

武汉大学国家网络安全学院

# 信息隐藏实验报告

学 号 2021302181156

姓 名 赵伯侯

实验名称 基本图像信息隐藏方法实践

指导教师 任延珍

# 一、实验名称: 基本图像信息隐藏方法实践

## 二、实验目的:

- 1. 学习 matlab 的基本操作。
- 2. 实现图像的 LSB 隐写和隐藏信息的提取。
- 3. 使用两点法实现图像的 DCT 域隐写与隐写信息的提取。

## 三、实验原理:

### (一) LSB 隐写

将我们选取的像素点的最不重要位依次替换成秘密信息, 以达到信息隐秘的目的。嵌入过程包括选择一个图像载体像素点的子集, 然后在子集上执行替换操作像素把整个图像的 LSB 与秘密信息  $mi$  进行交换 ( $mi$  是秘密信息的二进制表示)。一个替换系统也可以修改载体图像像素点的多个比特。

在本次实验中选取一张图片提取出其图像二进制编码的最后一位替换为秘密信息进行信息隐藏。原理图如下所示。

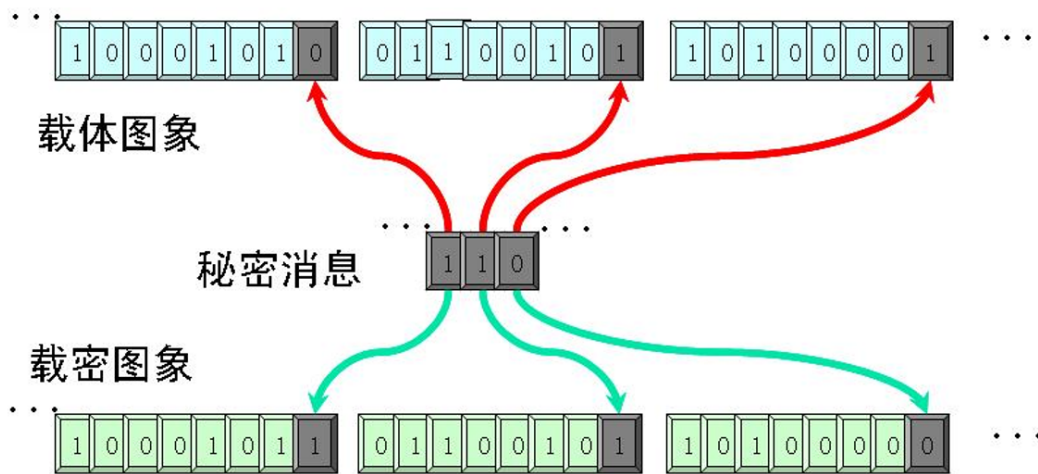


图 1: LSB 隐写原理

### (二) DCT 变换对图像进行压缩

在 DCT 变换中使用如下数学公式对图像进行离散余弦变换, 在压缩过后的图像中低频部分往往保存了图像的大部分信息, 高频部分保存的信息往往较少, 因此

可以通过删除图像的部分高频信息的方式对图像的大小进行压缩但不较大幅度改变图像质量。

## 二维离散余弦变换

$$F(u, v) = \frac{1}{\sqrt{MN}} c(u)c(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{\pi u(2x+1)}{2M}\right] \cos\left[\frac{\pi v(2y+1)}{2N}\right]$$

$$u = 0, 1, 2, \dots, M-1 \quad v = 0, 1, 2, \dots, N-1$$

## 逆变换

$$f(x, y) = \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} c(u)c(v) F(u, v) \cos\left[\frac{\pi u(2x+1)}{2M}\right] \cos\left[\frac{\pi v(2y+1)}{2N}\right]$$

$$x = 0, 1, 2, \dots, M-1 \quad y = 0, 1, 2, \dots, N-1$$

$$c(u) = \begin{cases} 1 & u = 0 \\ \sqrt{2} & u \neq 0 \end{cases} \quad c(v) = \begin{cases} 1 & v = 0 \\ \sqrt{2} & v \neq 0 \end{cases}$$

图 2: DCT 压缩原理

### (三) DCT 域隐写

对一幅图像进行离散余弦变换后,许多有关图像的重要可视信息都集中在 DCT 变换的一小部分系数中。可以将秘密信息隐藏在变换后的图像之中。在本次实验中采用两点法进行信息的隐藏,选取两个点,利用载体中两个特定数的相对大小来代表隐藏的信息。

