

信息隐藏技术第五次实验 —— 二值图像隐藏法

学号：2111033 姓名：艾明旭 专业：信息安全

一、实验要求

1.实验目的

- 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
- 提取：将秘密信息提取出来

2.实验环境

- 运行系统：Windows11
- 实验工具：Matlab2023a
- 数据：PNG 格式图像

3.要求

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

二、二值图像信息隐藏

1. 二值图像

二值图像是由黑白像素分布构成图像

2. 通常方法

处理二值图像的一般方法是利用图像区域中黑色像素的个数对秘密信息进行编码

三、隐藏思路

1. 思路一

简易算法概述

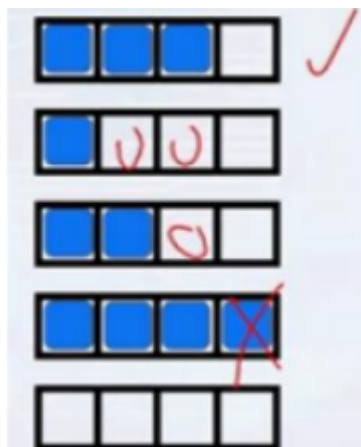
将原图划分为 1×4 的矩阵像素块，每个区域有连续四个像素点。像素点取值情况共有 5 类：

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

(1) 嵌入

- 需要将一个二值图像分成 $L(m)$ 个矩形图像区域 B_i ，若其中黑色像素的个数大于 $1/2$ ，则表示嵌入 0；如果白色像素的个数大于一半，则表示嵌入 1
- 当需要嵌入的比特与所选区域的黑白像素的比例不一致时，为了达到希望的像素关系，则改一些像素颜色
- 遵循一定的规则，并不引起感观察觉
- 应在黑白区域边缘进行



(2) 需注意的细节

- 考虑有一定的冗余度
- 确定两个阈值 $R1 > 50\%$ 和 $R0 < 50\%$ ，一个健壮性参数 λ 。隐藏 0 时，该块的黑色像素的个数应属于 $[R1, R1 + \lambda]$ 隐藏 1 时，该块的黑色像素的个数应属于 $[R0 - \lambda, R0]$
- 若为了适应所嵌入的比特，目标块必须修改太多的像素，就将该块设为无效
- 将无效块中的像素进行少量的修改，使得其中黑色像素的百分比大于 $R1 + 3\lambda$ ，或者小于 $R0 - 3\lambda$

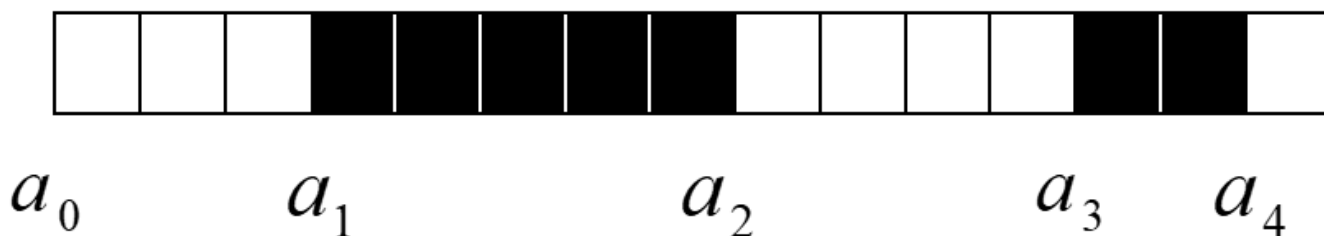
(3)提取

- 判断每一个图像块黑色像素的百分比，如果大于 $R1 + 3\lambda$ ，或者小于 $R0 - 3\lambda$ ，则跳过类似这样的无效块
- 若在 $[R1, R1 + \lambda]$ 或者 $[R0 - \lambda, R0]$ 的范围内，则正确提取出秘密信息 0 或 1



2. 思路二

使用游程编码的方法，如下图所示：



编码： $\langle a_0, 3 \rangle \langle a_1, 5 \rangle \langle a_2, 4 \rangle \langle a_3, 2 \rangle \langle a_4, 1 \rangle$

(1)嵌入

- 修改二值图像的游程长度
- 若秘密信息位是 0，则修改该游程长度为偶数
- 若为 1，则修改游程长度为奇数
- 若秘密信息的取值与游程长度的奇偶性相匹配，则不改变游程长度

(2)提取

根据游程长度奇偶性提取出秘密信息

四、实验代码

使用 MATLAB 实现了第二种方法，即 2.2. 中的利用游程编码的方法，用 decode.m 实现秘密信息嵌入，用 encode.m 实现秘密信息提取，嵌入时用到了 str2bit 和 Vector2Matrix 两个函数做待隐藏信息的编码处理，提取时用到了 bit2str 函数解码比特串为正常字符串

