第六章 变换域隐秘技术

在这一章里,主要讨论如何在变换域实现信息隐藏,以及在变换域实现信息隐藏的一些优点。由于信息隐藏技术从大的方面分为:隐写术和数字水印两个部分,数字水印我们将留在后几章重点阐述,所以本章只探讨变换域隐秘技术。

在上一章的 LSB(最低有效位)的信息隐藏方法中,我们很方便地通过改变图像像素的最低有效位实现了信息的隐藏,这种方法比较简单,实现起来也非常的容易,但是,用这种方法的隐藏信息是十分脆弱的,它连最简单的修改都不能容忍(如裁剪、压缩等),因为最低有效位同时也是最容易丢失的位。如果一个攻击者想要破坏秘密信息,是十分容易的,他只需对图像作最简单的处理即可达到目的。于是,我们需要一些更安全的域来实现隐藏。

前面已经提到过,在信号的频域嵌入信息要比在时域嵌入信息更具有鲁棒性。实际上,一幅图像经过时域到频域的变换,我们可以将待隐藏的信息藏入图像的显著区域,这种方法比 LSB 以及其他的一些空间域的隐藏更具抗攻击能力,例如,压缩、裁剪等图像处理技术。LSB 算法之所以能够实现,是因为我们改变最低有效位不会引起人们感官上的察觉,在变换域隐藏信息,人的肉眼就更不可能发现了。

实现一幅图像的时域到变换域的方法有很多,既可以用离散余弦变换的方法,也可以用小波变换的方法。在本章中我们对变换域的隐秘技术作一个简单的介绍。

6.1 DCT 域的信息隐秘的基本算法

在第三章中我们已经知道,用下面的公式实现一个 M× N 矩阵 A 的二维 DCT 变换:

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

$$0 \leq p \leq N-1 \qquad 0 \leq q \leq N-1 \tag{6.1}$$

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:

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

$$0 \le m \le M-1 \quad 0 \le n \le N-1 \qquad (6.2)$$

其中: a_p, a_q 同式 6.1 中的 a_p, a_q。

DCT 变换将图像信号从时域变换到了频域,它是现在广泛使用的有损数字图像压缩系统的核心步骤之一,该系统我们在第三章中已经作了简要的说明。下面我们先引用其 JPEG 压缩方案中的亮度量化表(如表 6.1 所示),以方便后面的分析。

表 6.1　　JPEG 压缩亮度量化表

(u, v)	1	2	3	4	5	6	7	8
1	16	11	10	16	24	40	51	61
2	12	12	14	19	26	58	60	55
3	14	13	16	24	40	57	69	56
4	14	17	22	29	51	87	80	62
5	18	22	37	56	68	109	103	77
6	24	35	55	64	81	104	113	92
7	49	64	78	87	103	121	120	101
8	72	92	95	98	112	100	103	99

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秘密信息进行编码。我们用 (u_1, v_1), (u_2, v_2) 来表示这两个系数的索引,算法描述如下:

对于第 i bit 秘密信息

if (要隐藏信息‘1’)

　　make $(u_1, v_1)_i > (u_2, v_2)_i$;

else

　　make $(u_1, v_1)_i < (u_2, v_2)_i$;

也就是说,在编码阶段我们是以秘密信息为主来使得 DCT 系数满足这一规则的。如果这两个系数的相对大小与要编码的信息比特不匹配,我们要‘强行’交换这两个系数,使之匹配,所以,make 的实质性操作要么为空,要么就是交换。

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 8×8 的图像块。在信息嵌入的时候,采用随机控制的办法选取图像块 b_i 以表示第 i 个消息比特的编码空间。

$B_i = DCT\{b_i\}$ 为图像块 b_i 经过 DCT 变换后的结果。

既然是通过比较变换后的两个 DCT 系数来完成信息的隐藏，那么在传递秘密信息前，通信的双方就必须对要比较的两个位置达成一致。对于每一个图像块，在8×8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。

