



南開大學
Nankai University

网络空间安全学院
信息隐藏技术实验报告

奇偶校验位隐藏法

姓名：孙璐

学号：2112060

专业：信息安全

2024 年 4 月 22 日

目录

1 实验要求	2
2 实验原理	2
2.1 利用奇偶校验位的方法 1	2
2.1.1 嵌入	2
2.1.2 提取	2
2.2 利用奇偶校验位的方法 2	2
2.2.1 嵌入	3
2.2.2 提取	3
3 实验步骤	3
3.1 HideAndExtract 函数	3
3.2 checksum 函数	4
3.3 Hide 函数	4
3.4 Extract 函数	5
3.5 完整 Matlab 源码	6
3.5.1 Matlab 中的 bitget 函数	8
3.5.2 Matlab 中的 bitset 函数	8
3.6 实验结果展示	9
4 实验心得体会	9

1 实验要求

实验 7: 奇偶校验位隐藏法

内容:

- 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。

2 实验原理

2.1 利用奇偶校验位的方法 1

把载体划分成几个不相重叠的区域, 在一个载体区域中储存 1 bit 的信息

2.1.1 嵌入

选择 $L(m)$ 个不相重叠区域, 计算出每一区域 I 所有最低比特的奇偶校验位 (即“1”的个数奇偶性), $b_i (i = 1, 2, \dots, n)$ 。

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

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 , 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:

下列区域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 $b_i = 1$, 从而满足了 $b = m_i$



2.1.2 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。

2.2 利用奇偶校验位的方法 2

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 1 bit 信息。

2.2.1 嵌入

选择 $L(m)$ 个不相重叠区域, 计算出每一区域 I 所有最低比特的奇偶校验位 (即 “1” 的个数奇偶性), $b_i (i = 1, 2, \dots, n)$ 。

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

区域 I 隐藏一个信息比特, 若 b_i 与 m_i 不同, 那么就将该区域中某个像素的最低比特位进行翻转, 从而使得奇偶校验位与 m_i 相同, 即 $b_i = m_i$ 。

2.2.2 提取

用同样的方法划分载体区域, 计算出奇偶校验位, 构成秘密信息。

例如: 下列区域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是 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 $b_i = 1$, 从而满足了 $b_i = m_i$ 。



3 实验步骤

3.1 HideAndExtract 函数

在函数 HideAndExtract 中, 读取原始图像和秘密信息图像, 并通过 imbinarize() 将秘密信息图像二值化转换成二值图像。创建一个 2x2 的窗口, 第一个子图显示载体图像, 第二个子图显示秘密信息图像。第三个子图显示调用了 Hide() 后嵌入了秘密图像的载体图像, 第四个子图显示调用 Extract() 提取出的秘密信息图像。

Listing 1: HideAndExtract

```

1
2 function HideAndExtract()
3     x=imread ('girl-gray(800).jpg'); %载体图像
4     y=imread ('boy-gray(400).jpg'); %秘密信息图像
      是灰度图像, 长宽均为载体图像的一半
5     y=imbinarize(y);
6     [m, n]= size(y);

```

```

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

```

3.2 checksum 函数

使用 bitget() 从一维向量 x 中提取四个比特位的值，并存储在 temp 向量中。其中 temp(1) 存储 $x(2i-1,2j-1)$ 的最低比特位，temp(2) 存储 $x(2i-1,2j)$ 的最低比特位，temp(3) 存储 $x(2i,2j-1)$ 的最低比特位，temp(4) 存储 $x(2i,2j)$ 的最低比特位。

使用 sum 计算这 4 个最低比特位的和，并使用 rem 函数进行模 2 运算，即为最终的校验和。

Listing 2: checksum

```

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

```

3.3 Hide 函数

使用 for 循环遍历秘密信息图像的每个像素，使用 checksum 计算载体图像对应区域的校验和。如果计算得到的校验和与秘密图像对应像素值不相等，那么需要在载体图像中任意反转一位嵌入秘密信息，这一位通过 rand() 随机生成来决定。

如果 rand() 生成的随机数的值是 0，那么反转 $x(2i-1,2j-1)$ 的最低比特位；如果 rand() 生成的随机数的值是 1，那么反转 $x(2i-1,2j)$ 的最低比特位；如果 rand() 生成的随机数的值是 2，那么反转

$x(2i, 2j-1)$ 的最低比特位；如果 `rand()` 生成的随机数的值是 3，那么反转 $x(2i, 2j)$ 的最低比特位

Listing 3: Hide

```

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

```

3.4 Extract 函数

for 循环遍历秘密矩阵的每个元素，使用 `checksum` 计算灰度图像对应区域的校验和，并存储在 `secret` 矩阵中。将秘密信息矩阵 `secret` 储存在 `out` 中。

Listing 4: Extract

```

1
2 function out=Extract()
3     c=imread('Hide-Image.bmp');
4     [m, n]= size(c);
5     secret = zeros(m/2 , n/2);
6     for i =1:m/2

```

```
7         for j =1: n/2
8             secret(i, j)= checksum(c, i, j);
9         end
10    end
11    out=secret;
12 end
```

