目 录

1 基于 LSB 的空域信息隐藏实现

1.1 问题描述

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。

1、要求实现信息嵌入算法。自选载体图像，嵌入内容为自己的学号。首先对原始图像中每个像素点的灰度值进行变换，由十进制转换为二进制；然后将秘密信息转换为二进制序列，并将原始图像中像素点的最低比特为替换为二进制序列中的每一比特信息；在上述替换结束后，将像素点的二进制数据转换回十进制数据，保存为含有秘密信息的图像，并与原始图像同框显示以对比视觉效果。

2、要求实现信息提取算法。从含有秘密信息的 BMP 图像中提取出自己的学号。首先对载密图像中每个像素点的灰度值进行变换，由十进制转换为二进制；然后提取图像像素点中最低有效位的数据，根据嵌入顺序进行组合得到嵌入秘密信息。

1.2 系统设计

嵌入基于.bmp 格式图片.

参考设计样例,以行顺序进行嵌入明文的二进制字节流.

1.3 系统实现

1.3.1 读取图片

使用 imread 函数读取图片,并获取相关信息.

```
1.      Picture=imread('kinkakuji01.bmp');
```

```
Double_Picture=Picture;
Double_Picture=double(Double_Picture);
    [m,n]=size(Double_Picture);
```

1.3.2 嵌入

以行顺序逐个嵌入最低位. 当嵌入完毕或图片被嵌入满时跳出.

```
            for f2=1:n
    for f1=1:m
        fprintf("[LOG] Embedding %d %d\t%d\n",f1,f2,Double_Picture(f1,f2));

        Double_Picture(f1,f2)=Double_Picture(f1,f2)-
mod(Double_Picture(f1,f2),2)+msg(p);
        if p==len
            break;
        end
        p=p+1;
    end
    if p==len
        break;
    end
  end
```

1.3.3 输出

整理数据, 输出到新的嵌入图片中. 并显示.

```
    Double_Picture=uint8(Double_Picture);
imwrite(Double_Picture,'kinkakuji01-steg.bmp');
subplot(121);imshow(Picture);title('Oroginal Pic');
    subplot(122);imshow(Double_Picture);title('Embedded Pic');
```

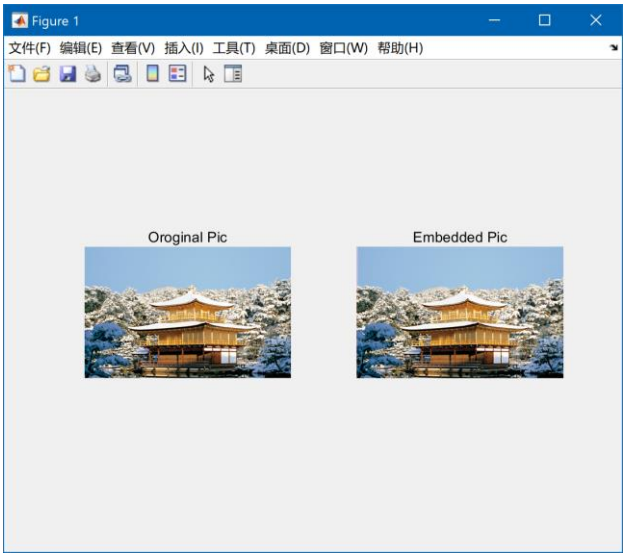

图 1-1 嵌入后的图片对比

1.3.4 提取

行顺序提取，并将结果添加到结果字符串中.

```
  for f2=1:n
    for f1=1:m
        if bitand(Picture(f1,f2),1)==1
            result=result+"1";
        else
            result=result+"0";
        end
        if p==len
            break;
        end
        if p<len
            p=p+1;
        end
    end
    if p==len
        break;
    end
  end
```

```
>> extract
[SUCCESS] Msg=0100000101100011011100100110111101110011011100110010000001110100011010000110010100100000011001110 1
```

图 1-2 运行结果,和明文字节流吻合.

明文为 Across the Great Wall We can Reach Every Corner of the World U201714886

1.4 实验小结

LSB 算法是对空域的 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。因此 LSB 算法最初是用于脆弱性水印的。

2 JPEG 图像变换域信息隐藏实现

2.1 问题描述

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;

⑶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。

采用 JPEG 格式灰度图像作为载体图像，用 MATLAB 实现 JPEG 图片变换域信息隐藏的嵌入与提取算法。自选 JPEG 载体图像，嵌入内容自选。

1、要求实现 JSTEG 信息嵌入与提取算法。并比较嵌入前后的视觉效果与 DCT 系数直方图。

2、要求实现 F3 信息嵌入与提取算法。并比较嵌入前后的视觉效果与 DCT 系数直方图。

3、要求实现 F4 信息嵌入与提取算法。并比较嵌入前后的视觉效果与 DCT 系数直方图。

4、要求实现 F5 信息嵌入与提取算法。并比较嵌入前后的视觉效果与 DCT 系数直方图。

2.2 系统设计

考虑到 JSTEG 系列算法差别较小,因此使用通用框架如下图:

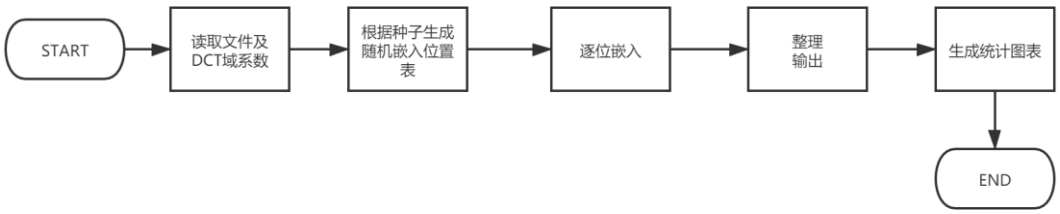

图 2-1 通用框架

其中读取文件 DCT 系数由工具执行. 下面主要描述嵌入部分的流程.

2.3 系统实现

2.3.1 行列解算

从随机数表中读取 index, 并根据图片尺寸解算为行向量和列向量.

```
location=randtable(RandTableIndex);
%Try to Resolve
row=ceil(location/400);
column=mod(location,400)+1;
```

2.3.2 Jsteg 算法

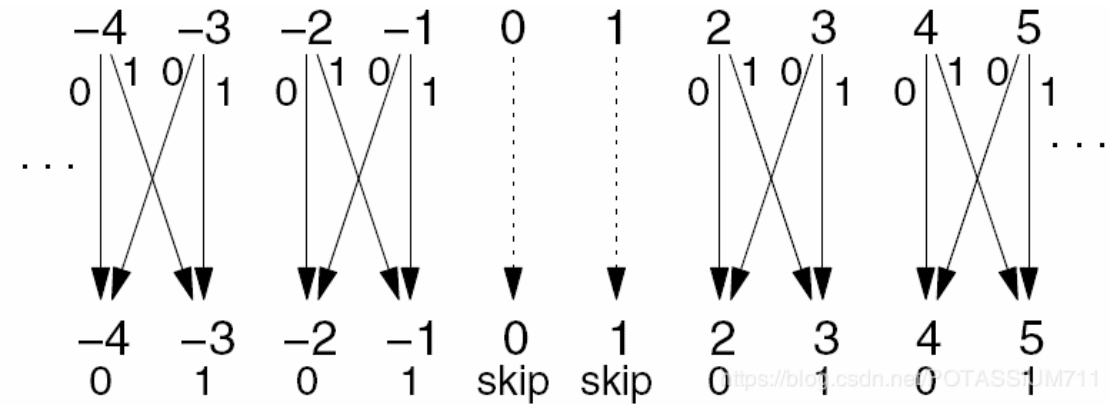

图 2-2 Jsteg 算法原理

读取系数和待嵌入比特流:

