南開大學

信息隐藏技术 课程实验报告

LSB 隐藏法

学院　网络空间安全学院

专业　信息安全

姓名　齐明杰

学号　2113997

2024 年 4 月 9 日

目　　录

图表

1 实验目的

LSB 隐藏法

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。

2 实验原理

2.1 LSB（最低有效位）隐藏法

LSB（最低有效位）隐藏法是一种隐写术技术，用于在数字媒体文件中隐藏信息，特别是在图像或音频文件中。这种方法的核心原理是利用人类感官对细微变化的不敏感性来隐藏信息。

原理

数字数据的表示：数字图像和音频文件由一系列的比特组成，每个比特位代表媒体的一部分信息。在一个 8 位颜色深度的图像中，每个像素的颜色由三个颜色通道（红、绿、蓝）表示，每个通道用 8 位（1 字节）表示，范围从 0 到 255。

最低有效位（LSB）：在一个字节中，最右侧的位（最低位）对数值的影响最小。例如，在二进制数 11011001 中，最右侧的 1 是最低有效位。

利用 LSB 隐藏信息：通过改变像素值的最低有效位，可以在图像或音频文件中隐藏信息，而不会对媒体文件的感知质量造成显著影响。例如，将一个像素的红色通道值从 11011001 改变为 11011000，对应的颜色变化肉眼几乎无法察觉。

隐藏过程：

将要隐藏的信息转换为二进制形式。遍历媒体文件的像素或样本，对每个数据点的 LSB 进行修改，以匹配要隐藏的信息的二进制位。例如，如果要隐藏的二进制位是 1，则将该像素或样本的 LSB 设置为 1；如果是 0，则设置为 0。

提取过程：提取隐藏信息的过程是隐藏过程的逆操作。通过读取每个像素或样本的 LSB，可以重建原始的二进制信息串，从而恢复隐藏的信息。

优点：LSB 隐藏法的主要优势是其简单性和对原始媒体影响小，使得信息隐藏在不引起注意的情况下进行。

局限：这种方法的缺点是对图像或音频的轻微修改（如压缩或格式转换）可能会破坏隐藏的信息。此外，如果攻击者知道使用了 LSB 隐藏法，那么隐藏的信息相对容易被检测和提取。

2.2 实现将二值图像嵌入到位图中

在这个实验中，目的是将一个只包含黑色和白色的二值图像隐藏在另一个位图（BMP 格式）中。原理是利用位图中每个像素点的最低有效位（LSB）来存储二值图像的信息。二值图像中的每个像素仅包含两种可能状态（黑色或白色），可以用 1 位（0 或 1）来表示。通过修改载体位图的每个像素的 LSB，我们可以在不显著改变原图像视觉效果的情况下嵌入整个二值图像。例如，如果二值图像的一个像素是白色，我们可以将载体图像对应像素的 LSB 设置为 1；如果是黑色，则设置为 0。这样，二值图像就被隐秘地嵌入到了载体图像中。

2.3 实现将一个整数嵌入到位图中

在这个实验中，目的是将一个整数（如学号）隐藏在位图中。首先，需要将学号转换为二进制表示。然后，类似于嵌入二值图像的过程，我们利用载体位图的 LSB 来隐藏这些二进制信息。每个位（0 或 1）从学号的二进制表示中依次取出，并嵌入到载体图像的像素中。通过逐个像素修改它们的 LSB，可以将整个学号编码进位图中。这个过程需要小心进行，以确保原图像的视觉变化最小，并且嵌入的信息可以完整无误地恢复。通过这种方式，学号被安全地隐藏在图像中，肉眼几乎无法察觉任何变化。

