第5章 图像正交变换

信号不仅可以在空间域表示,也可以在频率域表示,后者将有利于许多问题的分析讨论,因此,常需要对信号进行正交变换,变换到频率域进行处理。对图像进行正交变换在图像增强、图像复原、图像特征提取、图像编码等处理中都经常采用。常用的图像正交变换有多种,本章主要介绍图像的离散傅里叶变换、离散余弦变换、K-L 变换、Radon 变换和小波变换。

5.1 离散傅里叶变换

离散傅里叶变换(Discrete Fourier Transform,DFT)是直接处理离散时间信号的傅里叶变换,在数字信号处理中应用广泛。本节学习离散傅里叶变换的定义、性质、实现及其在图像处理中的应用。

5.1.1 离散傅里叶变换的定义

对于有限长数字序列 $f(x)$,$x=0,1,\cdots,N-1$,一维离散傅里叶变换及离散傅里叶反变换(IDFT)定义为:

$$\begin{cases} F(u)=\sum_{x=0}^{N-1} f(x)\mathrm{e}^{-\mathrm{j}\frac{2\pi ux}{N}} & u=0,1,2,\cdots,N-1 \\ f(x)=\frac{1}{N}\sum_{u=0}^{N-1} F(u)\mathrm{e}^{\mathrm{j}\frac{2\pi ux}{N}} & x=0,1,2,\cdots,N-1 \end{cases} \tag{5-1}$$

$f(x)$和 $F(u)$为离散傅里叶变换对,表示为 $\mathscr{F}[f(x)]=F(u)$或 $f(x)\Leftrightarrow F(u)$。

设 $W=\mathrm{e}^{-\mathrm{j}\frac{2\pi}{N}}$,则一维的离散傅里叶变换和离散傅里叶反变换可以表示为:

$$\begin{cases} F(u)=\sum_{x=0}^{N-1} f(x)W^{ux} & u=0,1,2,\cdots,N-1 \\ f(x)=\frac{1}{N}\sum_{u=0}^{N-1} F(u)W^{-ux} & x=0,1,2,\cdots,N-1 \end{cases} \tag{5-2}$$

数字图像为二维数据，二维离散傅里叶变换是由一维离散傅里叶变换推广而来的。二维离散傅里叶变换和离散傅里叶反变换定义为：

$$\begin{cases} F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{xu}{M}+\frac{yv}{N}\right)} & x,u=0,1,2,\cdots,M-1 \\ f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\mathrm{e}^{\mathrm{j}2\pi\left(\frac{xu}{M}+\frac{yv}{N}\right)} & y,v=0,1,2,\cdots,N-1 \end{cases} \tag{5-3}$$

其中，$f(x,y)$是二维离散信号，$F(u,v)$为 $f(x,y)$的频谱，u、v 为频域采样值；$f(x,y)$和 $F(u,v)$为二维离散傅里叶变换对，记为 $\mathscr{F}[f(x,y)]=F(u,v)$或 $f(x,y)\Leftrightarrow F(u,v)$。

$F(u,v)$一般为复数，表示为：

$$F(u,v)=R(u,v)+\mathrm{j}I(u,v)=|F(u,v)|\mathrm{e}^{\mathrm{j}\varphi(u,v)} \tag{5-4}$$

其中，$|F(u,v)|$称为 $f(x,y)$的傅里叶谱，如式(5-5)所示；$\phi(u,v)$称为 $f(x,y)$的相位谱，如式(5-6)所示；$f(x,y)$的功率谱定义为傅里叶谱的平方，如式(5-7)所示。

$$|F(u,v)|=\sqrt{R^2(u,v)+I^2(u,v)} \tag{5-5}$$

$$\phi(u,v)=\arctan\frac{I(u,v)}{R(u,v)} \tag{5-6}$$

$$E(u,v)=|F(u,v)|^2=R^2(u,v)+I^2(u,v) \tag{5-7}$$

5.1.2 离散傅里叶变换的实现

将二维离散傅里叶变换变换式做如下变换：

$$\begin{aligned} F(u,v)&=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi\frac{xu}{M}}\mathrm{e}^{-\mathrm{j}2\pi\frac{yv}{N}} \\ &=\sum_{x=0}^{M-1}\left[\sum_{y=0}^{N-1}f(x,y)\mathrm{e}^{-\mathrm{j}2\pi\frac{yv}{N}}\right]\mathrm{e}^{-\mathrm{j}2\pi\frac{xu}{M}} \\ &=\sum_{x=0}^{M-1}\{\mathscr{F}_y[f(x,y)]\}\mathrm{e}^{-\mathrm{j}2\pi\frac{xu}{M}} \\ &=\mathscr{F}_x\{\mathscr{F}_y[f(x,y)]\} \end{aligned} \tag{5-8}$$

式(5-8)称为二维离散傅里叶变换的可分性，表明二维离散傅里叶变换可用一维离散傅里叶变换来实现，即先对 $f(x,y)$的每一列进行一维离散傅里叶变换，得到$\mathcal{F}_y[f(x,y)]$，再对该中间结果的每一行进行一维离散傅里叶变换，得到 $F(u,v)$，运算过程中每个一维离散傅里叶变换可以采用快速傅里叶变换(Fast Fourier Transformation，FFT)实现快速运算。相反的顺序(先行后列)也可以。

MATLAB 中提供了实现离散傅里叶变换的相关函数，列举如下。

(1) fft 函数：实现一维离散傅里叶变换。

① fft(X)：对向量 X 进行离散傅里叶变换。若 X 为二维矩阵，则对它的每一列进行变

换；若为 N 维矩阵，则对其第一个长度非 1 的维进行变换。

② fft(X,N)：N 点离散傅里叶变换，若 X 长度小于 N，则补 0；若 X 长度大于 N，则截断。

③ fft(X,[],DIM)或者 fft(X,N,DIM)：在第 DIM 维进行离散傅里叶变换。

(2) ifft 函数：实现一维离散傅里叶反变换。

① ifft(X)：X 的离散傅里叶反变换。

② ifft(X,N)：通过补 0 或截断实现 X 的 N 点离散傅里叶反变换。

③ ifft(X,[],DIM)或者 ifft(X,N,DIM)：X 在 DIM 维的离散傅里叶反变换。

(3) fft2 函数：实现二维离散傅里叶变换。

① fft2(X)：对矩阵 X 实现二维离散傅里叶变换。

② fft2(X,MROWS,NCOLS)：通过补 0 或截断对 X 进行 MROWS×NCOLS 的二维离散傅里叶变换。

(4) ifft2 函数：实现二维离散傅里叶反变换。

① ifft2(F)：对矩阵 F 实现二维离散傅里叶反变换。

② ifft2(F,MROWS,NCOLS)：通过补 0 或截断对 F 进行 MROWS×NCOLS 的二维离散傅里叶反变换。

(5) fftshift(X)：将零频率分量移动到频谱中心。

(6) ifftshift(X)：将零频率分量移回原位，取消 fftshift 的效果。

【例 5-1】 对灰度图像进行离散傅里叶变换并显示频谱图。

程序如下：

```
clear,clc,close all;
grayI = imread('cameraman.tif');                 %打开灰度图像
DFT = fft2(grayI);                               %计算离散傅里叶变换
ADFT = abs(DFT);                                 %计算傅里叶谱
top = max(ADFT(:));
bottom = min(ADFT(:));
ADFT1 = (ADFT - bottom)/(top - bottom) * 100;    %把傅里叶谱系数规格化到[0 100],便于观察
ADFT2 = fftshift(ADFT1);                         %将规格化频谱图移位,低频移至频谱图中心
subplot(131),imshow(Image),title('原图');
subplot(132),imshow(ADFT1),title('原频谱图');
subplot(133),imshow(ADFT2),title('移位频谱图');
```

程序运行结果如图 5-1 所示。图 5-1(a)是原图；将傅里叶谱规格化到[0 100]的频谱图如图 5-1(b)所示，四角部分对应低频成分，中央部分对应高频成分；采用 fftshift 函数将频谱图进行移位，如图 5-1(c)所示，频谱图中间为低频部分，越靠外边频率越高。图像中的能量主要集中在低频区，高频能量很少或为 0。

【例 5-2】 对彩色图像进行离散傅里叶变换并显示频谱图。

彩色图像有 3 个色彩通道，其数据为 3 个二维矩阵，因此，需要进行 3 个二维离散傅里叶变换，将 3 个频谱图合成彩色频谱图。fft2 函数对于 M×N×3 的矩阵，即按照上述过程进行离散傅里叶变换，变换后的频谱矩阵同样为 M×N×3 的矩阵，为彩色频谱图。

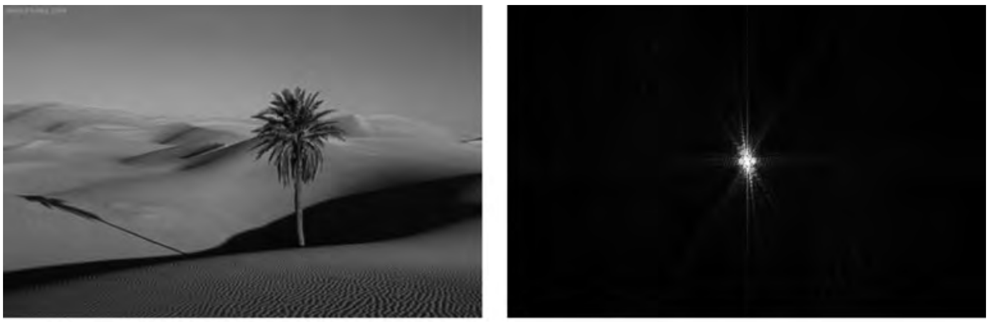
(a) 原图　　(b) 原频谱图　　(c) 移位频谱图

图 5-1　灰度图像傅里叶频谱图

程序如下：

```
clear,clc,close all;
Image = imread('desert.jpg');                    %彩色图像
DFT = fftshift(fft2(Image));                     %离散傅里叶变换并搬移频谱
DFT = abs(DFT);                                  %取傅里叶谱
ADFT = (DFT - min(DFT(:)))/(max(DFT(:)) - min(DFT(:))) * 100;
subplot(121),imshow(Image),title('原图');
subplot(122),imshow(ADFT),title('彩色频谱图');
```

程序运行结果如图 5-2 所示。从 MATLAB 工作区可以看到 Image 为 231×352×3 的 uint8 型矩阵，离散傅里叶变换同样是 231×352×3 的矩阵。



(a) 原图　　(b) 彩色频谱图

图 5-2　彩色图像傅里叶频谱图

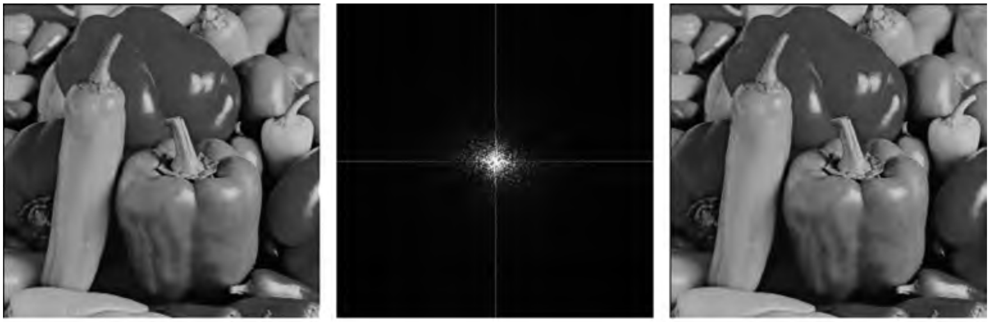
【例 5-3】 对图像进行离散傅里叶变换、显示频谱图和重建。

程序如下：

```
clear,clc,close all;
Image = imread('peppers.jpg');
DFT = fftshift(fft2(Image));
ADFT = abs(DFT);
```

```
top = max(ADFT(:));
bottom = min(ADFT(:));
ADFT = (ADFT - bottom)/(top - bottom) * 100;         % 计算傅里叶谱
recI = ifft2(ifftshift(DFT));                        % 离散傅里叶反变换
recI = abs(recI);                                    % 取复数的模
recI = uint8(recI);                                  % 变 double 型数据为整型数据,以便显示图像
subplot(131),imshow(Image),title('原图');
subplot(132),imshow(ADFT),title('彩色频谱图');
subplot(133),imshow(recI),title('DFT 重建图');
```

程序运行结果如图 5-3 所示。



(a) 原图　　(b) 彩色频谱图　　(c) 离散傅里叶变换重建图

图 5-3　离散傅里叶变换变换并重建

5.1.3　离散傅里叶变换的性质

离散傅里叶变换有许多重要性质,这些性质给离散傅里叶变换的运算和实际应用提供了极大的便利。这里主要介绍几个和二维离散傅里叶变换在图像处理中的应用密切相关的性质。

1. 线性和周期性

若 $\mathscr{F}[f(x,y)]=F(u,v)$,$0\leqslant x,u\leqslant M$,$0\leqslant y,v\leqslant N$,则

$$\mathscr{F}[a_1 f_1(x,y)+a_2 f_2(x,y)]=a_1\mathscr{F}[f_1(x,y)]+a_2\mathscr{F}[f_2(x,y)] \tag{5-9}$$

$$\begin{cases} F(u,v)=F(u+M,v)=F(u,v+N)=F(u+M,v+N) \\ f(x,y)=f(x+M,y)=f(x,y+N)=f(x+M,y+N) \end{cases} \tag{5-10}$$

式(5-10)表明,尽管 $F(u,v)$对无穷多个 u 和 v 的值重复出现,但只需根据在任一个周期里的值就可从 $F(u,v)$得到 $f(x,y)$;同样只需一个周期里的变换就可将 $F(u,v)$在频域里完全确定。

2. 几何变换性

1）共轭对称性

若 $\mathscr{F}[f(x,y)]=F(u,v)$，$F^*(-u,-v)$是 $f(-x,-y)$的离散傅里叶变换的共轭函数，则

$$F(u,v)=F^*(-u,-v) \tag{5-11}$$

2）平移性

若 $\mathscr{F}[f(x,y)]=F(u,v)$，则

$$\begin{aligned} f(x-x_0,y-y_0) &\Leftrightarrow F(u,v)\mathrm{e}^{-\mathrm{j}2\pi\left(\frac{x_0u}{M}+\frac{y_0v}{N}\right)} \\ f(x,y)\mathrm{e}^{\mathrm{j}2\pi\left(\frac{xu_0}{M}+\frac{yv_0}{N}\right)} &\Leftrightarrow F(u-u_0,v-v_0) \end{aligned} \tag{5-12}$$

式(5-12)表示对图像平移不影响其傅里叶变换的幅值，只改变相位谱。当 $u_0=M/2$，$v_0=N/2$ 时，$\mathrm{e}^{\mathrm{j}2\pi(u_0x/M+v_0y/N)}=\mathrm{e}^{\mathrm{j}\pi(x+y)}=(-1)^{x+y}$，则 $f(x,y)(-1)^{x+y}\Leftrightarrow F(u-M/2,v-N/2)$，频域的坐标原点从起始点(0,0)移至中心点，只要将 $f(x,y)$乘以$(-1)^{x+y}$ 因子，再进行傅里叶变换即可实现，即例 5-1 中的频谱搬移。

3）旋转性

把 $f(x,y)$和 $F(u,v)$表示为极坐标形式，若 $f(\gamma,\theta)\Leftrightarrow F(k,\varphi)$，则

$$f(\gamma,\theta+\theta_0)\Leftrightarrow F(k,\varphi+\theta_0) \tag{5-13}$$

若将空间域函数旋转角度 θ_0，那么在变换域此函数的离散傅里叶变换也旋转同样的角度；反之，若将变换域函数旋转某一角度，则空间域函数也旋转同样的角度。

4）比例变换特性

若 $\mathscr{F}[f(x,y)]=F(u,v)$，则

$$f(ax,by)\Leftrightarrow \frac{1}{|ab|}F\left(\frac{u}{a},\frac{v}{b}\right) \tag{5-14}$$