嵌入：

1. 0/1：跳过
2. 通过保持或 x±1x \ pm1x±1 进行调整以适合最低位数（0-偶数，1-奇数）
3. 2-1 不能达到 1，因此必须更改为 3。

提取：

1. 0/1：跳过
2. 只需读取最低位数（0 偶，1 奇）

2.3.3 F3 算法

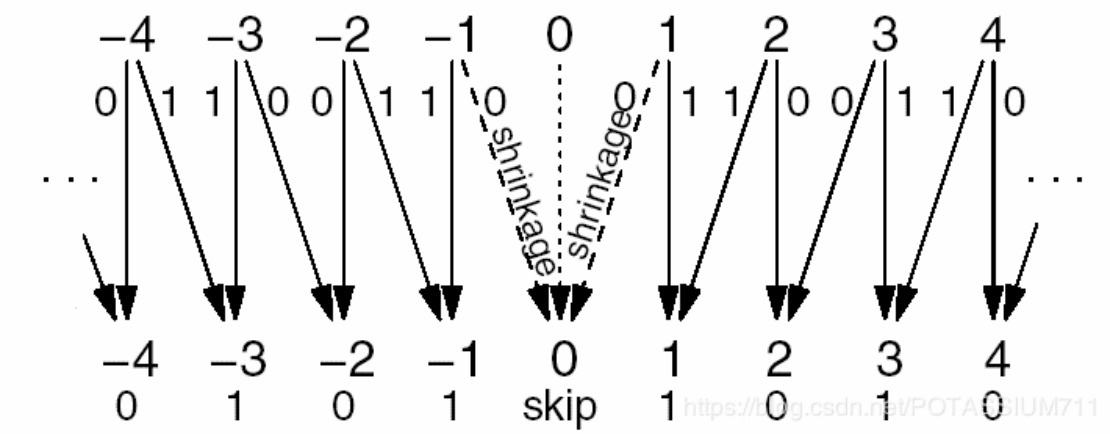

图 2-3 F3 算法原理

嵌入：

1. 0：跳过
2. 正负部分是对称的。
3. 1 / -1 和 0：shrinkage
4. 通过保持或 | x | − = 1 | x |-= 1 | x | − = 1 进行调整以适合最低位数（0-偶数，1-奇数）

提取：

1. 0：跳过
2. 只需读取最低位数（0 偶，1 奇）

2.3.4 F4 算法

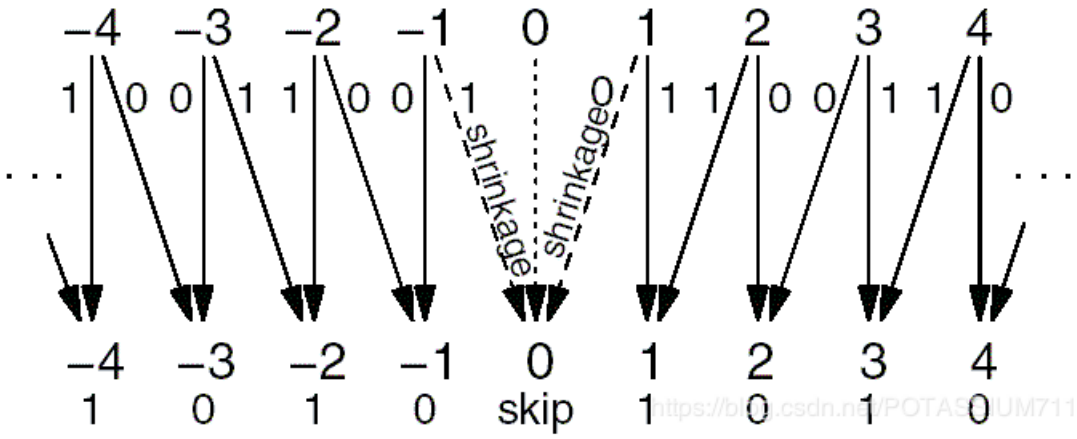

图 2-4 F4 算法

嵌入:

1. 0：跳过
2. 正负部分是对应的。
3. 1-0 和-1-1：shrinkage
4. 对于正数：奇数 1，偶数 0
5. 对于负数：奇数 0，偶数 1

提取：

1. 0：跳过
2. 对于正数：读取最低的数字
3. 对于负数：读取最低的数字并反转

2.3.5 F5 算法

矩阵编码:

1. bool p= (if x_1==lowbit($a_1 \oplus a_3$))
2. bool q= (if x_2==lowbit($a_2 \oplus a_3$))
3. `p && q?keep:pass;`
4. `!p&&q?Change a1:pass;`
5. `p&&!q?Change a2:pass;`
6. `!p&&!q?Change a3:pass;`

图 2-5 矩阵编码的逻辑描述

选择非 0 DCT 系数并洗混

对于每 3 个 DCT 系数，尝试使用矩阵编码嵌入

[注意]此过程中的 shrinkage：编码后，如果任何 DCT 系数变为 0，则取消此次嵌入并加载 3 个 DCT 系数，然后重新开始。

2.3.6 运行结果

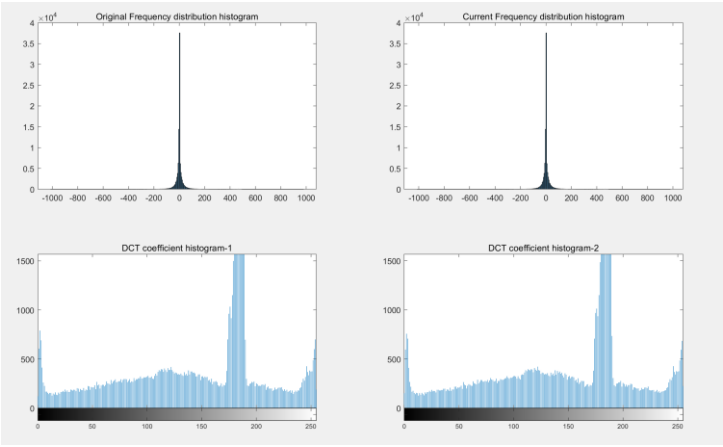

图 2-6 Jsteg 嵌入结果

ifequal	1
index	577
Index	577
locati...	54511
origin...	*1x1 struct*
plain	"0100000101100011011100100110111110...
ran	*1x1 struct*
randt...	*1x102400 double*
Rand...	805
result	"0100000101100011011100100110111110...