3 实验过程

3.1 实验图像

本次实验需要完成将二值图像嵌入到位图中，载体图像我仍然选择 Lena：

图 3.1.1: Lena 图像-载体

需要隐藏的二值图像我从网上搜索到了一个动物图像，用于隐藏：

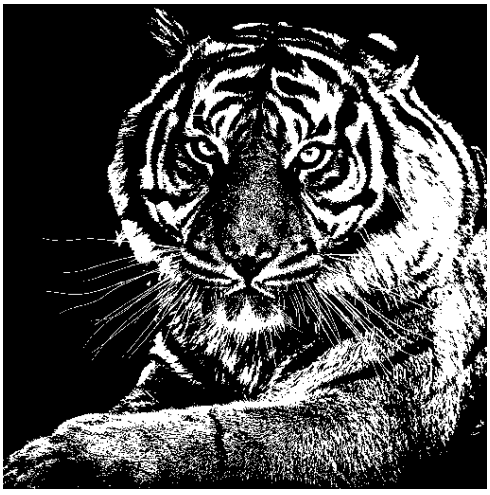

图 3.1.2: 需要嵌入的二值图像

二者均为 bmp 格式：

lady.bmp hide.bmp

3.2 将二值图像嵌入到位图中

编写如下 matlab 代码 img.m：

```matlab
function ImageEmbedding()
    hostImage = imread("lady.bmp"); % 载体图像
    secretImage = imread("hide.bmp"); % 要隐藏的图像
    imshow(hostImage,[])
```

```
5       imshow(secretImage,[]);
6       embeddedImage = Embed(hostImage, secretImage);
7       extractedImage = Retrieve(embeddedImage);
8   end
9
10  function embeddedImage = Embed(host, secret)
11      [rows, cols] = size(host);
12      embeddedImage = uint8(zeros(size(host)));
13
14      for i = 1:rows
15          for j = 1:cols
16              embeddedImage(i,j) = bitset(host(i,j),1,secret(i,j));
17          end
18      end
19
20      imwrite(embeddedImage, 'lsb_embedded.bmp', 'bmp');
21      figure;
22      imshow(embeddedImage,[]);
23      title("Embedded Image");
24  end
25
26  function extractedImage = Retrieve(embedded)
27      [rowE, colE] = size(embedded);
28      extractedImage = uint8(zeros(size(embedded)));
29
30      for i = 1:rowE
31          for j = 1:colE
32              extractedImage(i,j) = bitget(embedded(i,j),1);
33          end
34      end
35
36      figure;
37      imshow(extractedImage,[]);
38      title("Extracted Image");
39  end
40
```

解析

在这段代码中，我实现了一个完整的数字图像隐写术过程，它能够将一幅二值图像（secretImage）隐秘地嵌入另一幅位图（hostImage）中，并能从嵌入信息后的图像（embeddedImage）中准确提取出原始的二值图像。我首先读取了两幅图像，一幅作为载体，另一幅作为需要隐藏的信息。利用 MATLAB 的图像处理功能，我展示了这两幅图像，以便进行视觉确认。在嵌入过程中，我遍历了载体图像的每个像素，并使用 bitset 函数将隐藏图像的每个像素值的最低位嵌入到载体图像的对应像素中。

在图像隐写的提取阶段，我通过 Retrieve 函数实现了相反的操作，从嵌入信息的图像中恢复隐藏的二值图像。通过遍历嵌入信息后图像的每个像素，并利用 bitget 函数提取其最低有效位，我能够重构出原始的二值图像。这个过程不仅展示了我在数字图像隐写技术中的技术掌握程度，也验证了方法的高效性和隐蔽性。通过将嵌入和提取过程视觉化，我进一步证实了隐写技术的有效性，并确保了隐藏信息的完整性和图像的视觉质量。