1. encode.m

```

%% encode
clc;
clear;
d = imread('./pic.bmp');
d = im2bw(d);
imwrite(d, 'black1.bmp', 'bmp')
subplot (1, 2, 1); imshow (d, []); title (' 原始图片 ');
secret = 2111033;
for t = 1:24
    s(t) = bitget(secret, t);
end
num = 1;
t = 1;
while t < 24
    if s(t) == 0
        switch (CalculateBlack(d, num))
            case 0
                t = t - 1;
                num = num + 4;
            case 1
                temp = 1;
                startnum = num;
                while temp < 3
                    if d(startnum) == 1
                        d(startnum) = 0;
                        temp = temp + 1;
                        startnum = startnum + 1;
                    end
                end
                num = num + 4;
            case 2
                temp = 2;
                startnum = num;
                while temp < 3
                    if d(startnum) == 1
                        d(startnum) = 0;
                        temp = temp + 1;
                        startnum = startnum + 1;
                    end
                end
                num = num + 4;
            case 3
                num = num + 4;
            case 4
                temp = 4;
                startnum = num;
                while temp > 3
                    if d(startnum) == 0
                        d(startnum) = 1;
                        temp = temp - 1;
                    end
                end
            end
        end
        t = t + 1;
    end
end

```

```

        startnum = startnum + 1;
    end
end
num = num + 4;
end
else
    a = CalculateBlack(d, num)
    switch a
        case 0
            temp = 4;
            startnum = num;

            while temp > 3

                if d(startnum) == 1
                    d(startnum) = 0;
                    temp = temp - 1;
                    startnum = startnum + 1;
                end

            end

            num = num + 4;

        case 1
            num = num + 4;
        case 2
            temp = 2;
            startnum = num;

            while temp < 3

                if d(startnum) == 0
                    d(startnum) = 1;
                    temp = temp + 1;
                    startnum = startnum + 1;
                end
            end
            num = num + 4;
        case 3
            temp = 1;
            startnum = num;
            while temp < 3

                if d(startnum) == 0
                    d(startnum) = 1;
                    temp = temp + 1;
                    startnum = startnum + 1;
                end
            end
            num = num + 4;
        end
    end
end

```

```

        case 4
            t = t - 1;
            num = num + 4;
        end
    end
    t = t + 1;
end
imwrite(d, 'black2.bmp', 'bmp')
subplot (1, 2, 2); imshow (d, []); title (' 水印 ');

```

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 2111033）进行一个加密。我的学号展开为二进制为24位，所以至少需要 24 位，因此 for 循环执行 24 次。

然后，我们用 bitget() 得到的结果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，则我们需要将其修改为三白一黑，原理与上一种情况类似。

2. decode.m

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。

```

%% decode
clc;
clear;
d = imread('black2.bmp');

for t = 1:24
    s(t) = bitget(0, t);
end

t = 1;
num = 1;

while t < 24
    a = CalculateBlack(d, num);
    switch a
        case 0
            num = num + 4;
        case 1
            s(t) = 1;
            t = t + 1;
            num = num + 4;
        case 3
            s(t) = 0;
            t = t + 1;
            num = num + 4;
        case 4
            num = num + 4;
    end
end

sum = 0;

for t = 1:24
    sum = sum + s(t) * 2^(t - 1);
end

fprintf("秘密信息是:  %d\n", sum);

```

提取与隐藏类似，但只需要对隐藏信息的载体图片的游程进行统计以获得相应的二进制比特串，然后再解码为正常的字符串

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。

3. 计算色素数量

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

```
%% CalculateBlack
function out = CalculateBlack(x, i)
out = 0;
for a = i : i + 3
    if x(a) == 0
        out = out + 1;
    end
end
end
```

五、实验结果

1. 结果输出

原始图片



水印



2. 待隐藏秘密信息

隐藏信息.txt

2111033

3. 从图片提取的秘密信息

秘密信息是： 2111033

>>

六、总结与展望

本次实验利用 MATLAB 和对将秘密图像或文本转化为二进制码流，进一步利用黑白像素包含的区域冗余度进行隐藏，注意控制隐藏对象的大小和容量与秘密信息的匹配，注意图像行列的动态变化，需要发送者和接收者提前进行提取和隐藏方法的沟通。

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。

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。

通过对所学到的理论知识进行相应的应用，对 MATLAB 的应用也更加的熟练，最后期待自己未来更好的发展，在该门课程的学习过程当中更上一层楼。