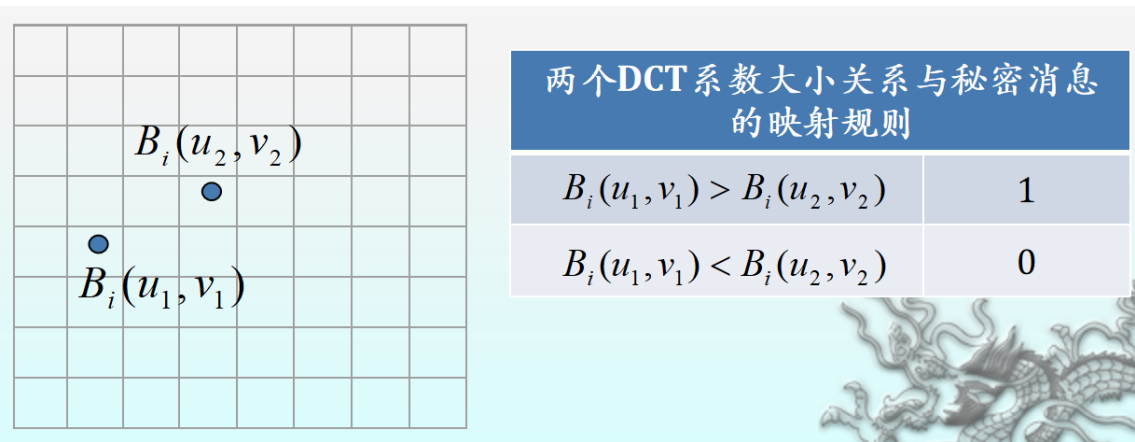


图 3: DCT 隐写原理

## 四、实验内容:

### (一) 实验一:LSB 隐写

LSB 信息隐藏的代码如下所示

```
1 %randlsbhide ( 'Lena.bmp', 'm.txt', 'scover.bmp', 2001 )
2 function [ ste_cover, len_total] = randlsbhide( input, file, output, key)
3 % 读入 图像矩阵
4 cover = imread( input) ;
5 ste_cover = cover;
6 ste_cover = double( ste_cover) ;
7 % 将文本文件转换为二进制
8 f_id = fopen( file, "r") ;
9 [ msg,len_total] = fread( f_id, 'ubitl' ) ;
10 % 判断嵌入消息量是否过大
11 [ m, n] = size( ste_cover) ;
12 if len_total > m*n
13     error( "嵌入消息量过大，请更换图像") ;
14 end
15 % p 表示当前隐写到的bit位数
16 p = 1;
17 % 调用随机间隔函数选取像素点
18 [ row, col] = randinterval( ste_cover, len_total, key) ;
19
20 % 在 LSB 隐秘消息
21 for i = 1:len_total
22     ste_cover(row( i) , col( i) ) = ste_cover(row(i) , col(i))-mod( ste_cover(
23         row(i) , col(i)),2)+ msg(p,1) ;
24     if p == len_total
25         break;
26     end
27 end
```

```

26     p = p + 1;
27 end
28
29 % 隐写后的图像输出
30 ste_cover = uint8( ste_cover) ;
31 imwrite( ste_cover, output) ;
32
33 % 显示实验结果
34 subplot( 1, 2, 1) ; imshow( cover) ; title( "原始图像") ;
35 subplot( 1, 2, 2) ; imshow( output) ; title("隐藏信息的图像") ;

```

代码 1: LSB 隐写代码

LSB 信息提取的代码如下图所示

```

1 %randlsbget( 'scover.bmp', 168, 'secret.txt', 2001)
2 function result = randlsbget( output, len_total, goalfile, key)
3 ste_cover = imread( output) ;
4 ste_cover = double( ste_cover) ;
5
6 frr = fopen( goalfile, 'a') ;
7 % p 代表提取进度
8 p = 1;
9 % 调用随机间隔函数选取像素点
10 [ row, col] = randinterval( ste_cover, len_total, key);
11 for i = 1:len_total
12     if bitand( ste_cover( row( i) , col( i) ) , 1) == 1
13         fwrite( frr, 1, 'ubit1' ) ;
14         result( p, 1 ) = 1;
15     else
16         fwrite( frr, 0, 'ubit1' ) ;
17         result( p, 1 ) = 0;

```

```

18     end
19     if p == len_total
20         break;
21     end
22     p = p + 1;
23 end
24 fclose( frr ) ;

```

代码 2: LSB 信息提取代码

调用的随机间隔函数选取像素点代码如下所示

```

1 function [row,col]=randinterval(matrix,count,key)
2 %计算间隔的位数
3 [m,n]=size(matrix);
4 interval1=floor(m*n/count)+1;
5 interval2= interval1-2;
6 if interval2==0
7     error('载体大小不能将秘密信息隐藏进去!');
8 end
9 %生成随机序列
10 rand('seed',key);
11 a=rand(1,count);
12 %计算row和col
13 r=1;
14 c=1;
15 row(1,1)=r;
16 col(1,1)=c;
17 for i=2:count
18     if a(i)>=0.5
19         c=c+interval1;
20     else

```

```

21         c=c+ interval2;
22     end
23     if c>n
24         r=r+1;
25     if r>m
26         error('载体大小 不能将秘密信息隐藏进去!');
27     end
28     c=mod(c,n);
29     if c==0
30         c=1;
31     end
32     end
33 row(1,i)=r;
34 col(1,i)=c;
35 end

```

代码 3: 随机间隔函数代码

## (二) 实验二: DCT 变换对图像进行压缩

DCT 变换对图像进行压缩的代码如下所示

```

1 %comimage = dctcom('Lena.bmp','bmp')
2 function comimage = dctcom( image, permission)
3 f = imread (image, permission) ;
4 f = double( f) /255;
5 T = dctmtx( 8) ;
6 B = blkproc( f, [ 8 8 ] , 'PI*x*P2', T, T') ;
7
8 %删除 每块中的 高频部分
9 mask = [1 1 1 1 0 0 0 0
10         1 1 1 0 0 0 0 0

```

```

11         1 1 0 0 0 0 0 0
12         1 0 0 0 0 0 0 0
13         0 0 0 0 0 0 0 0
14         0 0 0 0 0 0 0 0
15         0 0 0 0 0 0 0 0
16         0 0 0 0 0 0 0 0 ];
17
18 B2 = blkproc( B, [8 8] , 'P1.*x', mask) ;
19 I2 = blkproc( B2, [8 8] , 'P1*x*P2', T', T) ;
20
21 subplot(121) , imshow(f) ; title( '原始图像' ) ;
22 subplot(122) , imshow(I2) ; title( '逆DCT变换后的图像' ) ;

```

代码 4: DCT 变换图像压缩代码

### (三) 实验三: DCT 域隐写

在 DCT 域进行隐写的代码如下所示

```

1 function [count_false] = dchide2(a)
2
3 count_false=0;
4 A=imread('lena.png');
5 R=A(:,:,1);
6 G=A(:,:,2);
7 B=A(:,:,3);
8 f_id = fopen('m.txt', "r") ;
9 [ msg,len_total] = fread( f_id, 'ubit1' ) ;
10
11 %隐 写
12 I = im2double(R);
13 D = dctmtx(8);%生成变换矩阵