对图像 $f(x,y)$在空间尺度的缩放导致其傅里叶变换 $F(u,v)$在频域尺度的相反缩放。

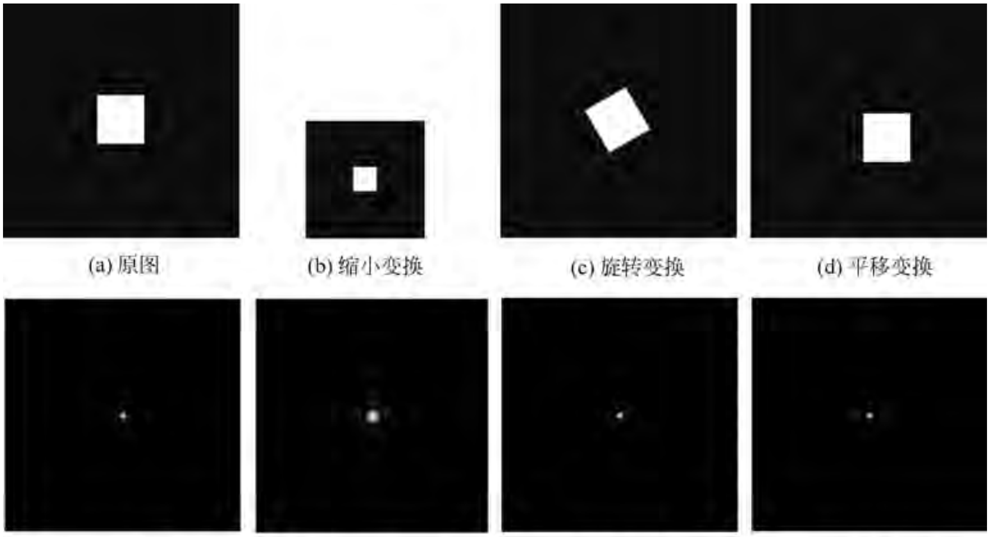
【例 5-4】 对一幅图像进行几何变换，再进行离散傅里叶变换运算，验证以上性质。

程序如下：

```
clear,clc,close all;
Image = rgb2gray(imread('block.bmp'));
[h,w,c] = size(Image);
scale = imresize(Image,0.5,'bilinear');                        %缩小变换
rotate = imrotate(Image,30,'bilinear','crop');                 %旋转变换
tform = affine2d([1 0 0;0 1 0;20 20 1]);
R = imref2d([h,w],[1 w],[1 h]);
trans = imwarp(Image,tform,'FillValue',0,'OutputView',R);      %平移变换
Originaldft = abs(fftshift(fft2(Image)));                      %原图离散傅里叶变换
Scaledft = abs(fftshift(fft2(scale)));                         %缩小图离散傅里叶变换
Rotatedft = abs(fftshift(fft2(rotate)));                       %旋转图离散傅里叶变换
```

```
Transdft = abs(fftshift(fft2(trans)));                    % 平移图离散傅里叶变换
figure,imshow(Image),title('原图');
figure,imshow(scale),title('缩小变换');
figure,imshow(rotate),title('旋转变换');
figure,imshow(trans),title('平移变换');
figure,imshow(Originaldft,[]),title('原图 DFT');
figure,imshow(Scaledft,[]),title('比例变换 DFT');
figure,imshow(Rotatedft,[]),title('旋转变换 DFT');
figure,imshow(Transdft,[]),title('平移变换 DFT');
```

程序运行结果如图 5-4 所示。可以看出，缩小变换后图像的频谱图尺度展宽，旋转后图像的频谱图随着旋转，平移后图像的频谱图没有变化。



(a) 原图 (b) 缩小变换 (c) 旋转变换 (d) 平移变换

(e) 原图傅里叶谱 (f) 缩小图傅里叶谱 (g) 旋转图傅里叶谱 (h) 平移图傅里叶谱

图 5-4 图像几何变换及其傅里叶频谱图

3. Parseval 定理

若 $\mathscr{F}[f(x,y)]=F(u,v)$，则

$$\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}|f(x,y)|^2=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}|F(u,v)|^2 \tag{5-15}$$

Parseval 定理也称为能量守恒定理，这个性质说明信号经傅里叶变换后不损失能量，只是改变了信号的表现形式，是变换编码的基本条件。

【例 5-5】 对图像进行离散傅里叶变换并计算时域和频域能量。

程序如下：

```
clear,clc,close all;
grayI = double(imread('cameraman.tif'));
```

```
[N,M,C] = size(grayI);
SI = grayI.^2;                          % 计算|f(x,y)|²
energyT = sum(SI(:));                   % 时域能量
DFT = abs(fft2(grayI));                 % 离散傅里叶变换
SDFT = DFT.^2;                          % 计算|F(u,v)|²
energyF = sum(SDFT(:));                 % 频域能量
energyF = energyF/(N * M);              % 频域能量除以采样点数,消除采样增益
diff = abs(energyF - energySI);         % 时域、频域能量差
```

运行程序,在 MATLAB 工作区可以看到 diff 值约为 0,即时域和频域能量一致。

4. 卷积定理

若 $\mathscr{F}[f(x,y)]=F(u,v)$,$\mathscr{F}[g(x,y)]=G(u,v)$,则

$$\begin{aligned} f(x,y)*g(x,y) &\Leftrightarrow F(u,v)\cdot G(u,v) \\ f(x,y)\cdot g(x,y) &\Leftrightarrow F(u,v)*G(u,v) \end{aligned} \tag{5-16}$$

在以上几个性质中,可分性使得二维离散傅里叶变换可以通过一维快速傅里叶变换快速实现;共轭对称性、平移性、旋转性、比例变换特性使得二维离散傅里叶变换具有一定的几何变换不变性,可以作为一种图像特征;Parseval 定理是变换编码的基本条件;卷积定理可以降低某些复杂图像处理算法的计算量,这几个性质在图像处理中应用较多。

5.1.4 离散傅里叶变换在图像处理中的应用

离散傅里叶变换在图像处理中的应用主要包括用于描述图像信息、滤波、压缩及便于运算几个方面。

1. 傅里叶描绘子

从原始图像中产生的数值、符号或者图形称为图像特征,反映了原图像的重要信息和主要特性,以便于让计算机有效地识别目标。这些表征图像特征的一系列符号称为描绘子。

描绘子应具有几何变换不变性,即在图像内容不变,仅产生几何变换(平移、旋转、缩放等)的情况下,描绘子不变,以保证识别结果的稳定性。离散傅里叶变换在图像特征提取方面应用较多,本小节主要介绍离散傅里叶变换直接作为特征的应用——傅里叶描绘子。

一个闭合区域,区域边界上的点(x,y)用复数表示为 $x+\mathrm{j}y$。沿边界跟踪一周,得到一个复数序列 $z(n)=x(n)+\mathrm{j}y(n)$,$n=0,1,\cdots,N-1$,$z(n)$为周期信号,其离散傅里叶变换系数用 $Z(k)$表示,$Z(k)$称为傅里叶描绘子。

根据离散傅里叶变换特性,$Z(k)$幅值具有旋转和平移不变性,相位信息具有缩放不变性,在一定程度上满足了描绘子的几何变换不变性,可以作为一种图像特征。

2. 离散傅里叶变换在图像滤波中的应用

经过离散傅里叶变换后,在傅里叶频谱中,中间部分为低频部分,越靠外边频率越高。

因此,可以在离散傅里叶变换后,设计相应的滤波器,实现高通滤波、低通滤波等的处理。

【例 5-6】 对图像进行离散傅里叶变换及频域滤波。

程序如下:

```
clear,clc,close all;
Image = imread('desert.jpg');
grayIn = rgb2gray(Image);
[h,w] = size(grayIn);
DFTI = fftshift(fft2(grayIn));                          % 离散傅里叶变换及频谱搬移
cf = 30;                                                % 截止频率
HDFTI = DFTI;
HDFTI(h/2 - cf:h/2 + cf,w/2 - cf:w/2 + cf) = 0;         % 低频置为 0
grayOut1 = uint8(abs(ifft2(ifftshift(HDFTI))));         % 离散傅里叶反变换
LDFTI = zeros(h,w);
LDFTI(h/2 - cf:h/2 + cf,w/2 - cf:w/2 + cf) = DFTI(h/2 - cf:h/2 + cf,w/2 - cf:w/2 + cf); % 高频置为 0
grayOut2 = uint8(abs(ifft2(ifftshift(LDFTI))));         % 离散傅里叶反变换
subplot(131),imshow(Image),title('原图');
subplot(132),imshow(uint8(grayOut1)),title('高通滤波');
subplot(133),imshow(uint8(grayOut2)),title('低通滤波');
```

程序运行结果如图 5-5 所示。程序中进行的实际是理想高通和低通滤波器,为提高滤波性能,也有其他类型的滤波器,详见第 6 章相关内容。

(a) 原图　(b) 高通滤波　(c) 低通滤波

图 5-5　图像频域滤波示例

3. 离散傅里叶变换在图像压缩中的应用

由 Parseval 定理知,信号经傅里叶变换前后能量不发生损失,只是改变了信号的表现形式,离散傅里叶变换系数表现的是各个频率点上的幅值。高频反映细节、低频反映景物概貌,往往认为可将高频系数置为 0,降低数据量;同时由于人眼的惰性,合理地设置高频系数为 0,可使图像质量一定范围内的降低不会被人眼察觉到。因此,离散傅里叶变换可以方便地进行压缩编码。

4. 离散傅里叶变换卷积性质的应用

抽象来看,图像处理算法可以认为是图像信息经过了滤波器的滤波(如平滑滤波、锐

化滤波等)，空间域滤波通常需要进行卷积运算。如果滤波器的结构比较复杂，可以利用离散傅里叶变换的卷积性质，把空间域卷积变为变换域的相乘，以简化运算，如式(5-17)所示。

$$\begin{cases} f_g = g * f \\ F_g(u,v) = G(u,v) \cdot F(u,v) \\ f_g = \mathrm{IDFT}(F_g) \end{cases} \tag{5-17}$$

式中，f 为原图像，g 为滤波器，利用 g 对 f 滤波，是用 g 和 f 卷积得到 f_g。这个过程可以改变为先对 f、g 进行离散傅里叶变换，把 G 和 F 相乘得 F_g，再进行傅里叶反变换，以降低卷积计算量。

注意：由于离散傅里叶变换和离散傅里叶反变换都是周期函数，因此在计算卷积时，需要让这两个离散函数具有同样的周期，否则将产生错误。利用快速傅里叶变换计算卷积，为防止频谱混叠误差，需对离散的二维函数补零，即周期延拓，两个函数同时周期延拓，使其具有相同的周期。

5.2 离散余弦变换

在傅里叶级数展开式中，如果被展开的函数是实偶函数，那么其傅里叶级数中只包含余弦项，再将其离散化可导出余弦变换，因此称为 DCT(Discrete Cosine Transform，离散余弦变换)。

5.2.1 离散余弦变换的定义

1. 一维 DCT

对于有限长数字序列 $f(x)$，$x=0,1,\cdots,N-1$，其一维 DCT 和 IDCT(离散余弦反变换)定义为：

$$\begin{cases} F(u) = C(u)\sqrt{\dfrac{2}{N}}\displaystyle\sum_{x=0}^{N-1} f(x)\cos\dfrac{(2x+1)u\pi}{2N} \\ f(x) = \sqrt{\dfrac{2}{N}}\displaystyle\sum_{u=0}^{N-1} C(u)F(u)\cos\dfrac{(2x+1)u\pi}{2N} \end{cases} \tag{5-18}$$

其中，$x,u=0,1,\cdots,N-1$，$C(u)=\begin{cases} 1/\sqrt{2} & u=0 \\ 1 & u=1,2,\cdots,N-1 \end{cases}$

将一维 DCT 变换表示成矩阵运算形式，即

$$\boldsymbol{F} = \boldsymbol{A}\boldsymbol{f} \tag{5-19}$$

$$
\boldsymbol{A}=\sqrt{\frac{2}{N}}\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \cos\frac{1}{2N}\pi & \cos\frac{3}{2N}\pi & \cdots & \cos\frac{(2N-1)}{2N}\pi \\ \vdots & \vdots & \ddots & \vdots \\ \cos\frac{N-1}{2N}\pi & \cos\frac{3(N-1)}{2N}\pi & \cdots & \cos\frac{(2N-1)(N-1)}{2N}\pi\end{bmatrix} \tag{5-20}
$$

其中，$\boldsymbol{F}$ 为变换系数矩阵，$\boldsymbol{A}$ 为正交变换矩阵，$\boldsymbol{f}$ 为时域数据矩阵。

一维 DCT 逆变换的矩阵形式表示为：

$$
\boldsymbol{f}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{F} \tag{5-21}
$$

2. 二维 DCT

数字图像为二维数据，把一维 DCT 推广到二维，二维 DCT 变换和反变换定义为：

$$
\begin{cases}F(u,v)=\dfrac{2}{\sqrt{MN}}C(u)C(v)\displaystyle\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\cos\left[\frac{\pi(2x+1)u}{2M}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right] \\ f(x,y)=\dfrac{2}{\sqrt{MN}}\displaystyle\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}C(u)C(v)F(u,v)\cos\left[\frac{\pi(2x+1)u}{2M}\right]\cos\left[\frac{\pi(2y+1)v}{2N}\right]\end{cases} \tag{5-22}
$$

式中，$\begin{cases}x,u=0,1,2,\cdots,M-1 \\ y,v=0,1,2,\cdots,N-1\end{cases}$，$C(u),C(v)=\begin{cases}1/\sqrt{2} & u,v=0 \\ 1, & \text{其他}\end{cases}$

二维 DCT 的矩阵形式表示为：

$$
\boldsymbol{F}=\boldsymbol{A}\boldsymbol{f}\boldsymbol{A}^{\mathrm{T}} \tag{5-23}
$$

二维 IDCT 的矩阵形式表示为：

$$
\boldsymbol{f}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{A} \tag{5-24}
$$

其中，$\boldsymbol{F}$ 为 DCT 变换系数矩阵，$\boldsymbol{f}$ 为空域数据矩阵，$\boldsymbol{A}$ 为正交变换矩阵。

5.2.2 离散余弦变换的实现

【例 5-7】 用矩阵运算对图像进行 DCT 变换。

程序如下：

```
clear,clc,close all;
Image = double(imread('cameraman.tif'));
[N,M,C] = size(Image);                    %图像尺寸,本例中图像为 256×256,N 和 M 相等
A = zeros(N,N);
A(1,1:N) = 1/sqrt(2);                     %系数矩阵 A 的第 1 行
for j = 2:N
    for i = 1:N
```

```
            A(j,i) = cos((2 * i - 1) * pi * (j - 1)/(2 * N));
        end
    end
    A = sqrt(2/N) * A;                              % 生成系数矩阵 A
    DCT = A * Image * A';                           % 二维 DCT 矩阵运算
    DCT = abs(DCT);
    top = max(DCT(:));
    bottom = min(DCT(:));
    DCT = (DCT - bottom)/(top - bottom) * 100;      % DCT 系数规格化
    imshow(DCT),title('DCT 频谱图');
```

程序运行结果如图 5-6 所示。从所得的结果可以看出，离散余弦变换具有使信息集中的特点，对图像进行 DCT 变换后，在变换域中，矩阵左上角低频的幅值大，而右下角高频的幅值小。

(a) 原灰度图　　(b) DCT频谱图

图 5-6　图像 DCT 变换

MATLAB 中提供了实现 DCT 变换的相关函数，列举如下。

(1) dct 函数：实现 DCT 变换。

① Y=dct(X)：对向量 X 进行 DCT 变换，返回 Y；若 X 是二维矩阵，则对其每一列进行 DCT 变换；若 X 是 N 维矩阵，则对其第 1 个长度非 1 的维进行变换。

② Y=dct(X,N)：通过补 0 或截断对向量 X 进行 N 点 DCT 变换。

③ Y=dct(X,[],DIM)或 Y=dct(X,N,DIM)：对 X 的第 DIM 维进行 DCT 变换。

④ Y=dct(…,'Type',K)：指定计算方式执行 DCT 变换，K 可取 1、2、3、4，分别代表 dct-I、dct-II、dct-III、dct-IV 变换，默认值为 2。

(2) idct 函数：实现 IDCT 变换。

① X=idct(Y)：IDCT 变换。

② X=idct(Y,N)：通过补 0 或截断对向量 Y 进行 N 点 IDCT 变换。

③ X=idct(Y,[],DIM)或 X=idct(Y,N,DIM)：对 Y 的第 DIM 维进行 IDCT 变换。

④ X=idct(…,'Type',K)：指定计算方式执行 IDCT 变换，K 可取 1、2、3、4，分别代表 idct-I、idct-II、idct-III、idct-IV 变换，默认值为 2。

