

浙江大学



课程名称:	多媒体安全
姓 名:	
学 院:	计算机学院
专 业:	计算机科学与技术
学 号:	
指导老师:	黄劲
完成时间:	2023 年 6 月 18 日

实验四：F3/F4 隐写术分析

一、实验目的

1. 了解数字图像隐写术中的典型算法：F3, F4 的基本原理。
2. 了解图像 DCT 系数的计算。
3. 了解数字水印与隐写术的异同。

二、实验内容与要求

1. 实现 F3、F4 的隐写系统，包括信息嵌入与信息检测。
2. 设计一份 3KB 左右的文本信息数据，使用上述两种隐写方法进行信息的嵌入。绘制原图与两种隐写方法嵌入得到的结果图的 DCT 系数直方图，并分析两种方法对原图 DCT 系数的影响。
3. 阐述隐写系统与数字水印系统的异同。

三、实验环境

语言版本：MATLAB R2020b

四、实验过程

4.1 Jpeg 量化处理

此部分主要是实现图像转换为 DCT 系数行向量，以及从该行向量重建图像。

将图像转换为对应的 DCT 系数需要对每个图像的 8*8 小块单独进行 DCT 变换，取整后得到 DCT 量化系数。

定义函数 getJpegCoe() 由图像数值矩阵获取其 DCT 量化系数矩阵，如下：

```
1 function [mat] = getJpegCoe(mat)
2
3     Quan = [
4         16, 11, 10, 16, 24, 40, 51, 61;
5         12, 12, 14, 19, 26, 58, 60, 55;
6         14, 13, 16, 24, 40, 57, 69, 56;
```

```
7         14, 17, 22, 29, 51, 87, 80, 62;
8         18, 22, 37, 56, 68, 109, 103, 77;
9         24, 35, 55, 64, 81, 104, 113, 92;
10        49, 64, 78, 87, 103, 121, 120, 101;
11        72, 92, 95, 98, 112, 100, 103, 99;
12    ];
13    for i = 1 : 64
14        for j = 1 : 64
15            part = mat((i * 8 - 7) : (i * 8), (j * 8 - 7) : (j * 8));
16            part = round(dct2(part) ./ Quan);
17            mat((i * 8 - 7) : (i * 8), (j * 8 - 7) : (j * 8)) = part;
18        end
19    end
20 end
```

之后, 通过一个排列矩阵把所有高频系数排列在行向量的最前面, 以方便隐写术植入信息的同时最小化对图像的扰动。

定义函数 `rearrangeCoe()` 由 DCT 量化系数矩阵获取排列后的 DCT 系数行向量, 如下:

```
1 function [lst] = rearrangeCoe(image)
2
3     Zigzag = [
4         64, 63, 59, 58, 50, 49, 37, 36;
5         62, 60, 57, 51, 48, 38, 35, 22;
6         61, 56, 52, 47, 39, 34, 23, 21;
7         55, 53, 46, 40, 33, 24, 20, 11;
8         54, 45, 41, 32, 25, 19, 12, 10;
9         44, 42, 31, 26, 18, 13, 9, 4;
10        43, 30, 27, 17, 14, 8, 5, 3;
11        29, 28, 16, 15, 7, 6, 2, 1;
12    ];
13    lst = zeros([1, 512 * 512]);
14    for i = 1 : 64
15        for j = 1 : 64
16            for k = 1 : 8
```

```
17         for l = 1 : 8
18             lst(((i - 1) * 64 + j - 1) * 64 + Zigzag(k, l)) =
19                 image((i - 1) * 8 + k, (j - 1) * 8 + l);
20         end
21     end
22 end
23 end
24 end
```

之后再实现以上两个函数的逆变换 deJpeg() 和 restoreCoe() 即可, 代码略。

4.2 F3 隐写及检测

此部分是实现 F3 隐写术植入信息以及对信息进行 F3 检测。

F3 隐写术的思想是: 对每个系数, 按照 LSB 植入信息 (若 LSB 不对应正确的码字则让该系数朝 0 的方向进 1), 按照 LSB 读取信息。具体实现上, 所有 0 系数均不应读出信息, 故需要在植入时对 ± 1 和 0 进行特殊处理。

