

# 实验六 图像的LSB隐藏法实验

---

学号：2110688 姓名：史文天 专业：信息安全

## 实验要求

---

- 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。

## 实验原理

---

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。

优点：简单，容易实现，容量大；

缺点：安全性不高，不能抵抗叠加噪声，有损压缩等破坏。

## 原图像展示

---



# 将二值图像嵌入到位图中

## 实验步骤

1. 将秘密信息转换为二进制码。
2. 将数字图像的像素点转换为二进制码。
3. 将秘密信息的二进制码依次嵌入到数字图像像素点的最低有效位中。
4. 将修改后的像素点重新组合成数字图像。
5. 接收方通过提取数字图像的最低有效位，即可获取秘密信息的二进制码，再将其转换为原始信息。

## 代码部分

- 首先，获取嵌入水印后的图像的大小，即行数和列数，分别为Mw 和Nw。
- 然后，创建一个与嵌入水印后的图像大小相同的全零矩阵WaterMark，用于存储提取出的水印图像。
- 接着，使用两层for 循环遍历嵌入水印后的图像的每个像素点，提取出每个像素点的最低有效位，并将其存储到WaterMark 矩阵的对应位置中。具体实现是使用bitget 函数获取嵌入水印后的图像的对应像素点的最低有效位。
- 提取完成后，使用imshow 函数显示提取出的水印图像，并设置title 为“WaterMark”。

```
function ImageHiding()
    x=imread("Lena.bmp"); %载体图像
    m=imread("logo.bmp"); %水印图像
    imshow(x, [])
    imshow(m, []);
    waterMarked=Hide(x,m);
    watermark=Extract(waterMarked);
end

function waterMarked = Hide(origin,watermark)
    [Mc,Nc]=size(origin);
    waterMarked=uint8(zeros(size(origin)));

    for i=1:Mc
        for j=1:Nc
            waterMarked(i,j)=bitset(origin(i,j),1,watermark(i,j));
        end
    end

    imwrite(waterMarked,'lsb_watermarked.bmp','bmp');
    figure;
    imshow(waterMarked, []);
    title("waterMarked Image");
end

function watermark=Extract(waterMarked)
    [Mw,Nw]=size(waterMarked);
    watermark=uint8(zeros(size(waterMarked)));

    for i=1:Mw
        for j=1:Nw
```

```

        WaterMark(i,j)=bitget(WaterMarked(i,j),1);
    end
end

figure;
imshow(WaterMark, []);
title("WaterMark");
end

```

## 结果分析

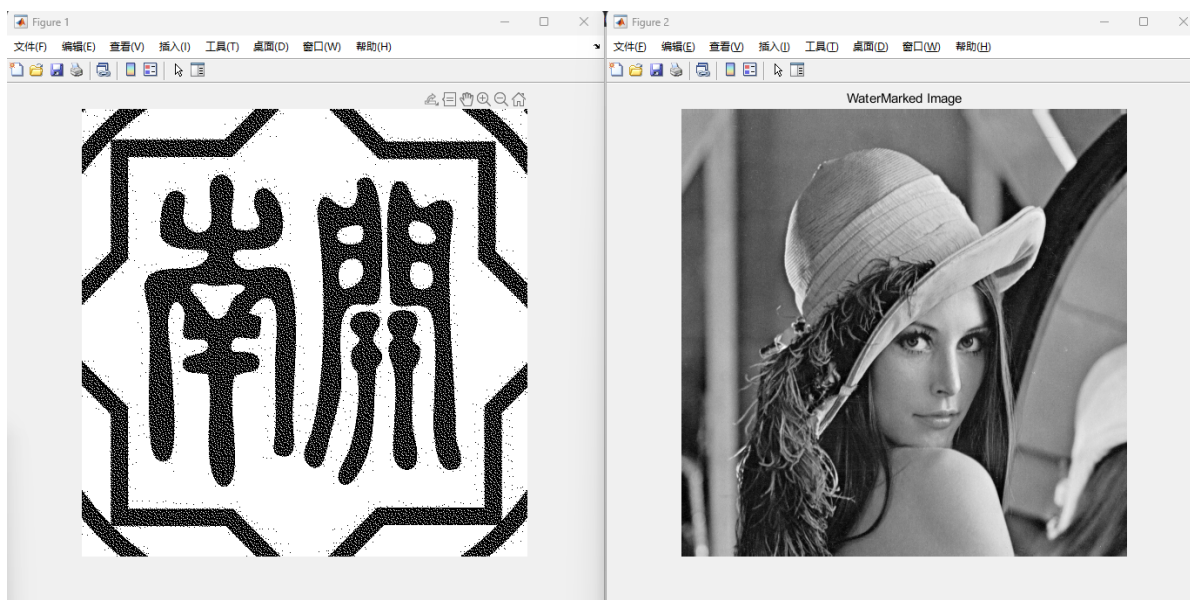
运行ImageHiding

```

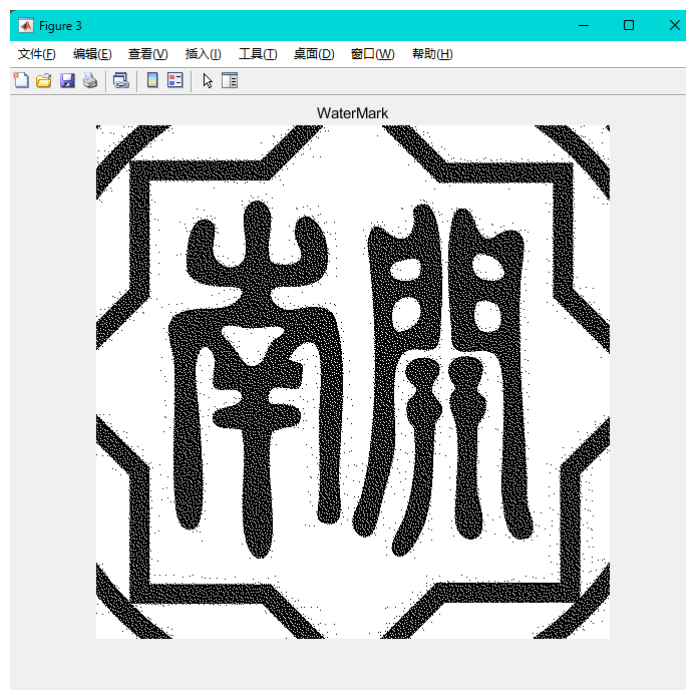
>> ImageHiding
fx
>>

