

# 信息隐藏技术实验报告

二值图像隐藏法  
2113662      张丛

实验内容  
二值图像隐藏法  
原理  
实现  
实验结果  
小结

## 实验内容

- 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。

## 二值图像隐藏法

### 原理

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

黑色像素个数多于一半：嵌入'0'。

黑色像素个数少于一半：嵌入'1'。

无效块：如果为了适应所嵌入的比特，目标块需要修改的像素太多，设为无效。无效块需要跳过。

简易算法如下：

## 5

## 简易算法

将原图划分为 $1 \times 4$ 的矩形像素块，每个区域有四个连续的像素点。这些像素点的取值情况可以分为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

我们要隐藏的信息为文本文档中的字符串，需要注意的是：

- 1 嵌入的信息长度不能过大，不能超过图片大小所能负担的度量；
- 2 为了简化过程，可以规定接收者已知秘密信息的长度。

## 6

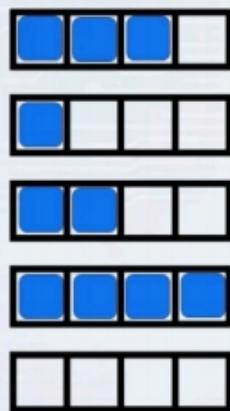
## 简易算法的嵌入过程(1)

嵌入过程

遍历原图中的每个 $1 \times 4$ 矩形区域。

(1) 如果我们要嵌入的信息为0，则需要将当前区域的黑像素点数量调整到3个。

- 如果当前区域黑像素点数量正好为3个，则不需要修改；
- 如果当前区域黑像素点数量为1或2或4个，则需进行修改，使其黑色像素点数量变为3个，同时要注意的是，对原有黑色像素直接利用，位置不做修改，为的是嵌入秘密信息的过程中，对图片的修改尽量少。
- 如果原区域全白，则舍弃这一块不做修改，否则变化可能会被直观视觉所感受到。

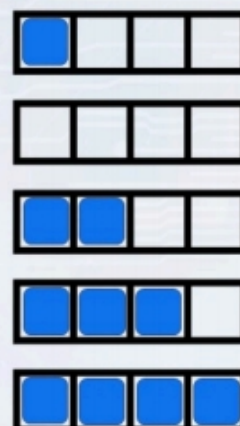


## 7

### 简易算法的嵌入过程(2)

(2) 如果我们要嵌入的信息为1，则需要将当前区域的黑像素点数量调整到1个。

- 如果当前区域黑像素点数量正好为1个，则不需要修改；
- 如果当前区域黑像素点数量为0或2或3个，则需进行修改，使其黑色像素点数量变为1个，同时要注意的是，对原有黑色像素直接利用，多余的翻转为白像素，为的是嵌入秘密信息的过程中，对图片的修改尽量少。
- 如果原区域全黑，则舍弃这一块不做修改，否则变化可能会被直观视觉所感受到。



## 8

### 简易算法的提取过程

嵌入过信息的图像中每个区域的黑色像素点个数只有4个取值：

0

1

3

4

#### 提取过程

遍历携密图像的每个 $1 \times 4$ 区域，如果黑色像素点个数为1或3则提取信息，1个黑像素点对应‘1’，3个黑像素点对应‘0’，黑色像素为0或4为未嵌入信息的区域。