```



```

14 C = blkproc(I,[8,8], 'P1*x*P2',D,D');
15 mask1=[16 11 10 16 24 40 51 61
16         12 12 14 19 26 58 60 55
17         14 13 16 24 40 57 69 56
18         14 17 22 29 51 87 80 62
19         18 22 37 56 68 109 103 77
20         24 35 55 64 81 104 113 92
21         49 64 78 87 103 121 120 101
22         72 92 95 98 112 100 103 99];
23
24 C = blkproc(C,[8,8], 'x./P1',mask1);%量化
25 [m,n]=size(C);
26 u=[5 2];%[2 3]
27 v=[4 3];%[4 1]
28
29 x=1; %当前处理的字符指针
30 len_total; %隐写信息长度
31 capacity=m*n/64;
32
33 if len_total>capacity
34     disp('隐藏信息过多');
35     exit;
36 end
37
38 %隐藏
39 for i=1:8:m
40     for j=1:8:n
41         if x<=len_total
42             if msg(x)==1 %隐写信息1 调整v的部分比u大一个a

```

```

43         if  $C(i+u(1)-1, j+u(2)-1) > C(i+v(1)-1, j+v(2)-1)$ 
44              $medium = C(i+u(1)-1, j+u(2)-1) + a;$ 
45              $C(i+u(1)-1, j+u(2)-1) = C(i+v(1)-1, j+v(2)-1);$ 
46              $C(i+v(1)-1, j+v(2)-1) = medium;$ 
47              $x = x + 1;$ 
48         elseif  $C(i+u(1)-1, j+u(2)-1) = C(i+v(1)-1, j+v(2)-1)$ 
49              $C(i+v(1)-1, j+v(2)-1) = C(i+v(1)-1, j+v(2)-1) + a;$ 
50              $x = x + 1;$ 
51         else
52              $x = x + 1;$ 
53             continue;
54         end
55     elseif  $msg(x) == 0$     %隐写信息0
56         if  $C(i+u(1)-1, j+u(2)-1) < C(i+v(1)-1, j+v(2)-1)$ 
57              $medium = C(i+u(1)-1, j+u(2)-1);$ 
58              $C(i+u(1)-1, j+u(2)-1) = C(i+v(1)-1, j+v(2)-1) + a;$ 
59              $C(i+v(1)-1, j+v(2)-1) = medium;$ 
60              $x = x + 1;$ 
61         elseif  $C(i+u(1)-1, j+u(2)-1) = C(i+v(1)-1, j+v(2)-1)$ 
62              $C(i+u(1)-1, j+u(2)-1) = C(i+u(1)-1, j+u(2)-1) + a;$ 
63              $x = x + 1;$ 
64         else
65              $x = x + 1;$ 
66             continue;
67         end
68     end
69 else
70     break;
71 end

```

```

72     end
73 end
74 C = blkproc(C,[8,8], 'x.*P1',mask1);
75 %图像重构
76 C = blkproc(C,[8,8], 'P1*x*P2',D',D);
77 figure(1)
78 C=im2uint8(C);
79 Mix=cat(3,C,G,B);
80
81 imwrite(C,"scover.bmp");
82 subplot(1,2,1),imshow(A),title('原图');
83 subplot(1,2,2),imshow(Mix),title('嵌入信息');
84
85 %提取
86 R=Mix(:,:,1);
87 G=Mix(:,:,2);
88 B=Mix(:,:,3);
89 img=im2double(R);
90 img = blkproc(img,[8,8], 'P1*x*P2',D,D');
91 s=1; %当前提取到的bit个数
92 frr = fopen("result.txt", 'a');
93
94 for i=1:8:m
95     for j=1:8:n
96         if s<=len_total
97             if img(i+u(1)-1,j+u(2)-1)>img(i+v(1)-1,j+v(2)-1)
98                 fwrite(frr, 0, 'ubit1') ;
99                 result_msg(s)=0;
100                 s=s+1;

```

```

101         elseif img(i+u(1)-1,j+u(2)-1)<img(i+v(1)-1,j+v(2)-1)
102             fwrite( frr , 1, 'ubit1' ) ;
103             result_msg(s)=1;
104             s=s+1;
105         end
106     else
107         break;
108     end
109 end
110 end
111 fclose(frr) ;
112
113 %输出误码率
114 for i=1:len_total
115     if result_msg(i) ~= msg(i)
116         count_false=count_false+1;
117     end
118 end

```

代码 5: DCT 域隐写与信息提取算法

对误码率进行分析的代码如下图所示

```

1 A=[0.1,0.01,0.001,0.0005,0.0004,0.0001,0.00001];
2 [x,len]=size(A);
3 result=zeros(1, len);
4 for i=1:len
5     result(i)=dcthide2(A(i));
6 end
7 resultt=result./504;

```

代码 6: DCT 域隐写误码率分析函数

## 五、实验环境:

1.win11 操作系统

2.MATLAB R2022a

## 六、实验步骤:

### (一) 实验一:LSB 隐写

#### 6.1.1 设计随机取点的算法, 随机选取像素点嵌入秘密信息

首先先将保存有秘密信息的文本文件转换成二进制文件存入数组 `msg` 中, 并将其长度保存在变量 `len_total` 中。然后调用随机间隔函数使其按照提供的密钥选取一系列的随机点坐标, 然后针对密文的每一个字符 `msg(p,1)`, 选定其要保存的像素块 `ste_cover(row(i),col(i))`, 将该像素块的二进制最后一位清零然后加上 `msg(p,1)`, 将所有的密文隐藏入图像中之后将图片输出为 `scover.bmp` 的形式。

