

信息隐藏技术第四次实验 —— LSB隐藏法

学号：2111033 姓名：艾明旭

专业：信息安全

一、实验要求

1.实验要求

1. 在 MATLAB 中调试完成
2. 编程实现，写出实验报告，含程序代码和截图，word/pdf 格式
3. QQ 群提交作业

2.实验目的

1. 实现将二值图像嵌入到位图中
2. 实现将学号嵌入到位图中

二、实验原理

1.特点

1. 简单，易实现，容量大
2. 安全性不高，不能抵抗叠加噪声、有损压缩等破坏

2.基本原理

lsb方法：

LSB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。

对空域的LSB做替换，用来替换LSB的序列就是需要加入的水印信息、水印的数字摘要或者由水印生成的伪随机序列。由于水印信息嵌入的位置是LSB，为了满足水印的不可见性，允许嵌入的水印强度不可能太高。然而针对空域的各种处理，如游程编码前的预处理，会对不显著分量进行一定的压缩，所以LSB算法对这些操作很敏感。

安全性：

主要有如下措施：

1. 对秘密信息进行加密后再隐藏
2. 多次重复嵌入
3. 引入纠错编码技术。先进行纠错编码，再进行隐藏。

LSB 算法相关函数

1. 获取图像 x 的行数和列数
2. 嵌入/提取图像 bit 位的值

流载体的LSB方法

嵌入 选择一个载体元素的子集 $\{j_1, j_2, \dots, j_{L(m)}\}$ ，其中共有 $L(m)$ 个元素，用以隐藏秘密信息的 $L(m)$ 个比特。然后在这个子集上执行替换操作，把 C_{j_i} 的最低比特用 m_i 来替换。

提取 找到嵌入信息的伪装元素的子集 $\{j_1, j_2, \dots, j_{L(m)}\}$ ，从这些伪装对象 S_{j_i} 中抽出它们的最低比特位，排列之后组成秘密信息。

3.基本步骤

1. 将原始载体图像的空域像素值由十进制转换成二进制
2. 用二进制秘密信息中的每一比特信息替换与之相对应的载体数据的最低有效位
3. 将得到的含秘密信息的二进制数据转换为十进制像素值，从而获得含秘密信息的图像

三、实验步骤

水印图像是二值图像，载体图像是与水印图像大小相同的 256 级灰度图像，进行 LSB 水印图像的嵌入和提取读取

1.将二值图像嵌入到位图中

读取 Lena 图像和 bmp 图像

其中 Lena 图像是灰度图像，NK 图像是水印(二值)图像

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

- 函数原型：y=imread(filename,fmt)
- 功能：读取 fmt 指定格式的图像文件内容
- 输入参数：filename 表示图像文件名，字符串

原始图像如下所示：



利用bitset函数将隐藏图像嵌入到载体图像的最低位平面，实现信息的隐藏

```

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

