

拓展资源 4.3 实验指导



4.3.1 图像的傅里叶变换—(平移性质)

1. 实验内容

对图 4.1 (a) 进行平移, 观察原图的傅里叶频谱与平移后的傅里叶频谱的对应关系。

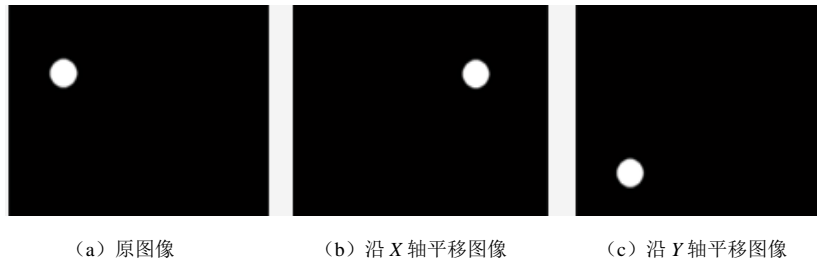


图 4.1 实验 4.3.1 所需图像

2. 实验原理

如果 $F(u, v)$ 的频率变量 u, v 各移动了 u_0, v_0 距离, $f(x, y)$ 的变量 x, y 各移动了 x_0, y_0 距离, 则傅里叶变换对有下列的形式。

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(u x_0 + v y_0)/N} \quad (4.34)$$

因此, 傅里叶变换的平移性质表明了函数与一个指数项相乘等于将变换后的空域中心(如式 (4.34)) 移到新的位置, 从 (4.34) 还可知, 对 $f(x, y)$ 的平移将不改变频谱的幅值(amplitude)。

3. 实验方法及程序

选取一幅图像, 进行离散傅里叶变换, 再对其分别进行 X 轴与 Y 轴上的平移, 得其离散傅里叶变换, 观察 3 幅结果图, 实验程序如下。

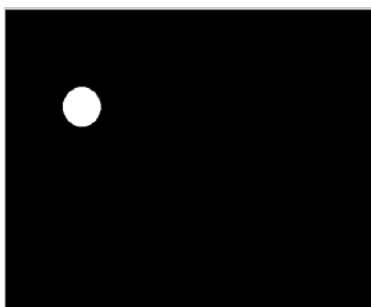
```
I=imread('1.bmp');
figure(1)
imshow(real(I));
I=I(:,:,3);
fftI=fft2(I);
sfftI=fftshift(fftI); %求离散傅里叶频谱
%对原始图像进行二维离散傅里叶变换, 并将其坐标原点移到频谱图中央位置
RRfdp1=real(sfftI);
IIfdp1=imag(sfftI);
a=sqrt(RRfdp1.^2+IIfdp1.^2);
a=(a-min(min(a)))/(max(max(a))-min(min(a)))*225;
```

```

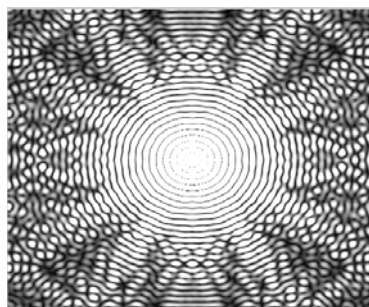
figure(2)
imshow(real(a));
I=imread('2.bmp');
figure(3),
imshow(real(I));
I=I(:,:,3);
fftI=fft2(I);
sfftI=fftshift(fftI); %求离散傅里叶频谱
%对原始图像进行二维离散傅里叶变换,并将其坐标原点移到频谱图中央位置
RRfdp1=real(sfftI);
IIfdp1=imag(sfftI);
a=sqrt(RRfdp1.^2+IIfdp1.^2);
a=(a-min(min(a)))/(max(max(a))-min(min(a)))*225;
figure(4),
imshow(real(a));
I=imread('3.bmp');
figure(5),
imshow(real(I));
I=I(:,:,3);
fftI=fft2(I);
sfftI=fftshift(fftI); %求离散傅里叶频谱
%对原始图像进行二维离散傅里叶变换,并将其坐标原点移到频谱图中央位置
RRfdp1=real(sfftI);
IIfdp1=imag(sfftI);
a=sqrt(RRfdp1.^2+IIfdp1.^2);
a=(a-min(min(a)))/(max(max(a))-min(min(a)))*225;
figure(6),
imshow(real(a));
    
```