#### 6.1.2 提取秘密信息

输入隐藏有密文的图像和密文长度之后提取图像 LSB 的最后一位二进制信息, 逐个随机点坐标进行提取即可提取到密文长度数量的二进制信息, 将信息输出成为一个保存有秘密信息的 `txt` 文件。

#### 6.1.3 画出随机位置

编写 `compare` 函数求经过 LSB 隐写后的文件与原图像的差值, 将差值作为图像绘制出来就能够得到由密钥 `key` 生成的所有的随机位置。`compare` 比较函数的代码如下所示。

```
1 function F = compare( original, hided)
2 % 读取原始载体图像矩阵
3 W = imread( original) ;
4 W = double( W) /255;
5 % 读取隐秘后图像矩阵
```

```

6 E = imread( hided) ;
7 E = double( E) /255;
8 % 将两图像矩阵相减，显示效果
9 F = E-W; % 注意，MATLAB 中矩阵相减只支持 double 型
10 imshow(mat2gray( F) );

```

代码 7: 绘制所有随机位置的函数代码

#### 6.1.4 对比隐写前后图像直方图，分析 LSB 隐写导致的值对效应

编写绘制加密前后图像的直方图的函数对图像的值对效应进行比较，绘制图像直方图的函数如下所示。

```

1 %原图
2 f=imread( 'Lena.bmp');%读入图片
3 [ m, n] = size(f) ;
4 result_1= zeros(1, 256);
5 for i=1:m
6     for j=1:n
7         for k=1:256
8             if f(i,j)-1==k
9                 result_1(k)=result_1(k)+1;
10            end
11        end
12    end
13 end
14
15 result=0;
16 for x=1:255
17     y=x+1;
18     result=result+abs(result_1(x)-result_1(y));

```

```

19 end
20 subplot(211),imhist(f),title("隐写之前的文件");
21 axis([0 255 0 6000]);%设置x、y轴坐标范围
22
23 %隐写后图像
24 f2=imread('scover.bmp');%读入图片
25 result_2= zeros(1, 256);
26 for i=1:m
27     for j=1:n
28         for k=1:256
29             if f2(i,j)-1==k
30                 result_2(k)=result_2(k)+1;
31             end
32         end
33     end
34 end
35
36 result2=0;
37 for x=1:255
38     y=x+1;
39     result2=result2+abs(result_2(x)-result_2(y));
40 end
41
42 subplot(212),imhist(f2),title("隐写之后的文件");
43 axis([0 255 0 6000]);%设置x、y轴坐标范围

```

代码 8: 绘制图像直方图函数代码

## (二) 实验二：DCT 变换对图像进行压缩

### 6.2.1 图像分块

首先将图像进行分块处理, 在本次实验中, 将整个图像分成了  $8*8$  大小的块分别进行处理

### 6.2.2 删除部分 DCT 系数

设置矩阵 `mask` 对 DCT 系数进行量化, 将矩阵的左上角部分置为 1, 其余部分置为 0, 即保留原图像的低频部分, 删除掉图像的高频部分

### 6.2.3 逆 DCT 变换

将经过 DCT 变换后逆 DCT 变换的块拼到一起后形成操作后的图像, 将原图和操作后的图像进行输出比较。

## (三) 实验三：DCT 域隐写

### 6.3.1 图像预处理

提取 RGB 图像的红色位所有数据作为信息隐藏的载体, 以满足对于所有的信息载体都能够处理的要求。保留原图像的其他位的数据,

### 6.3.2 图像分块后做二维 DCT 变换

将图像进行分块后得到  $8*8$  的分块, 设计量化矩阵, 将矩阵的左上角数值设置得尽可能小, 右下角数值尽可能大, 以实现尽可能多的保存图像的低频数据对图像实行二维 DCT 变换。

### 6.3.3 隐藏信息

对于第  $x$  个 bit 的信息, 如果要隐藏的信息为 1, 则将  $C(i+u(1)-1, j+u(2)-1)$  设置为大于  $C(i+v(1)-1, j+v(2)-1)$ , 其中  $C(i+u(1)-1, j+u(2)-1)$  和  $C(i+v(1)-1, j+v(2)-1)$  表示在该分块下约定好的两块的位置。



#### 6.3.4 逆 DCT 变换，还原图像

将隐藏好信息的块合并到一起形成完整的原图红色色域的图像，将原图其他两个色域的图像与其叠加形成三维 RGB 图像，将原图和经过处理的图像进行输出和保存操作。

七、实验结果与分析:

(一) 实验一:LSB 隐写

• 隐写结果 LSB 隐写之后的图片和原图的对比如下图所示，通过肉眼比较几乎不能够看出其中的差异。

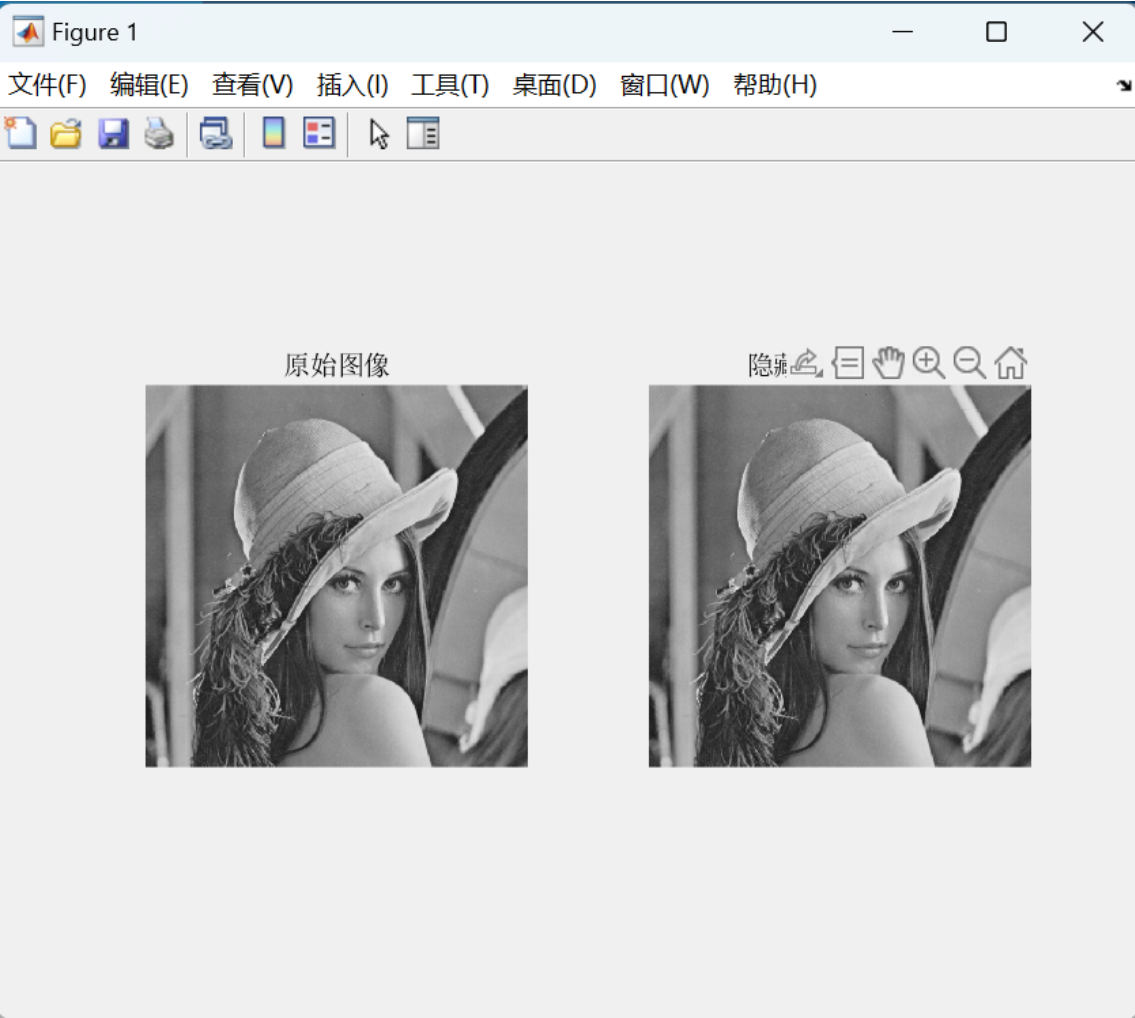


图 4: LSB 隐写前后图像

- 画出随机位置的结果运行比较函数比较原图和 LSB 隐写之后的图片结果如下图所示，可以看到在图像中存在一些较为明显的白色像素点和黑色像素点。

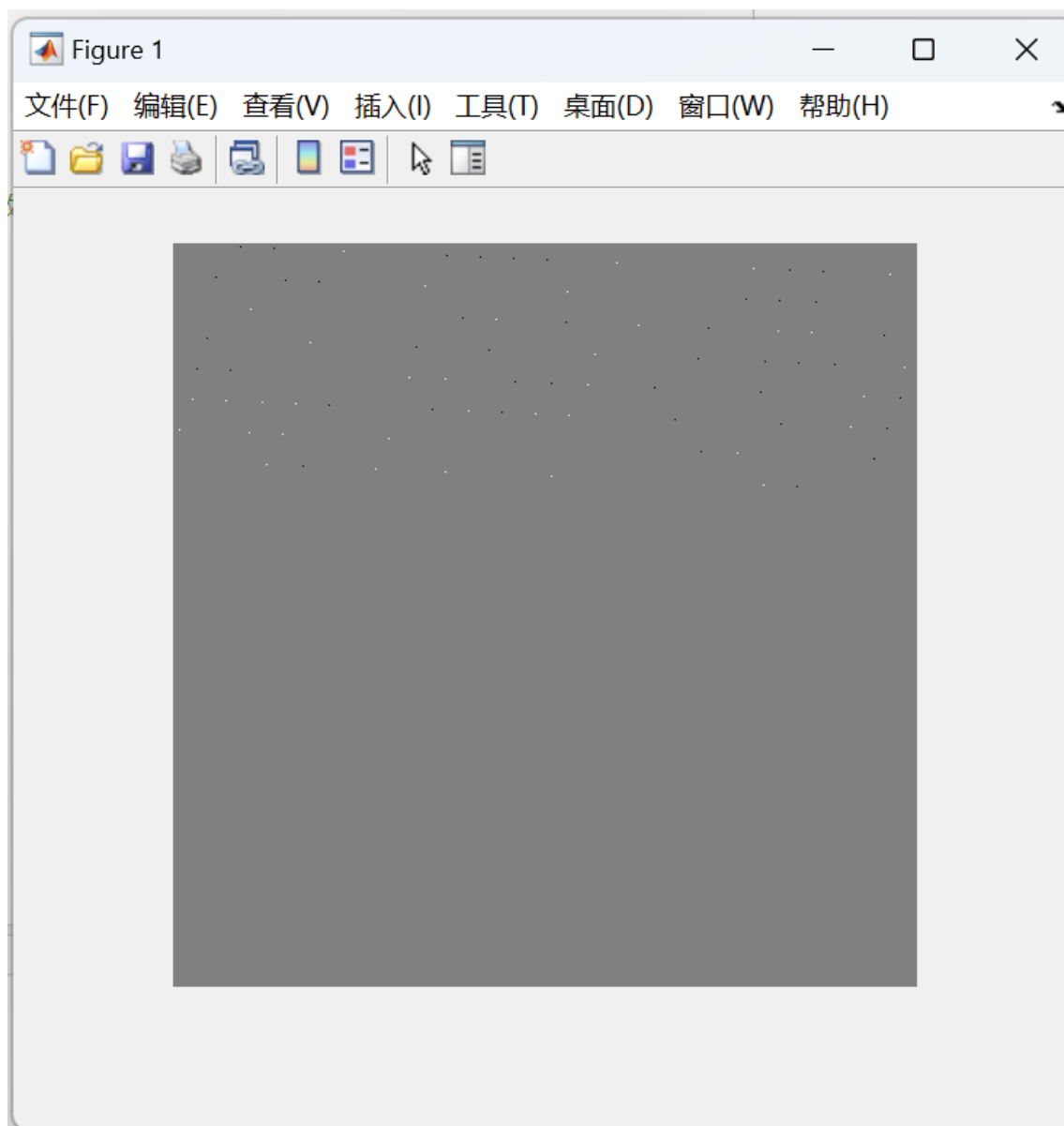