图 2-7 Jsteg 提取结果

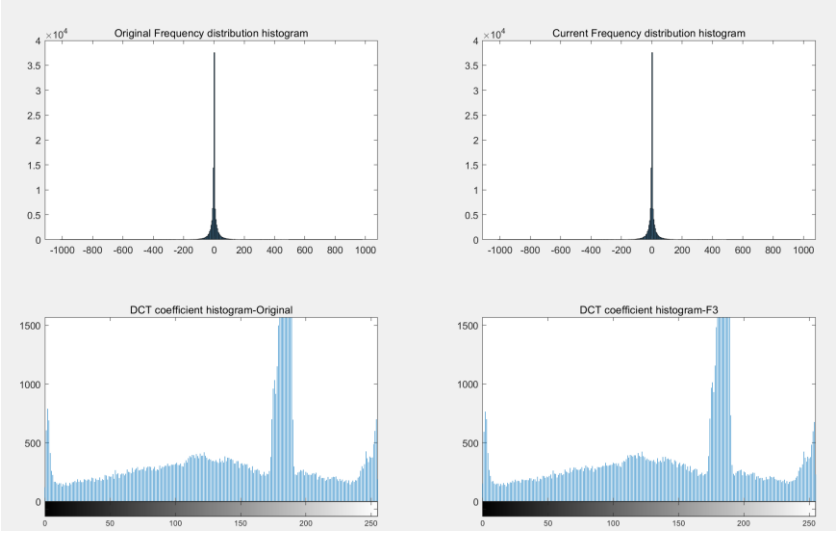

图 2-8 F3 嵌入

ifequal	1
index	577
Index	577
kp	1
locati...	13377
origin...	*1x1 struct*
plain	"0100000101100011011100100110111110... <预览内容在 128 个字符处截断>
ran	*1x1 struct*
randt...	*1x102400 double*
Rand...	797
result	"0100000101100011011100100110111110... <预览内容在 128 个字符处截断>
row	34