```

右图中添加了左图的水印



最后从右图中成功分离出水印



# 将学号（一个整数）嵌入到位图中

## 实验步骤

1. 将要嵌入的整数转换为二进制码。
2. 读取载体图片，并将其转换为二进制码。
3. 将要嵌入的整数的二进制码依次嵌入到载体图片像素点的最低有效位中。具体实现是使用bitset 函数将载体图片的对应像素点的最低有效位替换为要嵌入的整数的对应二进制位。
4. 将修改后的像素点重新组合成图片。
5. 嵌入完成后，保存修改后的图片。
6. 接收方通过提取图片的最低有效位，即可获取嵌入的整数的二进制码，再将其转换为原始整数。

## 代码部分

- 首先，使用imread 函数读取载体图像，即"Lena.bmp"。
- 然后，将要嵌入的信息m 设置为2110688（一个整数）。
- 接着，使用imshow 函数显示载体图像。其中，[] 表示将图像的像素值映射到[0,1] 之间，以便于显示。
- 然后，调用Hide 函数将整数m 嵌入到载体图像中，生成一个新的嵌入信息后的图像。Hide函数是一个自定义函数，实现了将整数通过LSB 算法嵌入到数字图像中的过程。
- 嵌入完成后，调用Extract 函数从新生成的嵌入信息后的图像中提取出原始的整数。Extract函数是自定义函数，实现了从数字图像中提取出嵌入的整数的过程。
- 最后，将提取出的整数赋值给变量watermark，并输出到命令窗口。

```
function IntHiding()  
    x=imread("Lena.bmp"); %载体图像  
    m=2110688; %要嵌入的信息  
    imshow(x, [])  
    waterMarked=Hide(x,m);  
    watermark=Extract(waterMarked)  
end
```

- 首先，获取载体图像的大小，即行数和列数，分别为Mc 和Nc。
- 然后，创建一个与载体图像大小相同的全零矩阵WaterMarked，用于存储嵌入水印后的图像。
- 使用两层for 循环遍历载体图像的每个像素点，如果当前像素点在第一行且列数小于等于22，则将要嵌入的整数的对应二进制位嵌入到该像素点的最低有效位中。具体实现是使用bitset函数将载体图像的对应像素点的最低有效位替换为要嵌入的整数的对应二进制位。
- 如果当前像素点不在第一行或列数大于22，则直接将该像素点的值赋给WaterMarked 矩阵的对应位置。  
嵌入完成后，使用imwrite 函数将嵌入水印后的图像保存为"lsb\_int\_watermarked.bmp" 文件。
- 最后，使用imshow 函数显示嵌入水印后的图像，并设置title 为"WaterMarked Image"。

```
function waterMarked = Hide(origin,watermark)  
    [Mc,Nc]=size(origin);  
    waterMarked=uint8(zeros(size(origin)));
```

```

for i=1:Mc
    for j=1:Nc
        if i==1 && j<=22
            tem=bitget(watermark,j);
            waterMarked(i,j)=bitset(origin(i,j),1,tem);
        else
            waterMarked(i,j)=origin(i,j);
        end
    end
end

imwrite(WaterMarked,'lsb_int_watermarked.bmp','bmp');
figure;
imshow(WaterMarked,[]);
title("WaterMarked Image");
end

```

- 首先，将提取出的整数WaterMark 初始化为0。
- 使用for 循环遍历嵌入水印后的图像的第一行的前21 个像素点，即整数的二进制码的前22位。
- 对于每个像素点，使用bitget 函数获取其最低有效位，并将其存储到tem 变量中。
- 然后，使用bitset 函数将tem 存储到WaterMark 变量的对应二进制位中。
- 循环结束后，WaterMark 变量中存储的就是从嵌入水印后的图像中提取出的整数的二进制码的前 21 位。
- 最后，将提取出的整数输出。

```

function WaterMark=Extract(WaterMarked)
    WaterMark=0;
    for j=1:22
        tem=bitget(WaterMarked(1,j),1);
        WaterMark=bitset(WaterMark,j,tem);
    end
end

```

## 结果分析

运行IntHiding函数，输出了隐藏了水印的图片，并还原了水印内容

```

>> IntHiding

watermark =

    2110688

fx >>

```



## 实验总结

---

通过本次的实验，我了解了LSB算法这种最为基础的图像隐写术算法，实现了将二值图像和整数嵌入图像的实验。同时我也了解到LSB算法的局限性，例如隐蔽性较差，需要结合其他加密算法一起使用。