(3) dct2 函数：实现二维 DCT 变换。

① B=dct2(A)：对矩阵 A 进行二维 DCT 变换，返回矩阵 B。

② B=dct2(A,[M N])或 B=dct2(A,M,N)：通过补 0 或截断对矩阵 A 进行 M×N 的 DCT 变换。

(4) idct2 函数：实现二维 IDCT 变换。

① B=idct2(A)：对矩阵 A 进行二维 IDCT 变换。

② B=idct2(A,[M N])或 B=idct2(A,M,N)：通过补 0 或截断对矩阵 A 进行 M×N 的 IDCT 变换。

(5) D=dctmtx(N)：返回 N×N 的 DCT 变换矩阵 D(即式(5-20)中的变换矩阵 A)。

(6) B=blockproc(A,[M N],FUN)：对图像 A 中的每一 M×N 块执行 FUN 定义的操作。函数将图像 A 中的每一数据块打包为"block struct"传递给用户定义的函数 FUN，返回矩阵(向量、标量)Y，Y=FUN(BLOCK_STRUCT)。"block struct"是 MATLAB 中定义的结构体，包括该块的信息，相关字段如表 5-1 所示。

表 5-1 block struct 结构体字段

参数	含义
border	二维向量[V H]，指定数据块的水平、垂直边界
blockSize	二维向量[rows cols]，指定块大小，若'border'被指定，块大小不包括边界像素
data	M×N 或者 M×N×P 的块数据矩阵
imageSize	二维向量[rows cols]，指定输入图像大小
location	二维向量[row col]，指定输入图像中数据块的第 1 个像素(第 1 行第 1 列)，不包括边界像素

【例 5-8】 对灰度图像进行 DCT 变换并显示频谱图。

程序如下：

```
clear,clc,close all;
grayI = imread('cameraman.tif');
DCT = dct2(grayI);                              %进行离散余弦变换
ADCT = abs(DCT);                                %求模
top = max(ADCT(:));
bottom = min(ADCT(:));
result = (ADCT - bottom)/(top - bottom) * 100;  %把模规格化到[0 100]
subplot(121),imshow(grayI),title('原灰度图');
subplot(122),imshow(result),title('灰度图 DCT 频谱图');
```

程序运行结果如图 5-7 所示。从 3 幅灰度图的频谱可以看出，能量主要集中在左上角低频分量处。

【例 5-9】 对彩色图像进行 DCT 变换、显示频谱图并重建原图。

程序如下：

(a) 原灰度图

(b) 对应的DCT频谱图

图 5-7　灰度图像的 DCT 频谱图

```
clear,clc,close all;
RGBI = imread('peppers.jpg');                          % 打开彩色图像
[N,M,C] = size(RGBI);
DCTS = zeros(N,M,C);                                   % DCT 频谱图初始化
RGBOut = zeros(N,M,C);                                 % 重建图初始化
for i = 1:C                                            % 分色彩通道处理
    channel = RGBI(:,:,i);
    DCT = dct2(channel);                               % DCT 变换
    ADCT = abs(DCT);
    top = max(ADCT(:));       bottom = min(ADCT(:));
    DCTS(:,:,i) = (ADCT - bottom)/(top - bottom) * 100;
    RGBOut(:,:,i) = abs(idct2(DCT));                   % IDCT 变换
end
RGBOut = uint8(RGBOut);
subplot(131),imshow(RGBI),title('原彩色图');
subplot(132),imshow(DCTS),title('彩色图 DCT 频谱图');
subplot(133),imshow(RGBOut),title('DCT 重建图');
```

程序运行结果如图 5-8 所示。

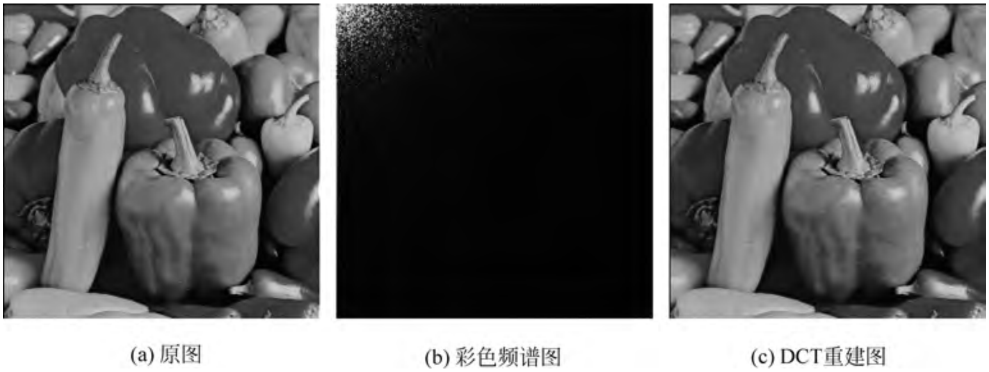


图 5-8 彩色图像 DCT 变换并重建

5.2.3 离散余弦变换在图像处理中的应用

离散余弦变换在图像处理中主要用于对图像(包括静止图像和运动图像)进行有损数据压缩。例如,静止图像编码标准 JPEG、运动图像编码标准 MPEG 中都使用了离散余弦变换,这是由于离散余弦变换具有很强的"能量集中"特性,大多数的能量都集中在离散余弦变换后的低频部分,压缩编码效果较好。

具体的做法一般是先把图像分成 8×8 的块,对每一个方块进行二维 DCT 变换,变换后的能量主要集中在低频区。对 DCT 系数进行量化,给高频系数大间隔量化,低频部分小间隔量化,舍弃绝大部分取值很小或为 0 的高频数据,降低数据量,同时保证重构图像不会发生显著失真。

【例 5-10】 对图像进行 DCT 变换,将高频置为 0,并进行反变换重建。

程序如下:

```
clear,clc,close all;
grayIn = imread('cameraman.tif');
[h,w] = size(grayIn);
DCTI = dct2(grayIn);
cf = 90;                                  %截止频率
FDCTI = zeros(h,w);
FDCTI(1:cf,1:cf) = DCTI(1:cf,1:cf);       %将高频系数置 0
grayOut = uint8(abs(idct2(FDCTI)));
subplot(121),imshow(grayIn),title('原图');
subplot(122),imshow(grayOut),title('压缩重建');
```

程序运行结果如图 5-9 所示。因为丢失了部分高频系数,所以重建图比原图模糊。程序中将大于截止频率的高频系数置 0,并没有考虑系数的大小,截止频率的指定也缺乏依据。

(a) 原图

(b) DCT压缩重建图像

图 5-9 DCT 压缩重建示例一

【例 5-11】 将彩色图像转换到 YCbCr 空间，分别对亮度和色度数据进行 DCT 变换，采用合理的方式将高频系数变为 0，并进行反变换重建。

本例中通过将 DCT 系数除以合适的数据以便将高频系数变为 0，称为量化；各 DCT 系数除以的数据称为量化步长，将在第 11 章具体学习。

程序如下：

```
clear,clc,close all;
ImageIn = imread('peppers.jpg');                    %读取彩色图像
YCbCrIn = double(rgb2ycbcr(ImageIn));               %转换到 YCbCr 空间
YQT = [16  11  10  16   24   40   51   61;
       12  12  14  19   26   58   60   55;
       14  13  16  24   40   57   69   56;
       14  17  22  29   51   87   80   62;
       18  22  37  56   68  109  103   77;
       24  35  55  64   81  104  113   92;
       49  64  78  87  103  121  120  101;
       72  92  95  98  112  100  103   99];          %亮度量化表,即块内各数据对应的量化步长
CQT = [17  18  24  47  99  99  99  99;
       18  21  26  66  99  99  99  99;
       24  26  56  99  99  99  99  99;
       47  66  99  99  99  99  99  99;
       99  99  99  99  99  99  99  99;
       99  99  99  99  99  99  99  99;
       99  99  99  99  99  99  99  99;
       99  99  99  99  99  99  99  99];                %色度量化表
blocksize = 8;                                       %定义块大小
A = dctmtx(blocksize);                               %计算变换矩阵
FUN1 = @(block_struct) A * block_struct.data * A';   %定义块 DCT 变换函数
FUN2 = @(block_struct) A' * block_struct.data * A;   %定义块 IDCT 变换函数
```

```
YCbCrOut = zeros(size(YCbCrIn));                              % 对输出的 YCbCr 图像初始化
for i = 1:3
    channel = YCbCrIn(:,:,i);                                 % 获取 YCbCr 各通道
    DCT = blockproc(channel,[blocksize,blocksize],FUN1);      % 块 DCT 变换
    if i == 1
        QT = YQT;
    else
        QT = CQT;
    End                                                       % 选择亮度或色度量化表
    FUN3 = @(block_struct) round(block_struct.data./QT);      % 定义量化函数,点除运算
    QDCT = blockproc(DCT,[blocksize,blocksize],FUN3);         % 块 DCT 系数量化
    FUN4 = @(block_struct) block_struct.data.*QT;             % 定义反量化函数,点乘运算
    IQDCT = blockproc(QDCT,[blocksize,blocksize],FUN4);       % 块量化后的数据反量化
    IDCT = blockproc(IQDCT,[blocksize,blocksize],FUN2);       % 块系数 IDCT 变换
    YCbCrOut(:,:,i) = IDCT;                                   % 存储 YCbCr 图像数据
end
YCbCrOut = uint8(YCbCrOut);
ImageOut = ycbcr2rgb(YCbCrOut);                               % 变换回 RGB 空间
figure,imshow(ImageIn),title('原图');
figure,imshow(ImageOut),title('重建图像');
```

程序运行结果如图 5-10 所示。程序中实现了色彩空间变换、DCT 变换、量化、反量化、IDCT 变换、色彩空间逆变换等操作,数据损失发生在量化一步。从图 5-10 可以看出,重建后的图像质量有所下降,但不明显,比直接指定截止频率更合理。

(a) 原图

(b) 重建图像

图 5-10 DCT 压缩重建示例二

5.3 K-L 变换

K-L 变换是建立在统计特性基础上的一种变换,又称为霍特林(Hotelling)变换或主成分分析(Principal Component Analysis,PCA),其突出优点是相关性好,是均方误差(Mean Square Error,MSE)意义下的最佳变换,在数据压缩技术中占有重要地位。

5.3.1 K-L 变换原理

1. K-L 变换的定义

用确定的正交归一向量系 $u_j, j=1,2,\cdots,\infty$ 展开向量 $\boldsymbol{X}$：

$$\boldsymbol{X}=\sum_{j=1}^{\infty}a_j u_j \tag{5-25}$$

用有限的 m 项来估计向量 $\boldsymbol{X}$，即

$$\hat{\boldsymbol{X}}=\sum_{j=1}^{m}a_j u_j \tag{5-26}$$

表示成矩阵形式为：

$$\hat{\boldsymbol{X}}=\boldsymbol{UA} \tag{5-27}$$

为了找到合适的变换矩阵 $\boldsymbol{U}$，计算用有限项展开代替无限项展开引起的均方误差，使均方误差最小的 $\boldsymbol{U}$ 最优。均方误差如下所示：

$$\overline{\varepsilon^2}=E[(\boldsymbol{X}-\hat{\boldsymbol{X}})^{\mathrm{T}}(\boldsymbol{X}-\hat{\boldsymbol{X}})]=E\left[\sum_{j=m+1}^{\infty}a_j u_j\cdot\sum_{j=m+1}^{\infty}a_j u_j\right]$$

利用已知条件求解均方误差。$\boldsymbol{u}$ 为正交归一向量系，$\boldsymbol{u}_i^{\mathrm{T}}\boldsymbol{u}_j=\begin{cases}1 & i=j\\ 0 & i\neq j\end{cases}$；且 $a_j=\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{X}$，所以

$$\overline{\varepsilon^2}=E\left[\sum_{j=m+1}^{\infty}a_j^2\right]=E\left[\sum_{j=m+1}^{\infty}u_j^{\mathrm{T}}\boldsymbol{XX}^{\mathrm{T}}u_j\right]=\sum_{j=m+1}^{\infty}u_j^{\mathrm{T}}E[\boldsymbol{XX}^{\mathrm{T}}]u_j$$

令 $\psi=E[\boldsymbol{XX}^{\mathrm{T}}]$，则

$$\overline{\varepsilon^2}=\sum_{j=m+1}^{\infty}\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{\psi}\boldsymbol{u}_j$$

利用拉格朗日乘数法求均方误差取极值时的 $\boldsymbol{u}$，拉格朗日函数为：

$$h(\boldsymbol{u}_j)=\sum_{j=m+1}^{\infty}\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{\psi}\boldsymbol{u}_j-\sum_{j=m+1}^{\infty}\lambda[\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{u}_j-1]$$

对 $\boldsymbol{u}_j$ 求导数，得

$$(\boldsymbol{\psi}-\lambda_j\boldsymbol{I})\boldsymbol{u}_j=0,\quad j=m+1,\cdots,\infty$$

其中，λ_j 是 $\boldsymbol{X}$ 的自相关矩阵 $\boldsymbol{\psi}$ 的特征值，$\boldsymbol{u}_j$ 是对应的特征向量。
则有：

$$\overline{\varepsilon^2}=\sum_{j=m+1}^{\infty}\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{\psi}\boldsymbol{u}_j=\sum_{j=m+1}^{\infty}\boldsymbol{u}_j^{\mathrm{T}}\lambda_j\boldsymbol{u}_j=\sum_{j=m+1}^{\infty}\lambda_j \tag{5-28}$$

得到以下结论：以 $\boldsymbol{X}$ 的自相关矩阵 $\boldsymbol{\psi}$ 的 m 个最大特征值对应的特征向量来逼近 $\boldsymbol{X}$ 时，其截断均方误差具有极小性质。这 m 个特征向量所组成的正交坐标系 $\boldsymbol{U}$ 称作 $\boldsymbol{X}$ 所在的 n 维空间的 m 维 K-L 变换坐标系。$\boldsymbol{X}$ 在 K-L 坐标系上的展开系数向量 $\boldsymbol{A}$ 称作 $\boldsymbol{X}$ 的 K-L 变

换，满足：

$$\begin{cases}\boldsymbol{A}=\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}\\ \boldsymbol{X}=\boldsymbol{U}\boldsymbol{A}\end{cases} \tag{5-29}$$

其中，$\boldsymbol{U}=(u_1 \quad u_2 \quad \cdots \quad u_m)$。

2. K-L 变换的性质

因$\boldsymbol{\psi}\boldsymbol{u}_j=\lambda_j\boldsymbol{u}_j$，则$\boldsymbol{\psi}\boldsymbol{U}=\boldsymbol{U}\boldsymbol{D}_\lambda$，$\boldsymbol{D}_\lambda$ 为对角矩阵，其互相关成分都应为 0，即

$$\boldsymbol{D}_\lambda=\begin{pmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_n\end{pmatrix} \tag{5-30}$$

因 $\boldsymbol{U}$ 为正交矩阵，所以$\boldsymbol{\psi}=\boldsymbol{U}\boldsymbol{D}_\lambda\boldsymbol{U}^{\mathrm{T}}$。

因 $\boldsymbol{X}=\boldsymbol{U}\boldsymbol{A}$，则$\boldsymbol{\psi}=E[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}]=E[\boldsymbol{U}\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}]=\boldsymbol{U}E[\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}]\boldsymbol{U}^{\mathrm{T}}$，所以，

$$E[\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}]=\boldsymbol{D}_\lambda \tag{5-31}$$

由式(5-31)可知，变换后的向量 $\boldsymbol{A}$ 的自相关矩阵$\boldsymbol{\psi}_A$ 是对角矩阵，且对角元素是 $\boldsymbol{X}$ 的自相关矩阵$\boldsymbol{\psi}$ 的特征值。显然，通过 K-L 变换，消除了原有向量 $\boldsymbol{X}$ 的各分量之间的相关性，即变换后的数据 $\boldsymbol{A}$ 的各分量之间的信息是相互独立的。

3. 信息量分析

通过前文的分析可知，数据 $\boldsymbol{X}$ 的 K-L 坐标系的产生矩阵采用的是自相关矩阵$\boldsymbol{\psi}=E[\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}]$，由于总体均值向量$\boldsymbol{\mu}$ 常常没有什么意义，因此也常常把数据的协方差矩阵作为 K-L 坐标系的产生矩阵，如式(5-32)所示。

$$\Sigma=E[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^{\mathrm{T}}] \tag{5-32}$$

已知 $a_1=\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{X}$，计算 a_1 的方差如下：

$$\mathrm{var}(a_1)=E[a_1^2]-E[a_1]^2=E[\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{u}_1]-E[\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{X}]E[\boldsymbol{X}^{\mathrm{T}}\boldsymbol{u}_1]=\boldsymbol{u}_1^{\mathrm{T}}\Sigma\boldsymbol{u}_1$$