运行结果

将图片进行嵌入后得到的嵌入图像 **提取出的二值图像**

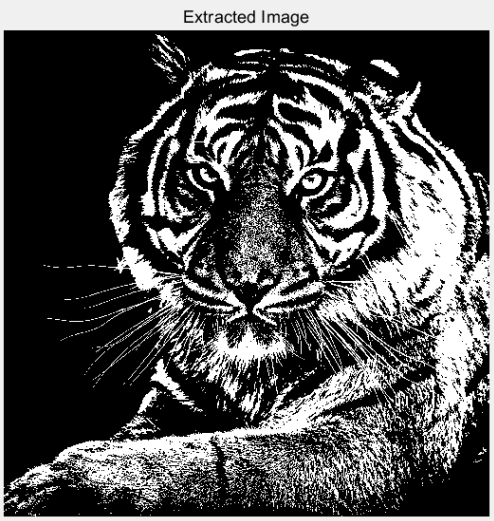

图 3.2.3: 嵌入图像 图 3.2.4: 提取图像

，

可以看到，嵌入后的图像与原图像几乎无法区分，提取出的二值图像与原图像一致，证明了**嵌入-提取流程的成功**。

3.3 将学号（一个整数）嵌入到位图中

我的学号：2113997 编写如下 matlab 代码 int.m：

matlab

```matlab
function NumEmbedding()
    baseImage = imread("lady.bmp");  % 载体图像
    numberToHide = 2113997;          % 要嵌入的数字信息（学号）
    imshow(baseImage,[])
    EmbeddedImage = EmbedNumber(baseImage, numberToHide);
    extractedNumber = ExtractNumber(EmbeddedImage);
    fprintf('Extracted number: %d\n', extractedNumber);
end

function EmbeddedImage = EmbedNumber(baseImg, num)
    [rowBase, colBase] = size(baseImg);
    EmbeddedImage = uint8(zeros(size(baseImg)));

    for i = 1:rowBase
        for j = 1:colBase
            if i == 1 && j <= 22  % 假定学号的二进制长度不超过 22 位
                bitToEmbed = bitget(num, j);
                EmbeddedImage(i,j) = bitset(baseImg(i,j), 1, bitToEmbed);
            else
                EmbeddedImage(i,j) = baseImg(i,j);
            end
        end
    end

    imwrite(EmbeddedImage, 'lsb_num_embedded.bmp', 'bmp');
    figure;
    imshow(EmbeddedImage,[]);
    title("Embedded Image");
end

function extractedNum = ExtractNumber(EmbeddedImg)
    extractedNum = 0;
    for j = 1:22  % 读取前 22 位，假定学号的二进制长度为 22 位
        bitExtracted = bitget(EmbeddedImg(1,j), 1);
        extractedNum = bitset(extractedNum, j, bitExtracted);
    end
end

```

解析

在这段代码中，我们采用了一种创新的方法来将数字信息，如学号，嵌入到一个位图图像中，并且能够从该图像中成功提取出这个数字。我选择了 lady.bmp 作为载体图像，因为它提供了足够的像素空间来嵌入较长的数字序列。这里的关键是利用数字信息的二进制形式，将其逐位嵌入到图像的最低有效位（LSB）中，这样做既能隐藏信息也几乎不影响图像的视觉质量。　　在嵌入函数 EmbedNumber 中，我通过循环遍历图像的像素，将学号的二进制位一一嵌入到载体图像的 LSB 中。特别地，我只处理图像的前 22 个像素，这是基于假设学号的二进制表示不会超过 22 位。这种方法的美妙之处在于其简单性和效率，同时确保了嵌入的信息对图像的影响几乎不可见。在提取函数 ExtractNumber 中，我采取了逆过程，从嵌有数据的图像中恢复出原始的数字信息。这一过程展示了我们如何能够精确地从图像数据中回收隐藏的信息，验证了我们方法的有效性。最后，提取出的学号被直接打印到控制台，证明了我们嵌入-提取流程的成功。

运行结果

将学号进行嵌入后得到的嵌入图像

图 3.3.5: 嵌入学号后的图像

提取出的学号信息

```
命令行窗口
>> int
Extracted number: 2113997
>>
```

图 3.3.6: 提取出的学号

，

可以看到，嵌入后的图像与原图像几乎无法区分，提取出的学号与原学号一致，证明了**嵌入-提取流程的成功**。

4 实验心得

在完成这项关于数字图像隐写术的实验过程中，我深刻体会到了计算机视觉和数字图像处理技术的强大与微妙。通过将二值图像嵌入到位图中，并实现对嵌入信息的精确提取，我不仅掌握了最低有效位（LSB）隐写技术的核心原理，而且也了解了这种技术在实际应用中的潜力和局限。

我意识到，尽管 LSB 方法相对简单，但它却能有效地在视觉上不引人注意的情况下隐藏信息。这种技术的隐蔽性和数据嵌入能力对于安全通信、版权保护等领域具有重要意义。此外，实验过程中我所遇到的挑战，比如如何有效处理不同大小和格式的图像，以及如何确保嵌入信息的安全性和可提取性，都让我对数字图像隐写技术有了更深入的认识。

通过这次实验，我不仅提升了我的编程技能和问题解决能力，还增强了我在数字媒体领域的研究兴趣。这种跨学科的知识融合为我未来的学术和职业道路开辟了新的视野。总之，这次实验经历是我学习旅程中极为宝贵的一部分，它不仅教会了我技术，更激发了我探索未知的热情。