首先，在假定嵌入过程不会导致载体严重降质的情况下，我们应该选择在 JPEG 压缩算法中(表 6.1) 亮度量化值一样的那些系数。其次，DCT 系数应该是中频系数为最优。之所以这样选择，其理由及优点是：

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，从而保证了算法的实施。试想一下，如果所选择的两个系数根本就不在一个数量级上，而我们仍然要强行地对其进行交换，对图像的破坏是多么大！

②这两个系数相应于 DCT 的中频部分，从而兼顾了机密信息隐藏的不可见性与鲁棒性。如果处理的是低频系数，由于其所相应拥有的能量过大，不利于信息隐藏不可见性的提高，而如果处理的是高频系数，则不具有鲁棒性——诸如有损压缩等操作一般都是针对能量低的高频部分展开的。基于上述考虑，我们应该选择(5,2) 和(4,3) 这一对系数或者(2,3) 和(4,1) 这一对系数。

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。对于 α的取值我们将放在后面讨论。这样一来在解码的过程中，问题就变得非常的简单，接收者只需要获得载有秘密信息的图像，也对图像做 DCT 变换和分块，按照随机控制的顺序直接比较 $B_i(u_1, v_1)$，$B_i(u_2, v_2)$ 的大小就能提取秘密信息了。

6.2 算法实现

编写函数 hidedctadv.m 和 extractdctadv.m 分别完成隐藏和提取实验。这里，秘密信息可以是任何形式的文件，我们的实验只用一个.txt 的文件做示范，这里还用到了随机函数 Randinterval.m。

1) 用于信息的隐藏：hidedctadv.m

```
% 文件名：hidedctadv.m
% 程序员：李鹏
```

```
% 编写时间: 2003. 11. 25
% 修改时间: 2004. 3. 8
% 函数功能: 本函数用于 DCT 域的信息隐藏
% 输入格式举例: [ count, msg, data] = hidedctadv( 'lenna. jpg', '1. jpg', '1. txt',
1982, 1) ;
% 参数说明:
% image 为载体图像
% imagegoal 为藏有秘密信息的载体, 即隐蔽载体
% msg 为待隐藏的信息
% key 为密钥, 用来控制随机选块
% alpha 为控制量, 用来保证编码的正确性
% count 为待隐藏信息的长度
% result 为隐藏结果
function [ count, msg, result] = 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) ;
% 将图像矩阵转为 double 型
data0 = double( data0) /255;
% 取图像的一层做隐藏
data = data0(:,:,1) ;
% 对图像分块
T = dctmtx( 8) ;
% 对分块图像做 DCT 变换
DCTrgb = 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( 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) ;
    2) 用于信息的提取: extractdctadv. m
    % 文件名: extractdctadv. m
    % 程序员: 李鹏
    % 编写时间: 2004. 3. 8
    % 函数功能: 本函数用于 DCT 隐藏信息的提取
```

```
% 输入格式举例: tt = extractdctadv('lennahide.jpg','jpg','2.txt',1982,40)
% 参数说明:
% image 为已经藏有信息的图像
% msg 为提取信息存放的位置
% key 为密钥,用来控制随机选块
% count 为信息的比特数,由藏入方给出
function result = extractdctadv(image,msg,key,count)
data0 = imread(image);
data0 = double(data0)/255;
% 用图像第一层做提取
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)
        fwrite(frr,0,'bit1');
        result(i,1) = 0;
    else
        fwrite(frr,1,'bit1');
        result(i,1) = 1;
    end
end
fclose(frr);
```

以下是选择 lenna 图像作为载体，隐藏的信息是一以. txt 文件保存的字符串进行的实验结果。图 6.1 为原始载体，图 6.2 为加入信息后的载体，我们取密钥为 1982，控制阈值 α取为 1。

图 6.1　原始图像

图 6.2　嵌入信息后的图像(α = 1)