## 实现

嵌入：

```
% 读取原始二值图像
d = imread('fox_binary_image.jpg');
subplot(1, 2, 1); imshow(d, []); title('原始图片');

% 要隐藏的秘密数字
secret = 2113662;

% 将秘密数字转换为二进制数组
s = dec2bin(secret, 24) - '0';
```

```

num = 1;
t = 1;

while t < 24
    if s(t) == 0
        % 当前位为0时的处理逻辑
        switch calculateBlack(d, num)
            case 0
                % 如果周围黑色像素数量为0，回退一位重新处理
                t = t - 1;
                num = num + 4;
            case {1, 2}
                % 如果周围黑色像素数量为1或2，选择合适的点插入数据
                temp = s(t) + 1;
                startnum = num;

                while temp < 3 && d(startnum) == s(t)
                    % 在连续的黑色像素中插入数据
                    d(startnum) = ~d(startnum);
                    temp = temp + 1;
                    startnum = startnum + 1;
                end

                num = num + 4;
            case 3
                % 如果周围黑色像素数量为3，直接跳过
                num = num + 4;
            case 4
                % 如果周围黑色像素数量为4，找到适合的点插入数据
                temp = 4;
                startnum = num;

                while temp > 3 && d(startnum) == s(t)
                    % 在连续的白色像素中插入数据
                    d(startnum) = ~d(startnum);
                    temp = temp - 1;
                    startnum = startnum + 1;
                end

                num = num + 4;
            end
        end
    else
        % 当前位为1时的处理逻辑
        a = calculateBlack(d, num);

        switch a
            case 0
                % 如果周围黑色像素数量为0，找到适合的点插入数据
                temp = 4;
                startnum = num;

                while temp > 3 && d(startnum) == s(t)
                    % 在连续的白色像素中插入数据
                    d(startnum) = ~d(startnum);
                    temp = temp - 1;
                    startnum = startnum + 1;
                end
            end
        end
    end
end

```

```

        num = num + 4;

    case {1, 2}
        % 如果周围黑色像素数量为1或2，直接跳过
        num = num + 4;
    case 3
        % 如果周围黑色像素数量为3，找到适合的点插入数据
        temp = 1;
        startnum = num;

        while temp < 3 && d(startnum) == s(t)
            % 在连续的白色像素中插入数据
            d(startnum) = ~d(startnum);
            temp = temp + 1;
            startnum = startnum + 1;
        end

        num = num + 4;
    case 4
        % 如果周围黑色像素数量为4，回退一位重新处理
        t = t - 1;
        num = num + 4;
    end

end

end
%disp(t);
t = t + 1;
end

% 保存带水印的图像
imwrite(d, 'scret.bmp', 'bmp')

subplot(1, 2, 2); imshow(d, []); title('水印');

```

流程如下：

1. 读取原始二值图像并显示在第一个子图中。
2. 将要隐藏的秘密数字2113662转换为24位的二进制数组。
3. 遍历二进制数组中的每一位，根据当前位的值（0或1）以及周围黑色像素的数量，选择合适的位置在二值图像中插入数据（黑色像素值取反），以隐藏秘密数字。
4. 将带有隐藏信息的图像保存为新的图像文件，并在第二个子图中显示。

**提取：**

```

% 读取带水印的二值图像
d = imread('scret.bmp');

% 初始化秘密信息数组
s = zeros(1, 24);

% 解析水印信息
t = 1;

```



```

num = 1;

while t < 24
    % 获取当前位置周围的黑色像素数量
    a = calculateBlack(d, num);

    % 根据黑色像素数量确定水印位的值
    switch a
        case 0
            % 周围没有黑色像素，当前水印位保持不变
            num = num + 4;
        case 1
            % 周围有一个黑色像素，水印位为1
            s(t) = 1;
            t = t + 1;
            num = num + 4;
        case 3
            % 周围有三个黑色像素，水印位为0
            s(t) = 0;
            t = t + 1;
            num = num + 4;
        case 4
            % 周围有四个黑色像素，跳过当前位
            num = num + 4;
    end
    disp(t);
end

% 将二进制数组转换为十进制数，恢复出原始的秘密信息
sum = 0;
for t = 1:24
    sum = sum + s(t) * 2^(t - 1);
end

% 打印出解码得到的秘密信息
fprintf("秘密信息是: %d\n", sum);

```

流程如下：

1. 读取带水印的二值图像。
2. 初始化一个长度为24的秘密信息数组。
3. 遍历二值图像中的每个像素，根据周围黑色像素的数量确定水印位的值，并将其存储在秘密信息数组中。
4. 将秘密信息数组转换为十进制数，恢复出原始的秘密信息。

## 实验结果



可见，完成了秘密信息的嵌入和提取。

## 小结

对课堂上的二值图像隐藏法的理论知识进行了实践，对信息隐藏技术有了进一步的理解。

实验实践的二值图像隐藏法的简易算法非常高明，即符合基本的规则，也能将无效块进行快速区分，需要修改时也很明确。