

**课 程 实 验 报 告**

**课程名称： 多媒体数据安全实验**

**专业班级： 信安1901**

**学 号：**

**姓 名： 李文重**

**指导教师： 马晓静**

**报告日期： 2022/4/18**

**网络空间安全学院**

目 录

[1 LSB图像变换域信息隐藏实现 2](#_Toc3487)

[1 实验目的 2](#_Toc13411)

[2实现思路 2](#_Toc2348)

[2.1 LSB算法 2](#_Toc23945)

[2.1.1 算法描述 2](#_Toc7802)

[2.1.2 实现思路 2](#_Toc16126)

[2.1.3实验结果及性能分析 3](#_Toc6943)

[3 实验小结 4](#_Toc22180)

[2 JPEG图像变换域信息隐藏实现 6](#_Toc22926)

[1 实验目的 6](#_Toc211)

[2实现思路 6](#_Toc18857)

[2.1 JSTEG算法 6](#_Toc1832)

[2.1.1 算法描述 6](#_Toc19153)

[2.1.2 实现思路 6](#_Toc32469)

[2.1.3实验结果及性能分析 7](#_Toc19327)

[2.2 F4算法 9](#_Toc14925)

[2.2.1 算法描述 9](#_Toc690)

[2.2.2 实现思路 9](#_Toc29061)

[2.2.3实验结果及性能分析 10](#_Toc31154)

[2.3 F5算法 11](#_Toc7467)

[2.3.1 算法描述 11](#_Toc4339)

[2.3.2 实现思路 11](#_Toc9329)

[2.3.3 实验结果及性能分析 12](#_Toc16678)

[3 实验小结 13](#_Toc14951)

[附录A JPEG图像变换域信息隐藏实现的源程序 15](#_Toc8631)

# 1 LSB图像变换域信息隐藏实现

## 1 实验目的

通过实验达到⑴加深对空域信息隐藏概念、系统的理解；⑵熟悉数字图片格式；(3)掌握MATLAB基础操作。

## 2实现思路

## 2.1 LSB算法

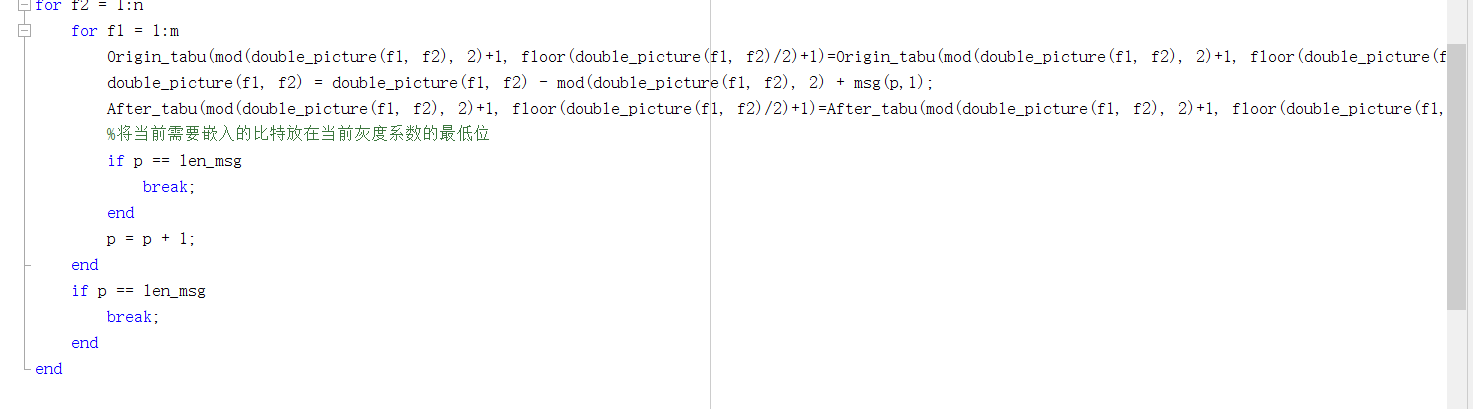
## 2.1.1 算法描述

大多数的嵌入算法都需要首先找到他的嵌入位置。而为了尽可能减少视觉上的影响，LSB选择在灰度图的上选择影响最小的最低比特这个平面上完成嵌入。

LSB空域信息隐藏算法关键步骤是将原始图像最低一个位平面替换为要隐藏的秘密信息。直接把所有像素排成一列，依次嵌入，而因为灰度图的灰度范围是0-255，所以直接通过xor 1 的方法来改变信息。考虑到嵌入信息的字符全部是0-128中的ASCII字符，直接变为比特流进行嵌入

## 2.1.2 实现思路

使用MATLAB的imread直接读入图像，然后获得其大小之后，直接通过两层循环依次枚举像素点，如果最低位不符合就直接xor 1。

图1-1 嵌入循环过程

而解密的时候首先要保证和加密时一样的枚举顺序，并且依次取出最低字节。这里因为我不太会把比特流转化为字符串，所以使用前面读取文件的逆过程，直接通过ubit1 写入文件里面，最后通过\*char型来读取文件获得最终的字符串。

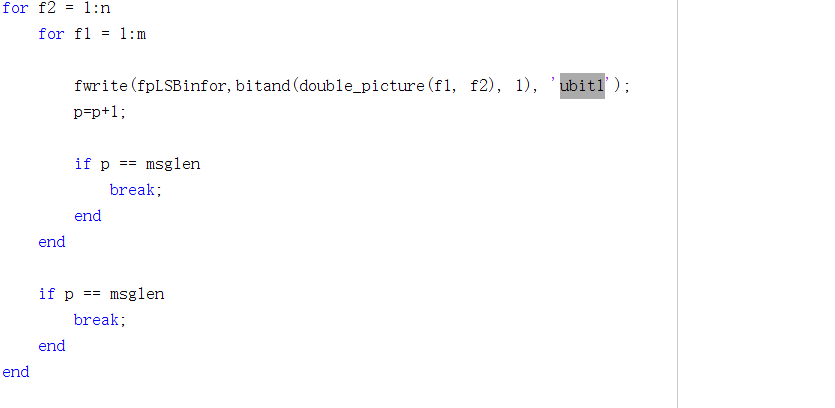


图1-2 提取循环过程

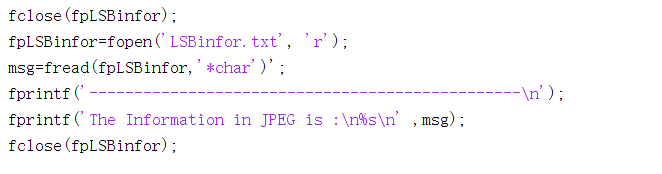


图1-3 获取字符串过程

除了完成嵌入和提取之外，还需要一种方式来体现值对现象，我是采用了matlab中的bar函数来完成，bar是绘制直方图的一个比较好用的工具，而且在了解中，我发现bar可以通过指定'stack'参数将一组数据统合到一个柱上，并且通过不同的染色来区分大小。同时为了进一步的让结果形象起来，还用一个值对中所占的比例绘制了一幅图。

## 2.1.3实验结果及性能分析

测试图选择的是经典的Lena的照片，而为了进一步的展示出值对的效果，采用的是随机数作为嵌入信息。可以看出从人眼观察来看，两幅图片并没有什么不同，而从二，三行的直方图可以看出明显值对出现了趋向于中心 0.5 的现象。

