图像中的信息,可以直接看到,例如公司的商标、版权信息等;不可见水印是指嵌入到图像中的信息是不可见的,需要使用特定的工具才能提取出来,例如数字签名、加密信息等。

其次,我学习了图像水印的嵌入和提取的基本流程和实现方法。图像水印的嵌入过程包括载体图像的读取、秘密信息图像的读取和处理、校验和的计算、比特位的反转和嵌入、以及嵌入后的图像的保存等步骤。图像水印的提取过程包括嵌入后的图像的读取、校验和的计算、秘密信息的提取和保存等步骤。在实现过程中,我掌握了 MATLAB 中图像处理的基本操作和函数的使用,例如 `imread`、`imwrite`、`imshow`、`imbinarize`、`bitset`、`bitget`、`rand`、`switch` 等函数。

最后,我深刻认识到信息安全的重要性。在数字化时代,信息已经成为了一种重要的资源和财富,而信息安全问题也越来越受到人们的关注。图像水印作为一种信息隐藏和保护的技术,可

以有效地保护图像的版权和安全，防止盗版和篡改。在实际应用中，图像水印可以应用于数字图书馆、数字图像库、数字版权管理、数字证书等领域，具有广泛的应用前景。

总之，本次实验让我深入了解了图像水印的概念和实现方法，掌握了 MATLAB 中图像处理的基本操作和函数的使用，同时也深刻认识到信息安全的重要性。通过实验的学习，我不仅提高了自己的实践能力和编程能力，也增强了自己的信息安全意识和保护意识，这对我今后的学习和工作都具有重要的意义。

NIJU