图 2-9 F3 提取

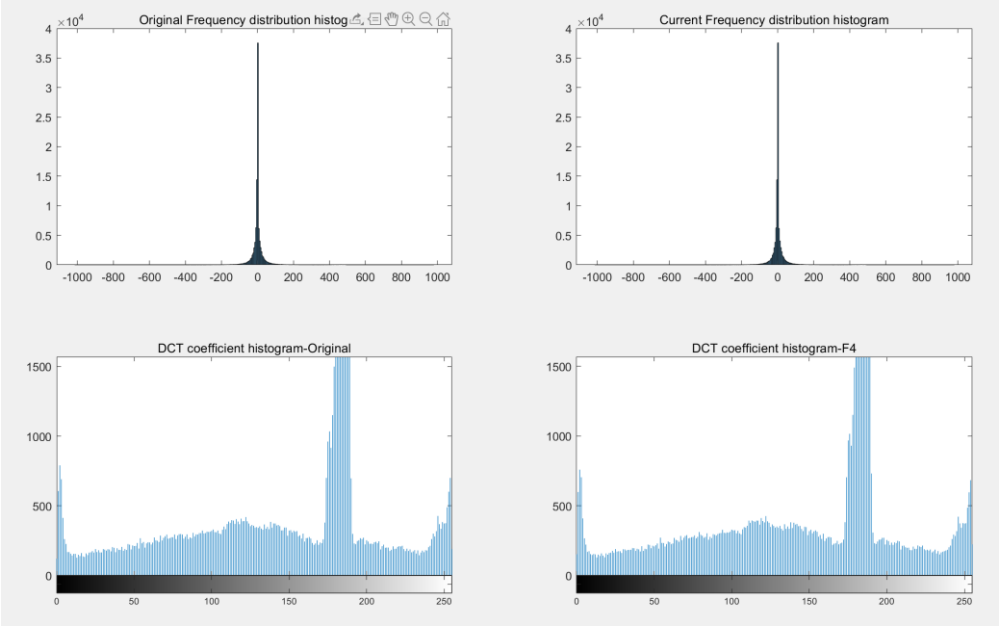

图 2-10 F4 嵌入

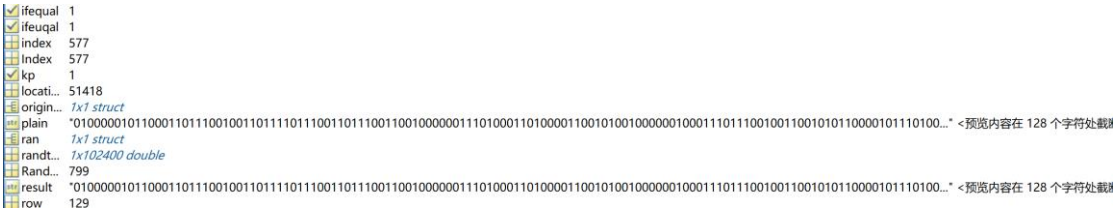

图 2-11 F4 提取

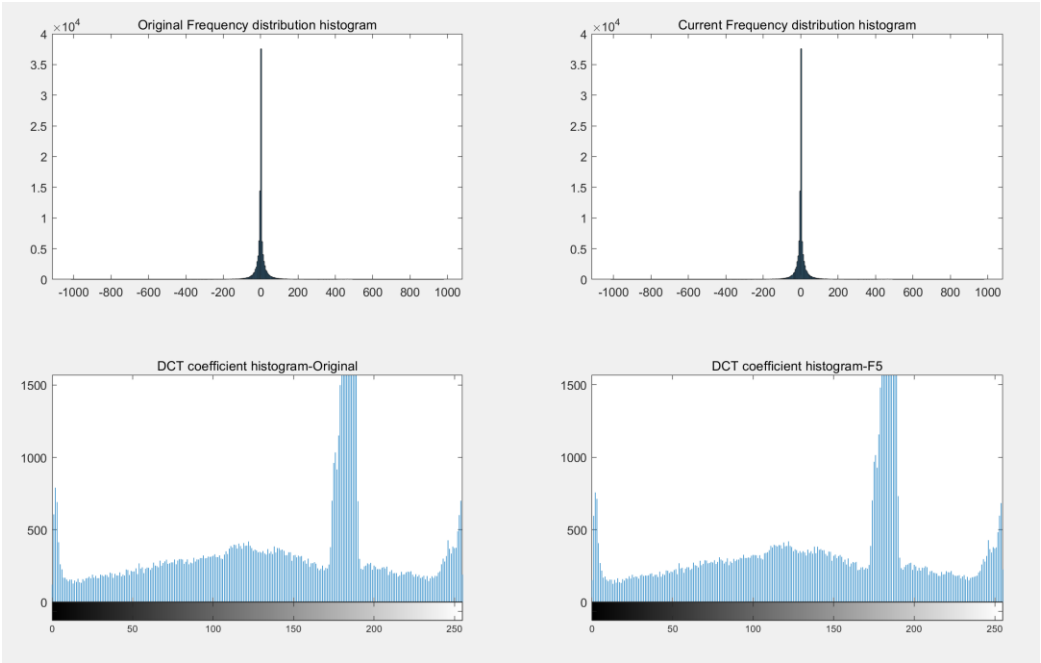

图 2-12 F5 嵌入

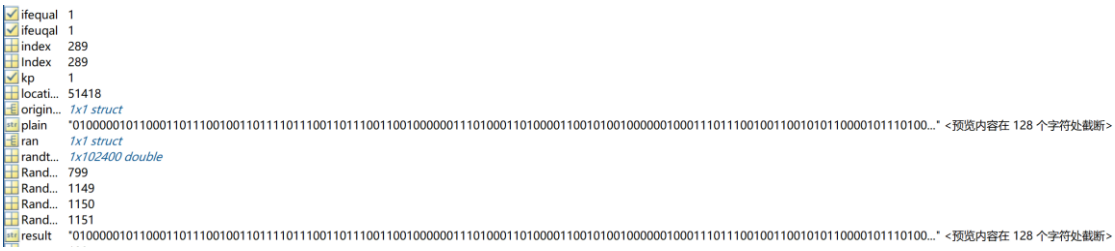

图 2-13 F5 提取

2.4 实验小结

本次实验主要是为了理解图片隐藏的两种算法,算法比较简单,但由于是第一次接触matlab,也花了不少时间摸索.通过这次实验,深刻理解了两种算法的过程并且逐步熟悉了 matlab 的使用.

个人原理总结已发在个人博客中:

https://blog.csdn.net/POTASSIUM711/article/details/105052334

代码已在 Github 上开源:

https://github.com/IRIDIUM-SUB/Steg_Experiment_HUST/tree/master/Jsteg

3 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							

4 附录 A 基于 LSB 的空域信息隐藏实现的源程序

4.1 嵌入

```
Picture=imread('kinkakuji01.bmp');
Double_Picture=Picture;
Double_Picture=double(Double_Picture);
%wen.txt_id=fopen('wen.txt','r');
%[msg,len]=fread(wen.txt_id,'ubit1');
msg='0100000101100011011100100110111101110011011100110010000001111010001101000011001010010000001000111011100100110010101100001011101000010000001010111011000001011011000011011000010000000101011101100101001000000110001101100001011011100010000010100100110010101100001011000110111010000010000000100010101110110011001010111001001111001001000000100001101101111011100100110111001100100101011100100010000011011111011001100010000001110100011010000110010100100000010101110110111001111011100100110110000110010000100000010101010011001000110000001100010011011100110001001101100001111000001111000001101100';%Across the Great Wall We can Reach Every Corner of the World U201714886
len=strlength(string(msg));
[m,n]=size(Double_Picture);
p=1;
for f2=1:n

    for f1=1:m
        fprintf("[LOG] Embedding %d %d\t%d\n",f1,f2,Double_Picture(f1,f2));

        Double_Picture(f1,f2)=Double_Picture(f1,f2)-mod(Double_Picture(f1,f2),2)+msg(p);
        if p==len
            break;
        end
        p=p+1;
    end
```

```
    if p==len
        break;
    end
end
Double_Picture=uint8(Double_Picture);
imwrite(Double_Picture,'kinkakuji01-steg.bmp');
subplot(121);imshow(Picture);title('Oroginal Pic');
subplot(122);imshow(Double_Picture);title('Embedded Pic');
```

4.2 提取

```
Picture=imread('kinkakuji01-steg.bmp');
Picture=double(Picture);
[m,n]=size(Picture);
result="";
len=568;
p=1;
for f2=1:n
    for f1=1:m
        if bitand(Picture(f1,f2),1)==1

            result=result+"1";
        else

            result=result+"0";
        end
        if p==len
            break;
        end
        if p<len
            p=p+1;
        end
    end
    if p==len
        break;
    end
end
fprintf("[SUCCESS] Msg=%s\n",result);
```

5 附录 B JPEG 图像变换域信息隐藏实现的源程序

5.1 Jsteg 嵌入

```
A=imread('kinkakuji01.jpg');
I=rgb2gray(A); %RGBÍ¼×ª»¯Î ª»Ò¶ÈÍ¼
imwrite(I,'kinkakuji01g.jpg','quality',100);
originobj= jpeg_read('kinkakuji01g.jpg');
activedobj=originobj;
subplot(2,2,1)
histogram(originobj.coef_arrays{1,1});%display Frequency distribution histog
ram
title("Original Frequency distribution histogram");
plain='0100000101100011011100100110111101110011011100110010000001110100011010
0000110010100100000010001110111001001100101011000010111010000100000010101110
1100001011011000110110000100000010101110110010100100000011000110110000010110
1100010000001010010011001010110000101100011011010000010000001000101011101100
1100101011100100111100100100000010000110110111101110010011011100110010101110
0100010000001101111011001100010000001110100011010000011001010010000001010111
1101111011100100110110001100100001000000101010100110010001100000011000100110
1110011000100110100001110000001110000011010';%Across the Great Wall We can R
each Every Corner of the World U201714886
ran=rng(7742085);%Seed No repeat in this seed

%DCT Table's Size=256*400
randtable=randperm(256*400);
FrequencyStatic=tabulate(randtable(:));
RandTableIndex=1;
index=1;
while index<=length(plain)
    %Get Location
    location=randtable(RandTableIndex);
    %Try to Resolve
    row=ceil(location/400);
    column=mod(location,400)+1;
    %Try to Replace
```

```
24.     if activedobj.coef_arrays{1,1}(row,column)==0||activedobj.coef_arrays{1,
    1}(row,column)==1
25.         %Fail to Fetch, Throw
26.         RandTableIndex=RandTableIndex+1;%Fetch the Next One
27.         continue
28.     end
29.
30.     if plain(index)=='1'
31.
32.         if mod(activedobj.coef_arrays{1,1}(row,column),2)==0%Even
33.
34.             activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arrays{1,
    1}(row,column)+1;%eg. -4->-3,2->3
35.
36.             %fprintf("1,+1");
37.         end
38.         %Odd Number Won't Be Changed
39.         index=index+1;
40.         %Debug
41.         fprintf("1-
    RandTableIndex%d\tLocation%d\n",RandTableIndex,randtable(RandTableIndex));
42.         RandTableIndex=RandTableIndex+1;
43.         continue%Warning: Unknown Bug, Patched temporarily
44.     end
45.
46.     if plain(index)=='0'
47.         location=randtable(RandTableIndex);%Unknown bug occurred. Patch it t
    emporarily
48.         if mod(activedobj.coef_arrays{1,1}(row,column),2)==1%Odd
49.             activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arrays{1,
    1}(row,column)-1;%eg.-3->-4,3->2
50.             %fprintf("0,-1");
51.         end
52.         %Even Number Won't Be Changed
53.         index=index+1;
54.         %Debug
55.         fprintf("0-
    RandTableIndex%d\tLocation%d\n",RandTableIndex,randtable(RandTableIndex));
56.         RandTableIndex=RandTableIndex+1;
57.     end
58.
```

```
end
subplot(2,2,2)
histogram(activedobj.coef_arrays{1,1});%display Frequency distribution histo
gram
title("Current Frequency distribution histogram");

result=isequal(originobj.coef_arrays{1,1},activedobj.coef_arrays{1,1});
jpeg_write(activedobj,'kinkakuji-Jsteg.jpg');
%Show DCT coefficient histogram
Aft=imread('kinkakuji-Jsteg.jpg');
subplot(2,2,3)
imhist(I);
title("DCT coefficient histogram-1");

subplot(2,2,4)
imhist(Aft);
 title("DCT coefficient histogram-2");
```