定义函数 F3_embed() 根据待植入向量 lst 和隐写信息 message 完成信息植入并返回。实现如下:

```
1 function [lst] = F3_embed(lst, message)
2
3     i = 1;
4     j = 1;
5     while j ~= length(message) + 1
6         if lst(i) ~= 0
7             if message(j) ~= mod(lst(i), 2)
8                 lst(i) = lst(i) - sign(lst(i));
9             end
10            if lst(i) ~= 0
11                j = j + 1;
12            end
13        end
14        i = i + 1;
15    end
16 end
```

定义函数 F3_decode() 根据待解码向量 lst 和信息长度 len 完成信息解码并返回解码信息 message。实现如下:

```
1 function [message] = F3_decode(lst , len)
2
3     message = zeros([1, len]);
4     i = 1;
5     j = 1;
6     while j ~= len + 1
7         if lst(i) ~= 0
8             message(j) = mod(lst(i), 2);
9             j = j + 1;
10        end
11        i = i + 1;
12    end
13 end
```

4.3 F4 隐写及检测

此部分是实现 F4 隐写术植入信息以及对信息进行 F4 检测。

F4 隐写术的思想是: 在 F3 的基础上, 对负系数反向处理, 即让 LSB 为 1 的系数对应码字 0, 其余与 F3 相同。

定义函数 F4_embed() 根据待植入向量 lst 和隐写信息 message 完成信息植入并返回。实现如下:

```
1 function [lst] = F4_embed(lst , message)
2
3     i = 1;
4     j = 1;
5     while j ~= length(message) + 1
6         if lst(i) < 0 && mod(lst(i), 2) == message(j)
7             lst(i) = lst(i) + 1;
8         elseif lst(i) > 0 && mod(lst(i), 2) ~= message(j)
9             lst(i) = lst(i) - 1;
10        end
11        if lst(i) ~= 0
12            j = j + 1;
```

```
13         end
14         i = i + 1;
15     end
16 end
```

定义函数 F4_decode() 根据待解码向量 lst 和信息长度 len 完成信息解码并返回解码信息 message。
实现如下:

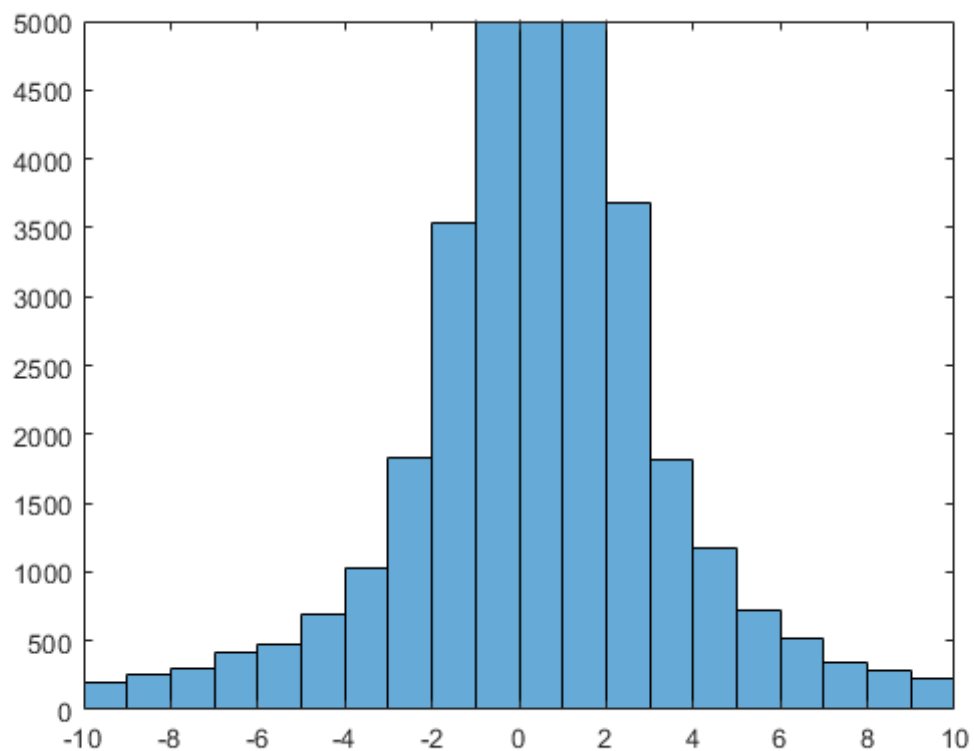
```
1 function [message] = F4_decode(lst , len)
2
3     message = zeros([1 , len]);
4     i = 1;
5     j = 1;
6     while j ~= len + 1
7         if lst(i) < 0
8             message(j) = 1 - mod(lst(i), 2);
9             j = j + 1;
10        elseif lst(i) > 0
11            message(j) = mod(lst(i), 2);
12            j = j + 1;
13        end
14        i = i + 1;
15    end
16 end
```