在此也给出中间过程中的一些结果，图 6.3、图 6.4 分别为密钥取 1982 时选块矩阵的随机行列值，操作过程中，图像被分成 8× 8 的块后，由 k1，k2 分别标识各块的首行地址和列地址来依次选择编码的块。

```
k1 =

   Columns 1 through 14

    1     1     1     1     1     1     1     1     1     2     2     2     2     2

   Columns 15 through 28

    2     2     2     2     2     3     3     3     3     3     3     3     3     3

   Columns 29 through 42

    3     4     4     4     4     4     4     4     4     4     4     4     5     5

   Columns 43 through 56

…
```

图 6.3　选块时的块首行地址序列

```
k2 =
   Columns 1 through 14
    1     5     7     9    13    17    21    25    29     1     5     7     9    13
   Columns 15 through 28
   17    21    23    27    31     1     3     7     9    13    17    19    23    25
   Columns 29 through 42
   29     1     5     7    11    13    15    19    23    25    29    31     3     7
   Columns 43 through 56
…
```

图 6.4　选块时的块首列地址序列

图 6.5, 图 6.6 分别是读取图像第一层的矩阵和对该层做分块 DCT 得到的矩阵(部分)。

Array Editor: data

File Edit View Web Window Help

Numeric format: shortG　Size: 256 by 256

	1	2	3	4	5	6
1	0.70196	0.69804	0.69804	0.70196	0.70196	0.69804
2	0.69804	0.69412	0.69412	0.69412	0.69804	0.69412
3	0.69804	0.69412	0.6902	0.6902	0.69412	0.6902
4	0.6902	0.68627	0.68235	0.68235	0.68627	0.68627
5	0.69804	0.6902	0.68235	0.68235	0.6902	0.6902
6	0.70196	0.6902	0.68627	0.68235	0.6902	0.6902

图 6.5　图像的第一层矩阵

为使大家直观地了解编码的过程, 图 6.7 和图 6.8 是隐藏信息前的 DCT 矩阵和隐藏信息后的 DCT 矩阵的局部截图, 这个局部对 1 比特的信息进行了编码, 注意比较它们数值的区别。

Array Editor: DCTrgb

File Edit View Web Window Help

Numeric format: shortG Size: 256 by 256

	1	2	3	4	5	6
1	5.5294	0.02495	-0.00090576	0.030216	-0.00098039	0.0018297
2	-0.011966	0.022077	-0.028234	1.9082e-005	-0.00027049	-0.0012524
3	0.033176	-0.0018085	-0.0021638	-0.0011417	-0.00053058	0.0021862
4	0.0031053	2.8559e-005	0.00071165	0.0017299	0.00077029	0.0023427
5	-0.0019608	-0.0017719	-0.00090576	-0.00073024	0.00098039	-0.0017682

图 6.6　做分块 DCT 后的矩阵

Array Editor: DCTrgb0

File Edit View Web Window Help

Numeric format: shortG Size: 256 by 256

	1	2	3	4	5	6
1	5.5294	0.02495	-0.00090576	0.030216	-0.00098039	0.0018297
2	-0.011966	0.022077	-0.028234	1.9082e-005	-0.00027049	-0.0012524
3	0.033176	-0.0018085	-0.0021638	-0.0011417	-0.00053058	0.0021862
4	0.0031053	2.8559e-005	0.00071165	0.0017299	0.00077029	0.0023427
5	-0.0019608	-0.0017719	-0.00090576	-0.00073024	0.00098039	-0.0017682

图 6.7　隐藏信息前的 DCT 矩阵

Array Editor: DCTrgb

File Edit View Web Window Help

Numeric format: shortG Size: 256 by 256

	1	2	3	4	5	6
1	5.5294	0.02495	-0.00090576	0.030216	-0.00098039	0.0018297
2	-0.011966	0.022077	-0.028234	1.9082e-005	-0.00027049	-0.0012524
3	0.033176	-0.0018085	-0.0021638	-0.0011417	-0.00053058	0.0021862
4	0.0031053	2.8559e-005	-0.011772	0.0017299	0.00077029	0.0023427
5	-0.0019608	0.00071165	-0.00090576	-0.00073024	0.00098039	-0.0017682