4. 实验结果与分析

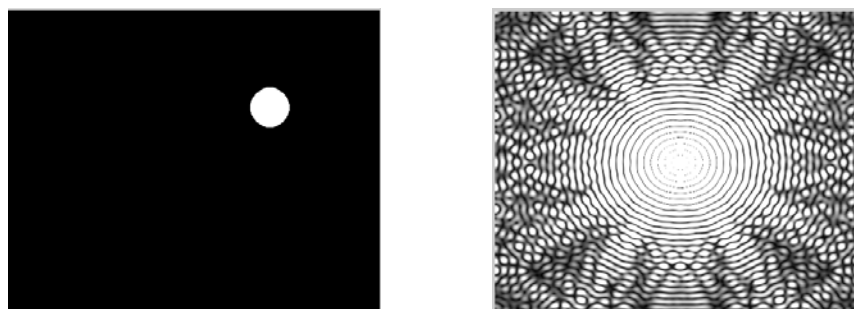
实验结果如图 4.2 所示。



(a) 原图像



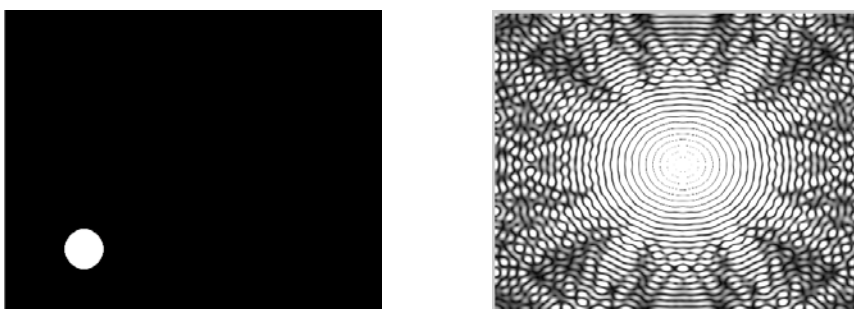
(b) 原图像傅里叶频谱



(c) 沿 X 轴平移图像

(d) 沿 X 轴平移后傅里叶频谱

图 4.2 实验 4.3.1 结果图



(e) 沿 Y 轴平移图像

(f) 沿 Y 轴平移后傅里叶频谱

图 4.2 实验 4.3.1 结果图 (续)

由所得结果可知，原图像图 4.2 (a) 分别经过 X 轴与 Y 轴上的平移后所得到的离散傅里叶变换频谱图 4.2 (d)、图 4.2 (f) 与原图像所得的傅里叶频谱图 4.2 (b) 基本相同。实验结果符合傅立叶变换平移性质，即函数与一个指数项相乘等于将变换后的空域中心移到新的位置，而且对 $f(x,y)$ 的平移将不改变频谱的幅值。

5. 思考题

将一幅图分别进行 X 轴与 Y 轴上的平移，所得的傅里叶频谱与原图像的傅里叶频谱有什么变化，请说明理由。



4.3.2 图像的傅里叶变换二 (旋转性质)

1. 实验内容

对图 4.3 的两幅图像分别做旋转，观察原图的傅里叶频谱与旋转后的傅里叶频谱的对应关系。



(a) 长方形

(b) 正方形

图 4.3 实验 4.3.2 所需图像

2. 实验原理

首先借助极坐标变换 $x = r \cos \theta$, $y = r \sin \theta$, $u = w \cos \phi$, $v = w \sin \phi$, 将 $f(x, y)$ 和 $F(u, v)$ 转换为 $f(r, \theta)$ 和 $F(w, \phi)$ 。

$$\begin{aligned} f(x, y) &\Leftrightarrow F(u, v) \\ f(r \cos \theta, r \sin \theta) &\Leftrightarrow F(w \cos \phi, w \sin \phi) \end{aligned}$$

