

## 实验三：DCT 系数隐写

综合评分：

## 【实验目的】：

掌握 MATLAB 函数编写，能够编写函数基于图像 DCT 实现二点法和三点法的嵌入和提取算法。

## 【实验内容】：（请将你实验完成的项目涂“■”）

- 一、分析隐蔽载体在 JPEG 压缩条件下的健壮参数  $a$  与隐藏鲁棒性的关系（两点法）。  
■二、编写两个函数，分别用来处理三点法的嵌入和提取操作。适应任意载体图像。

## 【实验工具及平台】：

☒ Windows+Matlab☐ 其它：（请注明）\_\_\_\_\_

## 【实验涉及到的相关算法】：

- 1、与实验内容选择的项目对应；
- 2、请使用流程图、伪代码、NS 图或文字方式描述，不要完全贴代码

■一、分析隐蔽载体在 JPEG 压缩条件下的健壮参数  $a$  与隐藏鲁棒性的关系（两点法）。

隐藏过程如下：

- 1.将载体图像分为  $8 \times 8$  的块，做二维 DCT 变换
- 2.用  $(u_1, v_1)(u_2, v_2)$  来表示选中的两个系数的坐标
- 3.对于第  $i$  bit 秘密信息  
if（要隐藏信息 ‘1’）  
    make  $B_i(u_1, v_1) > B_i(u_2, v_2)$   
else  
    make  $B_i(u_1, v_1) < B_i(u_2, v_2)$

- 4.逆 DCT 变换，还原图像

## function

```
[count,msg,data,result,DCTrgb]=hidedctadv(image,imagegoal,msg,key,alpha)
%按位读取秘密信息
frr=fopen(msg,'r');
[msg,count]=fread(frr,'ubit1');
fclose(frr);
data0=imread(image);
%将图象矩阵转为 double 型
data0=double(data0)/255;
%取图象的一层做隐藏
data=data0(:,:,1);
%对图象分块
T=dctmtx(8);
```

*%对分块图像做DCT变换*

```
DCTrgb=blkproc(data,[8 8],'P1*x*P2',T,T');
```

```
DCTrgb0=DCTrgb;
```

*%产生随机的块选择,确定图像块的首地址*

```
[row,col]=size(DCTrgb);
```

```
row=floor(row/8);
```

```
col=floor(col/8);
```

```
a=zeros([row col]);
```

```
[k1,k2]=randinterval(a,count,key);
```

```
for i=1:count
```

```
    k1(1,i)=(k1(1,i)-1)*8+1;
```

```
    k2(1,i)=(k2(1,i)-1)*8+1;
```

```
end
```

*%信息嵌入*

```
temp=0;
```

```
for i=1:count
```

```
    if msg(i,1)==0
```

```
        if DCTrgb(k1(i)+4,k2(i)+1)>DCTrgb(k1(i)+3,k2(i)+2)
```

```
            temp=DCTrgb(k1(i)+4,k2(i)+1);
```

```
            DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+3,k2(i)+2);
```

```
            DCTrgb(k1(i)+3,k2(i)+2)=temp;
```

```
        end
```

```
    else
```

```
        if DCTrgb(k1(i)+4,k2(i)+1)<DCTrgb(k1(i)+3,k2(i)+2)
```

```
            temp=DCTrgb(k1(i)+4,k2(i)+1);
```

```
            DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+3,k2(i)+2);
```

```
            DCTrgb(k1(i)+3,k2(i)+2)=temp;
```

```
        end
```

```
    end
```

```
    if DCTrgb(k1(i)+4,k2(i)+1)>DCTrgb(k1(i)+3,k2(i)+2)
```

```
        DCTrgb(k1(i)+3,k2(i)+2)=DCTrgb(k1(i)+3,k2(i)+2)-alpha;%将原本小的系数调整得更小
```

```
    else
```

```
        DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+4,k2(i)+1)-alpha;
```

```
    end
```

```
end
```

*%信息写回保存*

```
DCTrgb1=DCTrgb;
```

```
data=blkproc(DCTrgb,[8 8],'P1*x*P2',T',T);
```

```
result=data0;
```

```
result(:, :, 1)=data;
```

```
imwrite(result,imagegoal);
```

提取过程:

