

信息隐藏技术实验报告

奇偶校验位隐藏法

2113662 张丛

实验内容

奇偶校验位隐藏法

原理

实现

实验结果

小结

实验内容

- 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。

奇偶校验位隐藏法

原理

方法一：



利用奇偶校验位的方法一

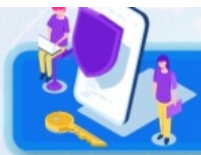
把载体划分成几个不相重叠的区域，在一个载体区域中存储一比特信息。

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

$$b_i (i=1, 2, \dots, n)。 \quad b_i = \sum_{j \in I} LSB(c_j) \bmod 2$$

嵌入

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 $b_i = m_i$ 。



利用奇偶校验位的方法一

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。

方法二：



利用奇偶校验位的方法二

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一比特信息。

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

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

嵌入

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 $b_i = m_i$ 。

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。

实现

代码如下：

```
function HideAndExtract()
    x = imread('nilu_gray_image.jpg'); % 载体图像
    y = imread('fox_binary_image.jpg'); % 秘密信息图像

    % 裁剪图像为相同尺寸
    [Mc, Nc, ~] = size(x);
    [Mm, Nm, ~] = size(y);
    min_rows = min(Mc, Mm);
    min_cols = min(Nc, Nm);
    start_row = floor((Mc - min_rows) / 2) + 1; % 计算裁剪后图像的起始行
```

```

start_col = floor((Nc - min_cols) / 2) + 1; % 计算裁剪后图像的起始列
x = x(start_row:start_row+min_rows-1, start_col:start_col+min_cols-1); % 裁剪载体
图像
y = y(1:min_rows, 1:min_cols); % 裁剪秘密信息图像

y = imbinarize(y); % 二值化秘密信息图像
[m, n] = size(y); % 获取秘密信息图像的尺寸

subplot(2, 2, 1);
imshow(x);
title('原始图像');

subplot(2, 2, 2);
imshow(y);
title('水印图像');

x = Hide(x, m, n, y);
subplot(2, 2, 3);
imshow(x, []);
title('伪装图像');

t = Extract();
subplot(2, 2, 4);
imshow(t, []);
title('提取出的水印图像');
end

% 计算特定一维向量的第m个区域的最低位的校验和
function out = checksum(x, i, j)
[rows, cols] = size(x);
indices = [2*i-1, 2*j-1, 2*i-1, 2*j, 2*i, 2*j-1, 2*i, 2*j];
temp = zeros(1, 4);
for k = 1:4
    row = indices(2*k-1);
    col = indices(2*k);
    if row >= 1 && row <= rows && col >= 1 && col <= cols
        temp(k) = bitget(x(row, col), 1); % 获取指定位置的最低位
    end
end
out = rem(sum(temp), 2); % 计算校验和
end

function result = Hide(x, m, n, y)
[rows, cols] = size(x);
for i = 1:m
    for j = 1:n
        row = 2*i-1;
        col = 2*j-1;
        if row >= 1 && row <= rows && col >= 1 && col <= cols && checksum(x, i,
j) ~= y(i, j) % 如果校验和不匹配
            random = int8(rand() * 3);
            switch random % 随机选择一个位置进行位反转
                case 0
                    x(row, col) = bitset(x(row, col), 1, ~bitget(x(row, col),
1));
                case 1
                    x(row, col+1) = bitset(x(row, col+1), 1, ~bitget(x(row,
col+1), 1));
            end
        end
    end
end

```

```

        case 2
            x(row+1, col) = bitset(x(row+1, col), 1, ~bitget(x(row+1,
col), 1));
        case 3
            x(row+1, col+1) = bitset(x(row+1, col+1), 1, ~bitget(x(row+1,
col+1), 1));
        end
    end
end
end
imwrite(x, 'watermarkedImage.bmp'); % 保存水印图像
result = x;
end

function out = Extract()
    c = imread('watermarkedImage.bmp');
    [m, n] = size(c);
    secret = zeros(m/2, n/2);
    for i = 1:m/2
        for j = 1:n/2
            secret(i, j) = checksum(c, i, j); % 提取水印信息
        end
    end
    out = secret;
end

```

整体流程如下：

1. HideAndExtract函数加载载体图像和水印图像，并调用Hide函数隐藏水印图像，然后调用Extract函数提取隐藏的水印图像。
2. Hide函数接受载体图像x、水印图像的行数m、列数n和水印图像y作为输入。该函数通过遍历水印图像的每个像素，并与载体图像的对应区域进行比较，如果需要反转一位，则随机选择一个像素点的最低位进行反转，然后将反转后的图像保存为新图像。最后，返回新图像。
3. Extract函数用于提取隐藏在伪装图像中的水印。该函数读取伪装图像，然后遍历伪装图像的每个像素，并根据最低位的校验和恢复水印图像。最后，返回提取出的水印图像。

实验结果



小结

此次信息隐藏技术实验使用了奇偶校验位隐藏法，将秘密图像嵌入到载体图像中，并成功提取出了隐藏的水印图像。

通过实验，我深入了解了奇偶校验位隐藏法的原理和实现方法，以及如何利用 MATLAB 实现该技术，对信息隐藏技术有了更深入的理解。