经过变换得

$$f(r, \theta + \theta_0) \Leftrightarrow F(w, \phi + \theta_0) \quad (4.35)$$

式 (4.35) 表明, 对 $f(x, y)$ 旋转一个角度 θ_0 对应于将其傅里叶变换 $F(u, v)$ 也旋转相同的角度 θ_0 。 $F(u, v)$ 到 $f(x, y)$ 也是一样。

3. 实验方法及程序

选取一幅图像, 进行离散傅里叶变换, 再对其进行一定角度的旋转, 进行离散傅里叶变换。

```
%构造原始图像
I = zeros(256,256);
I(88:168,124:132) = 1;      %图像范围是 256*256,前一值是纵向比,后一值
                             %是横向比

imshow(I)
%求原始图像的傅里叶频谱
J = fft2(I);
F = abs(J);
J1 = fftshift(F);figure
imshow(J1,[5 50])
%对原始图像进行旋转
J = imrotate(I,90,'bilinear','crop');
figure
imshow(J)
%求旋转后图像的傅里叶频谱
J1 = fft2(J);
F = abs(J1);
J2 = fftshift(F);figure
imshow(J2,[5 50])
```

4. 实验结果与分析

实验结果如图 4.4 所示。

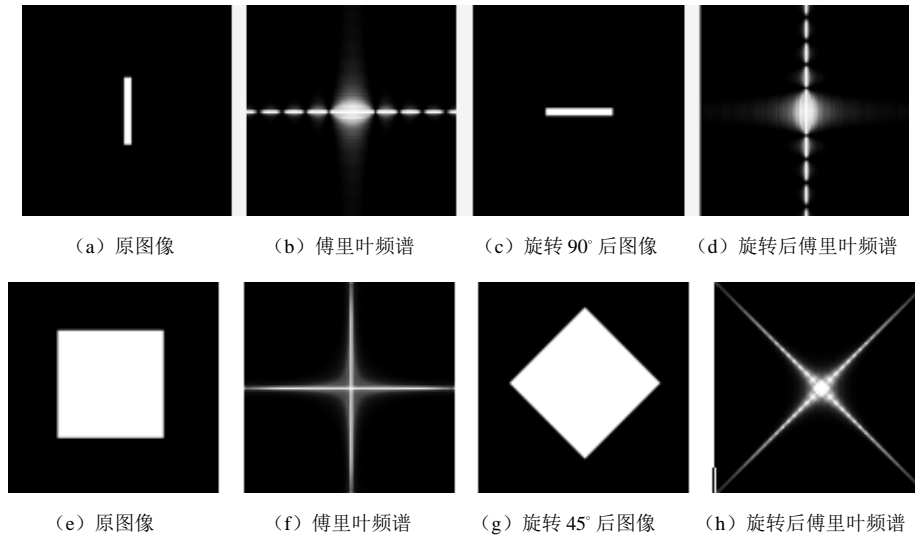


图 4.4 实验 4.3.2 结果图

由实验结果可知，首先从旋转性质来考虑，对比图 4.4 (b) 和图 4.4 (d)，时域中图像顺时针旋转 90° ，频域中图像也顺时针旋转 90° ；其次从尺度变换性质来考虑，如图 4.4 (a) 与图 4.4 (b) 可知，原图像与其傅里叶变换后的图像角度相差 90° ，由此可知，时域中信号被压缩，到频域中信号就被拉伸。