$\boldsymbol{u}_1$ 为 Σ 的特征向量，λ_1 为对应的特征值，则 $\mathrm{var}(a_1)=\boldsymbol{u}_1^{\mathrm{T}}\Sigma\boldsymbol{u}_1=\lambda_1\boldsymbol{u}_1^{\mathrm{T}}\boldsymbol{u}_1=\lambda_1$，即 a_1 的方差为 Σ 最大的特征值，a_1 称作第一主成分。

采用大特征值对应的特征向量组成变换矩阵，能对应地保留原向量中方差最大的成分，K-L 变换起到了减小相关性、突出差异性的效果，称之为主成分分析。

计算主成分的方差之和为：

$$\sum_{j=1}^{n}\mathrm{var}(a_j)=\sum_{j=1}^{n}\lambda_j=\mathrm{tr}(\Sigma)=\sum_{j=1}^{n}\mathrm{var}(X_j)$$

上式说明，n 个互不相关的主成分的方差之和等于原数据的总方差，即 n 个互不相关的主成分包含了原数据中的全部信息，各主成分的贡献率依次递减，第一主成分贡献率最大，数据的大部分信息集中在较少的几个主成分上。

主成分 a_i 的贡献率为：

$$\lambda_i \Big/ \sum_{j=1}^{n} \lambda_j, \quad i=1,2,\cdots,n \tag{5-33}$$

前 m 个主成分的累积贡献率反映前 m 个主成分综合原始变量信息的能力，定义如下：

$$\sum_{i=1}^{m} \lambda_i \Big/ \sum_{j=1}^{n} \lambda_j \tag{5-34}$$

【例 5-12】 设向量集为 $\begin{Bmatrix} \omega_1: (0 \quad 0 \quad 0)^{\mathrm{T}}, (1 \quad 0 \quad 1)^{\mathrm{T}}, (1 \quad 0 \quad 0)^{\mathrm{T}}, (1 \quad 1 \quad 0)^{\mathrm{T}} \\ \omega_2: (0 \quad 0 \quad 1)^{\mathrm{T}}, (0 \quad 1 \quad 1)^{\mathrm{T}}, (0 \quad 1 \quad 0)^{\mathrm{T}}, (1 \quad 1 \quad 1)^{\mathrm{T}} \end{Bmatrix}$，采用其自相关矩阵作为产生矩阵对其进行 K-L 变换，实现向二维降维，即求其前两个主成分。

程序如下：

```
clear,clc,close all;
X = [0 0 0;1 0 1;1 0 0;1 1 0;0 0 1;0 1 1;0 1 0;1 1 1]';
[n,N] = size(X);                                    % n 为维数,N 为样本数
V = X * X'/N;                                       % 求自相关矩阵
[coeff,D] = eigs(V);                                % 求特征值和特征向量
[D_sort,index] = sort(diag(D),'descend');           % 特征值降序排列
D = D(index,index);                                 % 按序调整特征值对角矩阵
coeff = coeff(:,index);                             % 按序调整特征向量矩阵
score = coeff' * X;                                 % K - L 变换
figure; plot(score(1,:),score(2,:),'ko'),title('K - L 变换');
xlabel('第一主成分得分');ylabel('第二主成分得分');
reconstructed = score' * coeff';                    % K - L 逆变换
```

程序运行结果如图 5-11 所示。

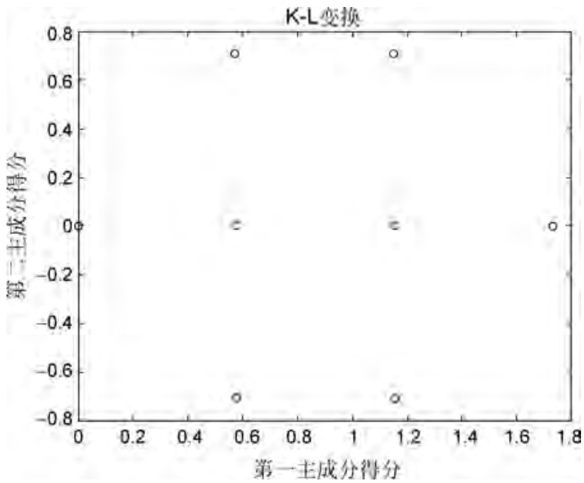


图 5-11 K-L 变换第一、二主成分

5.3.2 图像的 K-L 变换及其实现

图像的 K-L 变换是指将二维的图像转化为一维的向量，通常采用奇异值分解进行 K-L 变换。

1. 原理

图像 K-L 变换的原理是将二维图像采用行堆叠或列堆叠转换为一维处理。设一幅大小为 $M\times N$ 的图像 $f(x,y)$ 在某个传输通道上传输了 L 次，由于受到各种因素的随机干扰，接收的图像是一个图像集合，即

$$\{f_1(x,y),f_2(x,y),\cdots,f_L(x,y)\}$$

采用列堆叠将每一个 $M\times N$ 的图像表示为 MN 维的向量，即

$$\boldsymbol{f}_i=(\boldsymbol{f}_i(0,0)\quad \boldsymbol{f}_i(0,1)\quad \cdots\quad \boldsymbol{f}_i(M-1,N-1))^{\mathrm{T}}$$

图像向量 $\boldsymbol{f}=(f_1\quad f_2\quad \cdots\quad f_L)$ 的协方差矩阵和相应变换核矩阵为

$$\boldsymbol{\Sigma}_f=E[(\boldsymbol{f}-\boldsymbol{\mu}_f)(\boldsymbol{f}-\boldsymbol{\mu}_f)^{\mathrm{T}}]\approx\frac{1}{L}\left[\sum_{i=1}^{L}f_if_i^{\mathrm{T}}\right]-\boldsymbol{\mu}_f\boldsymbol{\mu}_f^{\mathrm{T}}\tag{5-35}$$

其中，$\boldsymbol{f}-\boldsymbol{\mu}_f$ 为原始图像 $\boldsymbol{f}$ 减去平均值向量 $\boldsymbol{\mu}_f$，称为中心化图像向量；Σ_f 是 $MN\times MN$ 维的矩阵。

设 λ_i 和 $\boldsymbol{u}_i$ 为 $\boldsymbol{\Sigma}_f$ 的特征值和特征向量，且降序排列，即

$$\lambda_1>\lambda_2>\lambda_3>\lambda_4>\cdots>\lambda_{M\times N}$$

K-L 变换矩阵 $\boldsymbol{U}$ 为

$$\boldsymbol{U}=(u_1,u_2,\cdots,u_{M\times N})=\begin{pmatrix} u_{11} & u_{21} & \cdots & u_{MN1}\\ u_{12} & u_{22} & \cdots & u_{MN2}\\ \vdots & \vdots & \ddots & \vdots\\ u_{1MN} & u_{2MN} & \cdots & u_{MNMN}\end{pmatrix}$$

则二维 K-L 变换表示为：

$$\boldsymbol{F}=\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{f}-\boldsymbol{\mu}_f)\tag{5-36}$$

离散 K-L 变换向量 $\boldsymbol{F}$ 是中心化向量 $\boldsymbol{f}-\boldsymbol{\mu}_f$ 与变换核矩阵 $\boldsymbol{U}$ 相乘所得的结果。

2. 奇异值分解

如前文所述，$\boldsymbol{f}$ 向量的协方差矩阵 $\boldsymbol{\Sigma}_f$ 是 $MN\times MN$ 维的矩阵，由于图像的维数 MN 一般很高，因此直接求解 $\boldsymbol{\Sigma}_f$ 的特征值和特征向量不现实。本小节简单介绍奇异值分解(Singular Value Decomposition，SVD)的方法，其详细的数学理论可以参看矩阵论的相关资料。

1) 原理

奇异值分解将一个大矩阵分解为几个小矩阵乘积，如式(5-37)所示。

$$\boldsymbol{B}=\boldsymbol{PDQ}^{\mathrm{T}}\tag{5-37}$$

其中，$\boldsymbol{B}$ 为 $m\times n$ 的矩阵；$\boldsymbol{P}$ 为 $m\times m$ 的方阵，其列向量正交，称为左奇异向量；$\boldsymbol{D}$ 为 $m\times n$ 的矩阵，仅对角线元素不为0，对角线上的元素称为奇异值；$\boldsymbol{Q}^{\mathrm{T}}$ 为 $n\times n$ 的方阵，其列向量正交，称为右奇异向量。

式(5-37)中小矩阵的求解可以采用下列方法。

设 $\boldsymbol{R}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}$，得到一个方阵，且 $\boldsymbol{R}^{\mathrm{T}}=(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})^{\mathrm{T}}=\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}=\boldsymbol{R}$，即 $\boldsymbol{R}$ 为 n 阶厄米特矩阵，可以证明 $\boldsymbol{R}$ 的特征值均为非负值。对矩阵 $\boldsymbol{R}$ 求特征值，如式(5-38)所示。

$$(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})q_i=\lambda_i q_i \tag{5-38}$$

右奇异矩阵 $\boldsymbol{Q}$ 由 q_i 组成。

由式(5-39)可得左奇异矩阵 $\boldsymbol{P}$。

$$\begin{cases}\sigma_i=\sqrt{\lambda_i}\\ p_i=\boldsymbol{B}q_i/\boldsymbol{\sigma}_i\end{cases} \tag{5-39}$$

其中，左奇异矩阵 $\boldsymbol{P}$ 由 p_i 组成；$\boldsymbol{\sigma}$ 即矩阵 $\boldsymbol{B}$ 的奇异值，在矩阵 $\boldsymbol{D}$ 中从大到小排列，且减小很快，可以用前 r 个大的奇异值来近似描述矩阵，如式(5-40)所示。

$$\boldsymbol{B}_{m\times n}\approx\boldsymbol{P}_{m\times r}\boldsymbol{D}_{r\times r}\boldsymbol{Q}_{n\times r}^{\mathrm{T}} \tag{5-40}$$

需注意，$\boldsymbol{Q}$ 为 $n\times r$ 的矩阵，$\boldsymbol{Q}^{\mathrm{T}}$ 为 $r\times n$ 的矩阵。

2）图像K-L变换实现

将中心化图像向量 $\boldsymbol{f}-\boldsymbol{\mu}_f$ 进行奇异值分解，即 $\boldsymbol{B}=\boldsymbol{f}-\boldsymbol{\mu}_f$，用前 r 个大的奇异值来近似描述，如式(5-41)所示。

$$\boldsymbol{B}_{MN\times L}\approx\boldsymbol{P}_{MN\times r}\boldsymbol{D}_{r\times r}\boldsymbol{Q}_{L\times r}^{\mathrm{T}} \tag{5-41}$$

将式(5-41)两边同时右乘 $\boldsymbol{Q}_{L\times r}$，即 $\boldsymbol{B}_{MN\times L}\boldsymbol{Q}_{L\times r}\approx\boldsymbol{P}_{MN\times r}\boldsymbol{D}_{r\times r}\boldsymbol{Q}_{L\times r}^{\mathrm{T}}\boldsymbol{Q}_{L\times r}$。

由于 $\boldsymbol{Q}$ 为正交矩阵，所以 $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q}$ 为单位阵，所以：

$$\boldsymbol{B}_{MN\times L}\boldsymbol{Q}_{L\times r}\approx\boldsymbol{P}_{MN\times r}\boldsymbol{D}_{r\times r}=\widetilde{\boldsymbol{B}}_{MN\times r} \tag{5-42}$$

由式(5-38)求出矩阵 $\boldsymbol{Q}$，进而求出 $\widetilde{\boldsymbol{B}}_{MN\times r}$，实现列压缩。

将式(5-41)两边同时左乘 $\boldsymbol{P}_{MN\times r}^{\mathrm{T}}$，即 $\boldsymbol{P}_{MN\times r}^{\mathrm{T}}\boldsymbol{B}_{MN\times L}\approx\boldsymbol{P}_{MN\times r}^{\mathrm{T}}\boldsymbol{P}_{MN\times r}\boldsymbol{D}_{r\times r}\boldsymbol{Q}_{L\times r}^{\mathrm{T}}$。

由于 $\boldsymbol{P}$ 为正交矩阵，所以 $\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}$ 为单位阵，所以：

$$\boldsymbol{P}_{MN\times r}^{\mathrm{T}}\boldsymbol{B}_{MN\times L}\approx\boldsymbol{D}_{r\times r}\boldsymbol{Q}_{L\times r}^{\mathrm{T}}=\widetilde{\boldsymbol{B}}_{r\times L} \tag{5-43}$$

由式(5-38)和式(5-39)求出矩阵 $\boldsymbol{Q}$、$\boldsymbol{D}$、$\boldsymbol{P}$，进而求出 $\widetilde{\boldsymbol{B}}_{r\times L}$，实现行压缩。

【例5-13】 打开人脸图像，采用SVD方法对其进行K-L变换，并显示变换结果。

程序如下：

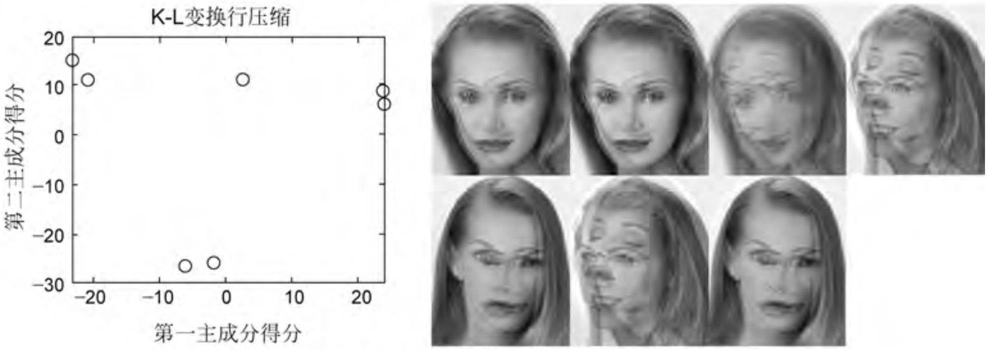
```
clear,clc,close all;
fmt = {'*.jpg','JPEG image(*.jpg)';'*.*','All Files(*.*)'};
[FileName,FilePath] = uigetfile(fmt,'导入数据','face*.jpg','MultiSelect','on');
                                        %利用uigetfile函数交互式选取训练样本图像
if ~isequal([FileName,FilePath],[0,0])
    FileFullName = strcat(FilePath,FileName);
```

```
else
    return;        % 若选择了图像,则获取图像文件的完整路径,否则退出程序,不再运行
end
N = length(FileFullName);
for k = 1:N
    Image = im2double(rgb2gray(imread(FileFullName{k})));
    X(:,k) = Image(:);                         % 把图像放在矩阵 X 的第 k 列
    InImage(:,:,:,k) = Image;
end
figure,montage(InImage,'size',[1 NaN]),title('原图');
[h,w,c] = size(Image);
averagex = mean(X')';                          % 计算图像的平均向量μ
X = X - averagex;                              % 求中心化图像向量
R = X' * X;                                    % 奇异值分解中的矩阵 R = B^T B
[Q,D] = eig(R);                                % 求矩阵 R 的特征值和特征向量
[D_sort,index] = sort(diag(D),'descend');      % 特征值从大到小排序
D = D(index,index);
Q = Q(:,index);                            % 按从大到小顺序重排特征值矩阵 D 和特征向量矩阵 Q
P = X * Q * (abs(D))^ - 0.5;               % 求左奇异矩阵 P
total = 0.0;
count = sum(D_sort);
for r = 1:N
    total = total + D_sort(r);
    if total/count > 0.95                  % 取占全部奇异值之和 95 % 的前 r 个奇异值
        break;
    end
end
KLCoefR = P' * X;
figure; plot(KLCoefR(1,:),KLCoefR(2,:),'ko'),title('K - L 变换行压缩');
xlabel('第一主成分得分');ylabel('第二主成分得分');
Y = P(:,1:2) * KLCoefR(1:2,:) + averagex;  % 基于前 2 个奇异值重建人脸图像
outImage = zeros(h,w,1,N);
for j = 1:N
    outImage(:,:,1,j) = reshape(Y(:,j),h,w);
end
figure,montage(outImage,'size',[2 NaN]),title('基于左奇异矩阵前两个奇异值重建图像');
Z = P(:,1:r) * KLCoefR(1:r,:) + averagex;          % 基于前 r 个奇异值重建人脸图像
for j = 1:N
    outImage(:,:,1,j) = reshape(Z(:,j),h,w);
end
figure,montage(outImage,'size',[1 NaN],'DisplayRange',[]);
        title('基于左奇异矩阵前 r 个奇异值重建的人脸图像');
KLCoefC = X * Q;                                   % 使用右奇异矩阵进行 K - L 变换
for j = 1:N
    outImage(:,:,1,j) = reshape(KLCoefC(:,j),h,w);
end
figure,montage(outImage,'size',[1 NaN],'DisplayRange',[]);
        title('基于右奇异矩阵的 K - L 变换');
```