接收者进行二维 DCT 变换，比较每一块中约定位置的 DCT 系数值，根据其相对大小，得到隐藏信息的比特串，从而恢复出秘密信息。

```
function result=extractdctadv(image,msg,key,count)
data0=imread(image);
data0=double(data0)/255;
%用图象第一层做提取
data=data0(:,:,1);
%分块做 DCT 变换
T=dctmtx(8);
DCTcheck=blkproc(data,[8 8],'P1*x*P2',T,T');
%产生随机的块选择,确定图像块的首地址
[row,col]=size(DCTcheck);
row=floor(row/8);
col=floor(col/8);
a=zeros([row col]);
[k1,k2]=randinterval(a,count,key);
for i=1:count
    k1(1,i)=(k1(1,i)-1)*8+1;
    k2(1,i)=(k2(1,i)-1)*8+1;
end
%准备提取并回写信息
frr=fopen(msg,'a');
for i=1:count
    if DCTcheck(k1(i)+4,k2(i)+1)<=DCTcheck(k1(i)+3,k2(i)+2)
        result(i,1)=0;
    else
        result(i,1)=1;
    end
end
end
fwrite(frr,result,'ubit1');
fclose(frr);
```

下面函数将自动取  $0.1 \sim 1$  十个等差为 0.1 的值作为控制阈值  $\alpha$ ，分别对同一文件进行隐藏。然后对隐藏结果进行压缩质量从 10% ~ 100% 十次 JPEG 压缩并分别从压缩后的结果中提取消息。比较每次提取的消息与原始秘密信息，将误码率反映到一组曲线上。

```
function result=jpgandalph(test,msg)
%定义压缩质量比从 10%到100%
quality=10:10:100;
alpha=0.1:0.1:1;
result=zeros([max(size(alpha)) max(size(quality))]');
resulttr=0;
resulttc=0;
for a=alpha
```

```
resultr=resultr+1;
[count,message,hideresult]=hidedctadv(test,'temp.jpg',msg,2019,a);
resultc=0;
different=0;
for q=quality
    resultc=resultc+1;
    imwrite(hideresult,'temp.jpg','jpg','quality',q);
    msgextract=extractdctadv('temp.jpg','temp.txt',2019,count);
    for i=1:count
        if message(i,1)~=msgextract(i,1)
            different=different+1;
        end
    end
    result(resultr,resultc)=different/count;
    different=0;
end
disp(['完成了第',int2str(resultr),'个(共 10 个) $\alpha$  的鲁棒性测试,请等待...']);
end
%return
for i=1:10
    plot(quality,result(i,:));
    hold on;
end
xlabel('jpeg 压缩率');
ylabel('提取的信息与原始信息不同的百分比例');
title('控制阈值  $\alpha$  在 JPEG 条件对隐藏鲁棒性的影响')
```

二、编写两个函数，分别用来处理三点法的嵌入和提取操作。适应任意载体图像。

选取三个位置：

嵌入 1 时：

$$\text{令 } B_i(u_1, v_1) > B_i(u_3, v_3) + D \quad B_i(u_2, v_2) > B_i(u_3, v_3) + D$$

嵌入 0 时：

$$\text{令 } B_i(u_1, v_1) < B_i(u_3, v_3) - D \quad B_i(u_2, v_2) < B_i(u_3, v_3) - D$$

只需将嵌入和提取部分代码改为：

```
%信息嵌入
%1,2,3 分别为(2,3)(3,1)(4,1)
temp=0;
for i=1:count
    if msg(i,1)==1
        if DCTrgb(k1(i)+4,k2(i)+1)>DCTrgb(k1(i)+2,k2(i)+3)
            temp=DCTrgb(k1(i)+4,k2(i)+1);
```

```
DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+2,k2(i)+3);
DCTrgb(k1(i)+2,k2(i)+3)=temp;
end
if DCTrgb(k1(i)+4,k2(i)+1)>DCTrgb(k1(i)+3,k2(i)+1)
    temp=DCTrgb(k1(i)+4,k2(i)+1);
    DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+3,k2(i)+1);
    DCTrgb(k1(i)+3,k2(i)+1)=temp;
end

%嵌入 0
else
    if DCTrgb(k1(i)+4,k2(i)+1)<DCTrgb(k1(i)+2,k2(i)+3)
        temp=DCTrgb(k1(i)+4,k2(i)+1);
        DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+2,k2(i)+3);
        DCTrgb(k1(i)+2,k2(i)+3)=temp;
    end
    if DCTrgb(k1(i)+4,k2(i)+1)<DCTrgb(k1(i)+3,k2(i)+1)
        temp=DCTrgb(k1(i)+4,k2(i)+1);
        DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+3,k2(i)+1);
        DCTrgb(k1(i)+3,k2(i)+1)=temp;
    end
end

if DCTrgb(k1(i)+4,k2(i)+1)>DCTrgb(k1(i)+2,k2(i)+3) &&
DCTrgb(k1(i)+4,k2(i)+1)>DCTrgb(k1(i)+3,k2(i)+1)
    DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+4,k2(i)+1)+d;%将原本小的系数调
整得更小
elseif DCTrgb(k1(i)+4,k2(i)+1)<DCTrgb(k1(i)+2,k2(i)+3) &&
DCTrgb(k1(i)+4,k2(i)+1)<DCTrgb(k1(i)+3,k2(i)+1)
    DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+4,k2(i)+1)-d;
end
end
```