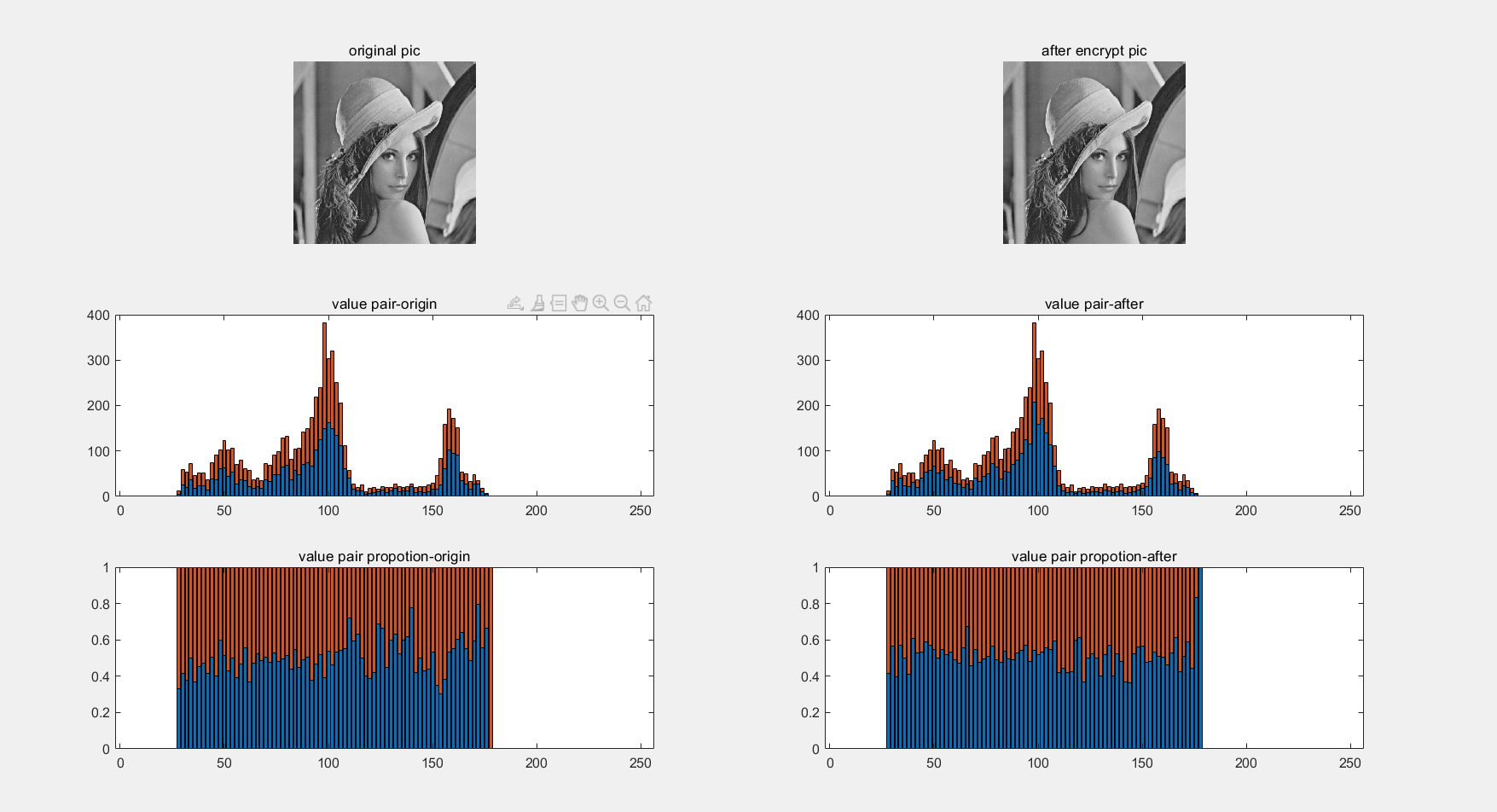


图1-4 嵌入前后图象效果，值对现象对照图

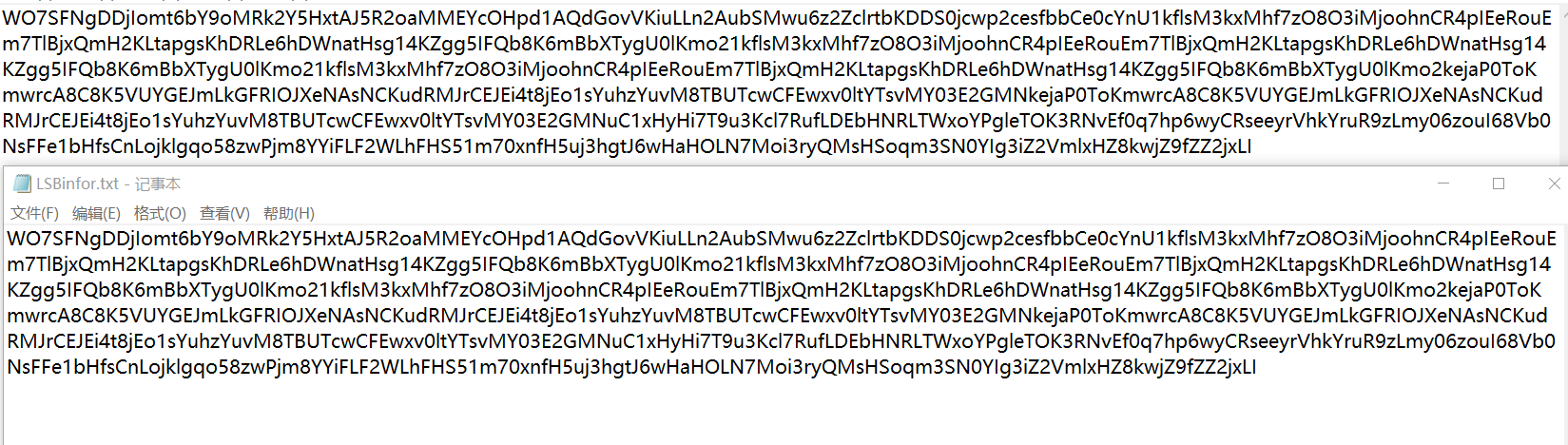


图1-4 嵌入提取信息对照图

可以看出嵌入的信息可以完全的解密出来。而且程序运行速度飞快。

## 3 实验小结

本次实验中不少时间都花在了学习使用matlab上，我在matlab的官网上看到了一个不错的类似闯关的教程，感觉对于我这种使用的不熟练的还是非常友好的。但在具体使用的过程中，还是不可避免地模仿了一些别人的方法。

而且在挑选图片上，也花了不少时间，我首先是试图自己把彩色图片转化为灰度图，但是转化完之后的有明显的失真，并且感觉不够有代表性。于是最终还是选择了Lena的高清图作为嵌入目标。

关于嵌入信息，我个人认为其实是没有必要使用随机数的，可以想见，只要嵌入的信息和图象是概率上独立的，那么他翻转一个值对中数量较多的那个的概率就要大于翻转值对中数量较小的。那么考虑对于每一个嵌入的比特都这样，那么最终值对中出现的频率会趋向于平衡。

# 2 JPEG图像变换域信息隐藏实现

## 1 实验目的

通过实验达到⑴加深对变换域信息隐藏算法原理的理解；⑵熟悉数字图片JPEG压缩格式；(3)比较不同变换域信息隐藏算法在嵌入前后的DCT系数直方图特征。

## 2实现思路

## 2.1 JSTEG算法

## 2.1.1 算法描述

和LSB算法不同，JSTEG的操作都是在DCT域中完成的。首先使用jpeg\_read读入函数，并且把所有的DCT系数读取到MATLAB里面。之后通过其二进制的末尾确定是否能够正确表示需要嵌入的一比特信息（忽略-1，0，1）。如果不符合，则将其变为值对中的另一个。具体的值对如下图：

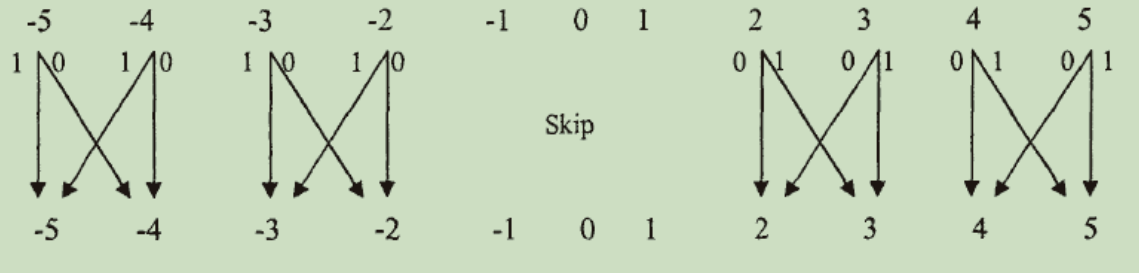


图2-1 JSTEG算法过程

## 2.1.2 实现思路

我设计的整体使用到了一个负责嵌入程序，一个提取程序，还有一个负责展示图像的程序。而因为负责展示图像的程序需要同时展示出嵌入前后的相关数据，所以让他用函数调用的方法来调用嵌入程序，并在执行完成之后通过读取对应的文件制作图表。

提取程序就是直接读取所有合法DCT系数，并且逐一提取出比特信息，并且写入到一个目标的文本文件中。最后以字符串形式读取文件得到信息。

而嵌入程序中，我的思路是嵌入一个提取程序，只有当从原始图像提取出的信息比特不同于需要嵌入的的信息时，才做修改。枚举全部的DCT系数，在跳过绝对值小于等于1的之后，其他的进行信息提取。如果不符合，则使用bitxor(原始值,1)的方式进行翻转。