程序运行结果如图 5-12 所示。

(a) 原始人脸图像

(b) 使用右奇异矩阵进行K-L变换



(c) 前两个奇异值对应左奇异向量对中心化人脸图像的降维及重建

(d) 前6个奇异值(和占总数的95%以上)对应左奇异向量对中心化人脸图像的重建

图 5-12　人脸图像 K-L 变换

3. K-L 变换函数

MATLAB 中对于 K-L 变换提供了相应的函数，列举如下。

(1) pcacov 函数：根据相关系数矩阵或协方差矩阵进行主成分分析。

① COEFF＝pcacov(V)：根据 n×n 的矩阵 V 进行主成分分析，返回主成分系数 COEFF。COEFF 为 n×n 的矩阵，每一列为一个主成分的系数向量，按主成分方差递减顺序排列，即原理中的变换矩阵 U。

② [COEFF,LATENT]=pcacov(V)：LATENT 是由主成分的方差构成的向量，即由 V 的 n 个特征值（从大到小）构成的向量。

③ [COEFF,LATENT,EXPLAINED]=pcacov(V)：EXPLAINED 是由 n 个主成分的百分比贡献率构成的向量。

（2）pca 函数：对原数据进行主成分分析。

① COEFF=pca(X)：返回 N×n 的数据矩阵 X 的主成分系数 COEFF，N 为样本数目，n 为样本维数，X 的每一行对应一个样本；COEFF 含义同 pcacov 函数中的参数。默认情况下，pca 函数将数据变为中心化数据，使用 SVD 算法。

② [COEFF,SCORE]=pca(X)：增加返回主成分得分 SCORE，是原理中矩阵 A 的转置。需要注意原理讲解中一般都是用列向量表示一个样本，N 个 n 维样本则是一个 n×N 的矩阵；但 pca 函数中输入的 X 为 N×n 的数据矩阵，是原理中数据矩阵的转置，SCORE 的计算采用 X×COEFF，实际是原理中矩阵 A 的转置；默认情况下，是先将 X 变为中心化数据再乘以 COEFF 计算出 SCORE，重建中心化数据则采用 SCORE×COEFF'。

③ [COEFF,SCORE,LATENT]=pca(X)：增加返回主成分方差 LATENT，即 X 的协方差矩阵的 n 个特征值构成的向量。

④ [COEFF,SCORE,LATENT,TSQUARED]=pca(X)：增加返回 X 中每个样本的 Hotelling T 平方统计量。TSQUARED 为有 N 个元素的向量，第 i 个元素是第 i 个样本与数据集的中心之间的距离。

$$T_i^2 = \sum_{j=1}^{n} \frac{\mathrm{SCORE}_{ij}^2}{\lambda_j} \quad i=1,2,\cdots,N$$

其中，SCORE_{ij}（$i=1,2,\cdots,N$；$j=1,2,\cdots,n$）为第 i 个样本的第 j 个主成分得分，λ_j 为样本协方差矩阵的 n 个降序排列的特征值。

⑤ [COEFF,SCORE,LATENT,TSQUARED,EXPLAINED]=pca(X)：增加返回各主成分的百分比贡献率组成的向量 EXPLAINED。

⑥ [COEFF,SCORE,LATENT,TSQUARED,EXPLAINED,MU]=pca(X)：当输入参数'Centered'设为 true 时，返回 X 中样本均值 MU；否则 MU 为全 0。

⑦ [⋯]=pca(⋯,'PARAM1',val1,'PARAM2',val2,⋯)：指定各参数实现主成分分析，用以控制计算、处理特殊数据类型。各参数含义如表 5-2 所示。

表 5-2 pca 函数参数

参数	含义
Algorithm	pca 函数实现主成分分析的算法，可选'svd'（奇异值分解）、'eig'（协方差矩阵特征值分解）、'als'（交替最小二乘法），默认值为'svd'
Centered	逻辑值，默认为 true，pca 函数将 X 变为中心化数据；为 false，不对数据进行中心化处理
Economy	逻辑值，默认为 true，函数仅返回结果的前一部分；为 false，则返回全部结果
NumComponents	正整数，指明需要的主成分数目 K，返回 COEFF 和 SCORE 的前 K 列

续表

参　数	含　义
Rows	X 中含有 NaN 情况时的处理方法，可取 'complete'（默认值，计算前去除值为 NaN 的样本，运算后在 SCORE 中的对应位置插入）；'pairwise'（切换 Algorithm 为 'eig'，计算协方差矩阵时用非 NaN 值取代 NaN）；'all'（确认 X 中没有数据缺失，采用所有数据进行运算，遇 NaN 终止运算）。采用 'als' 算法时本参数无效
Weights	样本权向量，元素均为正值
VariableWeights	分量权向量，元素均为正值
	配合 'als' 算法的参数略

（3）pcares 函数：重建数据，并求样本观测值矩阵中每个样本的每一个分量所对应的残差。

① RESIDUALS＝pcares(X，NDIM)：返回利用前 NDIM 个主成分重建数据时的残差。NDIM≤n；RESIDUALS 为 N×n 的矩阵，其元素为 X 中相应元素所对应的残差。

② [RESIDUALS，RECONSTRUCTED]＝pcares(X，NDIM)：增加返回用前 NDIM 个主成分的得分重建的数据 RECONSTRUCTED，是 X 的一个近似。pcares 只接受原始样本观测数据作为它的输入，不会自动对数据做标准化变换。

【例 5-14】 利用函数对不同字体数字的图像进行主成分分析并重建。

本例中对白色背景下不同字体的黑色数字的图像进行处理，设计分为以下 3 部分。

（1）生成样本。读取图像文件，反色，将数字目标变为白色；通过确定数字的外接矩形截取数字所在区域；将数字区域归一化为 16×16 的子图像；将 16×16 的数据变换为 1×256 的一维数据，生成样本。

```
clc;clear;close all;
fmt = {'*.jpg','JPEG (*.jpg)';'*.bmp','BMP (*.bmp)';'*.*','All Files(*.*)'};
[FileName,FilePath] = uigetfile(fmt,'导入外部数据','*.jpg','MultiSelect','on');
if ~isequal([FileName,FilePath],[0,0]);
    FileFullName = strcat(FilePath,FileName);
else
    return;
end                              % 选择数据并判断,同例 5-13
N = length(FileFullName);        % 选择的图像数目
Image = zeros(50);               % 存储原始图像数据,大小为 50×50
n = 16;n1 = 20;                  % 规格化图像为 n×n,显示用图像为 n1,外围增加了 2 行 2 列
BWI = zeros(n);                  % 存储二值化图像数据,大小为 16×16
training = zeros(1,n*n);         % 存储训练样本 1×256
showImage = zeros(n1,n1*N);      % 显示用图像,将各个样本小图像拼到一幅大图显示
for j = 1:N                      % 对每一幅图像进行预处理
    Image = imread(FileFullName{j});                 % 读取每一幅图像数据
    Image = 255 - Image;                             % 反色,将数字目标变为白色
    Image = im2bw(Image,0.4);                        % 二值化
    [y,x] = find(Image == 1);
    BWI = Image(min(y):max(y),min(x):max(x));        % 截取数字所在区域
    BWI = imresize(BWI,[n,n]);                       % 子区域归一化为 16×16
```

```
    showImage((n1 - n)/2 + 1:n + (n1 - n)/2,(j - 1) * n1 + 1:(j - 1) * n1 + n) = BWI;
                                                     % 当前图像拼到大图中
    training(j,:) = double(BWI(:)');                 % 转化为一维的样本
end
```

(2) 进行主成分分析。采用 pca 函数进行主成分分析,并通过绘图观察样本的第一、二主成分。

```
[coeff,score,latent,tsquare] = pca(training);        % 主成分分析
figure,plot(score(:,1),score(:,2),'ko');              % 绘制第一、二主成分得分图
title('主成分分析'),xlabel('第一主成分得分'),ylabel('第二主成分得分');
```

(3) 利用主成分重建图像。采用 pcares 函数分别基于第一、第一、二、第一、二、三主成分重建图像,并通过绘图观察重建图像与原始图像的区别。

```
[residuals1,reconstructed1] = pcares(training,1);     % 利用第一主成分重建数据
[residuals2,reconstructed2] = pcares(training,2);     % 利用第一、二主成分重建数据
[residuals3,reconstructed3] = pcares(training,3);     % 利用第一、二、三主成分重建数据
showResult1 = zeros(n1,n1 * N);
showResult2 = zeros(n1,n1 * N);
showResult3 = zeros(n1,n1 * N);
for j = 1:N
    tempI = reconstructed1(j,:);                      % 重建数据一
    tempI = reshape(tempI,n,n);                       % 转换为二维图像
    showResult1((n1 - n)/2 + 1:n + (n1 - n)/2,(j - 1) * n1 + 1:(j - 1) * n1 + n) = tempI;
    tempI = reconstructed2(j,:);
    tempI = reshape(tempI,n,n);
    showResult2((n1 - n)/2 + 1:n + (n1 - n)/2,(j - 1) * n1 + 1:(j - 1) * n1 + n) = tempI;
    tempI = reconstructed3(j,:);
    tempI = reshape(tempI,n,n);
    showResult3((n1 - n)/2 + 1:n + (n1 - n)/2,(j - 1) * n1 + 1:(j - 1) * n1 + n) = tempI;
end
figure,
subplot(221),imshow(showImage),title('原始二值数字图像');
subplot(222),imshow(showResult1),title('第一主成分重建的数字图像');
subplot(223),imshow(showResult2),title('第一、二主成分重建的数字图像');
subplot(224),imshow(showResult3),title('第一、二、三主成分重建的数字图像');
```

程序运行结果如图 5-13 所示,随着重建采用的主成分数目增多,重建图像清晰度逐渐提高。

(a) 原始二值数字图像　(b) 第一主成分重建的数字图像

(c) 第一、二主成分重建的数字图像　(d) 第一、二、三主成分重建的数字图像

图 5-13　图像主成分分析及重建

5.3.3 K-L 变换在图像处理中的应用

K-L 变换在图像处理中主要用于图像数据压缩和特征提取。

如前文所述，K-L 变换矩阵由特征向量组成，特征向量按特征值递减顺序排列。由于能量集中在特征值较大的系数中，因此丢掉特征值小的特征向量构成变换矩阵。K-L 变换的结果 F 是原图像 f 的低维投影，减少了数据量，在保留的主成分的贡献率不低于一定程度的情况下，不影响重建图像的质量。

K-L 变换常作为一种特征提取方法，从一组特征中计算出一组按重要性从大到小排列的新特征，是原有特征的线性组合，并且相互之间是不相关的，实现了数据的降维。例如，在人脸识别中，可以用 K-L 变换对人脸图像的原始空间进行转换，即构造人脸图像数据集的协方差矩阵，求出协方差矩阵的特征向量，再依据特征值的大小对这些特征向量进行排序，每一个特征向量的维数与原始图像一致，可以看作是一个图像，被称作特征脸。每一个人脸图像都可以确切地表示为一组特征脸的线性组合。

5.4 Radon 变换

图像的 Radon 变换也是一种重要的图像处理研究方法，将图像函数 $f(x,y)$ 沿其所在平面内的不同直线做线积分，即进行投影变换，可以获取图像在该方向上的突出特性，在去噪、重建、检测、复原中多有应用。本节介绍 Radon 变换的原理、实现及其应用。

5.4.1 Radon 变换的原理

如图 5-14(a)所示，直线 L 的方程可以表示为 $\rho = x\cos\theta + y\sin\theta$，其中，$\rho$ 代表坐标原点到直线 L 的距离，$\theta \in [0,\pi]$是直线法线与 x 轴的夹角。要将函数 $f(x,y)$沿直线 L 做线积分，即进行 Radon 变换，变换式表示为：

$$R(\rho,\theta) = \int_L f(x,y)\mathrm{d}s \tag{5-44}$$

采用 Delta 函数求解该线积分。Delta 函数是一个广义函数，在非零点取值为 0，而在整个定义域的积分为 1，用最简单的 Delta 函数表示，如式(5-45)所示。

$$\delta(t) = \begin{cases} 0 & t \neq 0 \\ 1 & t = 0 \end{cases} \tag{5-45}$$

对于直线 L，直线上的点(x,y)满足 $\delta(t)=1$，非直线上的点满足 $\delta(t)=0$，即

$$\delta(x\cos\theta + y\sin\theta - \rho) = \begin{cases} 0 & x\cos\theta + y\sin\theta - \rho \neq 0 \\ 1 & x\cos\theta + y\sin\theta - \rho = 0 \end{cases} \tag{5-46}$$

Radon 变换表达式可更改为：

$$R(\rho,\theta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta+y\sin\theta-\rho)\mathrm{d}x\,\mathrm{d}y \tag{5-47}$$

其中，$R(\rho,\theta)$是 $f(x,y)$的 Radon 变换，表示为 $\mathscr{R}[f(x,y)]=R(\rho,\theta)$。

给定一组(ρ,θ)，即可得出一个沿 $L_{\rho,\theta}$ 的积分值。如图 5-14(b)所示，n 条与直线 L 平行的线具有相同的 θ 角，但 ρ 不同，对每一条线都做 $f(x,y)$的线积分，有 n 条投影线，即对一幅图像在某一特定角度下的 Radon 变换会产生 n 个线积分值，构成一个 n 维的向量，称为 $f(x,y)$在角度 θ 下的投影。

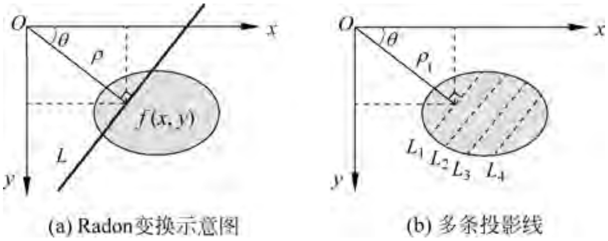


图 5-14　Radon 变换坐标系图

Radon 变换可以看成是 xy 空间向 $\rho\theta$ 空间的投影，$\rho\theta$ 空间上的每一点对应 xy 空间的一条直线。图像中高灰度值的线段会在 $\rho\theta$ 空间形成亮点，而低灰度值的线段在 $\rho\theta$ 空间形成暗点。因而，对图像中线段的检测可转化为在变换空间对亮点、暗点的检测。

二维 Radon 变换的反变换如式(5-48)所示。

$$f(x,y)=\frac{1}{2\pi^2}\int_0^{\pi}\mathrm{d}\theta\int_{-\infty}^{\infty}\frac{\partial R/\partial\rho}{x\cos\theta+y\sin\theta-\rho}\mathrm{d}\rho \tag{5-48}$$

5.4.2　Radon 变换的实现

在给定 θ 方向的情况下，数字图像 $f(x,y)$沿直线 L 的线积分，可以通过坐标系旋转后按列累加来实现。如图 5-15 所示，θ 是直线 L 法线与 x 轴的夹角，坐标系 xoy 顺时针旋转 θ 角变为 $x'oy'$，x'轴与 L 垂直，y'轴与 L 平行，将 $f(x',y')$沿 y'方向求和，实现图像在 θ 方向上的 Radon 变换。

由图 5-15 可知，坐标系旋转前后点的对应关系如式(5-49)所示。

$$\begin{cases} x'=x\cos\theta+y\sin\theta \\ y'=y\cos\theta-x\sin\theta \end{cases} \tag{5-49}$$