图 6.8　隐藏信息后的 DCT 矩阵

6.3　对算法参数的讨论

α是为了避免图像在传输过程中使 $B_i(u_1, v_1)$ 和 $B_i(u_2, v_2)$ 的相对大小发生错位从而导致编码发生错误而引入的控制量，α越大，编码越不容易出错，图像的鲁棒性更强，但是 α的取值的增大将带来载体视觉上的降质，因为他使 $B_i(u_1, v_1)$ 与 $B_i(u_2, v_2)$ 差的绝对值太大，在交换 $B_i(u_1, v_1)$ 和 $B_i(u_2, v_2)$ 的时候会出现更大的误差。这种偏差表现在图像的能量的变化上，它可能影响图像的整个部分。下面我们分别就 α与隐藏的鲁棒性和不可感知性的关系加以探讨。

(1) α与隐藏鲁棒性的关系

我们仍然仅用 JPEG 压缩的手段来探查不同控制阈值 α下的隐藏的鲁棒性。采用秘密信息存放于文件 secret. txt，编写函数 jpgandalpha. m 完成实验。该函数将自动取 0.1 ~1 十个等差为 0.1 的值作为控制阈值 α，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。函数代码如下：

```
% 文件名: jpgandalpha. m
% 程序员: 郭迟
% 编写时间: 2004. 3. 11
% 函数功能: 本函数将探讨 DCT 隐藏中的控制阈值 α在 JPEG 条件对隐藏鲁棒性的影响
% 输入格式举例: result = jpgandalpha( 'c: \lenna. jpg', 'c: \secret. txt')
% 参数说明:
% test 为原始图像
% msg 为待隐藏的信息
function result = jpgandalpha( test, msg)
% 定义压缩质量比从 10% ~100%
quality = 10: 10: 100;
alpha = 0. 1: 0. 1: 1;
result = zeros( [ max( size( alpha) )  max( size( quality) ) ] ) ;
resultr = 0;
resultc = 0;
for a = alpha
    resultr = resultr + 1;
    [ count, message, hideresult] = hidedctadv( test, 'temp. jpg', msg, 2003, a) ;
    resultc = 0;
```

```
        different = 0;
        for q = quality
            resultc = resultc + 1;
            imwrite( hideresult, 'temp. jpg', 'jpg', 'quality', q) ;
            msgextract = extractdctadv( 'temp. jpg', 'temp. txt',2003, 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) α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 α在 JPEG 条件下对隐藏鲁棒性的影响')
```

我们以 lenna. jpg 为载体,秘密信息取为宋人宋祁的一句词句:绿杨烟外晓寒轻,红杏枝头春意闹。执行 jpgandalpha 函数,得到实验结果如图 6.9 与表 6.2 所示。表 6.2 是图 6.9 的具体数据。

表 6.2 中若数值为 0,则表示其横向相应的 α在纵向压缩率的条件下有强鲁棒性,若数值不为 0 则表示相应的误码率,数值越大表示信息提取的越不真实。从表 6.2 可以看出,α取 0.1 时,完全不能保证信息提取的正确性,α在[0.2,0.4]区间内的抗 JPEG 压缩性能一般,当 α取 1 时,基本可以认为是不受 JPEG 压缩干扰的。