为了方便作图，我会在嵌入程序中直接把嵌入前后的DCT系数直接保留下来。然后通过其来做分布直方图和相关的值对分析。具体见下一部分。

## 2.1.3实验结果及性能分析

首先展示两幅图片，之后展示所有值出现的频数。

再在频数的基础上进行值对划分和展示，具体见图：

左侧为嵌入前，右侧为嵌入后。

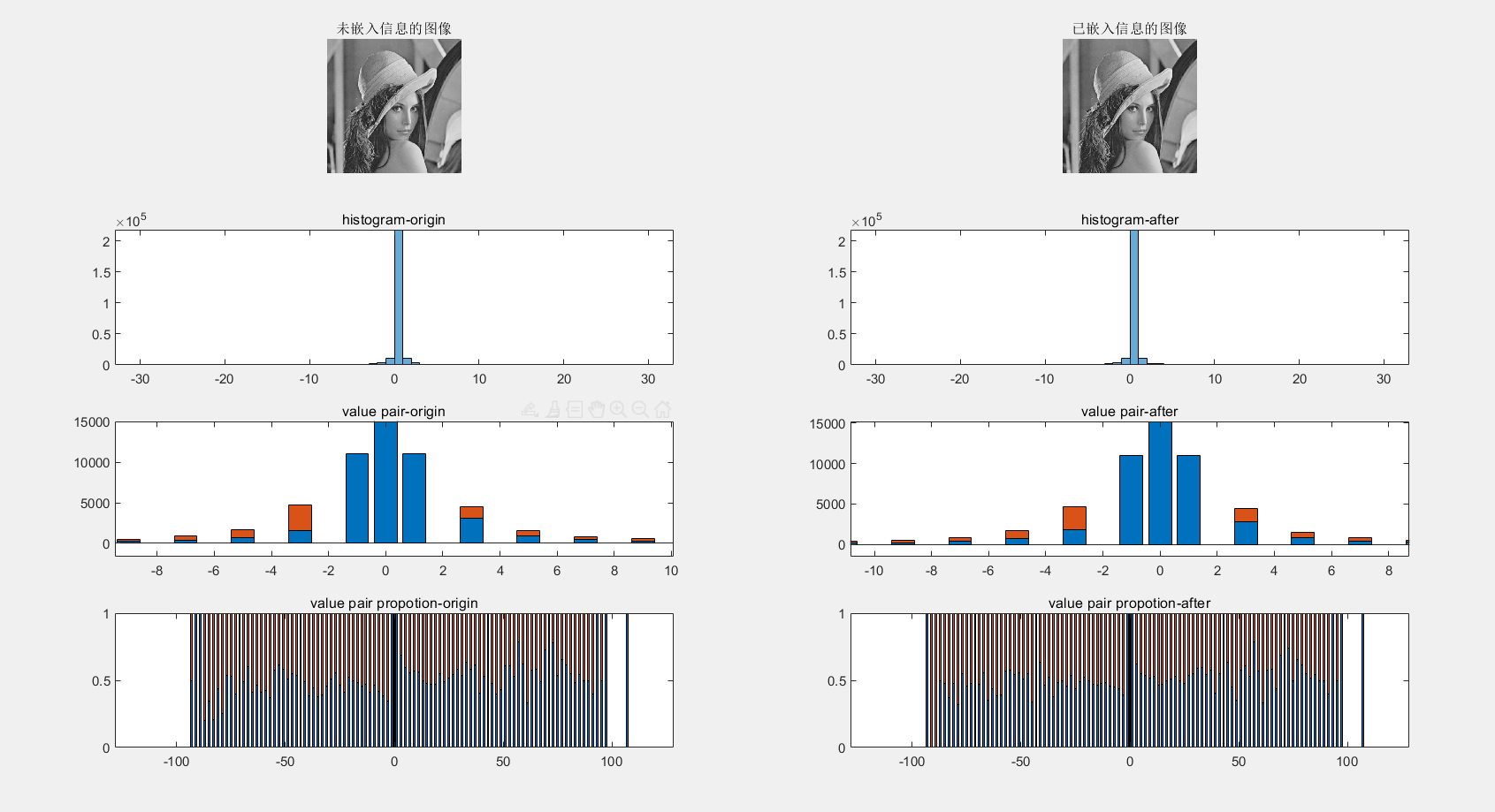


图2-2 基础信息展示

其中第三行的图象为逐值对的分析，蓝色表示其中较小的一个的数量，红色表示较大的一个。

第四行则是进行了标准化，统一使用占值对总数的比例而不是绝对数量来绘制图片。可以看出来存在值对现象，但是并不是非常明显。

于是我又制作了下面这张图，蓝色柱表示对于当前的值对，其嵌入前的值对中的两个频数的差的绝对值大于等于嵌入后的值对中的两个频数的差的绝对值。也就是说体现出了值对现象。红色则表示出现了不符合预期的值对偏离中心。

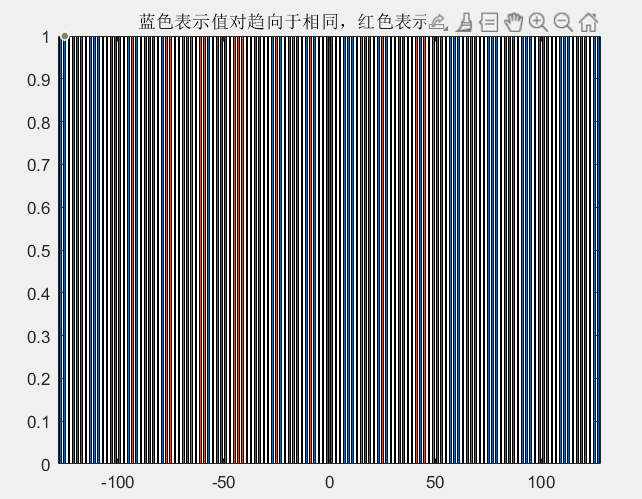


图2-3 值对偏移展示

其具体的数量图下，可以看出来在[-127,127]这个范围内，一共126（129 - {-1,0,1}）个值对中，有103个体现出了值对现象，并有23个不符合，可以认为确实产生了值对现象。

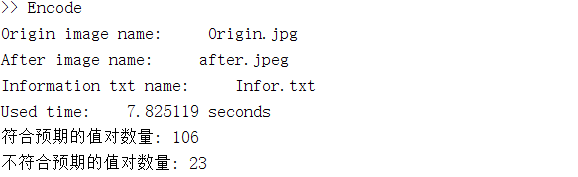


图2-4 值对偏移计数

而在运行效率这一方面，由于我的写法进行了大量的逐比特的文件写入（对前后DCT系数的写），所以程序的运行效率并不是很高，一般进行嵌入要跑5秒以上的时间。而提取因为不需要存储信息，所以效率要快得多，可以在一秒内跑出结果。

