



南开大学
Nankai University

南 开 大 学

网 络 空 间 安 全 学 院

信息隐藏技术实验八

二值图像隐藏法实验

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 18 日

摘要

利用 MATLAB 对二值图像进行简易算法的信息隐藏处理, 利用位图隐藏一段秘密文本信息。

关键字: 二值图像, 信息隐藏, 隐藏

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) 二值图像隐藏法	1
(二) 简易算法概述	1
(三) 嵌入过程	2
(四) 提取过程	2
三、 实验步骤	2
(一) 嵌入过程	2
1. 实验代码	3
2. 实验结果	6
(二) 提取过程	6
1. 实验代码	6
2. 实验结果	7
四、 实验心得体会	8

一、实验要求

(一) 实验目的

1. 隐藏：利用二值图像隐藏法将秘密信息 (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

(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) 二值图像隐藏法

二值图像: 黑白两种像素组成的图像, 1bit 表示存储像素颜色 (0 或 1)。

通常方法: 利用图像区域中黑色像素个数相对于区域中全部像素个数的百分比来对秘密信息进行编码。

(二) 简易算法概述

将原图划分为 1×4 的矩阵像素块, 每个区域有连续四个像素点。像素点取值情况共有 5 类: 全白, 1 个黑像素点, 2 个黑像素点, 3 个黑像素点和全黑。

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 1: 算法

当隐藏文本文档中的字符串时需要注意:

1. 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

2. 为了简化过程, 可规定接收者已知秘密信息的长度

(三) 嵌入过程

如果要嵌入 0, 需要嵌入后为三黑一白, 对应五种情况的处理: 全白, 不处理视为无效块, 一黑, 两黑, 全黑改为三黑, 三黑不用改。如果要嵌入 1, 需要嵌入后为一黑三白, 对应五种情况的处理: 全白, 两黑, 三黑改为一黑三白, 一黑不用改, 全黑, 视为无效块。



图 2: 嵌入

(四) 提取过程

每四个像素点为一块, 检测其中黑像素点个数。若是三个黑色, 则秘密图像中对应位置为黑色, 若是一个黑色 (即三个白色), 则对应白色。如果是全黑, 说明在隐藏时要隐藏白色, 如果是全白, 说明隐藏时要隐藏黑色。

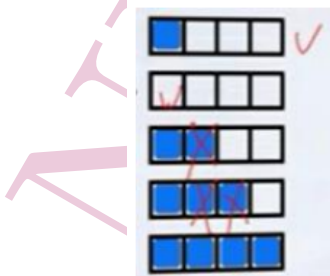


图 3: 提取

三、 实验步骤

(一) 嵌入过程

目标: 在 Lena 位图中嵌入我的学号 2012026

实验步骤

主要有如下三个步骤:

- (1) 给二值图像划定区域, 以 1x4 为一个区域进行位图的划分, 另外注意一下行列的变化
- (2) 统计区域内黑块的个数

(3) 循环嵌入秘密文本信息：分情况讨论，注意情况分类需要全面否则极易提取出错

1. 实验代码

首先是工具函数 CalculateBlack，计算 1*4 像素块中黑色像素的数量。

```
1 %% CalculateBlack
2 function out = CalculateBlack(x, i)
3 out = 0;
4 for a = i : i + 3
5     if x(a) == 0
6         out = out + 1;
7     end
8 end
9 end
```

函数的作用是计算 x 中从第 i 个元素开始的连续 4 个元素中，值为 0 的元素个数，并将结果返回。该函数以四格为单位遍历原始图像然后输出黑色像素的数量。

随后是嵌入秘密信息的函数 InsertSecret。

```
1 d = imread('./pic.bmp');
2 d = im2bw(d);
3 imwrite(d, 'black1.bmp', 'bmp')
4 subplot(1, 2, 1); imshow(d, []); title('原始图片');
5 secret = 2012026;
```

该函数读取图片并转换为二值图像，然后就是将秘密信息（我的学号 2012026）进行一个加密。

然后，我们使用 `bitget()` 函数，将我的学号进行二进制提取。

```
1 for t = 1:24
2     s(t) = bitget(secret, t);
3 end
```

首先观察我的学号 2012026 转换为二进制后的结果



图 4: 二进制的 2012026

可以看到，至少需要 24 位，因此 for 循环执行 24 次。
然后，看我们用 bitget() 得到的结果

A screenshot of a MATLAB array named '1x24 double'. The array contains the following binary sequence: 1 0 1 0 1 1 1 1 0 1 1 0 0 1 1 0 1 0 1 1 1 1 0 0 0. The sequence is displayed in a table with columns numbered 1 to 24.