4.4 F3/F4 隐写术效果比较

原图为:



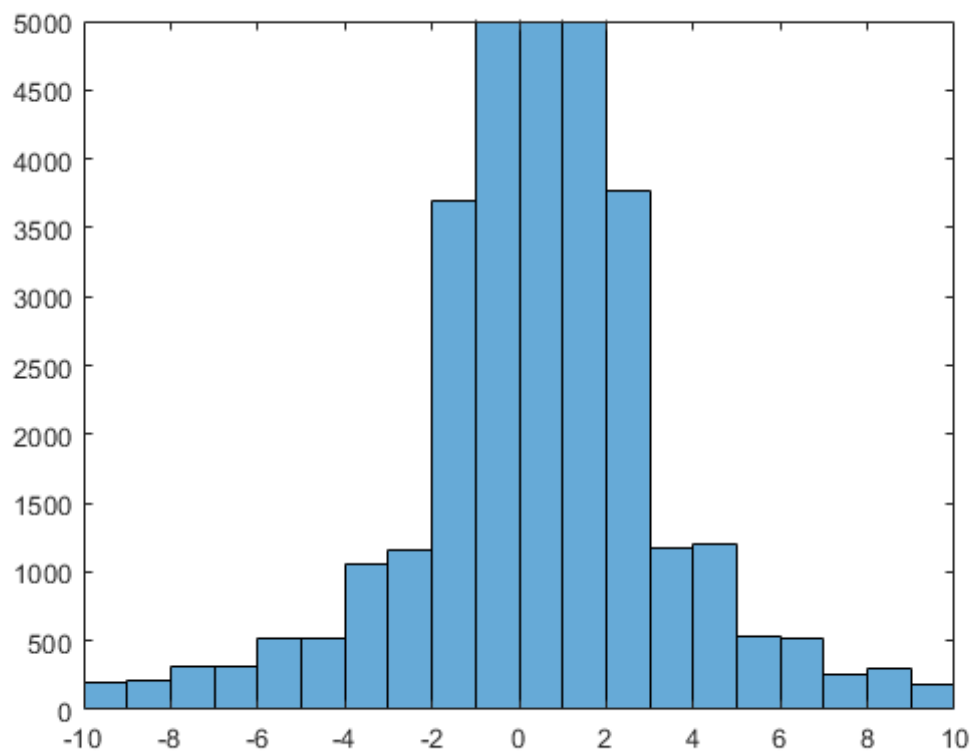
使用 `histogram()` 绘制原图 DCT 系数的直方图, 限制 x 轴为 $[-10, 10]$, y 轴为 $[0, 5000]$ (忽略 ± 1 和 0 的高度), 得到的直方图如下。



对原图进行 F3 隐写, 所得图片如下:



使用 `histogram()` 绘制 F3 隐写后 DCT 系数的直方图, 限制 x 轴为 $[-10, 10]$, y 轴为 $[0, 5000]$ (忽略 ± 1 和 0 的高度), 得到的直方图如下。