## 2.2 F4算法

## 2.2.1 算法描述

F4算法也是DCT域的一种隐写算法，只不过不同于JSTEG使用值对之间转化，F4采用的是每个数向它自身或者绝对值小于他的那个奇偶性不同的数值之间转化，具体如图：

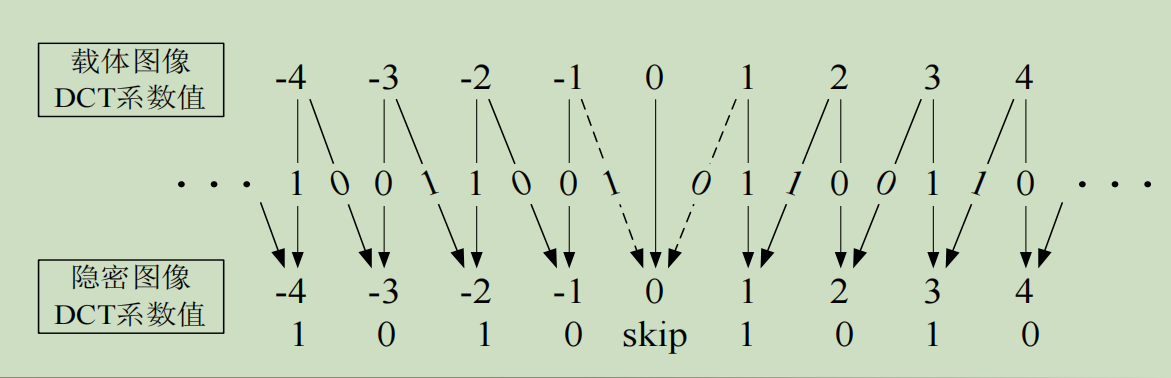


图2-5 F4算法过程

## 2.2.2 实现思路

基本沿用了的前文中的设计，使用三个模块完成任务，因为次算法不存在值对现象，所以就没有后面的值对现象展示。

以下代码是嵌入模块的核心代码：

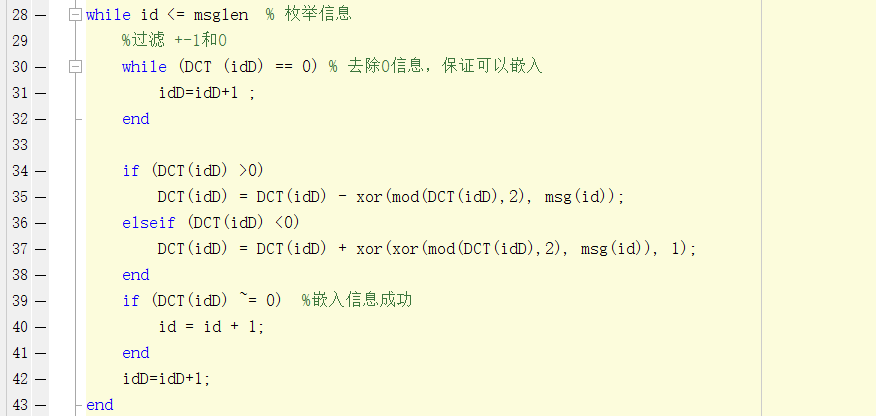


图2-6 F4算法matlab嵌入核心代码

提取过程中，需要分符号讨论，但是可以通过异或来简写。

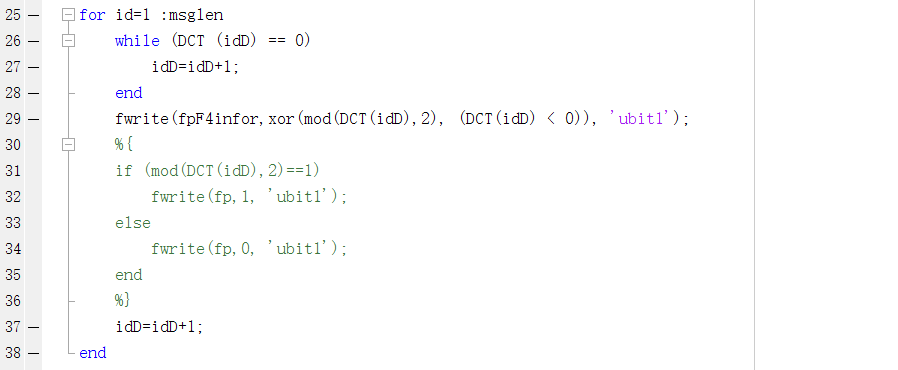


图2-7 F4算法matlab提取核心代码

## 2.2.3实验结果及性能分析

实验效果如图，此处特别展示的是1这个位置的数目在嵌入后有增加：

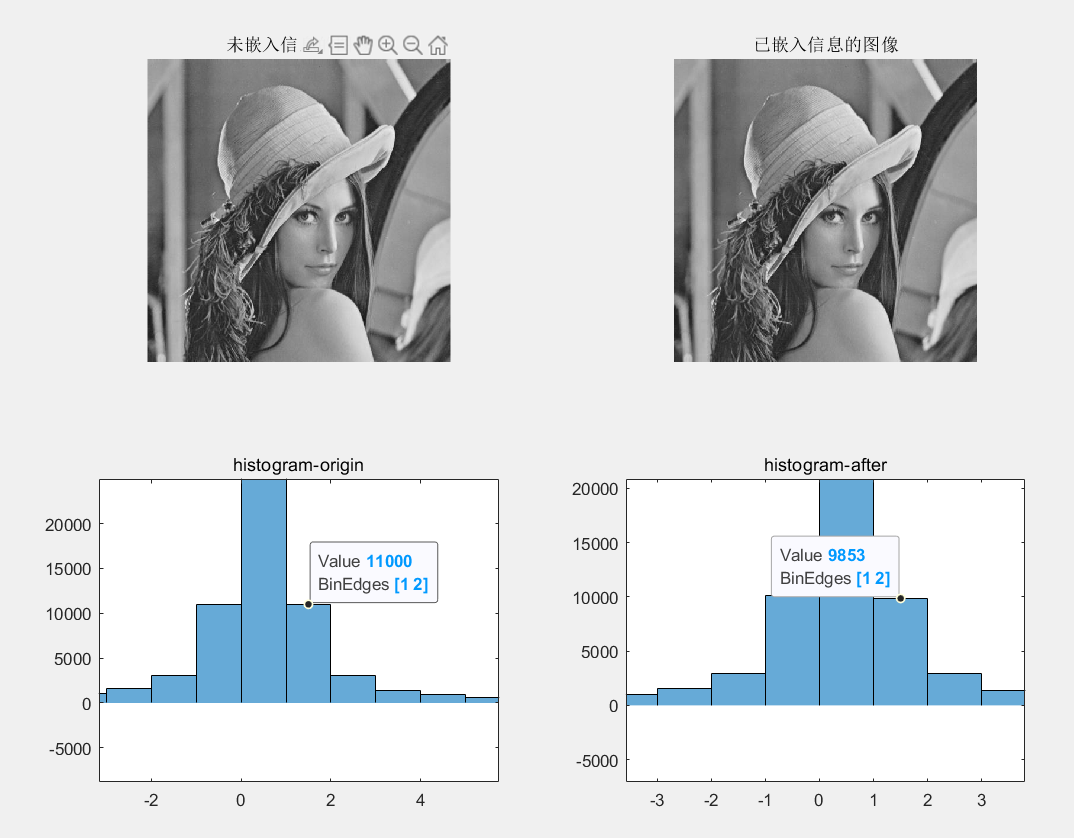


图2-8 F4算法图片和频率直方图

以下是正确性的展示，上方为提取出的信息，下方为原始信息。

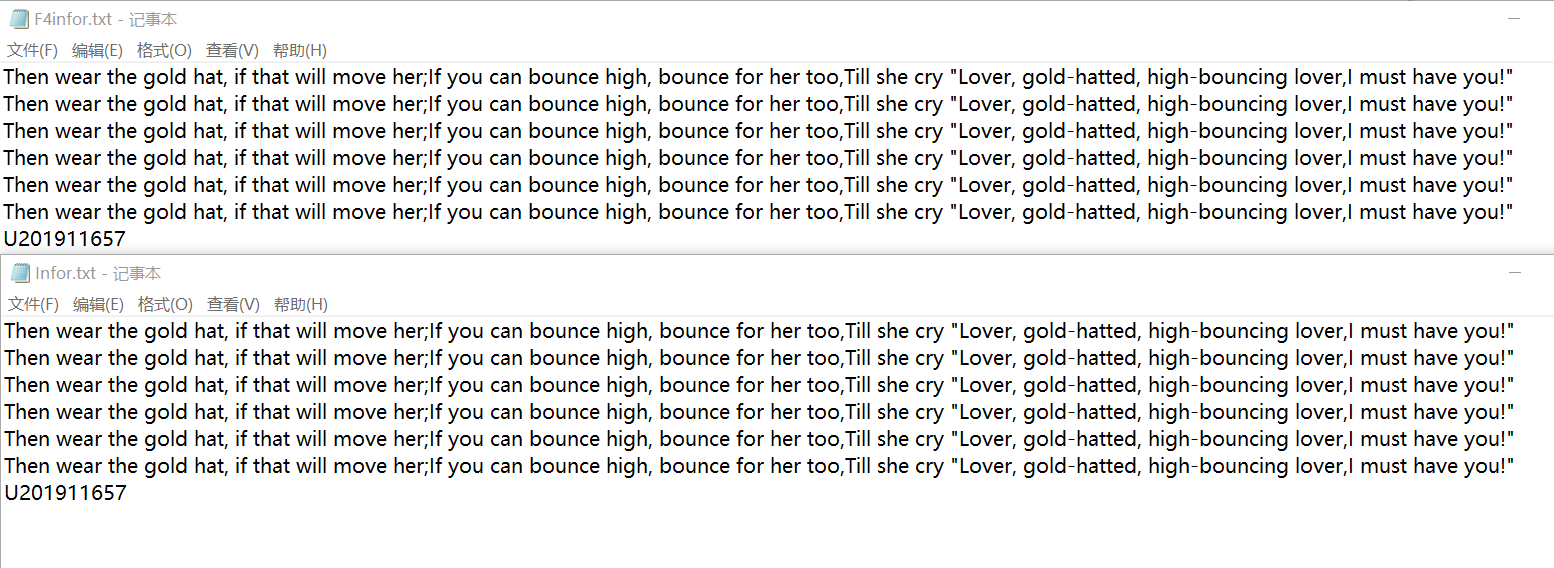


图2-9 F4算法正确性展示

## 2.3 F5算法

## 2.3.1 算法描述

F5算法也是DCT域的一种隐写算法，但相比于使用每一个比特来编码一个比特，而是使用个比特来编码k比特信息，这样可以做到任何情况下，都只需要调整原图DCT系数中的一个比特就可以完成嵌入。虽然嵌入率大幅下降，但是降低了嵌入的复杂度和比特偏差。

## 2.3.2 实现思路

整体设计依旧和之前一样，主要修改的是嵌入和提取部分。但因为我沿用了之前的思路，即嵌入前先提取，不同再改。

首先是我的比特表示的具体设计，表示嵌入信息，表示图像中的最低比特。则使用以下公式,其中的Sigma符号表示不进位加法，即异或:

所以程序实现的关键在于提取出原始图像中的信息。然后根据不同的地方，选择出需要修改的那一个字节，并且进行修改，根据上面的公式容易推算出，需要修改的字节为：

之后修改此字节即可。

还有一点需要注意的是，如果嵌入之后，目标字节被修改为0，那么必须保持目标字节被修改的情况下，整个进行回滚。而且因为很难提前判断是否嵌入点不够用或者是否无法完全嵌入，所以使用了assert来加强程序的鲁棒性。