	1	2	3	4	5	6	7	8	9	24
1	0	1	0	1	1	1	1	0	1	1	0	0	1	1	0	1	0	1	1	1	1	0	0	0

图 5: 提取

和上面计算器得到的结果相同。**这就是我们要隐藏的信息！**

然后是进行秘密的嵌入操作。

在这一步，我们首先需要判断嵌入的秘密信息是 0 还是 1，如果是 0 的话就需要三个黑色像素点和一个白色像素点。这一步可以通过实现定义好的 CalculateBlack 函数进行获取。如果输出了 0，那么我们就知道这个信息块是白色，应该是无效块。此时应当访问下一个矩阵。如果是 1 的话那么就将其中的两个白色像素点修改为黑色。依次类推，保证这个像素块中由三个黑色的像素点。同理，如果是嵌入信息是 1，则需要将其修改为三白一黑，原理与上一种情况类似。给出部分代码如下

```
1 num = 1;
2 t = 1;
3 while t < 24
4     if s(t) == 0
5         switch (CalculateBlack(d, num))
6             case 0
7                 t = t - 1;
8                 num = num + 4;
9             case 1
10                 temp = 1;
11                 startnum = num;
```

```
12         while temp < 3
13             if d(startnum) == 1
14                 d(startnum) = 0;
15                 temp = temp + 1;
16                 startnum = startnum + 1;
17             end
18         end
19         num = num + 4;
20         ..... 其他同理
21     end
22 else
23     a = CalculateBlack(d, num)
24     switch a
25         case 0
26             temp = 4;
27             startnum = num;
28             while temp > 3
29                 if d(startnum) == 1
30                     d(startnum) = 0;
31                     temp = temp - 1;
32                     startnum = startnum + 1;
33                 end
34             end
35             num = num + 4;
36         case 1
37             num = num + 4;
38         case 2
39             temp = 2;
40             startnum = num;
41             while temp < 3
42                 if d(startnum) == 0
43                     d(startnum) = 1;
44                     temp = temp + 1;
45                     startnum = startnum + 1;
46                 end
47             end
48             num = num + 4;
49             ..... 其他同理
50         end
51     end
52     t = t + 1;
53 end
```

注意，这里假设前提已经知道了图像大小可以承载信息量和秘密信息的长度。最后，分别保存原图和加了水印的图。

```
1  
2 imwrite(d, 'black2.bmp', 'bmp')  
3  
4 subplot (1, 2, 2); imshow (d, []); title (' 水印 ');
```