表 6.3 是在对应于表 6.2 中加框(JPEG 压缩率为 50%)的列下实际信息提取的内容。可以发现,在 α>0.3 时,信息开始变得可读了。

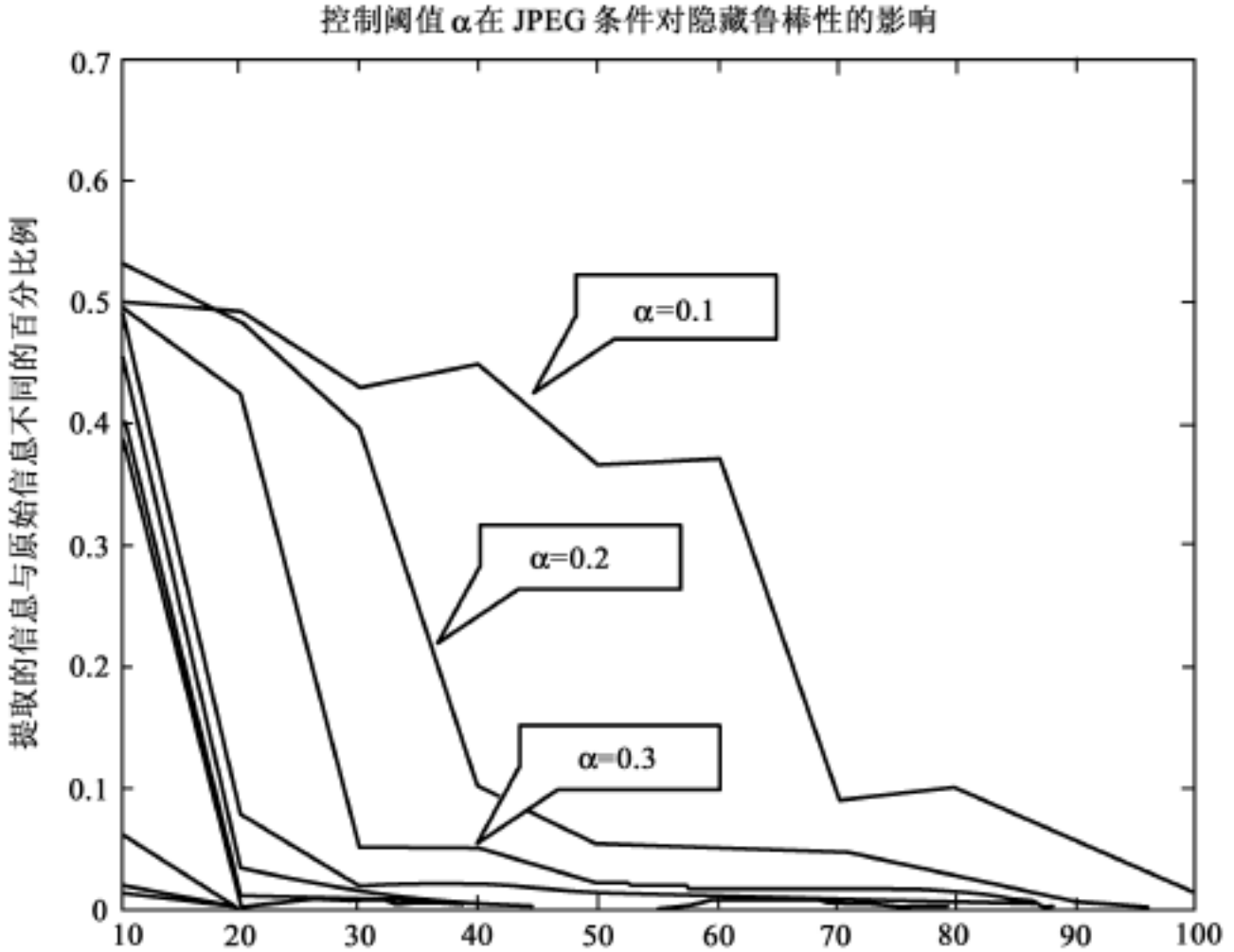

图 6.9　控制阈值 α在 JPEG 条件下对隐藏鲁棒性的影响

表 6.2　控制阈值 α在 JPEG 条件对隐藏鲁棒性的影响

	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
α=0.1	0.5	0.49167	0.42917	0.45	0.36667	0.37063	0.091667	0.1	0.054167	0.0125
α=0.2	0.53333	0.48333	0.39583	0.095833	0.054167	0.05	0.045833	0.025	0.0041667	0
α=0.3	0.49583	0.425	0.05	0.05	0.020833	0.016667	0.016667	0.0125	0	0
α=0.4	0.5	0.079167	0.016667	0.020833	0.0125	0.0083333	0.0083333	0.0041667	0	0
α=0.5	0.4625	0.033333	0.0125	0.0041667	0	0.0041667	0.0083333	0	0	0
α=0.6	0.41667	0.0063333	0.0041667	0.0041667	0	0	0	0	0	0
α=0.7	0.38167	0	0.0083333	0.0041667	0	0	0	0	0	0
α=0.8	0.0625	0	0	0.0041667	0	0	0	0	0	0
α=0.9	0.0125	0	0	0	0	0	0	0	0	0
α=1	0.016667	0	0	0	0	0	0	0	0	0