5. 思考题

将一幅图进行离散傅里叶变换，得到其傅里叶频谱图，再对原图像进行一定角度的旋转，得到的频谱图与原图的频谱图进行比较，以及原图像与其傅里叶谱存在的何种角度关系，说出符合哪些性质。



4.3.3 图像的离散余弦变换一

1. 实验内容

对图 4.5 进行离散余弦变换，观察其余弦变换系数及余弦反变换后恢复图像。

2. 实验原理

二维离散余弦变换由式 (4.36) 表示。

$$\begin{aligned}
 F(0,0) &= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \\
 F(0,v) &= \frac{\sqrt{2}}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cdot \cos \frac{(2y+1)v\pi}{2N} \\
 F(u,0) &= \frac{\sqrt{2}}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cdot \cos \frac{(2x+1)u\pi}{2N} \\
 F(u,v) &= \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cdot \cos \frac{(2x+1)u\pi}{2N} \cdot \cos \frac{(2y+1)v\pi}{2N} \quad (4.36)
 \end{aligned}$$



图 4.5 实验 4.3.3 所需图像

式 (4.36) 是正变换公式。式中， $f(x,y)$ 是空间域二维向量的元素， $x,y=0,1,2,\dots,N-1$ ，

$F(u,v)$ 是变换系数阵列的元素。式中表示的阵列为 $N \times N$ 。

二维离散余弦反变换由式 (4.37) 表示。

$$f(x,y) = \frac{1}{N} F(0,0) + \frac{\sqrt{2}}{N} \sum_{v=1}^{N-1} F(0,v) \cos \frac{(2y+1)v\pi}{2N} + \frac{\sqrt{2}}{N} \sum_{u=1}^{N-1} F(u,0) \cos \frac{(2x+1)u\pi}{2N} + \frac{2}{N} \sum_{u=1}^{N-1} \sum_{v=1}^{N-1} F(u,v) \cos \frac{(2x+1)u\pi}{2N} \cdot \cos \frac{(2y+1)v\pi}{2N} \quad (4.37)$$

3. 实验方法及程序

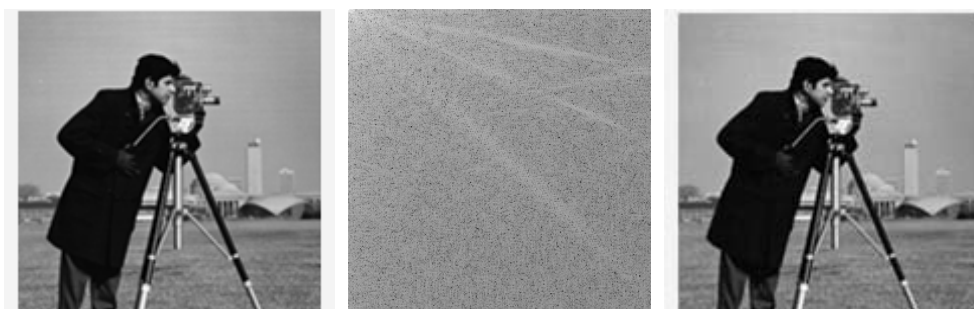
选取一幅图像，进行离散余弦变换，并对其进行离散余弦反变换，观察其结果。

```
%对 cameraman.tif 文件计算二维 DCT 变换
RGB = imread('cameraman.tif');
figure(1)
imshow(RGB)
I = rgb2gray(RGB);
%真彩色图像转换成灰度图像
J = dct2(I);
%计算二维 DCT 变换
figure(2)
imshow(log(abs(J)), [])
%图像大部分能量集中在左上角处
figure(3);
J(abs(J) < 10) = 0;
%把变换矩阵中小于 10 的值替换为 0，然后用 idct2 重构图像
K = idct2(J)/255;
imshow(K)
```