5.2 Jsteg 提取

```
 %Read image
originobj= jpeg_read('kinkakuji-Jsteg.jpg');
I=imread('kinkakuji-Jsteg.jpg');
activedobj=originobj;

ran=rng(7742085);%Seed No repeat in this seed
result="";%Store 0-1 string
randtable=randperm(256*400);%Shuffle
FrequencyStatic=tabulate(randtable(:));
RandTableIndex=1;
Index=1;

%Debug


while Index<=576%Len of plain
    %Get Location
    location=randtable(RandTableIndex);
    %Try to Resolve
    row=ceil(location/400);
```

```
    column=mod(location,400)+1;
    %Try
    if activedobj.coef_arrays{1,1}(row,column)==0||activedobj.coef_arrays{1,1}(row,column)==1
        %Fail to Fetch, Throw
        RandTableIndex=RandTableIndex+1;%Fetch the Next One
        continue
    end
    %For any vavild cell
    steg=activedobj.coef_arrays{1,1}(row,column);
    %Debug

    fprintf("%d-%d-RandTableIndex%d\tLocation%d\n",steg,mod(steg,2),RandTableIndex,location);
    %Recover
    %steg=odd->1
    %steg=even->0
    result=result+int2str(mod(steg,2));

    RandTableIndex=RandTableIndex+1;
    Index=Index+1;
end
9.  plain='0100000101100011011110010011011110111001101110011001000000011101000
1101000011001010010000000100011101110010011001010110000101110100001000000001010
1110110000101101100011011000010000000101011101100101001000000011000110110000010
1101110001000000010100100110010101100001011000110110100000010000000100010101110
1100110010101110010011110010010000000100001101101111011100100110111001100101010
1110010001000000011011110110011000100000001110100011010000110010100100000001010
1110110111101110010011011000110010000100000001010101001100100011000000001100010
01101110011000100110100001110000011100000110110';%Across the Great Wall We can Reach Every Corner of the World U201714886
ifequal=isequal(plain,result);%Final Comparison
```

5.3 F3 嵌入

```
A=imread('kinkakuji01.jpg');
I=rgb2gray(A); %RGBÍ¼×ª»¯Îª»Ò¶ÈÍ¼
imwrite(I,'kinkakuji01g.jpg','quality',100);
originobj= jpeg_read('kinkakuji01g.jpg');
activedobj=originobj;
```

```
subplot(2,2,1)


histogram(originobj.coef_arrays{1,1});%display Frequency distribution histogram
title("Original Frequency distribution histogram");
 plain='0100000101100011011100100110111101110011011100110010000001110100011010000110001100101001000000100011101110010011001010110000010111010000100000010101110110000010110110001101100001000000101011101100100100000011000110110000101100001011101000010000000101011101100001011000010000001010111011100010000001010010011001010110000010110001101101000001000000100010101110110011010111001001111001001000000100001101101111011100100110111001100101011100100010000001101111011001100010000001110100011010000110010100100000001010111010111101110010011011000110010000100000010101010011001000110000001100010011011100010011010000111000001110000011011000011010',%Across the Great Wall We can Reach Every Corner of the World U201714886
ran=rng(7742085);%Seed No repeat in this seed

%DCT Table's Size=256*400
randtable=randperm(256*400);
FrequencyStatic=tabulate(randtable(:));
RandTableIndex=1;
index=1;
while index<=length(plain)
    %Get Location
    location=randtable(RandTableIndex);
    %Try to Resolve
    row=ceil(location/400);
    column=mod(location,400)+1;
    %Try to Replace
    if activedobj.coef_arrays{1,1}(row,column)==0
        %Fail to Fetch, Throw
        fprintf("Fail to fetch,%d\t%d\n",index,RandTableIndex);
        RandTableIndex=RandTableIndex+1;%Fetch the Next One

        continue
    end
    % if plain=1 and even->abs-1
    % if plain=0 and odd->abs-1
    % if plain=1 and odd->Keep
    % if plain=0 and even->Keep
    % if plain=0 and 1/-1->Pass
```

```
    if plain(index)=='1'
        if mod(activedobj.coef_arrays{1,1}(row,column),2)==0
            if activedobj.coef_arrays{1,1}(row,column)>0
                activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arrays{1,1}(row,column)-1;
            else
                activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arrays{1,1}(row,column)+1;
            end
        fprintf("1-RandTableIndex%d\tLocation%d\n",RandTableIndex,randtable(RandTableIndex));
        end
        index=index+1;
        RandTableIndex=RandTableIndex+1;
    else
        if activedobj.coef_arrays{1,1}(row,column)==1||activedobj.coef_arrays{1,1}(row,column)==-1
            %Pass
            %Shrink to 0
            activedobj.coef_arrays{1,1}(row,column)=0;
             RandTableIndex=RandTableIndex+1;
             fprintf("(1/-1,0),Fail to fatch,%d\t%d\n",RandTableIndex,randtable(RandTableIndex));
             continue;
        end
        if  mod(activedobj.coef_arrays{1,1}(row,column),2)==1
            if activedobj.coef_arrays{1,1}(row,column)>0
                activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arrays{1,1}(row,column)-1;
            else
                activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arrays{1,1}(row,column)+1;
            end

        end
        fprintf("0-RandTableIndex%d\tLocation%d\n",RandTableIndex,randtable(RandTableIndex));
        index=index+1;
        RandTableIndex=RandTableIndex+1;

    end
```

```
end
subplot(2,2,2)
histogram(activedobj.coef_arrays{1,1});%display Frequency distribution histo
gram
title("Current Frequency distribution histogram");

result=isequal(originobj.coef_arrays{1,1},activedobj.coef_arrays{1,1});
jpeg_write(activedobj,'kinkakuji-F3.jpg');
%Show DCT coefficient histogram
Aft=imread('kinkakuji-F3.jpg');
subplot(2,2,3)
imhist(I);
title("DCT coefficient histogram-Original");

subplot(2,2,4)
imhist(Aft);
 title("DCT coefficient histogram-F3");
```

5.4 F3 提取

```
 %Read image
originobj= jpeg_read('kinkakuji-F3.jpg');
I=imread('kinkakuji-Jsteg.jpg');
activedobj=originobj;
ran=rng(7742085);%Seed No repeat in this seed
result="";%Store 0-1 string
randtable=randperm(256*400);%Shuffle
FrequencyStatic=tabulate(randtable(:));
RandTableIndex=1;
Index=1;

%Debug


while Index<=576%Len of plain
    %Get Location
    location=randtable(RandTableIndex);
    %Try to Resolve
    row=ceil(location/400);
    column=mod(location,400)+1;
```

```
    %Try
    if activedobj.coef_arrays{1,1}(row,column)==0
        %Fail to Fetch, Throw
        RandTableIndex=RandTableIndex+1;%Fetch the Next One
        continue
    end
    %For any vavild cell
    steg=activedobj.coef_arrays{1,1}(row,column);
    %Debug

    fprintf("%d-%d-
RandTableIndex%d\tLocation%d\n",steg,mod(steg,2),RandTableIndex,location);
    %Recover
    %steg=odd->1
    %steg=even->0
    result=result+int2str(mod(steg,2));

    RandTableIndex=RandTableIndex+1;
    Index=Index+1;
end
 plain='010000010110001101111001001101111011100110111001100100000011101000110
100001100101001000000100011101110010011001010110000101110100001000000101011
101100001011011000110110000100000010101110110010100100000011000110110000101
101110001000000101001001100101011000010110001101101000001000000100010101110110
011001010111001001111001001000000100001101101111011100100110111001100101011
100100010000000110111101100110001000000111010001101000011001010010000001010111
011011110111001001101100011001000010000000101010100110010001100000011000100
11011100110001001101000011100000011100000110110';%Across the Great Wall We can
Reach Every Corner of the World U201714886
 kp=isequal(plain,result);%Final Comparison
```