实际是图像的逆时针旋转。

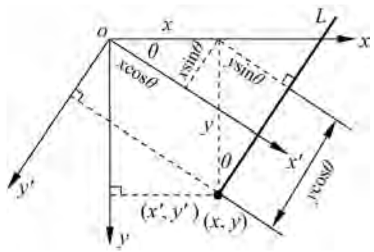


图 5-15　坐标系旋转示意图

因此，图像的 Radon 变换可以按下列步骤实现。

(1) 计算图像对角线的长度，即是 ρ 的取值范围；

(2) 设定旋转方向 $\theta\in[0,\pi]$；

(3) 将图像中的点按式(5-49)变为新坐标系中的点(旋转变换)；

(4) 将 $f(x',y')$沿 y'方向求和，即新图像的垂直投影。

【例 5-15】 对一幅图像进行指定方向上的 Radon 变换，并显示变换结果。

程序如下：

```
clc;clear;close all;
Image = rgb2gray(imread('block.bmp'));
[N,M] = size(Image);
len = floor(sqrt(N^2 + M^2) + 1);          %对角线长度,ρ的取值范围
x = - len/2:len/2 - 1;                     %投影后向量的横坐标范围,用于绘图
R = zeros(1,len);                          %初始化投影向量
free = floor((len - M + 1)/2);             %将投影值置于向量中间,求出前后空置的元素数目
R(free + 1:free + M) = sum(Image,1);       %原图垂直投影,即 0°方向上的 Radon 变换
    %以下为 45°方向上的 Radon 变换
theta1 = 45;
Image1 = imrotate(Image,theta1);
[N1,M1] = size(Image1);
free1 = floor((len - M1 + 1)/2);
R1 = zeros(1,len);
R1(free1 + 1:free1 + M1) = sum(Image1,1);
    %以下为 70°方向上的 Radon 变换
theta2 = 70;
Image2 = imrotate(Image,theta2);
[N2,M2] = size(Image2);
free2 = floor((len - M2 + 1)/2);
R2 = zeros(1,len);
R2(free2 + 1:free2 + M2) = sum(Image2,1);
subplot(231),imshow(Image),title('原图');
subplot(232),imshow(Image1),title('旋转 45°');
subplot(233),imshow(Image2),title('旋转 70°');
subplot(234),plot(x,R),title('0°方向上的 Radon 变换');
subplot(235),plot(x,R1),title('45°方向上的 Radon 变换');
subplot(236),plot(x,R2),title('70°方向上的 Radon 变换');
```

程序运行结果如图 5-16 所示。

MATLAB 提供了如下实现 Radon 变换和反变换的函数。

(1) R=radon(I,THETA)：对灰度图像 I 进行 Radon 变换。THETA 为投影夹角，若为一标量，则 R 为一列向量；若 THETA 为一向量，则 R 为一矩阵，其每一列对应 THETA 中一个角度的 Radon 变换值，默认情况下 THETA 取值为 0:179。

(2) [R,Xp]=radon(…)：向量 Xp 是 R 每一行对应的径向坐标，以图像的中心像素为原点。

(3) I=iradon(R,THETA)：利用二维矩阵 R 中的投影数据重建图像 I。THETA 指定投影的角度，若为向量，其中的元素为等间距角度；若为一个标量 D_theta，投影进行的角度为 m×D_theta(m=0,1,2,…,size(R,2)-1)；若输入为空矩阵，D_theta 默认为 180/size(R,2)。

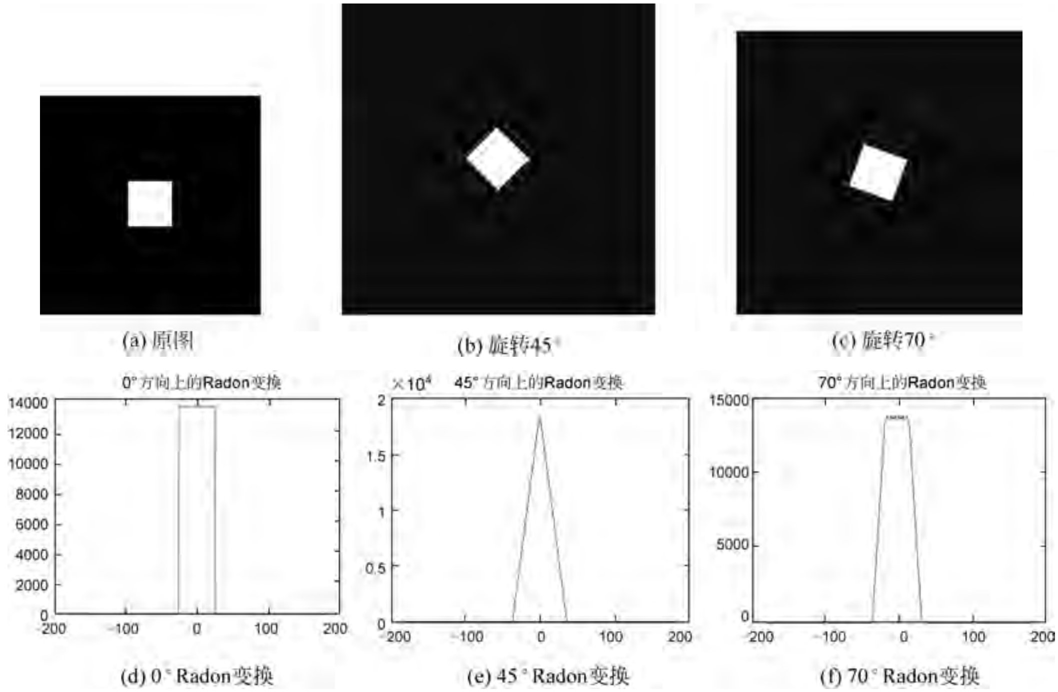


图 5-16　Radon 变换的实现

(4) I＝iradon(R,THETA,INTERPOLATION,FILTER,FREQUENCY_SCALING,OUTPUT_SIZE)：指定参数实现 Radon 反变换，参数如表 5-3 所示。

表 5-3　iradon 函数参数

参　　数	含　　义
INTERPOLATION	插值运算方法，可取'nearest'、'linear'、'spline'、'pchip'，默认为'linear'
FILTER	指定在频率域滤波的滤波器，可取'Ram-Lak'、'Shepp-Logan'、'Cosine'、'Hamming'、'Hann'、'none'，默认为'Ram-Lak'
FREQUENCY_SCALING	(0,1]之间的数，用于修正滤波器，默认为 1，若小于 1，滤波器被修正为适于频域范围[0,FREQUENCY_SCALING]，大于 FREQUENCY_SCALING 的频率被置为 0
OUTPUT_SIZE	指定重建图像的尺寸，若未指定，重建图像尺寸为 2×floor(size(R,1)/(2×sqrt(2)))

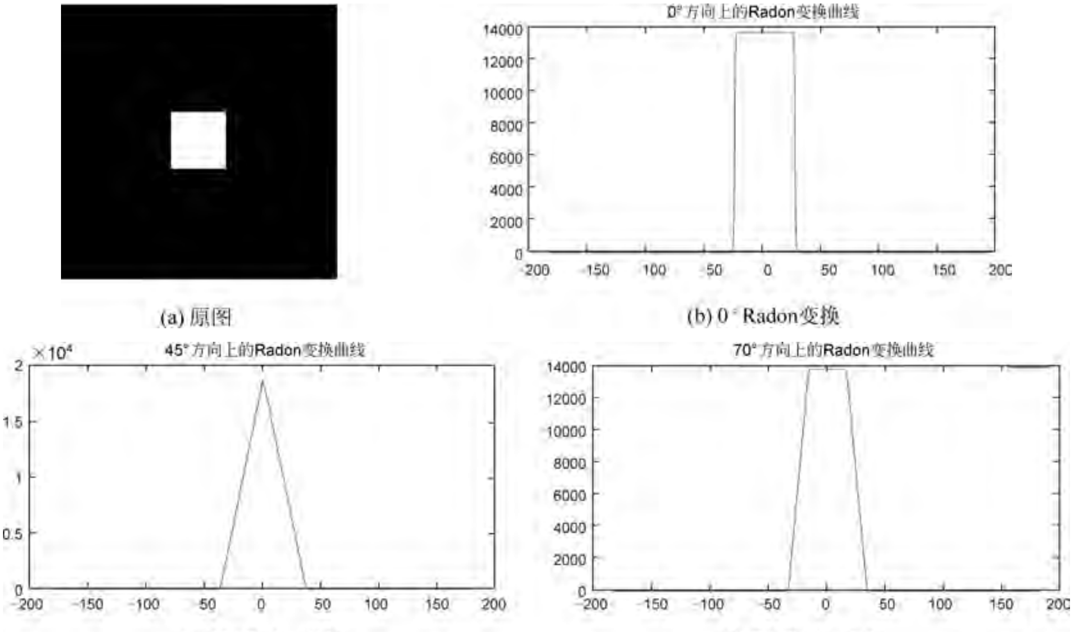
【例 5-16】　采用 radon 函数对图像进行 Radon 变换，并显示变换结果。

程序如下：

```
clear,clc,close all;
Image = rgb2gray(imread('block.bmp'));
[R1,X1] = radon(Image,0);
[R2,X2] = radon(Image,45);
```

```
[R3,X3] = radon(Image,70);
subplot(221),imshow(Image),title('原图');
subplot(222),plot(X1,R1),title('0°方向上的 Radon 变换曲线');
subplot(223),plot(X2,R2),title('45°方向上的 Radon 变换曲线');
subplot(224),plot(X3,R3),title('70°方向上的 Radon 变换曲线');
```

程序运行结果如图 5-17 所示。



(a) 原图　(b) 0° Radon变换　(c) 45° Radon变换　(d) 70° Radon变换

图 5-17　指定方向上的 Radon 变换

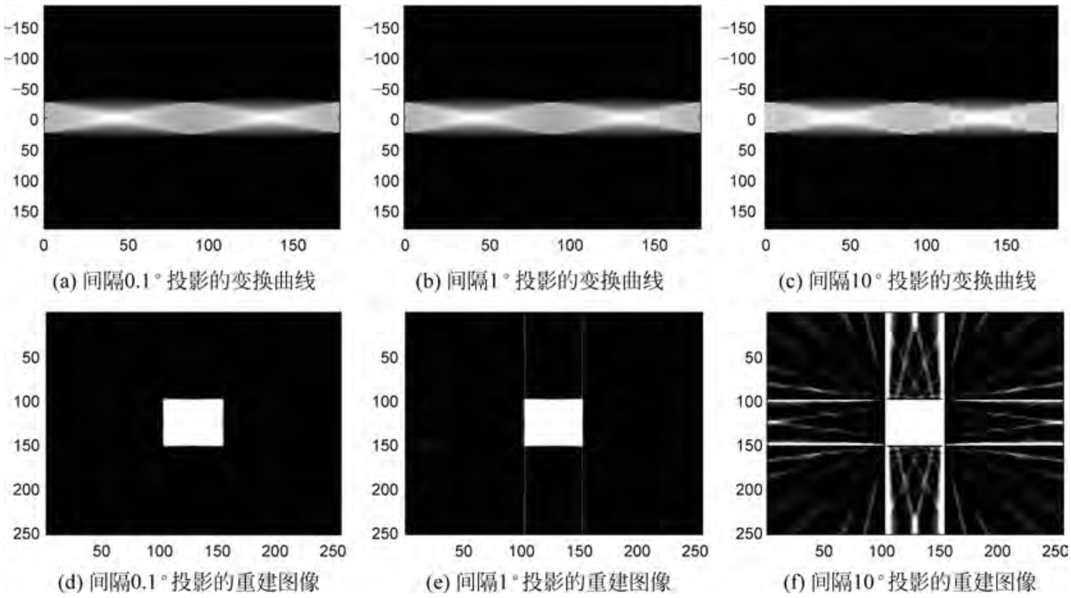
【例 5-17】　对一幅图像进行 Radon 正变换和逆变换,并显示变换结果。

程序如下:

```
clear,clc,close all;
Image = rgb2gray(imread('block.bmp'));
theta1 = 0:0.1:180;                %间隔 0.1°进行 Radon 变换
[R1,X1] = radon(Image,theta1);
result1 = iradon(R1,theta1);       %采用默认设置进行 Radon 逆变换
theta2 = 0:1:180;                  %间隔 1°进行 Radon 变换
[R2,X2] = radon(Image,theta2);
result2 = iradon(R2,theta2);
theta3 = 0:10:180;                 %间隔 10°进行 Radon 变换
[R3,X3] = radon(Image,theta3);
result3 = iradon(R3,theta3);
figure,imshow(Image),title('原图');
figure,colormap(gray);
subplot(231),imagesc(theta1,X1,R1),title('间隔 0.1°投影的 Radon 变换曲线集合');
subplot(232),imagesc(theta2,X2,R2),title('间隔 1°投影的 Radon 变换曲线集合');
```

```
subplot(233),imagesc(theta3,X3,R3),title('间隔 10°投影的 Radon 变换曲线集合');
subplot(234),image(result1),title('间隔 0.1°投影的重建图像');
subplot(235),image(result2),title('间隔 1°投影的重建图像');
subplot(236),image(result3),title('间隔 10°投影的重建图像');
```

程序运行结果如图 5-18 所示。R1、R2、R3 为二维矩阵，表示多条变换后的曲线，如图 5-18(a)～(c)所示，横轴表示 180°，纵轴表示每条曲线的高度，将 R 中的元素数值按大小转化为不同颜色，在坐标轴对应位置处以该颜色染色；投影间隔角度越小，投影后的变换曲线越多，染色越细腻。图 5-18(d)～(f)为重建图，与原图有差别，是由于 Radon 变换过程中数据损失造成的，投影间隔越小，损失越小，重建图像和原图越接近。



(a) 间隔0.1°投影的变换曲线　(b) 间隔1°投影的变换曲线　(c) 间隔10°投影的变换曲线

(d) 间隔0.1°投影的重建图像　(e) 间隔1°投影的重建图像　(f) 间隔10°投影的重建图像

图 5-18　Radon 变换和逆变换

5.4.3　Radon 变换的应用

Radon 变换可用来检测图像中的线段。将原来的 xy 平面内的点映射到 $\rho\theta$ 平面上，原 xy 平面一条线段上所有的点都将投影到 $\rho\theta$ 平面上的同一点。记录 $\rho\theta$ 平面上点的累积程度，累积程度足够的点所对应的 $\rho\theta$ 值即是 xy 平面上线段的参数。Radon 变换与第 9 章要讲的 Hough 变换检测线段的原理一样，可用于需要进行线检测的相关应用中，如线轨迹检测、滤波、倾斜校正等。

采用 Radon 变换计算出原图中各方向上的投影值，可以作为方向特征用于目标检测和识别，如应用于掌纹、静脉识别。

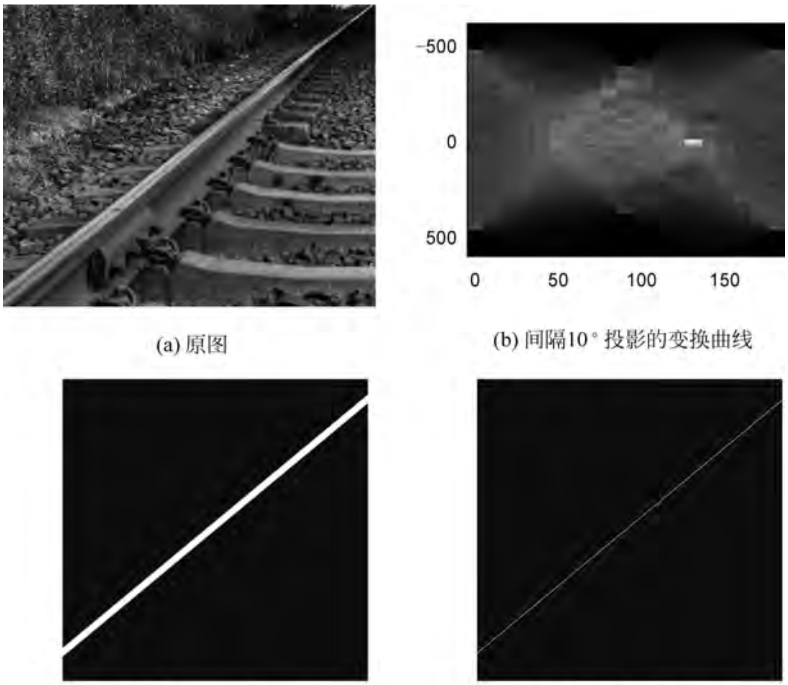
Radon 变换改变图像的表现形式，为相关处理提供便利，如图像复原中在 Radon 域用高阶统计量估计点扩散函数，提高了算法的运算速度。

【例 5-18】 对一幅图像进行 Radon 正变换，检测其中的线段。

程序如下：

```
clear,clc,close all;
Image = rgb2gray(imread('rail.jpg'));
theta = 0:10:180;
[R1,X1] = radon(Image,theta);                % Radon 变换
[N,M] = size(R1);                            % 矩阵 R1 的尺寸,M 为 theta 中元素的个数
R2 = reshape(R1,1,N * M);                    % 变换 R1 的形式,为找 M 个最大值做准备
sortR = sort(R2,'descend');                  % 降序排列
R2 = R1;
R2(R1 < sortR(M)) = 0;                       % 只保留最大的 M 个值,其余置为 0
R3 = R1;
R3(R1 < max(R1(:))) = 0;                     % 只保留最大的一个值,其余置为 0
result1 = iradon(R2,theta);
result2 = iradon(R3,theta);                  % 两种情况下的 Radon 变换
subplot(221),imshow(Image),title('原图');
colormap(gray);
subplot(222),imagesc(theta,X1,R1),title('间隔 10°投影的 Radon 变换曲线');
subplot(223),image(result1),title('根据 theta 个数保留最大值对应线段');
subplot(224),image(result2),title('仅保留 R 最大值对应线段');
```