4. 实验结果与分析

实验结果如图 4.6 所示。

由图 4.6 (b) 可知，离散余弦变换具有很强的“能量集中”特性，能量主要集中在左角处，因此在实际图像应用中，能量不集中的地方可在余弦编码中忽略，可通过对 `mask` 矩阵变换来实现，即将 `mask` 矩阵左上角置 1，其余全部置 0。然后通过离散余弦反变换后，图像得到恢复，图 4.6 (c) 恢复图像与图 4.6 (a) 原始图像基本相同。



(a) 原始图像

(b) 余弦变换系数

(c) 余弦反变换恢复图像

图 4.6 实验 4.3.3 结果图

5. 思考题

将一幅图进行离散余弦变换，得到其频谱图，观察其频谱图有何特点，再经过离散余弦反变换得到还原图像，比较与原图有何差别。



4.3.4 图像的离散余弦变换二

1. 实验内容

对图 4.5 进行离散余弦变换，做图像压缩解压，取不同的 DCT 系数，并观察其结果。

2. 实验原理

二维离散余弦变换与反变换原理见 4.5.3 实验原理。

二维离散余弦变换也可以写成矩阵式：

$$\begin{aligned} (F(u,v)) &= (A)(f(x,y))(A)' \\ (f(x,y)) &= (A)'(F(u,v))(A) \end{aligned} \quad (4.38)$$

式 (4.38) 中， $(f(x,y))$ 是空间域数据阵列； $(F(u,v))$ 是变换系数阵列； (A) 是系数阵列，变换矩阵 $(A)'$ 是 (A) 的转置。

离散余弦变换是先将整体图像分成 $N \times N$ 像素块，然后对 $N \times N$ 像素块逐一进行离散余弦变换。由于大多数图像的高频分量较小，相应于图像高频分量的系数经常为零，加上人眼对高频成分的失真不太敏感。所以可用更粗的量化。因此，传送变换系数的数码率要大大小于传送图像像素所用的数码率。到达接收端后通过反离散余弦变换回到样值。

3. 实验方法及程序

选取一幅图像，进行离散余弦变换，并对其进行压缩解压，观察其结果。

```

RGB = imread('camera.tif');
I=rgb2gray(RGB);
I = im2double(I);           %转换图像矩阵为双精度型
T = dctmtx(8);             %产生二维 DCT 变换矩阵，
%矩阵 T 及其转置 T' 是 DCT 函数 P1*X*P2 的参数
B = blkproc(I,[8 8], 'P1*x*P2', T, T');
mask1= [ 1 1 1 1 0 0 0 0
         1 1 1 0 0 0 0 0
         1 1 0 0 0 0 0 0
         1 0 0 0 0 0 0 0
         0 0 0 0 0 0 0 0
         0 0 0 0 0 0 0 0
         0 0 0 0 0 0 0 0
         0 0 0 0 0 0 0 0];
%二值掩模，用来压缩 DCT 系数
B2 = blkproc(B,[8 8], 'P1.*x', mask1); %只保留 DCT 变换的 10 个系数
    
```

```

I2 = blkproc(B2,[8 8], 'P1*x*P2',T',T); %重构图像
figure,imshow(I);
figure,imshow(B2);
figure,imshow(I2);

RGB = imread('camera.tif');
I=rgb2gray(RGB);
I = im2double(I); %转换图像矩阵为双精度型
T = dctmtx(8); %产生二维 DCT 变换矩阵,
%矩阵 T 及其转置 T' 是 DCT 函数 P1*X*P2 的参数
B = blkproc(I,[8 8], 'P1*x*P2',T,T');
mask2= [ 1 1 1 1 0 0 0 0
1 1 1 0 0 0 0 0
1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0]; %二值掩模, 用来压缩 DCT 系数
%只保留 DCT 变换的 10 个系数
B2 = blkproc(B,[8 8], 'P1.*x',mask2);
I2 = blkproc(B2,[8 8], 'P1*x*P2',T',T); %重构图像
figure,imshow(I);
figure,imshow(B2);
figure,imshow(I2);
    
```

