



南開大學  
Nankai University

网络空间安全学院  
信息隐藏技术报告

## 二值图像隐藏法实践

姓名：魏伯繁

学号：2011395

专业：信息安全

2023 年 4 月 20 日

## 目录

<b>1 实验要求</b>	<b>2</b>
<b>2 实验原理</b>	<b>2</b>
<b>3 编程实践</b>	<b>3</b>
3.1 隐藏过程的 pipeline . . . . .	3
3.2 信息提取过程的 pipeline . . . . .	3
3.3 比特位更改函数 . . . . .	4
3.4 黑块个数统计 . . . . .	6
3.5 实验结果 . . . . .	7
<b>4 实验心得</b>	<b>7</b>

## 1 实验要求

1. 隐藏：利用二值图像隐藏法，实现将秘密信息（可以是图像、文字等信息）嵌入到位图中；
2. 提取：将秘密信息提取出来。

## 2 实验原理

二值图像的隐藏法是一种数字水印技术，用于将水印信息嵌入到二值图像中而不影响其可视效果。其实验原理包括以下三个步骤：

1. 嵌入水印信息：首先将待隐藏的水印信息转化为二进制序列，并在二值图像的像素值最低位进行嵌入。这个过程可以通过修改具有嵌入水印信息的像素的最低位，实现对水印信息的嵌入。在这个过程中，需要保证修改操作对于目标图像在视觉和质量上影响最小，从而保证嵌入水印后的图像与原始图像的相似度高。

2. 提取水印信息：通过计算二进制序列的差异，水印信息可以被提取出来。水印信息的提取需要用到一种检测器，它会检测图像中特定位置的像素是否被修改以嵌入水印。如果像素被更改，则表示在该像素处嵌入的水印信息该位置为 1，否则，它为 0。

3. 检测水印信息：将提取到的水印信息与原始水印信息进行比较，根据比较结果来验证图像是否被篡改或者处理过。如果检测到图像被篡改，说明隐写信息已经失效或该图像不存在数字水印，从而确保图像的完整性和版权归属。

总之，二值图像的隐藏法的实验原理是将二值图像的像素最低位用于嵌入水印信息，通过特定的检测器来提取并验证水印信息，从而实现对图像内容的保护和检测。

实验的原理图如下图所示展示的十分清晰：

黑像素个数	0	1	2	3	4
像素分布	全白	1黑3白	两黑两白	3黑1白	全黑
含义	无效块	隐藏“1”	不能出现	隐藏“0”	无效块

图 2.1: 二值图像隐藏法原理

根据上面的实验原理，我们不难总结出在应用该技术时应注意的一些问题：当隐藏文本文档中的字符串时需要注意：嵌入信息的长度不可以过大，不能超过图像大小能负担的度量，为了简化过程，可规定接收者已知秘密信息的长度。

在编程实践中，我们采用下面的分类讨论方法来隐藏/提取秘密信息：

- 当秘密信息为 0，需要将当前区域黑像素点数量调整为 3 个。当黑像素点为 3 时，不需修改。当黑像素点为 1、2、4 时，需要进行修改。对原始的黑像素直接利用，位置不做修改，在嵌入秘密信息时减少对图片的修改。当黑像素点为 0 时，舍弃该预取不做修改，否则在直观上容易被发现
- 当秘密信息为 1，需要将当前区域黑像素点数量调整为 1 个。当黑像素点为 1 时，不需修改。当黑像素点为 0、2、3 时，需要进行修改。对原始的黑像素直接利用，多余的翻转为白像素，在嵌入秘密信息时减少对图片的修改。当黑像素点为 4 时，舍弃该预取不做修改，否则在直观视觉上可能被感受到。
- 提取过程：遍历原图的每个 1x4 的矩形区域嵌入信息的图像的每个区域的黑色像素只有 4 个取值：0, 1, 3, 4 黑像素个数为 1 或 3 时，提取信息 1 或 0。黑像素个数为 0 或 4 时，不提取信息。

## 3 编程实践

### 3.1 隐藏过程的 pipeline

信息异常的过程如下面的代码所示，这段代码在一个二值图像中嵌入一段水印信息。函数输入参数包括原始二值图像  $x$ 、图像的行数  $m$  和列数  $n$ 、以及要嵌入的二值水印图像  $y$ 。

函数中嵌入水印的原理是根据输入的水印图像  $y$  中的像素值 (0 或 1)，来判断需要对原始图像中的某些像素进行修改，从而实现水印信息的嵌入。具体而言，对于每一个像素，将其周围的黑色像素的数量统计出来（通过 `BlackNum` 函数实现），如果要嵌入的水印像素是 0（即黑色），并且该像素周围的黑色像素数量满足一定的条件（即 `tem` 值为 1 或 2 或 4），则将该像素修改为三个连续的黑色像素；如果要嵌入的水印像素是 1（即白色），并且该像素周围的黑色像素数量满足一定的条件（即 `tem` 值为 0 或 2 或 3），则将该像素修改为三个连续的白色像素。