表 6.3　　控制阈值 α在 50% JPEG 压缩条件的隐藏结果

原始信息	绿杨烟外晓寒轻,红杏枝头春意闹。
α= 0. 1	J? 鹳 wF 龒窩澌 |?? 频颭碉揃蚿媚
α= 0. 2	聪盐烟下摸氷珥,簂杏 R 锻反阂铐
α= 0. 3	聪杨烟外晓寒玳,红杏露头春意闹。
α= 0. 4	绿杨烟外晓寒玑,红杏枝头春意闹。
α= 0. 5	绿杨烟外晓寒轻,红杏枝头春意闹。
α= 0. 6	绿杨烟外晓寒轻,红杏枝头春意闹。
α= 0. 7	绿杨烟外晓寒轻,红杏枝头春意闹。
α= 0. 8	绿杨烟外晓寒轻,红杏枝头春意闹。
α= 0. 9	绿杨烟外晓寒轻,红杏枝头春意闹。
α= 1	绿杨烟外晓寒轻,红杏枝头春意闹。

需要说明的几个问题是:

①我们在这里分析的 α仅仅与我们实际的算法实现手段相一致,并不能严格理解为是对算法本身参数 α的分析。比如大家可能会对取 α=0.1 且不经过任何 JPEG 压缩(压缩率 100%) 时仍然有 1.25% 的误码率表示不理解。事实上,造成这一偏差的原因是我们使用的 MATLAB 在图像存储时精度转换造成的,对于这一事实的分析我们在 5.3.2 节中已有叙述,这里就不重复了。此外,α的理论作用是使 $|B_i(u_1, v_1) - B_i(u_2, v_2)| > \alpha$。实现这一目的的手段很多,我们这里采取的仅是其中较简单的一种(使小的系数通过减去 α变得更小) 。

②载体不同,相应的 α与性能的关系也不同。大家在选用不同载体实验时,也应该具体情况具体分析,否则当你看到提取出的信息是一团乱码时,我想无论如何你是无法产生“ 绿杨烟外晓寒轻,红杏枝头春意闹 ”的意境的。

③我们之所以取 α为[0.1,1] 进行分析,是因为结合实验的实际情况,当 α小于 0.1 时起不到保证隐藏鲁棒性的任何作用,而当 α大于 1 时,隐蔽载体的不可见性就值得研究了。

(2) α与隐藏不可见性的关系

给函数相同的入口参数(载体相同,信息相同,密钥相同) ,只取 α不同,输入:

```
>> [count, msg, data] = hidedctadv('lenna.jpg','1.jpg','jpg','1.txt',1982,algha);
```

图 6.10 ~图 6.15 分别对应 α的值为 0.01, 0.1, 1, 10, 100, 1000 的情况,可以明显地看到随着 α值的增大,对图像的破坏也越大。

这些图像反映出来的另一个现象我们也应该注意到,就是当 α= 100 和 α= 1 000的

图 6.10　α=0.01

图 6.11　α=0.1

图 6.12　α=1

图 6.13　α=10

图 6.14　α=100

图 6.15　α=1 000

两种情况下,对图的破坏是差不多的。也就是说,单纯的 α的影响是有限的,这是因为信息数量有限,对图像 DCT 系数的改变个数有限,加上我们所选用的点是中频系数,它并不含有图像的主要信息,对图像能量的改变是有限的。