5.5 F4 嵌入

```
 A=imread('kinkakuji01.jpg');
I=rgb2gray(A); %RGBÍ¼×ª»¯Îª»Ò¶ÈÍ¼
imwrite(I,'kinkakuji01g.jpg','quality',100);
originobj= jpeg_read('kinkakuji01g.jpg');
activedobj=originobj;
subplot(2,2,1)

```

```
8.
9.  histogram(originobj.coef_arrays{1,1});%display Frequency distribution histog
    ram
10. title("Original Frequency distribution histogram");
11. plain='0100000101100011011100100110111101110011011100110010000001110100011101
    0000110010100100000010001110111001001100101011000010111010000100000010101110
    1100001011011000110110000100000010101110110010100100000011000110110000010110
    1100010000001010010011001010110000101100011011010000010000001000101011101100
    1100101011100100111100100100000010000110110111101110010011011100110010101110
    0100010000001101111011001100010000001110100011010000110010100100000010101110
    1101111011100100110110001100100001000000101010100110010001100000011000100110
    1110011000100110100001110000011100000110110';%Across the Great Wall We can R
    each Every Corner of the World U201714886
12. ran=rng(7742085);%Seed No repeat in this seed
13. uosedindex=[];
14. throwedindex=[];
15. %DCT Table's Size=256*400
16. randtable=randperm(256*400);
17. FrequencyStatic=tabulate(randtable(:));
18. RandTableIndex=1;
19. index=1;
20. while index<=length(plain)
21.     %Get Location
22.     location=randtable(RandTableIndex);
23.     %Try to Resolve
24.     row=ceil(location/400);
25.     column=mod(location,400)+1;
26.     %Try to Replace
27.     if activedobj.coef_arrays{1,1}(row,column)==0
28.         %Fail to Fetch, Throw
29.         fprintf("Fail to fetch,%d\t%d\n",index,RandTableIndex);
30.         RandTableIndex=RandTableIndex+1;%Fetch the Next One
31.
32.         continue
33.     end
34.     % if plain=1 and positive odd->keep
35.     % if plain=1 and positive even->abs-1
36.     % if plain=1 and negative odd->abs-1
37.     % if plain=1 and negative even->keep
38.     %
39.     % if plain=0 and positive odd->abs-1
```

```
    % if plain=0 and positive even->keep
    % if plain=0 and negative odd->keep
    % if plain=0 and negative even->abs-1
    %
    % if plain=0 and 1->Shrinkage
    % if plain=1 and -1->Shrinkage
    if plain(index)=='1'
         if activedobj.coef_arrays{1,1}(row,column)==-1
            %Pass
            %Shrink to 0
            activedobj.coef_arrays{1,1}(row,column)=0;
             RandTableIndex=RandTableIndex+1;
             fprintf("(-
1,1),Fail to fatch,%d\t%d\n",RandTableIndex,randtable(RandTableIndex));
             continue;
         end

        if (mod(activedobj.coef_arrays{1,1}(row,column),2)==0&&activedobj.co
ef_arrays{1,1}(row,column)>0)||(mod(activedobj.coef_arrays{1,1}(row,column),
2)~=0&&activedobj.coef_arrays{1,1}(row,column)<0)
            if activedobj.coef_arrays{1,1}(row,column)>0
                activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arra
ys{1,1}(row,column)-1;
            else
                activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arra
ys{1,1}(row,column)+1;
            end
        end

        fprintf("1-
RandTableIndex%d\tLocation%d\n",RandTableIndex,randtable(RandTableIndex));
        index=index+1;
        RandTableIndex=RandTableIndex+1;
    else
        if activedobj.coef_arrays{1,1}(row,column)==1
            %Pass
            %Shrink to 0
            activedobj.coef_arrays{1,1}(row,column)=0;
             RandTableIndex=RandTableIndex+1;
             fprintf("(1,0),Fail to fatch,%d\t%d\n",RandTableIndex,randtable
(RandTableIndex));
```

```
74.              continue;
75.         end
76.         if  (mod(activedobj.coef_arrays{1,1}(row,column),2)~=0&&activedobj.c
    oef_arrays{1,1}(row,column)>0)||(mod(activedobj.coef_arrays{1,1}(row,column)
    ,2)==0&&activedobj.coef_arrays{1,1}(row,column)<0)
77.             if activedobj.coef_arrays{1,1}(row,column)>0
78.                 activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arra
    ys{1,1}(row,column)-1;
79.             else
80.                 activedobj.coef_arrays{1,1}(row,column)=activedobj.coef_arra
    ys{1,1}(row,column)+1;
81.             end
82.
83.         end
84.         fprintf("0-
    RandTableIndex%d\tLocation%d\n",RandTableIndex,randtable(RandTableIndex));
85.         index=index+1;
86.         RandTableIndex=RandTableIndex+1;
87.
88.     end
89. end
90. subplot(2,2,2)
91. histogram(activedobj.coef_arrays{1,1});%display Frequency distribution histo
    gram
92. title("Current Frequency distribution histogram");
93.
94. result=isequal(originobj.coef_arrays{1,1},activedobj.coef_arrays{1,1});
95. jpeg_write(activedobj,'kinkakuji-F4.jpg');
96. %Show DCT coefficient histogram
97. Aft=imread('kinkakuji-F4.jpg');
98. subplot(2,2,3)
99. imhist(I);
100. title("DCT coefficient histogram-Original");
101.
102. subplot(2,2,4)
103. imhist(Aft);
104. title("DCT coefficient histogram-F4");
```

