

信息隐藏技术实验报告

LSB隐藏法

2113662 张丛

实验内容
二值图像
原理
实现
实验结果
学号
原理
实现
实验结果
小结

实验内容

- 1、实现将二值图像嵌入到位图中；
- 2、实现将学号（一个整数）嵌入到位图中。

二值图像

原理

LSB(LeastSignificantBit) 方法是用秘密信息 (比特) 替换掉最低有效位的数据。具体来说, LSB 算法是一种隐写术, 用于将秘密信息嵌入到数字图像的最低有效位中, 以实现隐蔽传输。

该算法的基本思想是将秘密信息的二进制码嵌入到数字图像的最低有效位中, 因为最低有效位的变化对图像的视觉效果影响最小, 所以不容易被察觉。

需要注意的是, LSB 算法虽然隐蔽性较好, 但也容易被攻击者检测到。因此, 在实际应用中, 需要结合其他隐写术和加密算法, 以提高信息的安全性。

实现

```
function ImageHiding()  
    % 读取载体图像  
    x = imread("nilu_gray_image.jpg");  
    % 读取水印图像  
    m = imread("fox_binary_image.jpg");  
  
    % 裁剪图像为相同尺寸  
    [Mc, Nc, ~] = size(x);  
    [Mm, Nm, ~] = size(m);
```

```

min_rows = min(Mc, Mm);
min_cols = min(Nc, Nm);
x = x(1:min_rows, 1:min_cols);
m = m(1:min_rows, 1:min_cols);

% 显示载体图像
figure;
imshow(x, []);
title("Original Image");

% 显示水印图像
figure;
imshow(m, []);
title("watermark Image");

% 添加水印
waterMarked = Hide(x, m);

% 提取水印
watermark = Extract(waterMarked);
end

function waterMarked = Hide(origin, watermark)
% 获取原始图像尺寸
[Mc, Nc, ~] = size(origin);
% 创建用于存储水印图像的数组
waterMarked = uint8(zeros(Mc, Nc));

% 将水印嵌入到原始图像中
for i = 1:Mc
    for j = 1:Nc
        waterMarked(i, j) = bitset(origin(i, j), 1, watermark(i, j));
    end
end

% 将嵌入水印后的图像保存为文件
imwrite(waterMarked, 'lsb_watermarked.bmp', 'bmp');

% 显示嵌入水印后的图像
figure;
imshow(waterMarked, []);
title("watermarked Image");
end

function waterMark = Extract(waterMarked)
% 获取嵌入水印后的图像尺寸
[Mw, Nw, ~] = size(waterMarked);
% 创建用于存储提取水印后的数组
waterMark = uint8(zeros(Mw, Nw));

% 提取水印
for i = 1:Mw
    for j = 1:Nw
        waterMark(i, j) = bitget(waterMarked(i, j), 1);
    end
end

% 显示提取的水印图像

```

```
figure;  
imshow(waterMark, []);  
title("Extracted watermark");  
end
```

实验结果

原始图像：



Original Image



水印图像:



嵌入后的图像：



Watermarked Image



提取的图像：



Extracted Watermark



学号

原理

选择一幅图像作为载体图像，然后选取图像的第一行像素作为隐藏信息的嵌入位置。

具体来说，我们将一个整数2113662的每一位二进制表示，分别嵌入到图像第一行的每个像素的最低有效位中。嵌入完成后，我们得到了一个带有隐藏信息的水印图像。

实现

```
function IntHiding()
    % 读取载体图像
    x = imread('fox_gray_image.jpg');
    % 要嵌入的信息
    m = 2012026;

    % 显示原始图像
    figure;
    imshow(x, []);
    title('Original Image');

    % 嵌入水印并显示水印图像
    waterMarked = Hide(x, m);

    % 提取水印
    watermark = Extract(waterMarked);
end

function waterMarked = Hide(origin, watermark)
    % 获取图像尺寸
    [Mc, Nc] = size(origin);
    % 创建与原始图像大小相同的零矩阵
    waterMarked = uint8(zeros(size(origin)));

    % 遍历图像像素
    for i = 1:Mc
        for j = 1:Nc
            % 如果是第一行的像素，并且在21列之内
            if i == 1 && j <= 21
                % 获取要嵌入的信息中对应位置的比特值
                tem = bitget(watermark, j);
                % 将信息嵌入到原始图像的最低比特位
                waterMarked(i, j) = bitset(origin(i, j), 1, tem);
            else
                % 其他情况保持原始像素值不变
                waterMarked(i, j) = origin(i, j);
            end
        end
    end

    % 保存水印图像并显示
    imwrite(waterMarked, 'lsb_int_watermarked.bmp', 'bmp');
    figure;
    imshow(waterMarked, []);
    title('watermarked Image');
```



```
end
```

```
function waterMark = Extract(WaterMarked)
% 初始化水印信息
waterMark = 0;
% 遍历水印信息的比特位
for j = 1:21
    % 提取水印信息的比特值
    tem = bitget(WaterMarked(1, j), 1);
    % 将比特值设置到水印信息中的对应位置
    waterMark = bitset(waterMark, j, tem);
end
end
```

实验结果



```
10 | watermark = Extract(watermark)
命令行窗口
>> IntHiding
Extracted WaterMark: 2113662
fx >> |
```

小结

这次实验我们学习了LSB隐藏法，通过该方法可以将秘密信息嵌入到数字图像中，实现信息的隐蔽传输。

具体步骤包括选择一幅图像作为载体图像，然后选取图像的最低有效位作为隐藏信息的嵌入位置，将秘密信息的二进制码嵌入到最低有效位中。在实现过程中，我们需要注意保证隐藏信息的安全性和完整性，避免信息丢失和攻击。

通过本次实验，我对LSB隐藏法有了更深入的理解，并掌握了基本的实现方法。在今后的学习和工作中，我将继续探索信息隐藏技术的更多应用和改进方法，提高信息传输的安全性和效率。