提取：

```
%准备提取并回写信息
frr=fopen(msg,'a');
for i=1:count
    if DCTcheck(k1(i)+4,k2(i)+1)<=DCTcheck(k1(i)+2,k2(i)+3) &&
DCTcheck(k1(i)+4,k2(i)+1)<=DCTcheck(k1(i)+3,k2(i)+1)
        result(i,1)=1;
    else
        result(i,1)=0;
    end
end
end
```

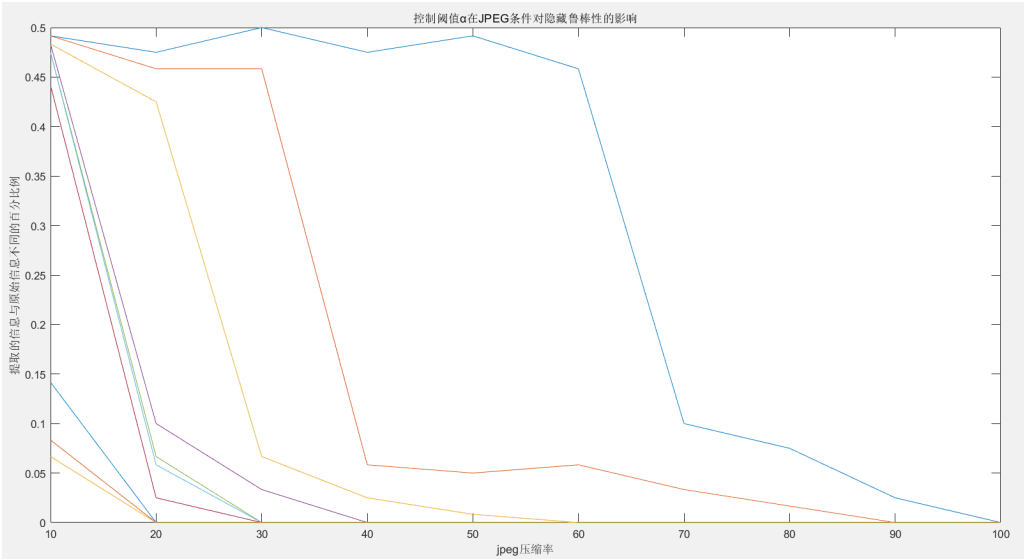
【实验分析】:

1、 请尽量使用曲线图、表等反映你的实验数据及性能

2、 对照实验数据从理论上解释原因

3、 如无明显必要，请不要大量粘贴实验效果图

■一、分析隐蔽载体在 JPEG 压缩条件下的健壮参数  $\alpha$  与隐藏鲁棒性的关系 （两点法）。  
原始图像：嵌入后图像：



	1	2	3	4	5	6	7	8	9	10
1	0.4917	0.4750	0.5000	0.4750	0.4917	0.4583	0.1000	0.0750	0.0250	0
2	0.4917	0.4583	0.4583	0.0583	0.0500	0.0583	0.0333	0.0167	0	0
3	0.4833	0.4250	0.0667	0.0250	0.0083	0	0	0	0	0
4	0.4833	0.1000	0.0333	0	0	0	0	0	0	0
5	0.4750	0.0667	0	0	0	0	0	0	0	0
6	0.4750	0.0583	0	0	0	0	0	0	0	0
7	0.4417	0.0250	0	0	0	0	0	0	0	0
8	0.1417	0	0	0	0	0	0	0	0	0
9	0.0833	0	0	0	0	0	0	0	0	0
10	0.0667	0	0	0	0	0	0	0	0	0

鲁棒性分析：

$\alpha$  取 0.1 时, 完全不能保证信息提取的正确性,  $\alpha$  在[ 0.2, 0.4] 区间内 的抗 JPEG 压缩性能一般, 当  $\alpha$  取 1 时, 基本可以认为是不受 JPEG 压缩干扰的。

$\alpha$  越大鲁棒性越好, 对图像的破坏越大。  $\alpha$  越小鲁棒性越差, 对图像的破坏越小。

加框( JPEG 压缩率为 50% ) 的列下实际信息提取的内容。可以发现, 在  $\alpha > 0.3$  时, 信息开始变得可读了。

## ■二、编写两个函数, 分别用来处理三点法的嵌入和提取操作。适应任意载体图像。

结果图像和两点法基本一致, 这里不再贴图。

其中参数 D 越大, 隐藏算法对于图像处理就越健壮, 但是对图像的改动就越大, 越容易引起察觉。

D=1:



D=5:



D=10:

D=100:



