

DCT变换

DCT原理

DCT和反DCT

根据DCT系数实现信息隐蔽

Code

randinterval

hidedctadv

extractdctadv

实验思考

DCT原理

DCT是一种实数域变换，其变换核为实数余弦函数，图像处理运用的是二维离散余弦变换对图像进行DCT，可以使得图像的重要可视信息都集中在DCT的一小部分系数中。

DCT 变换以后高频成分的系数一般很低，所以一般将水印信息嵌入在系数比较低的地方。嵌入后，再IDCT变换为图像，得到的图像与原图像相比，凭肉眼是很难觉察到差异的。

DCT和反DCT

$$B_{pq} = a_p a_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}$$
$$0 \leq p \leq M-1 \quad 0 \leq q \leq N-1$$

其中：

$$a_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases}, a_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N-1 \end{cases}$$

数值 B_{pq} 称为 A 的 DCT 系数。

逆 DCT 变换定义如下：

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} a_p a_q B_{pq} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}$$
$$0 \leq m \leq M-1 \quad 0 \leq n \leq N-1$$

上图中的公式对一个M*N的矩阵A进行了二维DCT变换和反DCT变换

根据DCT系数实现信息隐蔽

信息隐秘的思想是：通过调整图像块中两个 DCT 系数的相对大小来对密信息进行编码。为了在一幅图像中隐藏尽可能多的秘密信息，我们需要把图像分块，每一块中编码一个秘密信息。为了与JPEG压缩方案相一致，一般选择8x8的图像块。对于每一个图像块，在8x8的64个系数中选择两个系数a,b是有讲究的：我们需要考虑的是如何才能使这两个位置上的数据在图像经过处理后保持不变，至少是变化不大，因此选择应该遵从以下几点原则：

1. 量化系数一致，充分表明了位于这两个位置的 DCT 系数在数量级上是一致的，从而保证算法的实施。
2. 这两个系数相应于DCT的中频部分，从而兼顾机密信息的不可见性和鲁棒性。
3. 引入一个控制量，保证两个DCT值的差不会过小（插值过小可能会让图片上的微小扰动造成隐秘消息有误）

但同时要注意到，正是由于这样的一对系数大小相差很少，往往难以保证隐秘图像在保存、信道上传输以及提取信息时再次被读取等过程中不发生变化。我们任意读取一个块中的这两个位置的系数观察发现： $B(u_1, v_1) = 0.0044833$, $B(u_2, v_2) = 0.0018974$ 。显然，即使是像 10^{-3} 这么微小的变化都可以导致隐秘信息的丢失，这是不允许的。这两个系数的相对大小发生改变，将直接影响编码的正确性，因此，我们引入一个控制量 α 对系数差值进行放大。在编码的过程中，无论是 $B_i(u_1, v_1) > B_i(u_2, v_2)$ 或是 $B_i(u_1, v_1) < B_i(u_2, v_2)$ 我们都要使得 $|B_i(u_1, v_1) - B_i(u_2, v_2)| > \alpha$ ，这样，即使在变换过程中系数的值有轻微的改变，也不会影响编码的正确性。

Code

randinterval

```
1 function [row,col]=randinterval(matrix,count,key)
2 [m,n]=size(matrix);
3 interval1=floor(m*n/count)+1;
4 interval2=interval1-2;
5 if interval2==0
6     error('载体小不能秘信息藏进去');
7 end
8 rand('seed',key);
9 a=rand(1,count);
10 row=zeros([1 count]);
11 col=zeros([1 count]);
12 r=1;
13 c=1;
14 row(1,1)=r;
15 col(1,1)=c;
16 for i=2:count
17     if a(i)>=0.5
18         c=c+interval1;
19     else
20         c=c+interval2;
21     end
22     if c>n
23         r=r+1;
24         if r>m
25             error('载体小不能秘信息藏进去');
26         end
27         c=mod(c,n);
28         if c==0
29             c=1;
30         end
31     end
32     row(1,i)=r;
33     col(1,i)=c;
34 end
```

为了让DCT变换更加安全，我们要使用随机序列控制信息嵌入位，而上述算法具体使用的是随机间隔法。随机间隔的思想较为简单：确定要嵌入值的个数count，生成与其对应数量的（0，1）间随机数序列a，将a中的每一个随机值与0.5比较大小，若大于则取间隔interval1，若小于则取间隔interval2，其中interval1、interval2的选取都依赖于图片整体大小和嵌入值个数，保证不会溢出。

hidedctadv

```
1 function [count,msg,result]=hidedctadv(image,imagegoal,msg,key,alpha)
2 frr=fopen(msg,'r');
3 [msg,count]=fread(frr,'ubit1'); % 按位读取秘密信息
4 fclose(frr);
5 data0=imread(image);
6 data0=double(data0)/255; % 将图像矩阵转为double型
7 data=data0(:,:,1); % 取图像的一层做隐藏
8 T=dctmtx(8); % 做图像分块
9 DCTrgb=blkproc(data,[8 8],'P1*x*P2',T,T'); % 对分块图像做DCT变换
10
11 % 产生随机的块选择, 确定图像块的首地址
12 [row,col]=size(DCTrgb);
13 row=floor(row/8);
14 col=floor(col/8);
15 a=zeros([row col]);
16 [k1,k2]=randinterval(a,count,key);
17 for i=1:count
18     k1(1,i)=(k1(1,i)-1)*8+1;
19     k2(1,i)=(k2(1,i)-1)*8+1;
20 end
21
22 % 信息嵌入
23 temp=0;
24 p=0;
25 for i=1:count
26     if msg(i,1)==0
27         if DCTrgb(k1(i)+4,k2(i)+1)>DCTrgb(k1(i)+3,k2(i)+2)
28             temp=DCTrgb(k1(i)+4,k2(i)+1);
29             DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+3,k2(i)+2);
30             DCTrgb(k1(i)+3,k2(i)+2)=temp;
31         end
32     else
33         if DCTrgb(k1(i)+4,k2(i)+1)<DCTrgb(k1(i)+3,k2(i)+2)
34             temp=DCTrgb(k1(i)+4,k2(i)+1);
35             DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+3,k2(i)+2);
36             DCTrgb(k1(i)+3,k2(i)+2)=temp;
37         end
38     end
39     if DCTrgb(k1(i)+4,k2(i)+1)>DCTrgb(k1(i)+3,k2(i)+2)
40         DCTrgb(k1(i)+3,k2(i)+2)=DCTrgb(k1(i)+3,k2(i)+2)-alpha;
41     else
42         DCTrgb(k1(i)+4,k2(i)+1)=DCTrgb(k1(i)+4,k2(i)+1)-alpha;
43     end
44 end
45 DCTrgb1=DCTrgb;
```

```

46 data=blkproc(DCTrgb,[8 8],'P1*x*P2',T',T);
47 result=data0;
48 result(:,:,1)=data;
49 imwrite(result,imagegoal);

```

主要解释DCT变换部分：通过msg(i,1)获取信息流中第i位的取值，如果为0，比较第5个和第6个DCT系数，如果第5个系数大于第6个系数，则交换它们的值；如果为1，比较第5个和第6个DCT系数，如果第5个系数小于第6个系数，则交换它们的值。然后再将第5个或第6个系数减去alpha（alpha是控制隐藏信息强度的参数），这样即可完成一次信息的隐藏。

extractdctadv

extractdctadv.m

Plain Text

复制代码

```

1 function result=extractdctadv(image,msg,key,count)
2 data0=imread(image);
3 data0=double(data0)/255;
4 data=data0(:,:,1);
5 T=dctmtx(8);
6 DCTcheck=blkproc(data,[8 8],'P1*x*P2',T,T');
7 [row,col]=size(DCTcheck);
8 row=floor(row/8);
9 col=floor(col/8);
10 a=zeros([row col]);
11 [k1,k2]=randinterval(a,count,key);
12 for i=1:count
13     k1(1,i)=(k1(1,i)-1)*8+1;
14     k2(1,i)=(k2(1,i)-1)*8+1;
15 end
16 frr=fopen(msg,'w','n','GB2312');
17 for i=1:count
18     if DCTcheck(k1(i)+4,k2(i)+1) < DCTcheck(k1(i)+3,k2(i)+2)
19         fwrite(frr,0,'ubit1');
20         result(i,1)=0;
21     else
22         fwrite(frr,1,'ubit1');
23         result(i,1)=1;
24     end
25 end
26 fclose(frr);

```

由于预先知道了隐藏者用来生成随机数的种子seed和嵌入量count，在提取时很容易就可以知道应该在隐秘图像的哪些块中根据DCT提取信息。对于每块中的信息如果 $DCTcheck(k1(k)+4,k2(k)+1) < DCTcheck(k1(k)+3,k2(k)+2)$ ，则比特值设置为0，否则设置为1。

```
1 function test()
2 key=2002;
3 count=4;
4 alpha=0.5;
5 [count,msg,data]=hidedctadv('lena.bmp','lenahide.bmp','1.txt',key,alpha);
6 tt=extractdctadv('lenahide.bmp','2.txt',key,count);
```

嵌入信息为“复旦大学”，因此赋值count=4，上网查询资料一般alpha值取0.5，key随便赋值。

实验思考

1. alpha的取值通常介于0.1到1之间。alpha越小，量化范围较大，DCT系数保留较多的细节信息，但压缩比较小。而alpha越大，量化范围缩小，DCT系数的值被更多的聚集到相同的量化级别，导致丢失更多的细节信息，但压缩效率更高。通常，alpha的取值如0.2、0.5、0.7、1等比较常见。不同的图像和任务对alpha的取值有不同的要求。一些研究表明，对于大多数图像，alpha取0.5可以获得相对较好的压缩效果和质量。
2. DCT变换中一张图片承载的隐秘信息长度较低，因为一个8x8的分块才能隐藏1bit的信息。
3. DCT变换对于图片质量也有一定要求：如果载体图像噪声过高，在执行DCT变换时两个DCT值的计算可能偶然仍然产生错误，交换行为随之出错，进而从而降低了嵌入数据的成功率。