最后，将嵌入了水印信息的图像保存为文件 'watermarkedImage.bmp'，并返回修改后的二值数组 `result`，表示嵌入水印后的图像。

---

```
1  function result=Hide(x,m,n,y)
2      for index_row=1:m
3          for index_col=1:n
4              tem=BlackNum(x,index_row,index_col);
5              if y(index_row,index_col)==0 % 要隐藏黑像素
6                  if tem==1 || tem==2 || tem==4 % 改成三黑，三黑不用管？
7                      x=change_to_0(x,index_row,index_col,tem);
8                  end
9              elseif y(index_row,index_col)==1 % 要隐藏白像素
10                 if tem==0 || tem==2 || tem==3 % 改成三白三白（一黑）不用管？
11                     x=change_to_1(x,index_row,index_col,tem);
12                 end
13             end
14         end
15     end
16     imwrite(x, 'watermarkedImage.bmp');
17     result=x;
18 end
```

---

### 3.2 信息提取过程的 pipeline

下面的代码展示了信息提取过程所需要的的 pipeline，代码从一张嵌入了水印信息的二值图像中提取水印信息。函数没有输入参数，其内部实现的原理是根据嵌入水印信息的规则，从水印图像中提取出原始的二值水印信息。

函数开始的时候，读取了嵌入了水印信息的图像文件 'watermarkedImage.bmp'，并获取了该图像的行数  $m$  和列数  $n$ 。然后，函数新建了一个大小为  $m/2 \times n/2$  的空矩阵 `secret`，用于保存提取出的水印信息。

函数中的提取方法与嵌入方法恰好相反，对于每个嵌入了水印信息的像素，根据其周围的黑色像素数量统计出 tem 值，然后根据 tem 值来确定提取出来的水印像素是 0 还是 1（具体规则见函数代码）。这样遍历完整张图像后，就可以从 secret 矩阵中获取到完整的水印信息。

最后，将提取出来的二值水印信息保存为文件'watermark.bmp'，并返回提取出的水印信息，即矩阵 secret。

---

```

1  function out=Extract()
2      c=imread('watermarkedImage.bmp');
3      [m, n]= size(c);
4      secret = zeros(m/2 , n/2);
5
6      for index_row =1:m/2
7          for index_col =1:n/2
8              tem=BlackNum(c,index_row,index_col);
9              if tem==1
10                 secret(index_row,index_col)=1;
11             elseif tem==3
12                 secret(index_row,index_col)=0;
13             elseif tem==0
14                 secret(index_row,index_col)=0;
15             elseif tem==4
16                 secret(index_row,index_col)=1;
17             end
18         end
19     end
20     imwrite(secret , 'watermark.bmp');
21     out=secret;
22 end

```

---

### 3.3 比特位更改函数

下面以将对应的比特位更改为 1 为例来介绍代码的实现：改代码在一个二值图像中修改一些像素，用于实现水印信息的隐藏。函数的输入参数包括原始二值图像 x、图像的行数 m 和列数 n，以及水印信息的行数 index\_row、列数 index\_col 和状态信息 state

函数中的修改方法是根据水印信息的状态（即 state 值）来决定对原始图像中的某些像素进行怎样的修改。如果将要嵌入的水印像素是 0（即黑色），并且状态值为 0，则函数会在对应像素周围随机添加一个黑色像素；如果状态值为 2，则函数会在对应像素周围随机删除一个黑色像素；如果状态值为 3，则函数会在对应像素周围随机删除两个黑色像素。如果将要嵌入的水印像素是 1（即白色），则状态值为 0、2 或 3 时，函数会对原始图像中对应像素周围的黑色像素进行修改，从而隐藏水印信息。同理也可以编写改写周围白色像素的代码。

最后，函数返回修改后的二值数组 result，表示嵌入水印后的图像。

```
1  function result=change_to_1(x,index_row,index_col,state)
2      % 计算区域内黑色块的数量
3      [~,n]=size(x);
4      ny=n/2;
5      count_in_y=(index_row-1)*ny+index_col;
6      k=count_in_y*4-3; % 对应到 x 中的第几个像素?
7      % 计算水印图中第 i 行 j 列的像素对应的 x 中四个像素的首个像素
8      mx=idivide(int32(k),int32(n),"ceil");
9      my=k-(mx-1)*n;
10     if state==0
11         % 随机增加一个黑
12         randk=int32(rand()*3);
13         x(mx,my+randk)=0;
14     elseif state==2
15         % 随机减去一个黑
16         randk=int8(rand()+1);
17         t=0;
18         for q=0:3
19             if x(mx,my+q)==0
20                 t=t+1;
21                 if t==randk
22                     x(mx,my+q)=1;
23                 end
24             end
25         end
26     elseif state==3
27         % 随机减去两个黑
28         rand1=int8(rand()*2+1);
29         rand2=int8(rand()*2+1);
30         while rand1==rand2
31             rand1=int8(rand()*2+1);
32             rand2=int8(rand()*2+1);
33         end
34         t=0;
35         for q=0:3
36             if x(mx,my+q)==0
37                 t=t+1;
38                 if t==rand1 || t==rand2
39                     x(mx,my+q)=1;
40                 end
41             end
42         end
43     end
44 end
```

```
42         end
43
44     end
45     result=x;
46 end
```

---

### 3.4 黑块个数统计

代码实现了计算嵌入水印信息的二值图像中, 指定区域内黑色像素的数量。函数的输入参数包括原始二值图像  $x$ 、区域的行数  $\text{index\_row}$  和列数  $\text{index\_col}$ 。

函数中的计算方法是, 根据给定的区域行列数, 计算出对应水印图像中的像素位置 (即  $\text{count\_in\_y}$ ); 然后根据这个位置, 计算出该位置对应原始图像中的四个像素位置 (即乘以 4 并减去 3), 从而得到对应的像素行列数 (即  $\text{mx}$  行  $\text{my}$  列)。最后, 对于该位置周围的四个像素逐个判断, 如果是黑色像素, 则计数器  $\text{out}$  加 1。

最终, 函数返回区域内黑色像素的数量  $\text{out}$ 。可以通过这个函数计算出指定区域内的黑色像素数量, 从而实现根据一定的规则嵌入水印信息的目的。

---

```
1  function out = BlackNum (x, index_row, index_col)
2      % 计算区域内黑色块的数量
3      [~,n]=size(x);
4      ny=n/2;
5      count_in_y=(index_row-1)*ny+index_col;
6      k=count_in_y*4-3; % 对应到 x 中的第几个像素?
7      % 计算水印图中第 i 行 j 列的像素对应的 x 中四个像素的首个像素
8      mx=idivide(int32(k),int32(n),"ceil");
9      my=k-(mx-1)*n;
10     out=0;
11     for t=0:3
12         if x(mx,my+t)==0
13             out=out+1;
14         end
15     end
16 end
```

---

### 3.5 实验结果

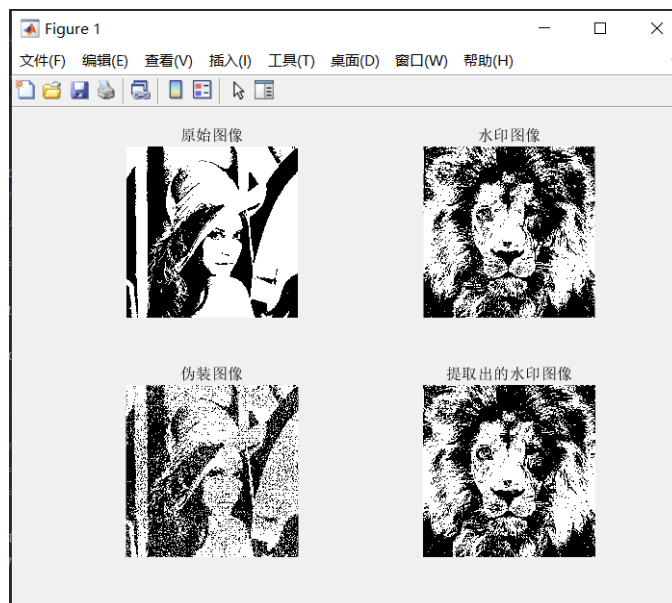


图 3.2: 实验效果

## 4 实验心得

在数字图像处理中，信息隐藏是一项非常有趣的技术，它可以让我们在不影响原始图像质量的情况下，在图像中嵌入不可见的信息。而二值图像信息隐藏则是其中重要的一部分，因为许多图像处理任务都可以通过将图像转换为二值图像来实现。

在进行二值图像信息隐藏的实验中，我首先学习了一些基本的概念和方法，例如二值图像的表示、嵌入水印信息的方式、映射函数的设计等。然后，我按照实验指导书中的要求，对给定的二值图像进行了嵌入水印信息的操作，实现了将指定的信息嵌入到二值图像中的功能。

通过这次实验，我深刻认识到了信息隐藏技术的重要性和应用广泛性。随着数字图像处理技术的发展，越来越多的应用场景需要对图像中嵌入额外的信息，例如数字版权保护、图像认证、信息交流等。在这些场景下，信息隐藏技术可以帮助我们保护图像中的信息，并在需要时将其提取出来，从而提高信息安全性和可信度。

同时，我也意识到了信息隐藏技术的一些局限性和挑战。例如，嵌入的信息必须足够隐蔽，不能被人轻易地发现和篡改；嵌入和提取过程中需要注意信息的准确性和完整性；不同的嵌入和提取算法可能会对图像质量产生不同程度的影响等等。因此，在实际应用中需要综合考虑多个因素，选择合适的嵌入方法和参数，以达到最佳的效果。

总的来说，这次二值图像信息隐藏实验让我受益匪浅，不仅提高了我的理论知识水平，也让我更深入地了解了信息隐藏技术的实际应用和挑战。我相信，通过不断学习和探索，我将能够更好地应用信息隐藏技术，在实际工程和研究中发挥更大的作用。