3.5 完整 Matlab 源码

载体图像为 boy.bmp, 水印图像为 snowman.bmp。Embedding 将水印图像嵌入到载体图像中, Decode 函数从嵌入后的图像中提取出原始水印图像。

Listing 5: 二值图像嵌入到位图

```
1
2 function HideAndExtract()
3     x=imread ('girl-gray(800).jpg'); %载体 图像
4     y=imread ('boy-gray(400).jpg'); %秘密 信息 图像
5     %是灰度图像, 长宽均为载体图像的一半
6     y=imbinarize(y);
7     [m, n]= size(y);
8
9     subplot(2, 2, 1);
10    imshow(x) ; title('原始 图像');
11
12    subplot(2, 2, 2);
13    imshow(y) ; title('水 印 图 像');
14
15    x=Hide(x,m,n,y);
16    subplot(2, 2, 3);
17    imshow(x ,[]) ; title('伪 装 图 像');
18
19    t=Extract();
20    subplot(2,2,4);
21    imshow(t,[]); title("提取出的水印图像");
22 end
23
24 function out = checksum (x, i, j)
25     %计 算 特 定 一 维 向 量 的 第m个 区 域 的 最 低 位 的 校 验 和
26     temp= zeros(1, 4);
27     temp(1) = bitget(x(2*i-1,2*j-1), 1);
28     temp(2) = bitget(x(2*i-1,2*j), 1);
```

```

29     temp(3) = bitget(x(2*i, 2*j-1), 1);
30     temp(4) = bitget(x(2*i, 2*j ), 1);
31     out=rem(sum(temp), 2);
32 end
33
34 function result=Hide(x,m,n,y)
35     for i =1:m
36         for j =1:n
37             if checksum(x, i, j) ~= y(i, j) %需要反转一位
38                 random= int8(rand()*3);
39                 switch random %任意反转一位
40                     case 0
41                         x(2*i-1,2*j-1)= bitset(x(2*i-1,2*j-1), 1, ~
42                             bitget(x(2*i-1,2*j-1), 1));
43                     case 1
44                         x(2*i-1,2*j)= bitset(x(2*i-1,2*j) , 1 , ~
45                             bitget(x(2*i-1,2*j), 1));
46                     case 2
47                         x(2*i, 2*j-1)= bitset(x(2*i, 2*j-1) ,1 ,~ bitget(x(2*i ,
48                             2*j-1) , 1));
49                     case 3
50                         x(2*i , 2*j)= bitset(x(2*i , 2*j) , 1 , ~ bitget(x(2*i ,
51                             2*j) , 1));
52                     end
53                 end
54             end
55         end
56     end
57     imwrite(x , 'Hide-Image.bmp');
58     result=x;
59 end
60
61 function out=Extract()
62     c=imread('Hide-Image.bmp');
63     [m, n]= size(c);
64     secret = zeros(m/2 , n/2);
65     for i =1:m/2
66         for j =1: n/2
67             secret(i, j)= checksum(c, i, j);
68         end
69     end
70     out=secret;

```


67 | end

3.5.1 Matlab 中的 bitget 函数

- 语法

```
b = bitget(A,bit)
```

```
b = bitget(A,bit,assumedtype)
```

- 说明

`b = bitget(A,bit)` 返回整数数组 `A` 中位于位置 `bit` 的位值。

`b = bitget(A,bit,assumedtype)` 假定 `A` 属于 `assumedtype`。

```
tmp(i,j) = bitget(X(i,j),k);
```

`bitget` 函数首先将 `X(i,j)` 处灰度值分解为二进制串，然后取第 `k` 位

3.5.2 Matlab 中的 bitset 函数

`bitset` 函数的功能是设置指定位置的位。

- 语法

```
intout = bitset(A,bit)
```

```
intout = bitset(A,bit,assumedtype)
```

```
intout = bitset(A,bit,V)
```

```
intout = bitset(A,bit,V,assumedtype)
```

- 说明

`intout = bitset(A,bit)` 返回 `A` 的值，并将位置 `bit` 设置为 1（打开）。

`intout = bitset(A,bit, assumedtype)` 假定 `A` 的类型为 `assumedtype`。

`intout = bitset(A,bit,V)` 返回 `A`，并将位置 `bit` 设置为 `V` 的值。

a) 如果 `V` 为零，则位的位置 `bit` 将设置为 0（关闭）。

b) 如果 `V` 为非零，则位的位置 `bit` 将设置为 1（打开）。

`intout = bitset(A,bit,V, assumedtype)` 假定 `A` 的类型为 `assumedtype`。

3.6 实验结果展示



4 实验心得体会

本次实验通过将二值秘密图像嵌入在载体图像中实现图像水印的嵌入和提取，实现对图像的保护。本次实验中通过对载体图像和秘密信息图像的读取、校验和的计算、比特位的翻转和嵌入等步骤实现了图像水印的嵌入和提取。

本次实验不仅帮助我加深了对数字水印嵌入和提取的理解，还提高了我在 MATLAB 编程方面的实践能力。通过实验，我深刻体会到了理论知识与实践操作的结合对于知识的巩固和应用的重要性，也更加深刻地认识到了信息安全在数字化时代的重要性。