```

嵌入水印后的图像如下所示：



可以看出，在视觉方面伪装对象和载体对象无法区分

利用bitget函数提取伪装对象的最低位平面，恢复隐藏的图像

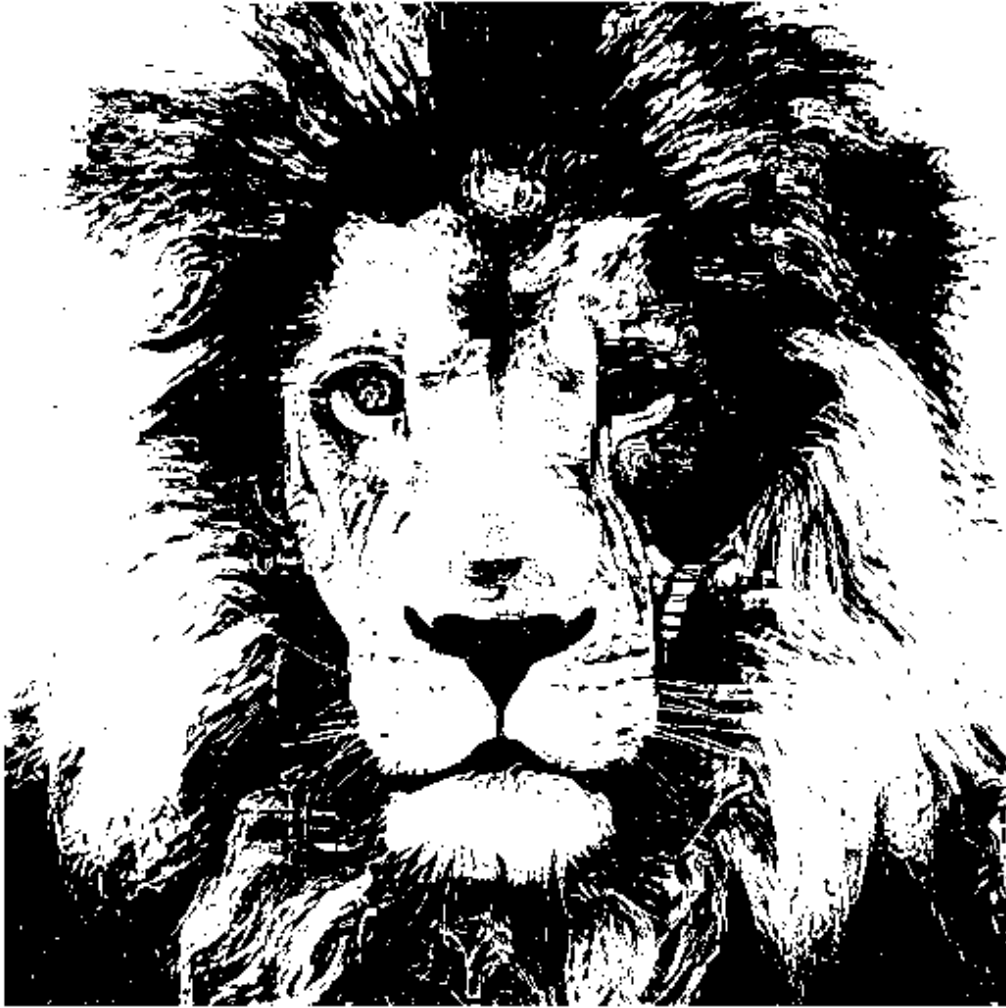
```
function WaterMark=Extract(WaterMarked)
    [Mw,Nw]=size(WaterMarked);
    WaterMark=uint8(zeros(size(WaterMarked)));

    for i=1:Mw
        for j=1:Nw
            WaterMark(i,j)=bitget(WaterMarked(i,j),1);
        end
    end

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

恢复后的水印图像如下所示：

WaterMark



其他的嵌入和提取方法

1.利用fwrite和fseek函数

```

function WaterMarked = Hide(origin,watermark)
    fidx=fopen(origin,'r');
    [x,xlength]=fread(fidx,inf,'uint8');
    fidm=fopen(watermark,'r');
    [m,mlength]=fread(fidm,inf,'uint8');
    %写入所有数据
    fid=fopen('Method2_watermarked.bmp','w');
    fwrite(fid,x);
    fwrite(fid,m);
    fclose(fid);
    fclose(fidx);
    fclose(fidm);
    WaterMarked=imread('Method2_watermarked.bmp');
    figure;
    imshow(WaterMarked,[]);
    title("WaterMarked Image");
    WaterMarked="Method2_watermarked.bmp";
end

function WaterMark=Extract(WaterMarked)
    fid=fopen(WaterMarked,'r');
    fseek(fid,2,"bof");
    moffset=fread(fid,1,'uint32');
    fseek(fid,moffset+2,"bof");
    mlen=fread(fid,1,"uint32");
    fseek(fid,moffset,"bof");
    fWaterMark=fread(fid,mlen,'uint8');
    fclose(fid);
    mfid=fopen("method2_watermark.bmp",'w');
    fwrite(mfid,fWaterMark,'uint8');
    fclose(mfid);
    WaterMark=imread('method2_watermark.bmp');
    figure;
    imshow(WaterMark,[]);
    title("WaterMark");
end

```

2.利用偏移量实现隐藏和提取

```

function Method3()
    x=input('请输入载体图像: ','s'); %载体图像
    m=input('请输入要隐藏的秘密信息图像: ','s'); %水印图像
    WaterMarked=Hide(x,m);
    watermark=Extract(WaterMarked);
end

```

```

function WaterMarked = Hide(origin,watermark)
    fidx=fopen(origin,'r');
    %获取图像数据的偏移
    fseek(fidx,10,"bof");
    xoffset=fread(fidx,1,"uint32");
    %读取出文件头
    fseek(fidx,0,"bof");
    [xhead,xlength]=fread(fidx,xoffset,'uint8');
    %读取出图像数据
    [xtail,xtlength]=fread(fidx,inf,"uint8");
    fidm=fopen(watermark,'r');
    [m,mlength]=fread(fidm,inf,'uint8');
    %写入所有数据，将水印信息放到文件头和图像数据之间的位置
    fid=fopen('Method3_watermarked.bmp','w');
    fwrite(fid,xhead);
    fwrite(fid,m);
    fwrite(fid,xtail);
    %还要更改图像数据偏移的字段，加上水印信息的文件长度
    fseek(fid,10,"bof");
    fwrite(fid,xoffset+mlength,"uint32");
    fclose(fid);
    fclose(fidx);
    fclose(fidm);
    WaterMarked=imread('Method3_watermarked.bmp');
    figure;
    imshow(WaterMarked,[]);
    title("WaterMarked Image");
    WaterMarked="Method3_watermarked.bmp";
end

```