最终代码如下图所示，去除掉修改字节部分就是提取过程。

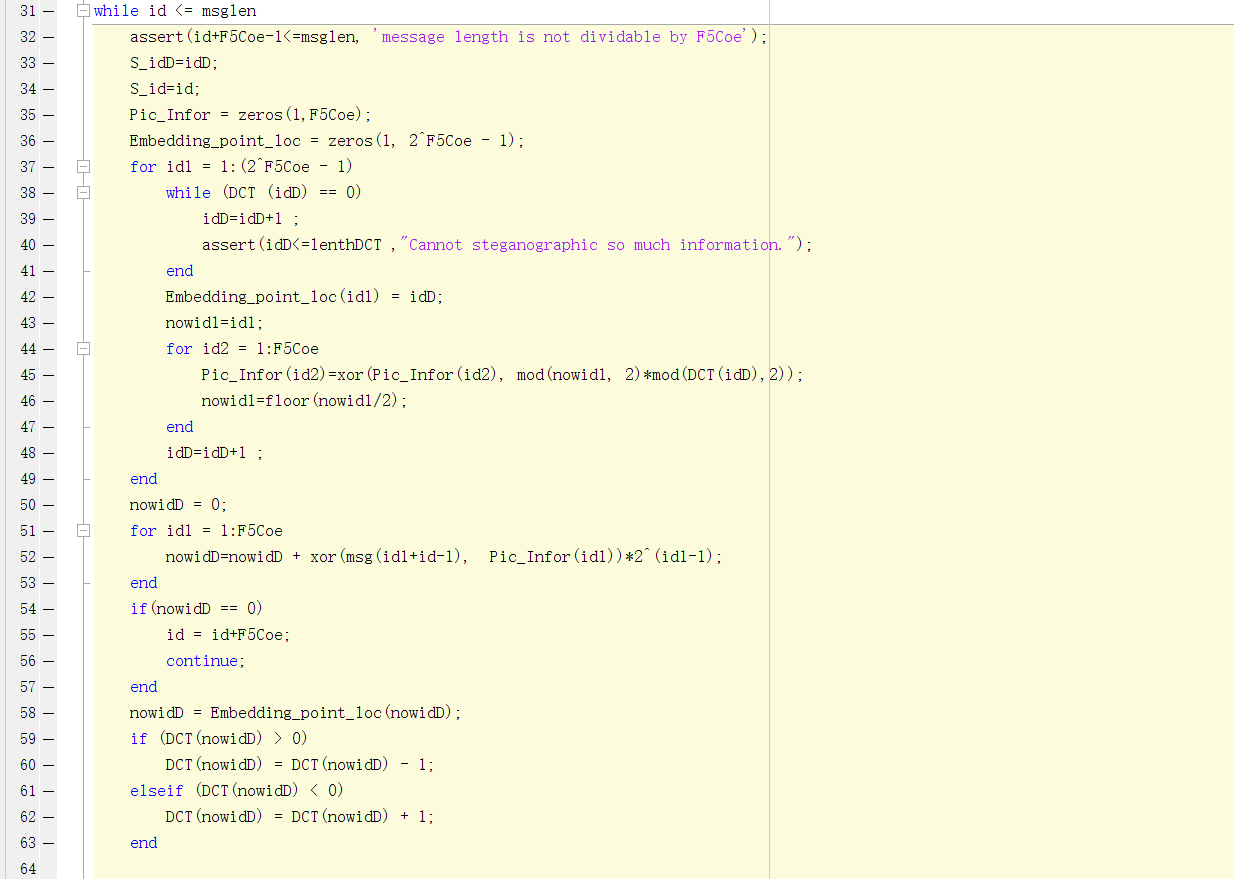


图2-10 F45算法matlab核心代码

## 2.3.3 实验结果及性能分析

首先是正确性验证：

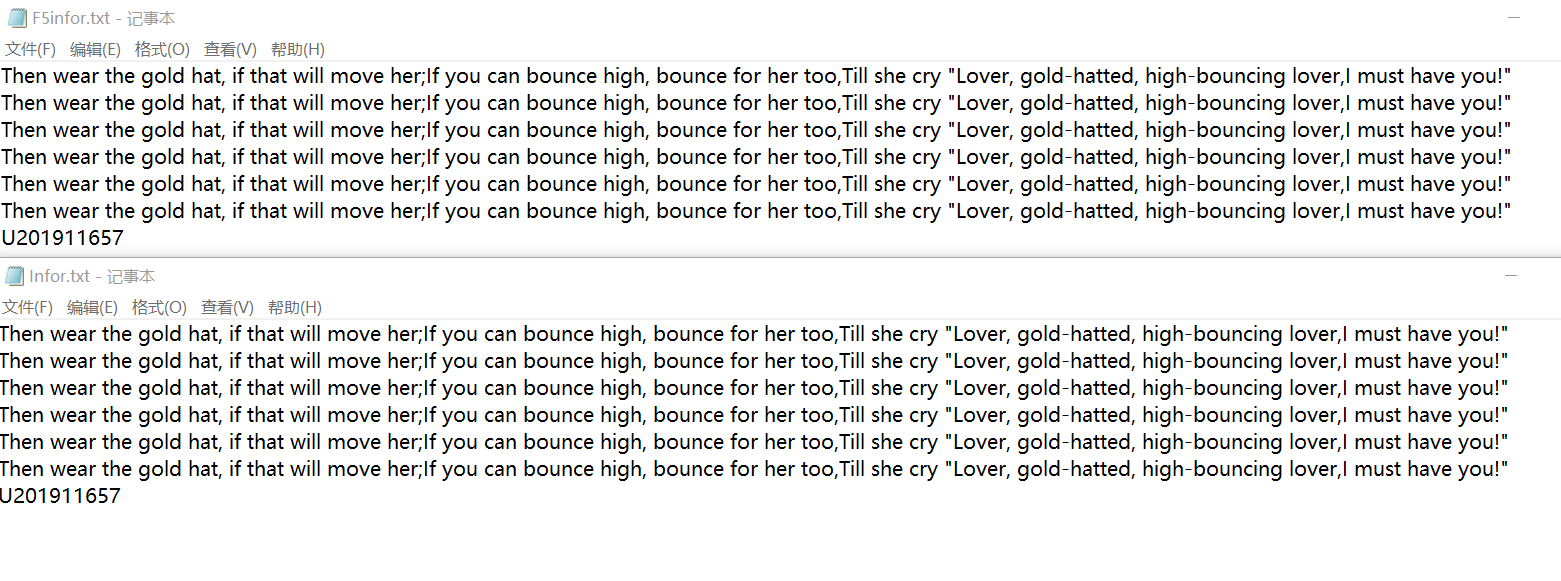


图2-11 F5算法正确性展示

可以看到可以正常提取，并且不会报错。

运行输出如图所示：

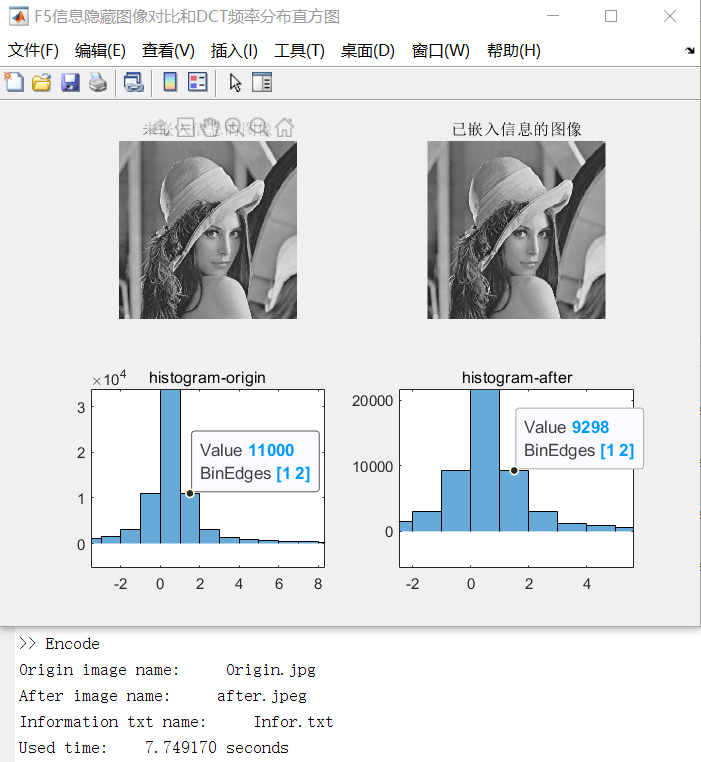


图2-12 F5算法图片比较和频率直方图

图象无法看出显著修改痕迹，但DCT系数频率直方图出现不同。

## 3 实验小结

这次实验能够亲手实现了几种嵌入和提取算法，这比仅仅上课学习理论的效果要好得多。而且顺便还学些了MATLAB这个工具，也算是有所收获。MATLAB本身的功能相当齐全，但是他的调试功能和开发界面确实不算很完善。