4. 实验结果与分析

实验结果如图 4.7 所示。

由上例可知，图像分成 8×8 像素块，对每一像素块进行块运算然后经过压缩系数，对其进行压缩，压缩率就是通过压缩系数来决定的，比较 `mask1` 和 `mask2`——压缩 DCT 系数，就可知，压缩率对其原始图像压缩解压尤为重要，当压缩率高时得到的图像就比较模糊（如图 4.7 (c) 所示）。在压缩后经过解压，再使用块运算 `blkproc()` 函数重构图像，就可得出以上结果。



(a) 原始图像 (b) 压缩 DCT 系数 mask1 解压图像 (c) 压缩 DCT 系数 mask2 解压图像

图 4.7 实验 4.3.4 结果图

5. 思考题

将一幅图进行离散余弦变换，再进行压缩解压，观察不同压缩 DCT 系数，以及解压后图像有何变化。



4.3.5 图像的哈达玛变换

1. 实验内容

对图 4.5 进行哈达玛变换，并对其进行压缩解压，编写 Matlab 程序，观察其结果。

2. 实验原理

原理见 4.2 节的公式 (4.33)。

3. 实验方法及程序

选取一幅图像，进行哈达玛变换，观察其结果。对一幅图像进行哈达玛变换。部分参考程序（以一幅 256×256 的图像为例，首先将其分割为 1024 个， 8×8 的子图像块后，然后对每个图像块进行变换，再按照每个系数的方差来排次序，保留方差较大的系数，舍去方差较小的系数。保留原系数的 $1/2$ ，即 32 个系数，进行 2:1 的压缩。）

```
cr=0.5;
I=imread('cameraman.tif');%输入图像
I=double(I)/255;%将读入的 unit8 类型的 RGB 图像 I 转换为 double 类型的数值
figure(1),imshow(I);%显示
%求图像大小
[m_I,n_I]=size(I); %提取矩阵 I 的行列数，m_I 为 I 的行数，n_I 为 I 的列数
sizi=8;
snum=64;
%分块处理
t=hadamard(sizi) %生成 8*8 的哈达玛矩阵
hdcoe=blkproc(I,[sizi sizi],'P1*x*P2',t,t);
%将图片分成 8*8 像素块进行哈达玛变换
%重新排列系数
CE=im2col(hdcoe,[sizi sizi],'distinct');
%将矩阵 hdcoe 分为 8*8 互不重叠的子矩阵，再将每个子矩阵作为 CE 的一列
[Y Ind]=sort(CE);%对 CE 进行升序排序
%舍去方差较小的系数，保留原系数的二分之一，即 32 个系数
[m,n]=size(CE); %提取矩阵 CE 的行列数，m 为 CE 的行数，n 为 CE 的列数
snum=snum-snum*cr;
for i=1:n
CE(Ind(1:snum),i) =0;
```

```

end
                                %重建图像
re_hdcoe=col2im(CE,[sizi sizi],[m_I n_I],'distinct');
                                %将矩阵的列重新组织到块中.
re_I=blkproc(re_hdcoe,[sizi sizi],'P1*x*P2',t',t);
                                %进行反哈达玛变换,得到压缩后图像
re_I=double(re_I)/64;
                                %转换为 double 类型的数据
figure(2);
imshow(re_I);
                                %计算原始图像和压缩后图像的误差
error=I.^2-re_I.^2;
MSE=sum(error(:))/prod(size(re_I));

```

运行以上程序，得到实验结果如图 4.8 所示。



(a) 原始图像



(b) 压缩哈达玛变换系数，解压后的图像

图 4.8 实验 4.3.5 结果图

沃尔什—哈达玛变换，是将一个函数变换成取值为+1 或-1 的基本函数构成的级数，用它来逼近数字脉冲信号时要比 FFT 有利。同时，它只需要进行实数运算，存储量比 FFT 要少得多，运算速度也快得多。因此，在图像传输、通信技术和数据压缩中被广泛使用。