另外,对 α与隐藏不可见性的关系,我们这里仅仅是从直观上给出了结果。具体涉及到知觉感知分析的相关内容,将在第八章叙述。

6.4 小波域信息隐秘的讨论

在变换信息隐秘技术中，还有一类常用的方法，即小波域的信息隐秘。二维信号小波分解的 Mallat 算法可用式(6.3)描述：

$$A_{j-1}f(x,y) = A_j f(x,y) + D_j^1 f(x,y) + D_j^2 f(x,y) + D_j^3 f(x,y) \quad (6.3)$$

其中，j 表示分解尺度，A 为低频系数，D_j^1，D_j^2，D_j^3 为 j 尺度下水平、垂直、对角方向上的三个高频系数，如图 6.16 所示。

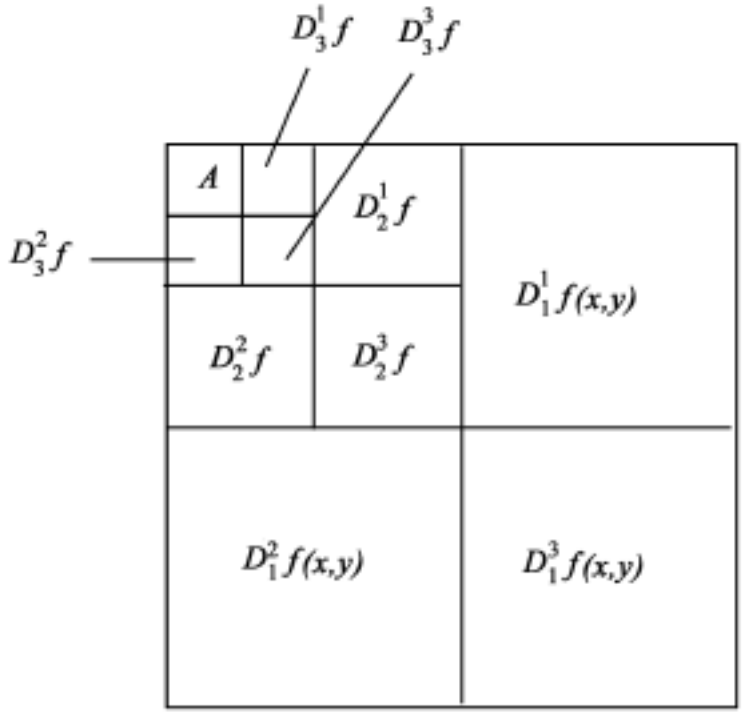

图 6.16 二维信号的小波分解

关于小波域的信息隐秘，有几点需要说明一下：

①二维小波分解的形态如图 6.16 所示，低频分量居于左上角。

②我们一般认为，将要隐藏的信息藏入低频系数会有较高的鲁棒性，但由于低频系数较少，嵌入信息的量有限，这就造成了鲁棒性与隐藏容量的矛盾。

③一种改进的设想是将信息藏入高尺度分解下的高频部分，即图 6.16 所示的

$$D_2^1 f(x,y),\ D_2^2 f(x,y),\ D_2^3 f(x,y),\ D_3^1 f(x,y),\ D_3^2 f(x,y),\ D_3^3 f(x,y)$$

等部分。做这种改进的原因是：

- 这些部分仍然是某一尺度下的低频部分，秘密信息隐藏在这些区域并不影响鲁棒性。
- 将秘密信息隐藏在这些区域，隐藏的不可见性会比单纯隐藏在低频部分更好。
- 这样可以扩大隐藏信息的容量。

至此，我们对信息隐藏中的一类重要应用——隐秘技术作了进一步的研究和探讨，我们在这一部分所做的实验还存在不少需要加强的地方，但我们希望我们所做的

努力和得到的结论能对大家理解隐秘技术起到积极的作用，虽然隐秘技术相对于我们后面将要涉及的数字水印技术来说应用得相对较少，也相对比较简单，但是它却是学习和研究水印技术不可缺少的一部分知识。因此，对于它的学习同样值得我们关注。