5.6 F4 提取

```
%Read image
originobj= jpeg_read('kinkakuji-F4.jpg');
I=imread('kinkakuji-Jsteg.jpg');
activedobj=originobj;
ran=rng(7742085);%Seed No repeat in this seed
result="";%Store 0-1 string
randtable=randperm(256*400);%Shuffle
FrequencyStatic=tabulate(randtable(:));
RandTableIndex=1;
Index=1;

%Debug



while Index<=576%Len of plain
    %Get Location
    location=randtable(RandTableIndex);
    %Try to Resolve
    row=ceil(location/400);
    column=mod(location,400)+1;
    %Try
    if activedobj.coef_arrays{1,1}(row,column)==0
        %Fail to Fetch, Throw
        RandTableIndex=RandTableIndex+1;%Fetch the Next One
        continue
    end
    %For any vavild cell
    steg=activedobj.coef_arrays{1,1}(row,column);
    %Debug
    %usedindex(end+1)=location;

    %Recover
    %positive odd and negative even->1
    %positive even and negative odd->0
    if (mod(steg,2)&&steg>0)||(mod(steg,2)==0&&steg<0)
        result=result+"1";
```

```
        fprintf("%d-1-RandTableIndex%d\tLocation%d\n",steg,RandTableIndex,location);
    else
        result=result+"0";
        fprintf("%d-0-RandTableIndex%d\tLocation%d\n",steg,RandTableIndex,location);
    end


    RandTableIndex=RandTableIndex+1;
    Index=Index+1;
end
plain='0100000101100011011100100110111101110011011100110010000001110100011010000001000111011100100110010101100001011101000010000001010111011000010111000011011000010000001010111011001010010000001100011011000010110110000101000000100001010011001010110000101100011011010000010000000100010101110110001011100100100000010000011011011110111001001101110011001010111001000111100100100000100000110111011001000010000001110100011010000011001010010000001010111011100100111011000110010000100000010101010011001000110000001100010011011100100011010000111000001110000011010';%Across the Great Wall We can Reach Every Corner of the World U201714886

ifeuqal=isequal(plain,result);%Final Comparison
```

5.7 F5 嵌入

```
A=imread('kinkakuji01.jpg');
I=rgb2gray(A); %RGBÍ¼×ª»¯Îª»Ò¶ÈÍ¼
imwrite(I,'kinkakuji01g.jpg','quality',100);
originobj= jpeg_read('kinkakuji01g.jpg');
activedobj=originobj;
subplot(2,2,1);


histogram(originobj.coef_arrays{1,1});%display Frequency distribution histogram
title("Original Frequency distribution histogram");
plain='0100000101100011011100100110111101110011011100110010000001110100011010000001000111011100100110010101100001011101000010000001010111011000010111000011011000010000001010111011001010010000001100011011000010110110000101000000100001010011001010110000101100011011010000010000000100010101110110001011100100100000010000011011011110111001001101110011001010111001000111100100100000100000110111011001000010000001110100011010000011001010010000001010111011100100111011000110010000100000010101010011001000110000001100010011011100100011010000111000001110000011010';%Across the Great Wall We can Reach Every Corner of the World U201714886
```

```
    11000100000010100100110010101100000101100011011010000010000001000101011101100
    11001010111001001111001001000000100001101101111011100100110111001100101011100
    01000100000011011110110011000100000011101000110100001100101001000000101011100
    11011110111001001101100011001000010000000101010100110010001100000011000100110
    11100110001001101000011100000011100000110110';%Across the Great Wall We can R
    each Every Corner of the World U201714886

ran=rng(7742085);%Seed No repeat in this seed
uosedindex=[];
throwedindex=[];
%DCT Table's Size=256*400
randtable=randperm(256*400);
FrequencyStatic=tabulate(randtable(:));
RandTableIndex1=1;
RandTableIndex2=2;
RandTableIndex3=3;
index=1;

% For Each 3 DCT coef and 2 plain
while index<=length(plain)/2

    a1=randtable(RandTableIndex1);
    row1=ceil(a1/400);
    column1=mod(a1,400)+1;

    a2=randtable(RandTableIndex2);
    row2=ceil(a2/400);
    column2=mod(a2,400)+1;

    a3=randtable(RandTableIndex3);
    row3=ceil(a3/400);
    column3=mod(a3,400)+1;


    x1=str2num(plain((index-1)*2+1));
    x2=str2num(plain((index-1)*2+2));

    %Check and Drop
    %Pretty Slow But Readable
    if activedobj.coef_arrays{1,1}(row1,column1)==0
        RandTableIndex1=RandTableIndex1+1;
```

```
47.         RandTableIndex2=RandTableIndex2+1;
48.         RandTableIndex3=RandTableIndex3+1;
49.         fprintf("[LOG] a1=0, move,%d\t%d\t%d\n",RandTableIndex1,RandTableInd
    ex2,RandTableIndex3);
50.         continue;%Restart
51.     else
52.         if activedobj.coef_arrays{1,1}(row2,column2)==0
53.         RandTableIndex2=RandTableIndex2+1;
54.         RandTableIndex3=RandTableIndex3+1;
55.         fprintf("[LOG] a2=0, move,%d\t%d\t%d\n",RandTableIndex1,RandTableInd
    ex2,RandTableIndex3);
56.         continue;%Restart
57.         else
58.             if activedobj.coef_arrays{1,1}(row3,column3)==0
59.                 RandTableIndex3=RandTableIndex3+1;
60.                 fprintf("[LOG] a3=0, move,%d\t%d\t%d\n",RandTableIndex1,Rand
    TableIndex2,RandTableIndex3);
61.                 continue;%Restart
62.             end
63.         end
64.     end
65.
66.     %Now it should be 3 non-0 DCT coef
67.     fprintf("[READY] Current DCT Coef Group:%d\t%d\t%d\n",activedobj.coef_ar
    rays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row2,column2),activedobj
    .coef_arrays{1,1}(row3,column3));
68.     if mod(bitxor(activedobj.coef_arrays{1,1}(row1,column1),activedobj.coef_
    arrays{1,1}(row3,column3),'int16'),2)==x1&&mod(bitxor(activedobj.coef_arrays
    {1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2)==x
    2%p && q?keep:pass;
69.         fprintf("[LOG] No Change,%d\t%d\t%d \t %d\t%d\n",activedobj.coef_arr
    ays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row2,column2),activedobj.
    coef_arrays{1,1}(row3,column3),x1,x2);
70.     else
71.         if mod(bitxor(activedobj.coef_arrays{1,1}(row1,column1),activedobj.c
    oef_arrays{1,1}(row3,column3),'int16'),2)~=x1&&mod(bitxor(activedobj.coef_ar
    rays{1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2
    )==x2%!p&&q?Change a1:pass;
72.             fprintf("[LOG] Change a1,%d\t%d\t%d \t %d\t%d\n",activedobj.coef
    _arrays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row2,column2),actived
    obj.coef_arrays{1,1}(row3,column3),x1,x2);
```

```
            if activedobj.coef_arrays{1,1}(row1,column1)>0
                activedobj.coef_arrays{1,1}(row1,column1)=activedobj.coef_arrays{1,1}(row1,column1)-1;
            else
                activedobj.coef_arrays{1,1}(row1,column1)=activedobj.coef_arrays{1,1}(row1,column1)+1;
            end
        else
            if mod(bitxor(activedobj.coef_arrays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2)==x1&&mod(bitxor(activedobj.coef_arrays{1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2)~=x2%p&&!q?Change a2:pass;
                fprintf("[LOG] Change a2,%d\t%d\t%d \t %d\t%d\n",activedobj.coef_arrays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3),x1,x2);
                if activedobj.coef_arrays{1,1}(row2,column2)>0
                activedobj.coef_arrays{1,1}(row2,column2)=activedobj.coef_arrays{1,1}(row2,column2)-1;
                else
                activedobj.coef_arrays{1,1}(row2,column2)=activedobj.coef_arrays{1,1}(row2,column2)+1;
                end
            else
                if mod(bitxor(activedobj.coef_arrays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2)~=x1&&mod(bitxor(activedobj.coef_arrays{1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2)~=x2%!p&&!q?Change a3:pass;
                    fprintf("[LOG] Change a3,%d\t%d\t%d \t %d\t%d\n",activedobj.coef_arrays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3),x1,x2);
                    if activedobj.coef_arrays{1,1}(row3,column3)>0
                        activedobj.coef_arrays{1,1}(row3,column3)=activedobj.coef_arrays{1,1}(row3,column3)-1;
                    else
                        activedobj.coef_arrays{1,1}(row3,column3)=activedobj.coef_arrays{1,1}(row3,column3)+1;
                    end
                end
            end
        end
    end
```

```
98.
99.     %Shrinkage
100.     if activedobj.coef_arrays{1,1}(row1,column1)==0
101.         fprintf("[LOG] %d\tShrinkage a1,%d\t%d\t%d\n",index,RandTableIndex1
    ,RandTableIndex2,RandTableIndex3);
102.         RandTableIndex1=RandTableIndex2;
103.         RandTableIndex2=RandTableIndex3;
104.         RandTableIndex3=RandTableIndex3+1;
105.         continue;
106.     else
107.         if activedobj.coef_arrays{1,1}(row2,column2)==0
108.             fprintf("[LOG] %d\tShrinkage a2,%d\t%d\t%d\n",index,RandTableIn
    dex1,RandTableIndex2,RandTableIndex3);
109.             RandTableIndex2=RandTableIndex3;
110.             RandTableIndex3=RandTableIndex3+1;
111.             continue;
112.         else
113.             if  activedobj.coef_arrays{1,1}(row3,column3)==0
114.                 fprintf("[LOG] %d\tShrinkage a3%d\t%d\t%d\n",index,RandTabl
    eIndex1,RandTableIndex2,RandTableIndex3);
115.                 RandTableIndex3=RandTableIndex3+1;
116.                 continue;
117.             end
118.         end
119.     end
120.     %Adjust DCT Coef Index
121.     %Start Over
122.     RandTableIndex1=RandTableIndex3+1;
123.     RandTableIndex2=RandTableIndex1+1;
124.     RandTableIndex3=RandTableIndex2+1;
125.     index=index+1;
126.     fprintf("[SUCCESS] Embedding Success:Next Group:%d\t%d\t%d\t%d\n",RandT
    ableIndex1,RandTableIndex2,RandTableIndex3,index);
127. end
128. subplot(2,2,2)
129. histogram(activedobj.coef_arrays{1,1});%display Frequency distribution hist
    ogram
130. title("Current Frequency distribution histogram");
131.
132. result=isequal(originobj.coef_arrays{1,1},activedobj.coef_arrays{1,1});
133. jpeg_write(activedobj,'kinkakuji-F5.jpg');
```

```
%Show DCT coefficient histogram
Aft=imread('kinkakuji-F5.jpg');
subplot(2,2,3)
imhist(I);
title("DCT coefficient histogram-Original");

subplot(2,2,4)
imhist(Aft);
title("DCT coefficient histogram-F5");
```