```

function WaterMark=Extract(WaterMarked)
    fid=fopen(WaterMarked,'r');
    %获取载体图像的实际大小
    fseek(fid,2,"bof");
    xlen=fread(fid,1,'uint32');
    %读取整个文件，得到整个文件的大小
    fseek(fid,0,"bof");
    [A,truelen]=fread(fid,inf,'uint8');
    %相减，即可得到秘密信息的大小
    mlen=truelen-xlen;
    %获取载体图像数据的偏移，减去秘密信息的大小，就得到了秘密信息的偏移
    fseek(fid,10,"bof");
    xoffset=fread(fid,1,"uint32");

```



```

moffset=xoffset-mlen;
%由此，即可读出秘密信息
fseek(fid,moffset,"bof");
m=fread(fid,mlen,"uint8");
fidm=fopen("method3_watermark.bmp","w");
fwrite(fidm,m,"uint8");
fclose(fidm);
fclose(fid);
WaterMark=imread('method3_watermark.bmp');
figure;
imshow(WaterMark,[]);
title("WaterMark");
end

```

2.将学号(一个整数)嵌入到位图中

利用 `LSB` 方法将学号 `2111033` 嵌入到载体图像中

读取 Lena 灰度图像并调整大小

```

function Method4()
    x=input('请输入载体图像: ','s'); %载体图像
    m=input('请输入要隐藏的秘密信息(int型): '); %秘密信息
    WaterMarked=Hide(x,m);
    watermark=Extract(WaterMarked);
end

```

灰度图像如下所示：

灰度图像



嵌入学号 2111033 后得到水印图像

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。

```

function WaterMarked = Hide(origin,watermark)
    fidx=fopen(origin,'r');
    fseek(fidx,0,"bof");
    [x,xlength]=fread(fidx,inf,'uint8');
    fid=fopen('Method4_watermarked.bmp','w');
    fwrite(fid,x);
    %加入秘密信息
    fseek(fid,6,"bof");
    fwrite(fid,watermark,"uint32");
    fclose(fid);
    fclose(fidx);
    WaterMarked=imread('Method4_watermarked.bmp');
    figure;
    imshow(WaterMarked,[]);
    title("WaterMarked Image");
    WaterMarked="Method4_watermarked.bmp";
end

```

水印图像如下所示：



从水印图像中提取学号

```
function WaterMark=Extract(WaterMarked)
    fid=fopen(WaterMarked,'r');
    %获取秘密信息
    fseek(fid,6,"bof");
    m=fread(fid,1,'uint32');
    fclose(fid);
    WaterMark=m;
    WaterMark
end
```

最终结果如下所示，可以看到结果正确：

```
>> Method4
请输入载体图像：lena.bmp
请输入要隐藏的秘密信息(int型)：2111033

WaterMark =

    2111033
```

四、总结与展望

本次实验利用 MATLAB 对秘密图像进行 LSB 隐藏，将秘密信息隐藏在载体较低比特的噪声位，实现秘密信息的水印嵌入和提取算法。实验过程中，我深刻地认识到了数字图像隐写术的重要性和应用价值。LSB算法是数字图像隐写术中最基础、最常用的一种方法，其原理简单、实现方便，可以用于数字图像的版权保护、身份认证等方面。在实验中，我通过编写 MATLAB 代码，实现了将一个整数通过 LSB 算法嵌入到数字图像中，并从嵌入后的图像中提取出原始整数的过程。同时，我也了解到了 LSB 算法的局限性，即隐蔽性较差，容易被攻击者检测到，因此需要结合其他隐写术和加密算法，以提高信息的安全性。

在实验中，我还学习了 MATLAB 的基本语法和图像处理函数的使用方法。通过编写代码，我深入理解了数字图像的存储方式和像素点的组成，以及如何对图像进行读取、显示、修改和保存等操作。同时，我也学会了如何使用 MATLAB 的位运算函数，如 bitget 和 bitset，来实现 LSB 算法的嵌入和提取过程。

最后期待自己未来更好的发展，能够继续在学习信息隐藏技术这么课程的道路上走得更高更远！