程序运行结果如图 5-19、图 5-20 所示。



(a) 原图　　(b) 间隔10°投影的变换曲线

(c) 根据theta个数保留最大值对应线段　　(d) 仅保留R最大值对应线段

图 5-19　Radon 变换检测线段示例一

(a) 原图　(b) 根据theta个数保留最大值对应线段　(c) 仅保留R最大值对应线段

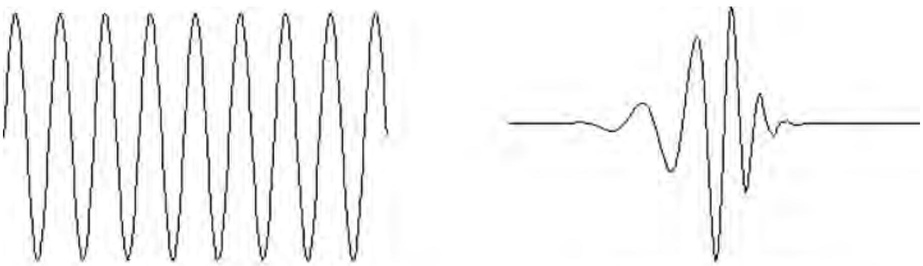
图 5-20　Radon 变换检测线段示例二

5.5　小波变换

作为重要的数学工具，小波变换被应用到数字图像处理的多个方面，如图像平滑、边缘检测、图像分析及压缩编码等。本节介绍小波变换的基本原理、特性及其在图像处理中的应用。

5.5.1　小波

波(Wave)被定义为时间或空间的一个振荡函数，如一条正弦曲线。小波(Wavelet)是“小的波”，具有在时间上集中能量的能力，是分析瞬变的、非平稳的或时变现象的工具。如图 5-21 所示，正弦曲线在$-\infty \leqslant t \leqslant \infty$上等振幅振荡，具有无限能量，而小波具有围绕一点集结的有限能量。



(a) 正弦波　(b) Daubechies小波db8

图 5-21　波和小波

1. 定义

定义 1：设函数 $\psi(t)$满足下列条件：

$$\int_R \psi(t)\mathrm{d}t = 0 \tag{5-50}$$

对其进行平移和伸缩产生如下函数族 $\psi_{a,b}(t)$。

$$\psi_{a,b}(t)=\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)\quad a,b\in R,a\neq 0 \tag{5-51}$$

$\psi(t)$称为基小波或母小波，a 称为伸缩因子(尺度因子)，b 为平移因子，$\psi_{a,b}(t)$称为 $\psi(t)$生成的连续小波。由傅里叶变换性质可得：

$$\Psi_{a,b}(\omega)=\sqrt{a}\Psi(a\omega)\mathrm{e}^{-\mathrm{j}\omega b} \tag{5-52}$$

定义 2：若函数 $\psi(t)$的傅里叶变换 $\Psi(\omega)$满足：

$$C_{\psi}=\int_{R}\frac{|\Psi(\omega)|^{2}}{|\omega|}\mathrm{d}\omega<\infty \tag{5-53}$$

则称 $\psi(t)$为允许小波，式(5-53)称为允许性条件。其中 $\Psi(\omega)=\int_{R}\psi(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$。

因 $\Psi(\omega)|_{\omega=0}=\int_{R}\psi(t)\mathrm{d}t=0$，故允许小波一定是基小波。

2. 特点

1) 紧支撑性

小波函数 $\psi(t)$满足式(5-50)，即均值为 0，$\psi(t)$应具有振荡性，即在图形上具有“波”的形状。$\psi(t)$满足 $\int_{R}|\psi(t)|\mathrm{d}t<\infty$、$\int_{R}|\psi(t)|^{2}\mathrm{d}t<\infty$，因此，$\psi(t)$仅在小范围内波动，且能量有限，即小波函数 $\psi(t)$的定义域是紧支撑的，超出一定范围时，波动幅度迅速衰减，具有速降性。

2) 变化性

小波函数 $\psi_{a,b}(t)$及它的频谱 $\Psi_{a,b}(\omega)$随尺度因子 a 的变化而变化。由式(5-51)可知，随着 a 的减小，$\psi_{a,b}(t)$的支撑区随之变窄，其幅值变大，如图 5-22 所示。

由傅里叶变换的尺度变换性质：若 $\mathscr{F}[f(t)]=F(\omega)$，则 $\mathscr{F}[f(\alpha t)]=F(\omega/\alpha)/|\alpha|$，可知，$\Psi_{a,b}(\omega)$随着 a 的减小而向高频端展宽。

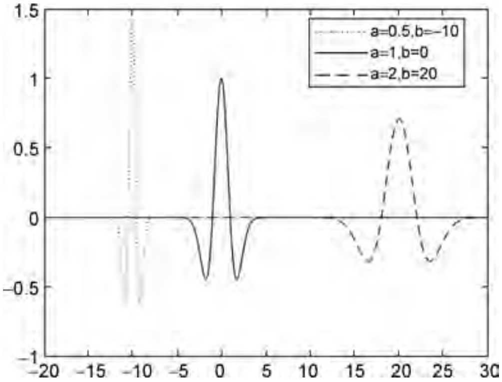


图 5-22　Marr 小波参数 a，b 取不同值的波形

3）消失矩

若小波 $\psi(t)$ 满足式(5-54)，则称该小波具有 K 阶消失矩。

$$\int_R t^k \psi(t)\mathrm{d}t = 0 \quad k = 0,1,\cdots,K-1 \tag{5-54}$$

这时，$\Psi(\omega)$ 在 $\omega=0$ 处 K 次可微，即 $\Psi^k(0)=0, k=1,2,\cdots,K$，小波 $\psi(t)$ 随着 K 的增加，波形振荡越来越强烈。

3．一维小波实例

1）Haar 小波

Haar 小波是最简单的小波，其表达式为：

$$\begin{cases} \psi_H(t) = \begin{cases} 1 & 0 \leqslant t < 1/2 \\ -1 & 1/2 \leqslant t < 1 \\ 0 & \text{其他} \end{cases} \\ \Psi_H(\omega) = \dfrac{1 - 2\mathrm{e}^{-\frac{\mathrm{i}\omega}{2}} + \mathrm{e}^{-\mathrm{i}\omega}}{\omega \mathrm{i}} \end{cases} \tag{5-55}$$

Haar 小波具有紧支性(长度为 1)和对称性，消失矩为 1，即 $\int_R \psi_H(t)\mathrm{d}t = 0$。 Haar 小波不是连续函数，应用有限，但结构简单，一般作为原理示意或说明。

2）Morlet 小波

Morlet 小波是用高斯函数构造的一种小波，其时域、频域可表示为：

$$\begin{cases} \psi(t) = \pi^{-1/4}(\mathrm{e}^{-\mathrm{i}\omega_0 t} - \mathrm{e}^{-\omega_0^2/2})\mathrm{e}^{-t^2/2} \\ \Psi(\omega) = \pi^{-1/4}[\mathrm{e}^{-(\omega-\omega_0)^2/2} - \mathrm{e}^{-\omega_0^2/2}\mathrm{e}^{-\omega^2/2}] \end{cases} \tag{5-56}$$

由上式可以看出，Morlet 小波满足允许条件，即 $\Psi(0)=0$。

当 $\omega_0 \geqslant 5$ 时，$\mathrm{e}^{-\omega_0^2/2} \approx 0$，所以，式(5-56)的第 2 项可以忽略，Morlet 小波可以近似表示为：

$$\begin{cases} \psi(t) = \pi^{-1/4}\mathrm{e}^{-\mathrm{i}\omega_0 t}\mathrm{e}^{-t^2/2} \\ \Psi(\omega) = \pi^{-1/4}\mathrm{e}^{-(\omega-\omega_0)^2/2} \end{cases} \tag{5-57}$$

Morlet 小波在时域、频域都具有较好的局部性，是很常用的小波。

3）Mexican 草帽小波

Mexican 草帽小波成比例于高斯函数二阶导数，也称为 Marr 小波，其表达式为：

$$\psi(t) = \left(\frac{2}{\sqrt{3}}\pi^{-1/4}\right)(1-t^2)\mathrm{e}^{-t^2/2} \tag{5-58}$$

Mexican 草帽小波具有对称性，支撑区间是无限的，有效支撑区间为[－5 5]，在视觉信息处理方面获得了很多应用。

MATLAB 提供了相应函数生成一维小波，其调用格式如下。

(1) [PSI,X]=morlet(LB,UB,N)：返回 Morlet 小波在区间[LB UB]内 N 个规则网格点的值 PSI。小波有效紧支集为[−4 4]，函数返回的 PSI 是按照 $\cos(5t)e^{-t^2/2}$ 计算的，即式(5-57)中 $\omega_0=5$，只取了实部，且未考虑系数 $\pi^{-1/4}$。

(2) [PSI,X]=mexihat(LB,UB,N)：返回 Mexican 草帽小波在区间[LB UB]内 N 个规则网格点的值 PSI。小波有效紧支集为[−5 5]。

【例 5-19】 绘制 Haar 小波、Morlet 小波及 Marr 小波的函数图形及其频谱图。

程序如下：

```
clear,clc,close all;
t = 0:0.01:1;
N = length(t);
Hpsi(1:N/2) = 1;  Hpsi(N/2:N) = - 1;                % Haar 小波
HDFT = abs(fftshift(fft(Hpsi))) * 2/N;
x = linspace(0,1,N);
subplot(121),plot(x,Hpsi,'k',[0,1.5],[0,0],':k',[1,1],[ - 1,0],':k'),…
            ylim([ - 2 2]),title('Haar 小波');
subplot(122),plot(x,HDFT,'k'),title('Haar 小波频谱');
t = - 4:0.01:4;
omega0 = 5;
Realpsi = pi^( - 0.25) * (cos(omega0 * t) - exp( - omega0^2/2)). * exp( - t.^2/2); % Morlet 小波实部
Impsi = pi^( - 0.25) * ( - sin(omega0 * t) - exp( - omega0^2/2)). * exp( - t.^2/2); % Morlet 小波虚部
figure;
subplot(121),plot(t,Realpsi,'k',t,Impsi,'k:'),title('Morlet 小波');
omega = 0:0.01:10;
Fm = pi^( - 0.25) * (exp( - (omega - omega0).^2/2) - exp( - omega0^2/2 - omega.^2/2));
subplot(122),plot(omega,Fm,'k'),title('Morlet 小波频谱');
[Mpsi,t] = mexihat( - 5,5,256);                                  % Mexican 小波
figure;subplot(121),plot(t,Mpsi,'k'),title('Mexican 小波');
MDFT = abs(fftshift(fft(Mpsi)));
x = linspace( - 5,5,length(MDFT));
subplot(122),plot(x,MDFT,'k'),title('Mexican 小波频谱');
```

程序运行结果如图 5-23 所示。

5.5.2　一维小波变换

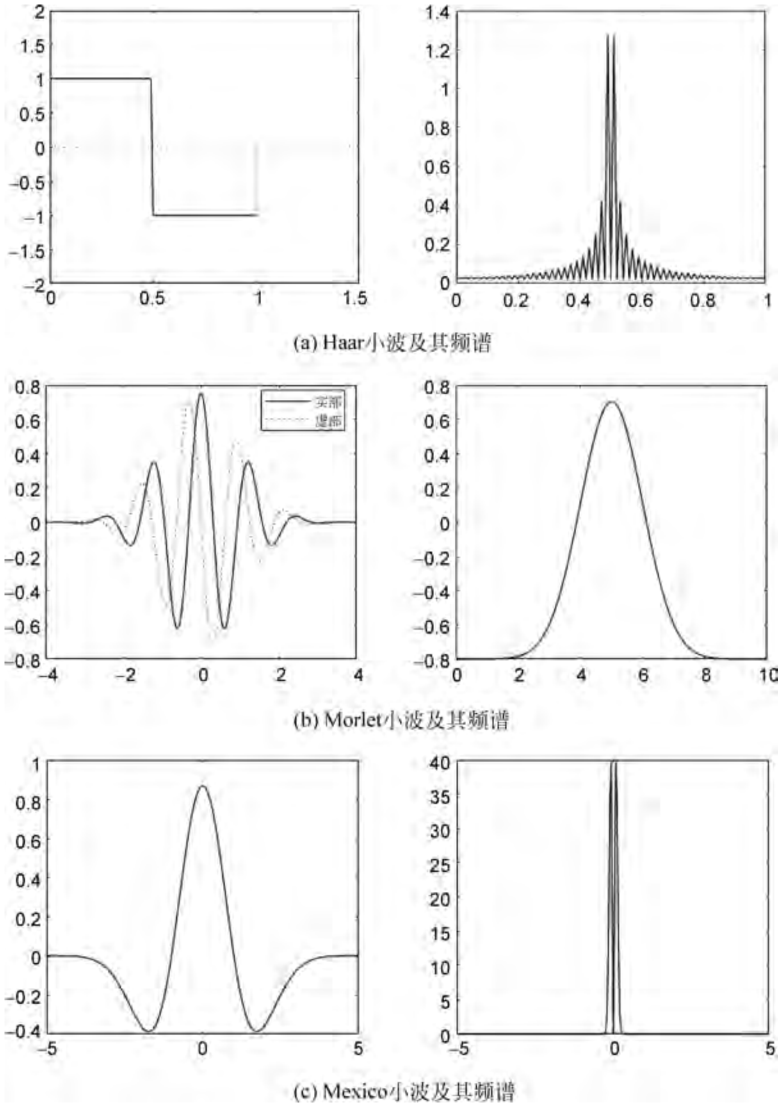
将 $\psi_{a,b}(t)$ 中的参数 a,b 离散化，取 $a=a_0^j, j=0,\pm1,\pm2,\cdots, b=ka_0^j, j,k\in Z$，则离散化后的小波函数为：

$$\psi_{j,k}(t)=a_0^{-j/2}\psi(a_0^{-j}t-k),\quad j,k\in Z \tag{5-59}$$

用 $\psi_{j,k}(t)$ 将函数 $f(t)$ 展开，即

$$f(t)=\sum_j\sum_k \alpha_{j,k}\psi_{j,k}(t) \tag{5-60}$$

展开系数 $\alpha_{j,k}$ 的集合称为 $f(t)$ 的离散小波变换(Discrete Wavelet Transform，DWT)，如



(a) Haar小波及其频谱

(b) Morlet小波及其频谱

(c) Mexico小波及其频谱

图 5-23　3 种常见小波及其频谱

式(5-61)所示。

$$\alpha_{i,k}=\langle f(t),\psi_{i,k}(t)\rangle=\int_R f(t)\psi_{j,k}^*(t)\mathrm{d}t \tag{5-61}$$

1. 正交小波变换与多分辨分析

定义 1：设平方可积函数空间 $L^2(R)$中的函数 $\psi(t)$是一个允许小波，若其二进伸缩平移系($a_0=2$)构成 $L^2(R)$的标准正交基，则称 $\psi(t)$为正交小波，称 $\psi_{j,k}(t)$是正交小波函数，称相应的离散小波变换 $\alpha_{j,k}=\langle f(t),\psi_{j,k}(t)\rangle$为正交小波变换，如式(5-62)所示。

$$\psi_{j,k}(t)=2^{-j/2}\psi(2^{-j}t-k),\quad j,k\in Z \tag{5-62}$$

例如，Haar 小波的二进伸缩平移系如下：

$$\psi_{j,k}(t)=2^{-j/2}\psi(2^{-j}t-k)=\begin{cases}2^{-j/2} & 2^{j}k\leqslant t<(2k+1)2^{j-1}\\ -2^{-j/2} & (2k+1)2^{j-1}\leqslant t\leqslant(k+1)2^{j}\\ 0 & 其他\end{cases} \tag{5-63}$$

可以验证构成 $L^2(R)$的一个标准正交基。

多分辨分析（Multi-Resolution Analysis，MRA）也称多尺度分析，是构造正交小波基的一般方法。

定义 2：若 $L^2(R)$中一个子空间序列$\{V_j\}_{j\in Z}$及一个函数$\varphi(t)$满足以下条件，则称其为一个正交多分辨分析。

（1）$V_j\subseteq V_{j-1}, j\in Z$；

（2）$f(t)\in V_j\Leftrightarrow f(2t)\in V_{j-1}$；

（3）$\bigcap\limits_{j\in Z}V_j=\{0\}, \bigcup\limits_{j\in Z}V_j=L^2(R)$；

（4）$\varphi(t)\in V_0$，且$\{\varphi(t-k)\}_{k\in Z}$是 V_0 的标准正交基，称 $\varphi(t)$是此多分辨分析的尺度函数。