这三个部分里面，比较麻烦的是JSteg的值对现象展示和F5的编写，而其他的部分基本都是一个模板用到底。

而在值对展示过程中。我使用了多张图片和多个嵌入信息，似乎符合预期的值对和不符合预期的值对之比不过2：1左右。这个比例要小于我之前的预期。

而F5的代码编写时，主要是我犯了一些比较低级的错误，所以花了非常多的时间去找Bug。

还有就是没仔细看要求把LSB的报告写了。

# 

**指导教师评定意见**

一、对实验报告的评语

|  |
| --- |
|  |

二、对实验报告评分

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 评分项目  (分值) | 程序内容  (36.8分) | 程序规范  (9.2分) | 报告内容  (36.8分) | 报告规范  (9.2分) | 考勤  （8分） | 逾期扣分 | 合 计  (100分) |
| 得分 |  |  |  |  |  |  |  |

# 附录A JPEG图像变换域信息隐藏实现的源程序

**一、JSTEG**

**1.JSTEG嵌入函数代码：**

1. function [AC]=jsteg\_simulation(Origin\_Pic, Aim\_Pic, Infortxt)
2. fpinfor = fopen(Infortxt,"r");
3. [msg,msglen]=fread(fpinfor,'ubit1');
4. fclose(fpinfor);
6. **try**
7. jobj=jpeg\_read(Origin\_Pic) ;   %JPEG image structure
8. DCT=jobj.coef\_arrays{1};  % DCT plane
9. **catch**
10. error(' ERROR (problem with the cover image)');
11. end
12. %%
14. f1=fopen('DCT1out.txt', 'w'); % 存储原始DCT系数
15. len\_DCT = length(DCT(:));
16. **for** i=1:len\_DCT
17. fprintf(f1,'%d ',DCT(i));
18. end
19. fclose(f1);
21. AC=numel(DCT)-numel(DCT(1:8:end, 1:8:end));  % 非零AC DCT系数的数量 此处不理解
22. **if**(msglen >AC)
23. error('ERROR (message too long to steganography) ');
24. end
26. idD=1;
28. %%
30. **for** id = 1:msglen
31. %过滤 +-1和0
32. **while** (abs (DCT (idD))<=1)
33. idD=idD+1 ;
34. end
35. **if** (DCT(idD) >0)
36. %{
37. **if** (message(id)==0 && mod(DCT(idD),2)~=0)
38. DCT(idD)=DCT(idD)-1;
39. end
40. **if** (message(id)==1 && mod(DCT(idD),2)==0)
41. DCT(idD)=DCT(idD)+1;
42. end
43. %}
44. % 尝试Xor？
45. **if** msg(id) ~= mod(DCT(idD),2) % 如果提取结果不等于嵌入信息
46. DCT(idD)=bitxor(DCT(idD),1); % 进行翻转
47. end
48. elseif (DCT(idD) <0)
49. **if** (msg(id)==0 && mod(DCT(idD),2)~=0)
50. DCT(idD)=DCT(idD)+1;
51. end
52. **if** (msg(id)==1 && mod(DCT(idD),2)==0)
53. DCT(idD)=DCT(idD)-1;
54. end
55. end
56. idD=idD+1;
57. end
58. %%
59. f2=fopen('DCT2out.txt', 'w'); % 存储修改后的DCT系数
60. **for** i=1:len\_DCT
61. fprintf(f2,'%d ',DCT(i));
62. end
63. fclose(f2);
64. %%
66. %%% save the resulting stego image
67. **try**
68. jobj.coef\_arrays{1}=DCT;
69. jobj.optimize\_coding=1;
70. jpeg\_write(jobj , Aim\_Pic) ;
71. **catch**
72. error('ERROR (problem with saving the stego image) ')
73. end
75. end

**2.JSTEG执行嵌入及绘图代码：**

1. %%% setup
2. Origin\_Pic='Origin.jpg';  % Origin image (grayscale JPEG image)
3. Aim\_Pic='after.jpeg';  % resulting image (grayscale JPEG image)
4. Infortxt = 'Infor.txt'; % Information that need to steganography
5. fprintf('Origin image name:     %s\n' ,Origin\_Pic);
6. fprintf('After image name:     %s\n' , Aim\_Pic) ;
7. fprintf('Information txt name:     %s\n' , Infortxt) ;
9. %%
11. tic;% tic用来保存当前时间，而后使用toc来记录程序完成时间
12. [nzAC]=jsteg\_simulation (Origin\_Pic, Aim\_Pic, Infortxt) ;
13. T=toc;
14. %%
15. fprintf('Used time:    %5f seconds\n',T);
17. max\_pic\_row = 4;  % 图片的行数
18. max\_pic\_column = 2; % 图片的列数
19. now\_pic\_id = 1; % 现在是第几幅图,每次输出图像之后加1
21. % 做图像对比图
22. fig = figure('numbertitle','off','name','JSTEG信息隐藏图像对比和DCT频率分布直方图');
23. set(gcf,'unit','centimeters','position',[0,0,40,30])
24. subplot(max\_pic\_row,max\_pic\_column,now\_pic\_id);imshow(Origin\_Pic);title('未嵌入信息的图像');now\_pic\_id = now\_pic\_id + 1;
25. subplot(max\_pic\_row,max\_pic\_column,now\_pic\_id);imshow(Aim\_Pic);title('已嵌入信息的图像');now\_pic\_id = now\_pic\_id + 1;
27. % 做直方图
28. data1=load('DCT1out.txt');
29. data2=load('DCT2out.txt');
30. subplot(max\_pic\_row,max\_pic\_column,now\_pic\_id);histogram(data1,-30:1:30);title('histogram-origin');now\_pic\_id = now\_pic\_id + 1;
31. subplot(max\_pic\_row,max\_pic\_column,now\_pic\_id);histogram(data2,-30:1:30);title('histogram-after');now\_pic\_id = now\_pic\_id + 1;
33. % 做值对条状图
34. % 横轴为 [-127:2:-1, 0, 1:2:127] 到 1:129 的顺序映射
35. normalize\_offset = 128; % 这里实际上不是一个变量，不能修改，因为后面的值不是使用normalize\_offset 推算出来的
36. Origin\_tabu = zeros(2, 129);
37. After\_tabu = zeros(2, 129);
39. data1=load('DCT1out.txt');
40. **for** i = 1:length(data1)
41. **if**(data1(i) <= -1\*normalize\_offset || data1(i) >= normalize\_offset)
42. **continue**;
43. end
44. normalize\_DCT = data1(i) + normalize\_offset;
45. **if**(normalize\_DCT < 127)
46. x = xor(mod(normalize\_DCT, 2), 1) + 1;
47. y = ceil(normalize\_DCT/2);
48. elseif(normalize\_DCT > 129)
49. x = mod(normalize\_DCT, 2) + 1;
50. y = floor(normalize\_DCT/2) + 2;
51. **else**
52. x = 1;
53. y = normalize\_DCT-63;
54. end
55. Origin\_tabu(x, y ) = Origin\_tabu(x, y ) + 1;
56. end
58. subplot(max\_pic\_row,max\_pic\_column,now\_pic\_id);bar([-127:2:-1, 0, 1:2:127],Origin\_tabu, 'stack');title('value pair-origin');now\_pic\_id = now\_pic\_id + 1;
60. data2=load('DCT2out.txt');
61. **for** i = 1:length(data2)
62. **if**(data2(i) <= -1\*normalize\_offset || data2(i) >= normalize\_offset)
63. **continue**;
64. end
65. normalize\_DCT = data2(i) + normalize\_offset;
66. **if**(normalize\_DCT < 127)
67. x = xor(mod(normalize\_DCT, 2), 1) + 1;
68. y = ceil(normalize\_DCT/2);
69. elseif(normalize\_DCT > 129)
70. x = mod(normalize\_DCT, 2) + 1;
71. y = floor(normalize\_DCT/2) + 2;
72. **else**
73. x = 1;
74. y = normalize\_DCT-63;
75. end
76. After\_tabu(x, y ) = After\_tabu(x, y ) + 1;
77. end
78. subplot(max\_pic\_row,max\_pic\_column,now\_pic\_id);bar([-127:2:-1, 0, 1:2:127],After\_tabu , 'stack');title('value pair-after');now\_pic\_id = now\_pic\_id + 1;