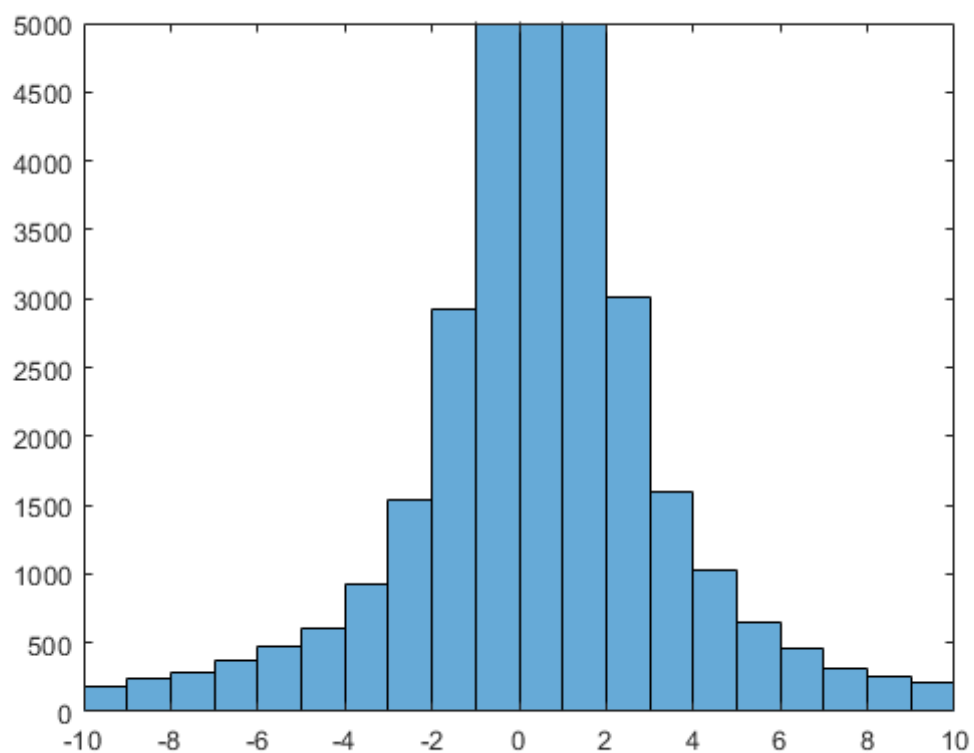


从隐写后的未保存图片读取隐写信息, 准确率为 100%。

对原图进行 F4 隐写, 所得图片如下:



使用 `histogram()` 绘制 F3 隐写后 DCT 系数的直方图, 限制 x 轴为 $[-10, 10]$, y 轴为 $[0, 5000]$ (忽略 ± 1 和 0 的高度), 得到的直方图如下。



从隐写后的未保存图片读取隐写信息, 准确率为 100%。

五、实验分析与结论

5.1 隐写方法对图片及 DCT 系数的影响

肉眼观察图片缩略图和原图差异并不大，但是放大之后可以显著看出植入后图片是以 8×8 块为分割的（可能是因为只有灰度通道，分块比较明显）。

使用 F3 和 F4 隐写术时，与原图对比，可见系数都往 0 靠拢。

使用 F3 隐写术时，明显可见偶数系数频率有异常升高，许多偶数系数频率与相对应的奇数系数频率持平，甚至更高。

使用 F4 隐写术时基本能保证与原图 DCT 系数直方图的相似，在一阶统计规律上难以分辨。

5.2 隐写系统与数字水印系统的异同

相同点：

隐写系统与水印系统都是往多媒体数据（如图像）中插入一些信息；

两者都尽可能让图像受到最小扰动。

不同点：

水印系统起的是版权保护作用，而隐写系统起到隐秘通信的作用；

水印需要能够被公共检测，而隐写目的就是不被公共发现地进行通信；

水印是基于图像的，一般是根据图像来选取水印。而隐写术是基于信息的，一般根据信息来选取图像；

等等。

六、实验感想

本次实验较为简单，也没有什么问题。唯一的问题是如果把图像保存后再读取则信息准确率会有较大下滑，不过这也是信道上的问题，和算法无关。

通过本实验，我理解并掌握了简单隐写术的基本方法及原理，有助于后续对课程的学习。