2. 实验结果

最终，观察嵌入秘密信息的二值图像如下：

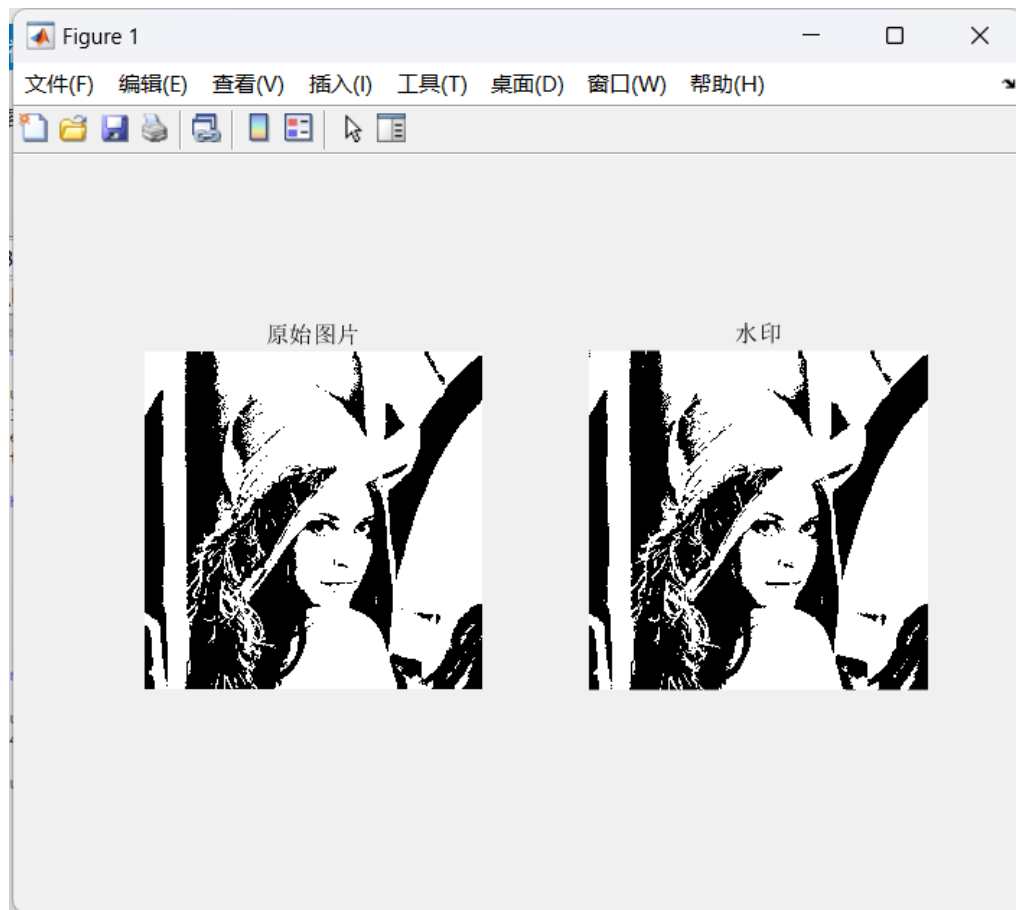


图 6: 隐藏后

由上图对比可见，位图加入了一段文本信息，但从感官上与原图没有差别。**可见嵌入实验取得圆满成功！**

(二) 提取过程

1. 实验代码

读取载入信息后的伪装图像数据矩阵，以同嵌入过程相同的遍历方式获得区域内黑色像素的个数，统计区域内黑像素的个数是 1, 3 还是 0, 4，从而恢复出秘密信息 0 或 1。


```
1 while t < 24
2     a = CalculateBlack(d, num);
3     switch a
4         case 0
5             num = num + 4;
6         case 1
7             s(t) = 1;
8             t = t + 1;
9             num = num + 4;
10        case 3
11            s(t) = 0;
12            t = t + 1;
13            num = num + 4;
14        case 4
15            num = num + 4;
16    end
17 end
```

对秘密信息进行提取需要事先知道信息长度，这里可以通过将信息编码到图片首部的冗余部分进行。为演示方便起见可以假设长度是事先知道的。首先读取图片，初始化存放密文的矩阵。然后进行提取操作，如果我们已知长度为 8，所以可以将遍历的步长设置为 8，然后计算其中黑色像素的数量，为 0 或 4 时代表为无效块，黑色为 1 时代表为 1，黑色为 3 时代表为 0。然后将 2 进制转换为 10 进制即可。

另一段代码则是把二进制的数转为原始十进制内容，如下所示：

```
1 sum = 0;
2
3 for t = 1:24
4     sum = sum + s(t) * 2^(t - 1);
5 end
```

2. 实验结果

最后，恢复出来的信息如下所示：



图 7: 提取结果

可见提取实验取得圆满成功!

四、 实验心得体会

二值图像隐藏法是一种常见的信息隐藏技术，它利用黑白像素包含的区域冗余度进行隐藏，将秘密图像或文本转化为二进制码流，嵌入到载体图像的像素中，从而实现信息的隐蔽传输。本实验旨在通过实践掌握二值图像隐藏法的基本原理和实现方法，加深对于数据存储方式的理解，并注意控制隐藏对象的大小和容量与秘密信息的匹配，以及图像行列的动态变化。

实验过程中，我们首先需要准备一张载体图像和一段秘密信息。载体图像应该是一张二值图像，可以使用 MATLAB 中的 `im2bw` 函数将彩色图像转化为二值图像。秘密信息可以是一段文本或一张二值图像，需要将其转化为二进制码流。在转化过程中，需要注意数据存储的方式，如大端小端方式等，以确保发送者和接收者使用相同的方式进行数据存储和解析。

接下来，我们需要将秘密信息嵌入到载体图像的像素中。具体方法是将秘密信息的二进制码流按照一定的规则嵌入到载体图像的像素中。常见的嵌入规则包括 LSB（最低有效位）嵌入法、PVD（像素值差分）嵌入法等。LSB 嵌入法是将秘密信息的二进制码流按照顺序依次嵌入到载体图像的像素的最低有效位中，从而实现信息的隐蔽传输。PVD 嵌入法是将秘密信息的二进制码流按照顺序依次嵌入到载体图像的像素的像素值差分中，从而实现信息的隐蔽传输。在嵌入过程中，需要注意控制隐藏对象的大小和容量与秘密信息的匹配，以及图像行列的动态变化，避免信息的丢失和损坏。

最后，我们需要将隐藏的秘密信息从载体图像中提取出来。具体方法是按照嵌入规则逆向操作，将载体图像中隐藏的秘密信息提取出来，并将其转化为原始的二进制码流，再根据数据存储的方式进行解析，得到原始的秘密信息。在提取过程中，需要注意发送者和接收者提前进行提取和隐藏方法的沟通，以确保信息的正确提取和解析。

总的来说，二值图像隐藏法是一种简单有效的信息隐藏技术，可以用于保护敏感信息的隐私和安全。在实践中，需要注意控制隐藏对象的大小和容量与秘密信息的匹配，以及图像行列的动态变化，避免信息的丢失和损坏。