81. % 做值对标准化图
82. % 横轴为 [-127:2:-1, 0, 1:2:127] 到 1:129 的顺序映射
83. Origin\_propotion = zeros(2, 129);
84. After\_propotion = zeros(2, 129);
85. **for** id = 1:129
86. Origin\_propotion(1, id) =  Origin\_tabu(1, id)/(Origin\_tabu(1, id)+Origin\_tabu(2, id));
87. Origin\_propotion(2, id) =  1.0-Origin\_propotion(1, id);
88. After\_propotion(1, id) =  After\_tabu(1, id)/(After\_tabu(1, id)+After\_tabu(2, id));
89. After\_propotion(2, id) =  1.0-After\_propotion(1, id);
90. end
92. subplot(max\_pic\_row,max\_pic\_column,now\_pic\_id);bar([-127:2:-1, 0, 1:2:127],Origin\_propotion, 'stack');title('value pair propotion-origin');now\_pic\_id = now\_pic\_id + 1;
93. subplot(max\_pic\_row,max\_pic\_column,now\_pic\_id);bar([-127:2:-1, 0, 1:2:127],After\_propotion , 'stack');title('value pair propotion-after');now\_pic\_id = now\_pic\_id + 1;

96. % 做值对是否符合预期的图
97. % 横轴为 [-127:2:-1, 0, 1:2:127] 到 1:129 的顺序映射
98. value\_move = zeros(2, 129);
99. cnt1 = 0;
100. cnt2 = 0;
101. **for** id = 1:129
102. **if** ( abs(Origin\_tabu(1, id)-Origin\_tabu(2, id)) >= abs(After\_tabu(1, id)-After\_tabu(2, id))) % 值对现象符合预期
103. value\_move(1, id) = 1.0;
104. value\_move(2, id) = 0.0;
105. cnt1 = cnt1 + 1;
106. **else**  % 值对现象不符合预期
107. value\_move(2, id) = 1.0;
108. value\_move(1, id) = 0.0;
109. cnt2 = cnt2 + 1;
110. end
111. end
112. fig = figure('numbertitle','off','name','JSTEG信息隐藏 值对现象展示');
113. bar([-127:2:-1, 0, 1:2:127],value\_move , 'stack');title('蓝色表示值对趋向于相同，红色表示趋向于不同');
114. fprintf("符合预期的值对数量: %d\n不符合预期的值对数量: %d\n",cnt1, cnt2 );
115. **JSTEG解密代码：**
116. Aim\_Pic='after.jpeg';  %resulting image (grayscale JPEG image)
117. Infortxt = 'Infor.txt'; % Information that need to steganography
118. fpjsteginfor=fopen('jsteginfor.txt', 'w'); %直接把比特写到文件里，再作为字符串读出来
120. % 此处偷个懒获取嵌入信息的长度。
121. fpinfor = fopen(Infortxt,"r");
122. [~,msglen]=fread(fpinfor,'ubit1');% 使用~表示忽略读入的具体信息
123. fprintf("The length of infor is :%d\n", msglen);
124. fclose(fpinfor);
126. **try**
127. jobj=jpeg\_read(Aim\_Pic) ;   %JPEG image structure
128. DCT=jobj.coef\_arrays{1};  % DCT plane
129. **catch**
130. error(' ERROR (problem with the after image)');
131. end
132. %%
133. AC=numel(DCT)-numel(DCT(1:8:end, 1:8:end));  %非零AC DCT系数的数量
134. **if**(msglen >AC)
135. error('ERROR (message too long to steganography) ');
136. end
137. idD=1;
138. %%
140. **for** id=1 :msglen
141. **while** (abs (DCT (idD))<=1)
142. idD=idD+1;
143. end
144. fwrite(fpjsteginfor,mod(DCT(idD),2), 'ubit1');
145. %{
146. **if** (mod(DCT(idD),2)==1)
147. fwrite(fp,1, 'ubit1');
148. **else**
149. fwrite(fp,0, 'ubit1');
150. end
151. %}
152. idD=idD+1;
153. end
154. fclose(fpjsteginfor);
155. %%
157. fpjsteginfor=fopen('jsteginfor.txt', 'r');
158. msg=fread(fpjsteginfor,'\*char')';
159. fprintf('-----\n');
160. fprintf('The Information in JPEG is :\n%s\n' ,msg);
161. fclose(fpjsteginfor);

**二、F4**

**1.F4嵌入函数代码：**

1. function [AC]=F4\_simulation(Origin\_Pic, Aim\_Pic, Infortxt)
2. fpinfor = fopen(Infortxt,"r");
3. [msg,msglen]=fread(fpinfor,'ubit1');
4. fclose(fpinfor);
6. **try**
7. jobj=jpeg\_read(Origin\_Pic) ;   %JPEG image structure
8. DCT=jobj.coef\_arrays{1};  % DCT plane
9. **catch**
10. error(' ERROR (problem with the cover image)');
11. end
12. %%
13. f1=fopen('DCT1out.txt', 'w');
14. len\_DCT = length(DCT(:));
15. **for** i=1:len\_DCT
16. fprintf(f1,'%d ',DCT(i));
17. end
18. fclose(f1);
20. AC=numel(DCT)-numel(DCT(1:8:end, 1:8:end));  % 非零AC DCT系数的数量 此处不理解
21. **if**(msglen >AC)
22. error('ERROR (message too long to steganography) ');
23. end
24. idD=1;
25. id=1;
26. %%
28. **while** id <= msglen  % 枚举信息
29. %过滤 +-1和0
30. **while** (DCT (idD) == 0) % 去除0信息，保证可以嵌入
31. idD=idD+1 ;
32. end
34. **if** (DCT(idD) >0)
35. DCT(idD) = DCT(idD) - xor(mod(DCT(idD),2), msg(id));
36. elseif (DCT(idD) <0)
37. DCT(idD) = DCT(idD) + xor(xor(mod(DCT(idD),2), msg(id)), 1);
38. end
39. **if** (DCT(idD) ~= 0)  %嵌入信息成功
40. id = id + 1;
41. end
42. idD=idD+1;
43. end
44. %%
45. f2=fopen('DCT2out.txt', 'w');
46. **for** i=1:len\_DCT
47. fprintf(f2,'%d ',DCT(i));
48. end
49. fclose(f2);
50. %%
52. %%% save the resulting stego image
53. **try**
54. jobj.coef\_arrays{1}=DCT;
55. jobj.optimize\_coding=1;
56. jpeg\_write(jobj , Aim\_Pic) ;
57. **catch**
58. error('ERROR (problem with saving the stego image) ')
59. end
61. end

**2.F4执行嵌入及绘图代码：**

1. %%% setup
2. Origin\_Pic='Origin.jpg';  % Origin image (grayscale JPEG image)
3. Aim\_Pic='after.jpeg';  % resulting image (grayscale JPEG image)
4. Infortxt = 'Infor.txt'; % Information that need to steganography
5. fprintf('Origin image name:     %s\n' ,Origin\_Pic);
6. fprintf('After image name:     %s\n' , Aim\_Pic) ;
7. fprintf('Information txt name:     %s\n' , Infortxt) ;
9. %%
11. tic;% tic用来保存当前时间，而后使用toc来记录程序完成时间
12. [nzAC]=F4\_simulation (Origin\_Pic, Aim\_Pic, Infortxt) ;
13. T=toc;
14. %%
15. fprintf('Used time:    %5f seconds\n',T);
17. % 做图像对比图
18. fig = figure('numbertitle','off','name','F4信息隐藏图像对比和DCT频率分布直方图');
19. subplot(2,2,1);imshow(Origin\_Pic);title('未嵌入信息的图像');
20. subplot(2,2,2);imshow(Aim\_Pic);title('已嵌入信息的图像');
21. % 做直方图
22. data1=load('DCT1out.txt');
23. data2=load('DCT2out.txt');
24. subplot(2,2,3);histogram(data1,-30:1:30);title('histogram-origin');
25. subplot(2,2,4);histogram(data2,-30:1:30);title('histogram-after');

**3.F4解密代码：**

1. Aim\_Pic='after.jpeg';  %resulting image (grayscale JPEG image)
2. Infortxt = 'Infor.txt'; % Information that need to steganography
3. fpF4infor=fopen('F4infor.txt', 'w'); %直接把比特写到文件里，再作为字符串读出来
5. fpinfor = fopen(Infortxt,"r");
6. [~,msglen]=fread(fpinfor,'ubit1');
7. fprintf("The length of infor is :%d\n", msglen);
8. fclose(fpinfor);
10. **try**
11. jobj=jpeg\_read(Aim\_Pic) ;   %JPEG image structure
12. DCT=jobj.coef\_arrays{1};  % DCT plane
13. **catch**
14. error(' ERROR (problem with the after image)');
15. end
16. %%
18. AC=numel(DCT)-numel(DCT(1:8:end, 1:8:end));  %非零AC DCT系数的数量
19. **if**(msglen >AC)
20. error('ERROR (message too long to steganography) ');
21. end
22. idD=1;
23. %%
25. **for** id=1 :msglen
26. **while** (DCT (idD) == 0)
27. idD=idD+1;
28. end
29. fwrite(fpF4infor,xor(mod(DCT(idD),2), (DCT(idD) < 0)), 'ubit1');
30. %{
31. **if** (mod(DCT(idD),2)==1)
32. fwrite(fp,1, 'ubit1');
33. **else**
34. fwrite(fp,0, 'ubit1');
35. end
36. %}
37. idD=idD+1;
38. end
39. fclose(fpF4infor);
40. %%
42. fpF4infor=fopen('F4infor.txt', 'r');
43. msg=fread(fpF4infor,'\*char')';
44. fprintf('------------------------------------------------\n');
45. fprintf('The Information in JPEG is :\n%s\n' ,msg);
46. fclose(fpF4infor);