图 5: 生成的所有的随机位置

- 画出隐写前后的图像直方图，分析值对效应在运行函数 `draw` 之后生成原图和隐写后图像的直方图，观察两张图片的值对效应

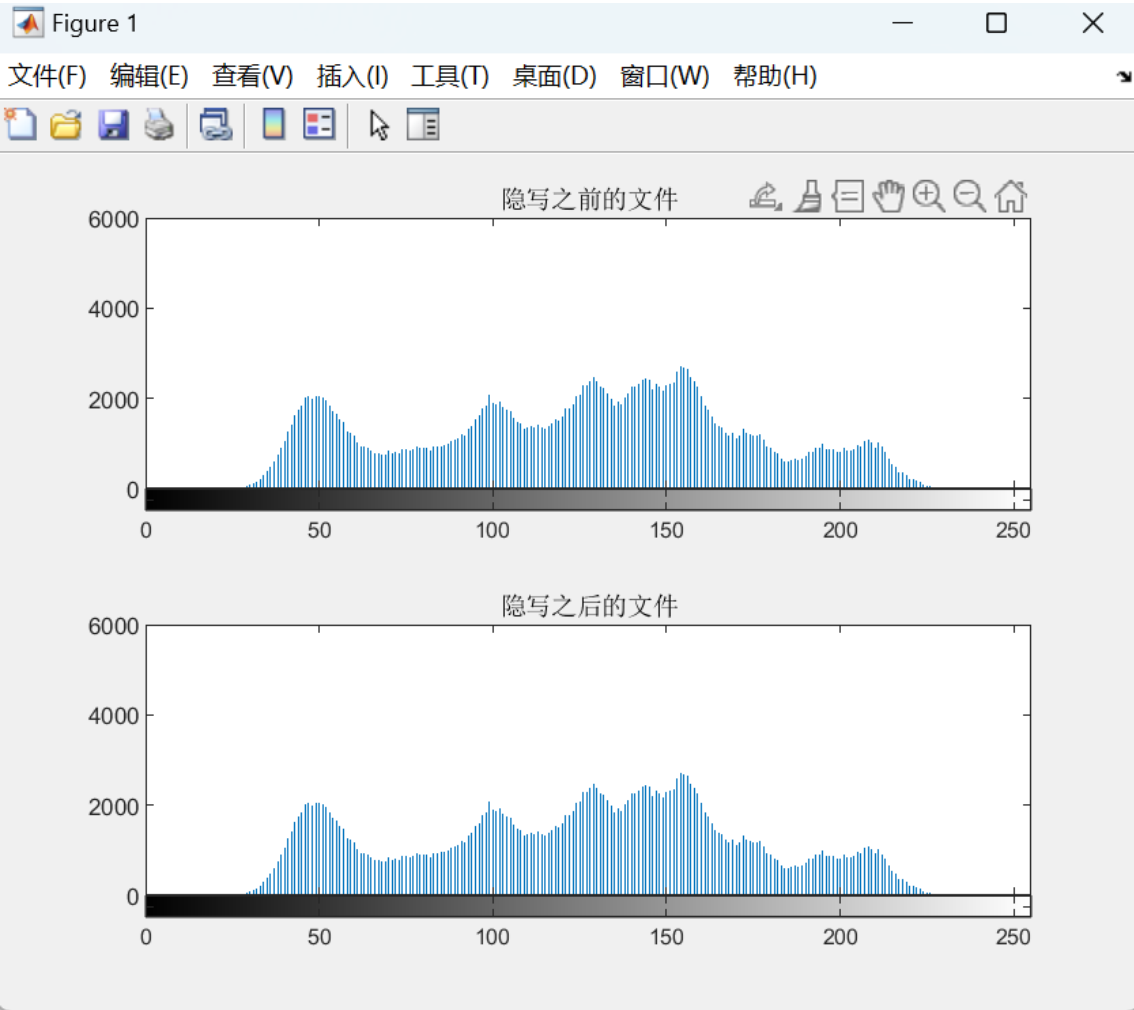


图 6: 隐写前后图像直方图

通过肉眼观察很难发现两个直方图中的值对效应，所以在代码中对每一个图像的像素值进行了统计，统计方法为将像素值为  $i$  的像素个数与像素值为  $i+1$  的像素值做差取绝对值后求整个图像的总和。对隐写前后的统计结果分析如下表所示