5.8 F5 提取

```
%Read image
originobj= jpeg_read('kinkakuji-F5.jpg');
I=imread('kinkakuji-Jsteg.jpg');
activedobj=originobj;
ran=rng(7742085);%Seed No repeat in this seed
result="";%Store 0-1 string
randtable=randperm(256*400);%Shuffle
FrequencyStatic=tabulate(randtable(:));
RandTableIndex1=1;
RandTableIndex2=2;
RandTableIndex3=3;
Index=1;

%Debug

while Index<=576/2%Len of plain
    a1=randtable(RandTableIndex1);
    row1=ceil(a1/400);
    column1=mod(a1,400)+1;

    a2=randtable(RandTableIndex2);
    row2=ceil(a2/400);
    column2=mod(a2,400)+1;

    a3=randtable(RandTableIndex3);
    row3=ceil(a3/400);
    column3=mod(a3,400)+1;

```

```
    if activedobj.coef_arrays{1,1}(row1,column1)==0
        RandTableIndex1=RandTableIndex1+1;
        RandTableIndex2=RandTableIndex2+1;
        RandTableIndex3=RandTableIndex3+1;
        fprintf("[LOG] a1=0, move,%d-%d-%d\n",RandTableIndex1,RandTableIndex2,RandTableIndex3);
        continue;%Restart
    else
        if activedobj.coef_arrays{1,1}(row2,column2)==0
        RandTableIndex2=RandTableIndex2+1;
        RandTableIndex3=RandTableIndex3+1;
        fprintf("[LOG] a2=0, move,%d-%d-%d\n",RandTableIndex1,RandTableIndex2,RandTableIndex3);
        continue;%Restart
        else
            if activedobj.coef_arrays{1,1}(row3,column3)==0
                RandTableIndex3=RandTableIndex3+1;
                fprintf("[LOG] a3=0, move,%d-%d-%d\n",RandTableIndex1,RandTableIndex2,RandTableIndex3);
                continue;%Restart
            end
        end
    end
    %Now it should be 3 non-0 DCT coef
    fprintf("[READY] Current DCT Coef Group:%d-%d-%d\n",activedobj.coef_arrays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3));
    result=result+num2str(mod(bitxor(activedobj.coef_arrays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2));
    fprintf("[INFO] %d\tReveal[1]:%d\t%d->%d\n",Index,activedobj.coef_arrays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row3,column3),mod(bitxor(activedobj.coef_arrays{1,1}(row1,column1),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2));
    result=result+num2str(mod(bitxor(activedobj.coef_arrays{1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2));
    fprintf("[INFO] %d\tReveal[2]:%d\t%d->%d\n",Index,activedobj.coef_arrays{1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3),mod(bitxor(activedobj.coef_arrays{1,1}(row2,column2),activedobj.coef_arrays{1,1}(row3,column3),'int16'),2));
    %Prepare for the next Group
    RandTableIndex1=RandTableIndex3+1;
```

```
57.     RandTableIndex2=RandTableIndex1+1;
58.     RandTableIndex3=RandTableIndex2+1;
59.     Index=Index+1;
60.     fprintf("[SUCCESS] Revaling Success:Next Group:%d-%d-%d\t%d\n",RandTable
    Index1,RandTableIndex2,RandTableIndex3,Index);
61. end
62. plain='0100000101100011011100100110111101110011011100110010000001110100011010
    0000110010100100000010001110111001001100101011000010111010000100000010101110
    1100001011011000110110000100000010101110110010100100000011000110110000010110
    1110001000000101001001100101011000010110001101101000001000000100010101110110
    1100101011100100111100100100000010000110110111101110010011011100110010101110
    0100010000001101111011001100010000001110100011010000110010100100000010101110
    1101111011100100110110001100100001000000101010100110010001100000011000100110
    1110011000100110100001110000011100000110110';%Across the Great Wall We can R
    each Every Corner of the World U201714886
63.
64. ifeuqal=isequal(plain,result);%Final Comparison
```