**三、F5**

**1.F5嵌入函数代码：**

1. function [AC]=F5\_simulation(Origin\_Pic, Aim\_Pic, Infortxt, F5Coe)
2. fpinfor = fopen(Infortxt,"r");
3. [msg,msglen]=fread(fpinfor,'ubit1');
4. fclose(fpinfor);
5. fperrinfor = fopen('Error2.txt',"w");
7. **try**
8. jobj=jpeg\_read(Origin\_Pic) ;   %JPEG image structure
9. DCT=jobj.coef\_arrays{1};  % DCT plane
10. **catch**
11. error(' ERROR (problem with the cover image)');
12. end
14. %%
15. % 存储原始DCT系数
16. f1=fopen('DCT1out.txt', 'w');
17. len\_DCT = length(DCT(:));
18. **for** i=1:len\_DCT
19. fprintf(f1,'%d ',DCT(i));
20. end
21. fclose(f1);
23. AC=numel(DCT)-numel(DCT(1:8:end, 1:8:end));  % 非零AC DCT系数的数量 此处不理解
24. lenthDCT = numel(DCT);
25. **if**(msglen >AC)
26. error('ERROR (message too long to steganography) ');
27. end
28. %%
29. idD=1;
30. id=1;
31. **while** id <= msglen
32. assert(id+F5Coe-1<=msglen, 'message length is not dividable by F5Coe');
33. S\_idD=idD;
34. S\_id=id;
35. Pic\_Infor = zeros(1,F5Coe);
36. Embedding\_point\_loc = zeros(1, 2^F5Coe - 1);
37. **for** id1 = 1:(2^F5Coe - 1)
38. **while** (DCT (idD) == 0)
39. idD=idD+1 ;
40. assert(idD<=lenthDCT ,"Cannot steganographic so much information.");
41. end
42. Embedding\_point\_loc(id1) = idD;
43. nowid1=id1;
44. **for** id2 = 1:F5Coe
45. Pic\_Infor(id2)=xor(Pic\_Infor(id2), mod(nowid1, 2)\*mod(DCT(idD),2));
46. nowid1=floor(nowid1/2);
47. end
48. idD=idD+1 ;
49. end
50. nowidD = 0;
51. **for** id1 = 1:F5Coe
52. nowidD=nowidD + xor(msg(id1+id-1),  Pic\_Infor(id1))\*2^(id1-1);
53. end
54. **if**(nowidD == 0)
55. id = id+F5Coe;
56. **continue**;
57. end
58. nowidD = Embedding\_point\_loc(nowidD);
59. **if** (DCT(nowidD) > 0)
60. DCT(nowidD) = DCT(nowidD) - 1;
61. elseif (DCT(nowidD) < 0)
62. DCT(nowidD) = DCT(nowidD) + 1;
63. end
65. %%
66. %{
67. **if** (DCT(nowidD) == 0)
68. idD=S\_idD;
69. id=S\_id;
70. **continue**;
71. end
73. Pic\_Infor =zeros(1, F5Coe);
74. **for** id1 = 1:(2^F5Coe - 1)
75. idD=Embedding\_point\_loc(id1);
76. nowid1=id1;
77. idD
78. fprintf(fperrinfor, " %d ",idD);
79. **for** id2 = 1:F5Coe
80. Pic\_Infor(id2)=xor(Pic\_Infor(id2), mod(nowid1, 2)\*mod(DCT(idD),2));
81. nowid1=floor(nowid1/2);
82. end
83. end
85. **for** id1 = 1:F5Coe
86. assert(Pic\_Infor(id1)==msg(id1));
87. end
89. %}
91. %%






99. **if** (DCT(nowidD) == 0)
100. idD=S\_idD;
101. id=S\_id;
102. **else**
103. id = id+F5Coe;
104. end
105. end
106. %%
107. f2=fopen('DCT2out.txt', 'w');
108. **for** i=1:len\_DCT
109. fprintf(f2,'%d ',DCT(i));
110. end
111. fclose(f2);
112. %%
114. %%% save the resulting stego image
115. **try**
116. jobj.coef\_arrays{1}=DCT;
117. jobj.optimize\_coding=1;
118. jpeg\_write(jobj , Aim\_Pic) ;
119. **catch**
120. error('ERROR (problem with saving the stego image) ')
121. end
122. fclose(fperrinfor);
123. end

**2.F5执行嵌入及绘图代码：**

1. %%% setup
2. Origin\_Pic='Origin.jpg';  % Origin image (grayscale JPEG image)
3. Aim\_Pic='after.jpeg';  % resulting image (grayscale JPEG image)
4. Infortxt = 'Infor.txt'; % Information that need to steganography
5. fprintf('Origin image name:     %s\n' ,Origin\_Pic);
6. fprintf('After image name:     %s\n' , Aim\_Pic) ;
7. fprintf('Information txt name:     %s\n' , Infortxt) ;
9. %%
11. tic;% tic用来保存当前时间，而后使用toc来记录程序完成时间
12. [nzAC]=F5\_simulation (Origin\_Pic, Aim\_Pic, Infortxt, 2) ;
13. T=toc;
14. %%
15. fprintf('Used time:    %5f seconds\n',T);
17. % 做图像对比图
18. fig = figure('numbertitle','off','name','F5信息隐藏图像对比和DCT频率分布直方图');
19. subplot(2,2,1);imshow(Origin\_Pic);title('未嵌入信息的图像');
20. subplot(2,2,2);imshow(Aim\_Pic);title('已嵌入信息的图像');
21. % 做直方图
22. data1=load('DCT1out.txt'); % 前的DCT系数
23. data2=load('DCT2out.txt'); % 后的DCT系数
24. subplot(2,2,3);histogram(data1,-30:1:30);title('histogram-origin');
25. subplot(2,2,4);histogram(data2,-30:1:30);title('histogram-after');

**3. F5解密代码：**

1. Aim\_Pic='after.jpeg';  %resulting image (grayscale JPEG image)
2. Infortxt = 'Infor.txt'; % Information that need to steganography
3. F5Coe = 2; % 需要和加密保持一致
4. fpF5infor=fopen('F5infor.txt', 'w'); %直接把比特写到文件里，再作为字符串读出来
6. fpinfor = fopen(Infortxt,"r");
7. [~,msglen]=fread(fpinfor,'ubit1');
8. fprintf("The length of infor is :%d\n", msglen);
9. fclose(fpinfor);
10. fperrinfor = fopen('Error.txt','w');
11. **try**
12. jobj=jpeg\_read(Aim\_Pic) ;   %JPEG image structure
13. DCT=jobj.coef\_arrays{1};  % DCT plane
14. **catch**
15. error(' ERROR (problem with the after image)');
16. end
17. %%
19. AC=numel(DCT)-numel(DCT(1:8:end, 1:8:end));  %非零AC DCT系数的数量
20. lenthDCT = numel(DCT);
21. **if**(msglen >AC)
22. error('ERROR (message too long to steganography) ');
23. end
24. %%
25. idD=1;
26. id=1;
27. % 使用2位的F5
28. **while** id <= msglen
29. assert(id+F5Coe-1<=msglen, 'message length is not dividable by F5Coe');
30. Pic\_Infor = zeros(1,F5Coe);
32. **for** id1 = 1:(2^F5Coe - 1)
33. **while** (DCT (idD) == 0)
34. idD=idD+1 ;
35. assert(idD<=lenthDCT ,"Cannot steganographic so much information.");
36. end
37. nowid1=id1;
38. **for** id2 = 1:F5Coe
39. Pic\_Infor(id2)=xor(Pic\_Infor(id2), mod(nowid1, 2)\*mod(DCT(idD),2));
40. nowid1=floor(nowid1/2);
41. end
42. idD=idD+1 ;
43. end
45. **for** id1 = 1:F5Coe
46. fwrite(fpF5infor,Pic\_Infor(id1), 'ubit1');
47. end
48. id = id+F5Coe;
49. end
50. fclose(fpF5infor);
51. %%
53. fpF5infor=fopen('F5infor.txt', 'r');
54. msg=fread(fpF5infor,'\*char')';
55. fprintf('------------------------------------------------\n');
56. fprintf('The Information in JPEG is :\n%s\n' ,msg);
57. fclose(fpF5infor);
58. fclose(fperrinfor);