表 1: 隐写前后图像值对效应统计

图像	值对差值总和
原图	15288
隐写后图像	15282

由此可见隐写后的图像值对插值总和相对较低，具有较强的值对效应

## (二) 实验二：DCT 变换对图像进行压缩

- 图像压缩前后的结果如下图所示

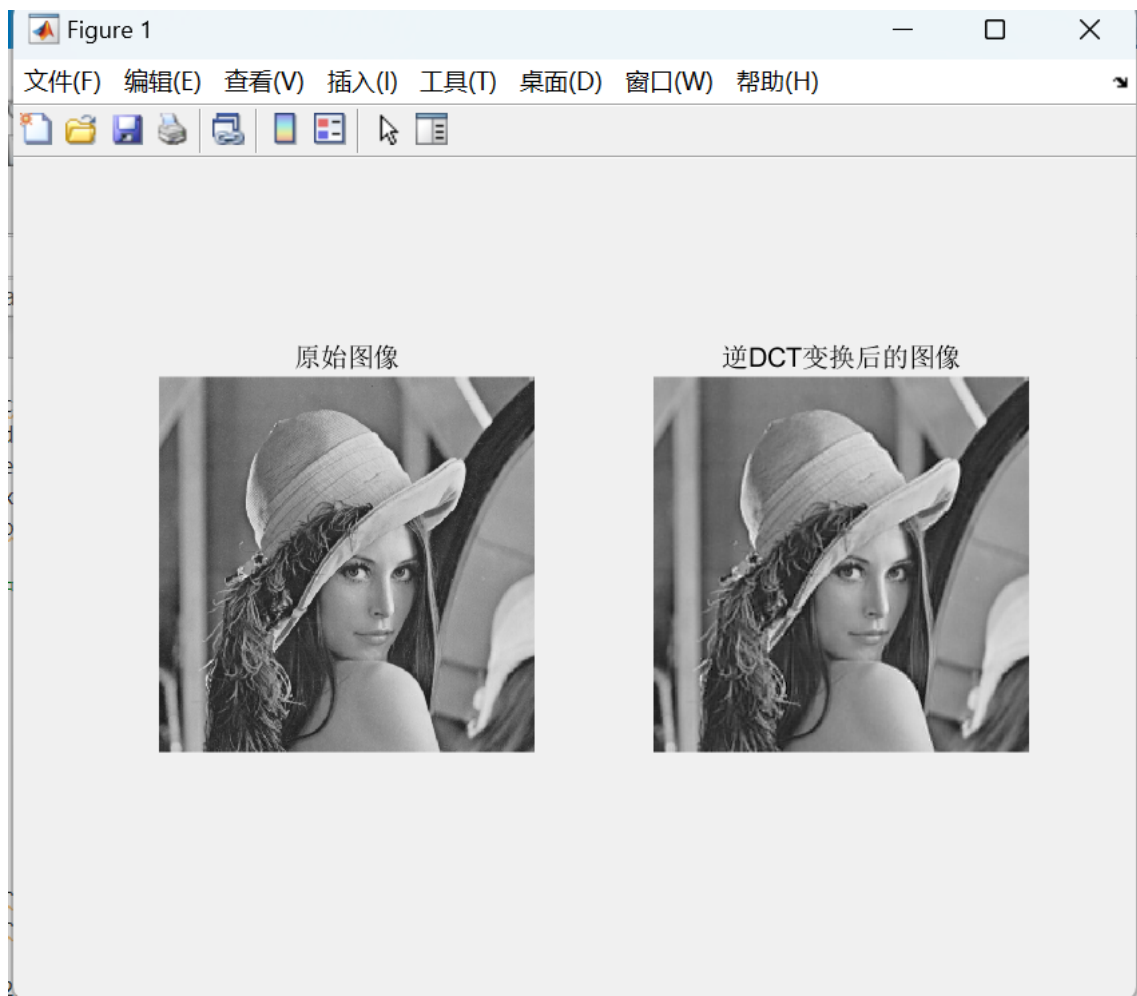


图 7: DCT 变换前后图像对比

- 结果分析

通过对 DCT 压缩前后的图像的视觉质量进行分析可以看出即使删除了图像 DCT 变换后的高频系数，在肉眼观察上仍然不能够发现两张图片的差异。

### (三) 实验三：DCT 域隐写

- DCT 域隐写结果

DCT 隐写后的图像和原图如下图所示，经过肉眼观察并不能够发现其中存在的变化。

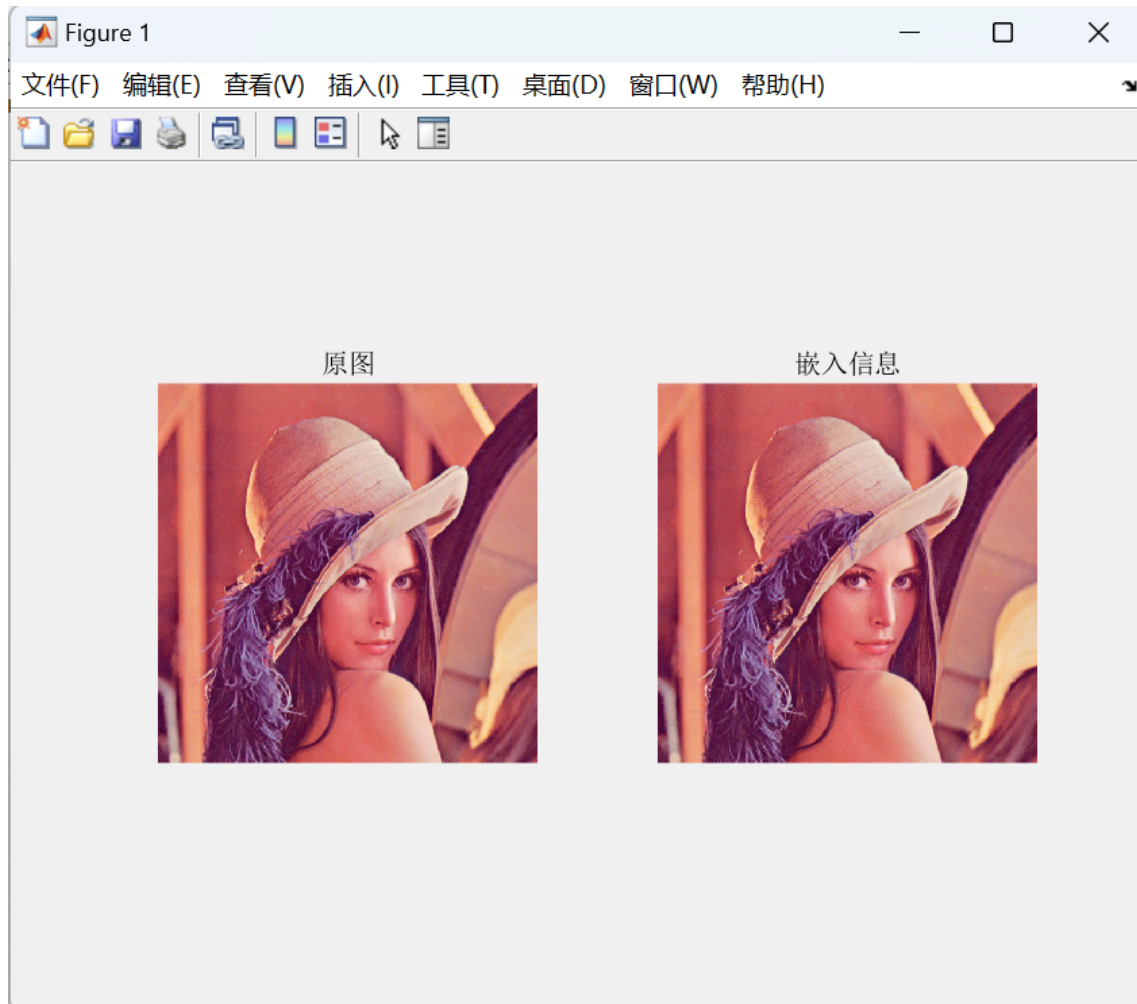


图 8: DCT 隐写前后图像

• DCT 数据提取结果

DCT 数据提取结果与原始秘密信息的对比如下图所示。

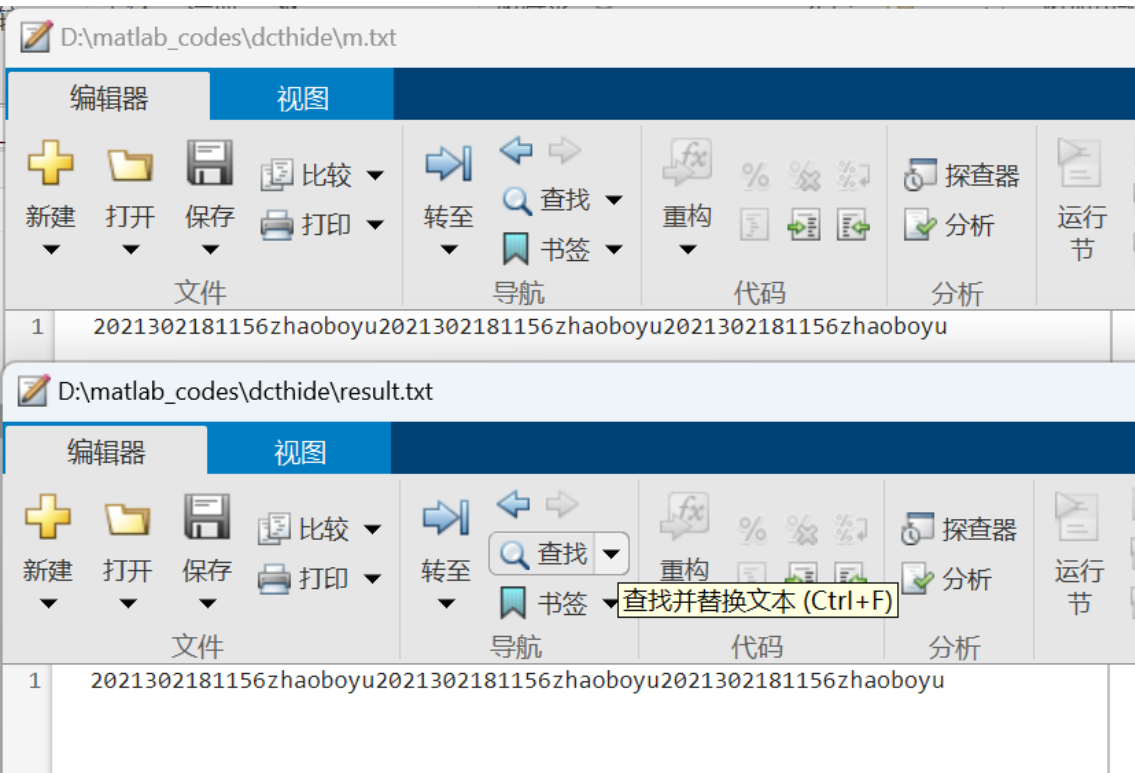


图 9: DCT 提取结果

• 分析健壮参数和鲁棒性运行健壮参数和鲁棒性分析代码后的结果如下图所示：

表 2: 分析健壮参数和鲁棒性

健壮参数	误码个数	误码率
0.1	0	0.00%
0.01	0	0.00%
0.001	0	0.00%
0.0005	2	0.40%
0.0004	10	1.98%
0.0001	59	11.70%
0.00001	75	14.88%

经过上表的结果可以看出健壮参数的值越小，提取信息的误码率就越高，图像的鲁棒性就越差。

## 八、总结及心得体会:

1. 在分析 DCT 隐写的鲁棒性时仅仅列出了实验表格，没有将实验数据化成直方图的形式。
2. 实验中仅对一张 Lena 图片进行了操作，没有针对更多其他格式其它类型的图片进行操作。
3. 实验代码运行的过程中没能处理好函数间的关系，导致写出的函数可观性较差，运行较为复杂，不能直接运行