2. 尺度函数与小波函数

由多分辨分析性质(2)、(4)可知，对于任何 $\varphi(t)\in V_0$，有 $\varphi(2^{-j}t)\in V_j$；$\{\varphi(t-k)\}_{k\in Z}$是 V_0 的标准正交基，函数系$\{2^{-j/2}\varphi(2^{-j}t-k)\}_{k\in Z}$则构成了 V_j 的一组标准正交基；即$\{\varphi(t-k)\}_{k\in Z}$张成 $L^2(R)$的子空间 V_0，$\{2^{-j/2}\varphi(2^{-j}t-k)\}_{k\in Z}$张成了 V_j。

$\varphi(t)\in V_0$，而 $V_0\subseteq V_{-1}$，因此，$\varphi(t)\in V_{-1}$。因函数系$\{2^{1/2}\varphi(2t-k)\}_{k\in Z}$构成了 V_{-1} 的一组标准正交基，因此，$\varphi(t)$可以借助于$\{2^{1/2}\varphi(2t-k)\}_{k\in Z}$的加权和表示。

$$\varphi(t)=\sum_{k\in Z}h_k\sqrt{2}\,\varphi(2t-k) \tag{5-64}$$

其中，系数$\{h_k\}_{k\in Z}$称为尺度函数（尺度滤波器）系数，满足

$$\begin{cases}h_k=\dfrac{1}{\sqrt{2}}\displaystyle\int_R\varphi(t)\varphi^*(2t-k)\mathrm{d}t\\ H(\omega)=\dfrac{1}{\sqrt{2}}\displaystyle\sum_k h_k\mathrm{e}^{-\mathrm{i}k\omega}\end{cases} \tag{5-65}$$

式(5-64)称为双尺度方程，其频域形式为：

$$\Phi(2\omega)=H(\omega)\Phi(\omega) \tag{5-66}$$

定义函数 $\psi_{j,k}(t)$，张成尺度函数在不同尺度下张成的空间之间的差空间为$\{W_j\}_{j\in Z}$，称 W_j 为尺度为 j 的小波空间，V_j 为尺度为 j 的尺度空间。

由于 $V_j\subseteq V_{j-1}$，即 $V_{j-1}=V_j+W_j$，且 $W_j\perp V_j$，$j\in Z$，显然，当 $m,n\in Z$，$m\neq n$ 时，有 $W_m\perp W_n$，如式(5-67)所示。

$$V_{j-1}=V_j+W_j=V_{j+1}+W_{j+1}+W_j=\cdots$$

$$=V_{j+s}+W_{j+s}+W_{j+s-1}+\cdots+W_{j+1}+W_j \tag{5-67}$$

令 $s\to+\infty$，则 $V_{j-1}=\bigoplus\limits_{m=j}^{+\infty}W_m$

令 $j\to-\infty$，则 $L^2(R)=\bigoplus\limits_{m=-\infty}^{+\infty}W_m$

因 $W_0\subset V_{-1}$，张成 W_0 的小波函数 $\psi(t)$ 可以由 V_{-1} 的标准正交基 $\{2^{1/2}\varphi(2t-k)\}_{k\in Z}$ 表示。

$$\psi(t)=\sum_{k\in Z}g_k\sqrt{2}\varphi(2t-k) \tag{5-68}$$

式(5-68)也称为双尺度方程，其频域表示为：

$$\Psi(2\omega)=G(\omega)\Phi(\omega) \tag{5-69}$$

$\{g_k\}_{k\in Z}$ 满足：

$$\begin{cases} g_k=\dfrac{1}{\sqrt{2}}\displaystyle\int_R\psi(t)\varphi^*(2t-k)\mathrm{d}t \\ G(\omega)=\dfrac{1}{\sqrt{2}}\displaystyle\sum_k g_k\mathrm{e}^{-\mathrm{i}k\omega} \end{cases} \tag{5-70}$$

式(5-64)和式(5-68)这两个双尺度方程是多分辨分析赋予尺度函数 $\varphi(t)$ 和小波函数 $\psi(t)$ 的最基本性质。

综上所述，多分辨分析的基本思想其实就是：为有效地寻找空间 $L^2(R)$ 的基底，从 $L^2(R)$ 的某个子空间出发，在这个子空间中建立基底，然后利用简单的变换把该基底扩充到 $L^2(R)$ 中去。

3. 常见小波函数的尺度函数计算

MATLAB 提供了 wavefun 函数，用于计算小波函数的近似值及对应的尺度函数；提供了 waveinfo 函数查看小波的相关信息，其部分调用如下。

(1) [PHI,PSI,XVAL]＝wavefun('wname',ITER)：对正交小波、Meyer 小波适用，返回尺度函数 PHI 和小波函数 PSI 的取值；ITER 为迭代次数；XVAL 为取值点，支撑区间内有 2^{ITER} 个点。

(2) [PSI,XVAL]＝wavefun('wname',ITER)：对没有尺度函数的小波适用，如 Morlet 小波、Mexican 草帽小波、高斯导数小波或复小波。输出 PSI 为实数或复数向量。

(3) …＝wavefun(…,'plot')：计算并绘制函数。

(4) waveinfo('wname')：提供 wname 指定的小波的信息，包括 haar、db、sym、coif、bior、rbio、meyr、dmey、gaus、mexh、morl、cgau、cmor、shan、fbsp、fk 等。

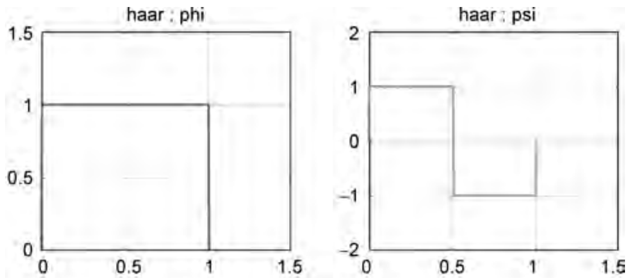
【例 5-20】 使用 wavefun 函数绘制 Haar 小波、Morlet 小波、Meyer 小波、Daubechies (db*N*) 小波的尺度函数及小波函数图形。

程序如下：

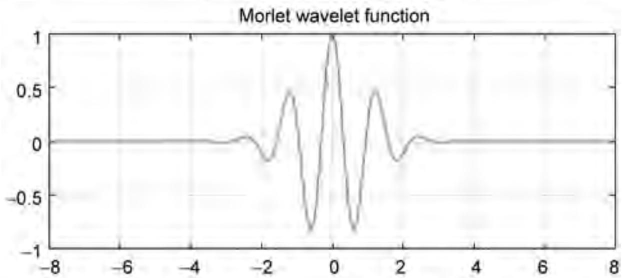
```
clear,clc,close all;
figure,[phi1,psi1,xval1] = wavefun('haar',8,'plot');
```

```
figure,[psi2,xval2] = wavefun('morl',8,'plot');
figure,[phi3,psi3,xval3] = wavefun('meyr',8,'plot');
figure,[phi4,psi4,xval4] = wavefun('db4',8,'plot');
```

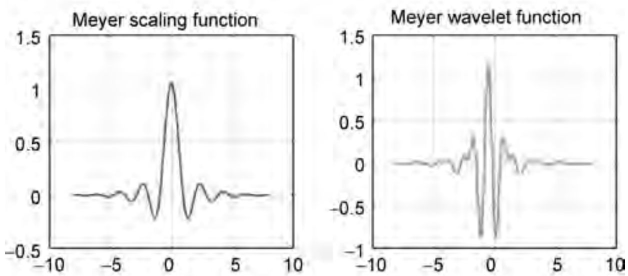
程序运行结果如图 5-24 所示。



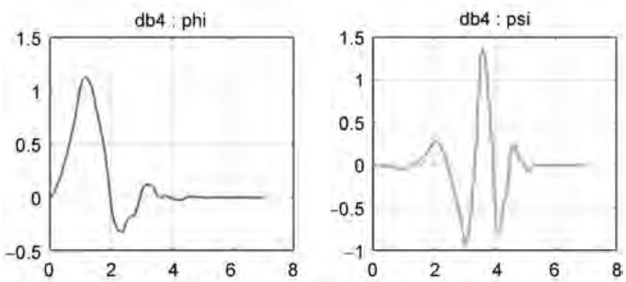
(a) Haar小波尺度函数和小波函数



(b) Morlet小波函数



(c) Meyer小波尺度函数和小波函数



(d) db4小波尺度函数和小波函数

图 5-24　常见小波的尺度函数和小波函数

4. 函数的正交小波分解

由以上讨论可知，给定一个多分辨分析($\{V_k\}_{k\in Z}$，$\varphi(t)$)，可确定一个小波函数 $\psi(t)$和其伸缩系$\{\psi_{j,k}(t)=2^{-j/2}\psi(2^{-j}t-k)\}_{j,k\in Z}$，并张成小波空间$\{W_j\}_{j\in Z}$，因 $W_i\perp W_j(i\neq j)$，且 $L^2(R)=\bigoplus\limits_{j\in Z}W_j$，所以$\{\psi_{j,k}(t)=2^{-j/2}\psi(2^{-j}t-k)\}_{j,k\in Z}$ 构成 $L^2(R)$的标准正交基。因此，对任何 $f(t)\in L^2(R)$，有：

$$f(t)=\sum_{j,k}d_{j,k}\psi_{j,k}(t) \tag{5-71}$$

其中，$d_{j,k}=\langle f(t),\psi_{j,k}(t)\rangle_{j,k\in Z}$ 是 $f(t)$的离散小波变换，且是正交小波变换，式(5-71)是 $f(t)$的重构公式，也称为 $f(t)$的正交小波分解。

由多分辨分析性质(2)，即 $f(t)\in V_j\Leftrightarrow f(2t)\in V_{j-1}$ 可知：V_j 的频率范围是 V_{j-1} 的一半，且是 V_{j-1} 中的低频表现部分，而 $V_{j-1}=V_j+W_j$，所以，W_j 的频率表现在 V_j 与 V_{j-1} 之间的部分，而且 W_j 的频带互不重叠，因此，通常认为 V_j 表现了 V_{j-1} 的“概貌”，W_j 表现了 V_{j-1} 的不同频带中的“细节”。记 $d_{j,k}\psi_{j,k}(t)=w_j(t)$，则 $w_j(t)\in W_j$，式(5-71)可写成：

$$f(t)=\sum_{j}w_j(t) \tag{5-72}$$

式(5-72)说明任何一个函数 $f(t)\in L^2(R)$可以分解成不同频带的细节之和。

实际情况中，函数 $f(t)\in L^2(R)$仅有有限的细节。由式(5-67)得 $V_0=V_s+W_s+W_{s-1}+\cdots+W_1$。设一个函数 $f(t)\in V_0$，则 $f(t)$可分解为 $f(t)=f_s(t)+w_s(t)+w_{s-1}(t)+\cdots+w_1(t)$，即

$$f(t)=\sum_{k\in Z}c_{s,k}\varphi_{s,k}(t)+\sum_{j=1}^{s}\sum_{k\in Z}d_{j,k}\psi_{j,k}(t) \tag{5-73}$$

其中，$c_{s,k}=\langle f(t),\varphi_{s,k}(t)\rangle$，$k\in Z$，$d_{j,k}=\langle f(t),\psi_{j,k}(t)\rangle$，$k\in Z$。称式中第 1 项为 $f(t)$的不同尺度 $s(s\geqslant 1)$下的近似式，是 $f(t)$中频率不超过 2^{-s} 的成分；称第 2 项中的 $\sum\limits_{k\in Z}d_{j,k}\psi_{j,k}(t)$ 为 $f(t)$的不同尺度 j 下的细节，是 $f(t)$中频率 2^{-j} 到 2^{-j+1} 之间的细节成分。

根据以上分析可知，当尺度函数 $\varphi(t)$、小波函数 $\psi(t)$确定后，通过计算$\{c_{s,k}\}_{k\in Z}$、$\{d_{j,k}\}_{k\in Z}$，即可得到函数 $f(t)\in L^2(R)$的近似和细节。

$$\begin{aligned}
c_{j+1,k}&=\langle f(t),\varphi_{j+1,k}(t)\rangle=\int_R f(t)\varphi_{j+1,k}^{*}(t)\mathrm{d}t=\int_R f(t)\{2^{-(j+1)/2}\varphi^{*}(2^{-(j+1)}t-k)\}\mathrm{d}t\\
&=\int_R f(t)2^{-(j+1)/2}\sum_{n\in Z}h_n^{*}\sqrt{2}\varphi^{*}[2(2^{-(j+1)}t-k)-n]\mathrm{d}t\\
&=\int_R f(t)2^{-j/2}\sum_{n\in Z}h_n^{*}\varphi^{*}[2^{-j}t-(2k+n)]\mathrm{d}t\\
&=\sum_{m\in Z}h_{m-2k}^{*}\int_R f(t)2^{-j/2}\varphi^{*}(2^{-j}t-m)\mathrm{d}t\\
&=\sum_{m\in Z}h_{m-2k}^{*}\int_R f(t)\varphi_{j,m}^{*}(t)\mathrm{d}t
\end{aligned}$$

$$= \sum_{m \in Z} h_{m-2k}^{*} \langle f(t), \varphi_{j,m}(t) \rangle$$

$$= \sum_{m \in Z} h_{m-2k}^{*} c_{j,m}$$

同理，得到 $d_{j+1,k} = \langle f(t), \psi_{j+1,k}(t) \rangle = \sum_{m \in Z} g_{m-2k}^{*} c_{j,m}$

因此，得到正交小波分解的 Mallat 快速算法，如式(5-74)所示。

$$\begin{cases} c_{j+1,k} = \sum_{n \in Z} h_{n-2k}^{*} c_{j,n} \\ d_{j+1,k} = \sum_{n \in Z} g_{n-2k}^{*} c_{j,n} \end{cases} \quad k \in Z \tag{5-74}$$

因此，只要知道双尺度方程中的传递系数 $\{h_k\}_{k \in Z}$($g_k = (-1)^k h_{1-k}^{*}$)，就可计算出一系列正交小波分解系数，过程如图 5-25 所示。

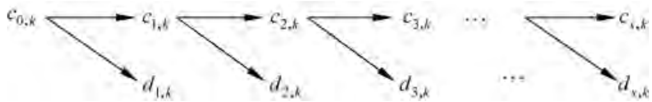


图 5-25　正交小波分解算法示意图

初始值 $c_{0,k} = \langle f(t), \varphi_{0,k}(t) \rangle = \int_R f(t) \varphi^{*}(t-k) \mathrm{d}t$，对于离散序列，$f(t) \to f_n = f(n\Delta t)$，因此，

$$c_{0,k} \approx \sum_{n} f_n \varphi(n-k) \tag{5-75}$$

根据信号处理理论，利用序列 $\{h_k\}_{k \in Z}$ 对一个离散信号 $\{x_n\}_{n \in Z} \in l^2(Z)$ 进行滤波，则

$$y_k = h_k * x_k = \sum_{n \in Z} h_{k-n} x_n \tag{5-76}$$

比较式(5-74)和式(5-76)发现，式(5-76)卷积式中 k 对所有的 n 值做卷积运算，而式(5-74)卷积式中是 $2k$ 对所有的 n 值做卷积运算，缺少了奇数($2k+1$)的部分，即卷积运算或滤波处理之后所得的序列抽去了 k 的奇数部分，只剩下偶数部分，这一过程称为再抽样，抽样率为 2。所以，分辨率 j 的近似分量 $c_{j,k}$ 分解为分辨率为 $j+1$ 的近似分量 $c_{j+1,k}$ 和细节分量 $d_{j+1,k}$ 的分解方法可以用图 5-26 所示的滤波过程来表示。

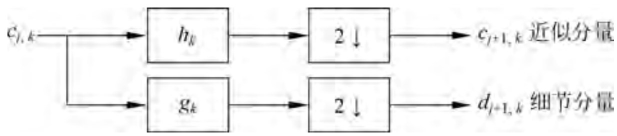


图 5-26　近似分量 $c_{j,k}$ 分解为 $c_{j+1,k}$ 和 $d_{j+1,k}$(2↓代表再抽样，抽样率为 2)

5. 函数的正交小波重构

所谓重构，即已知近似序列 $\{c_{j+1,k}\}_{k \in Z}$ 和细节序列 $\{d_{j+1,k}\}_{k \in Z}$，求出序列 $\{c_{j,k}\}_{k \in Z}$。

由正交小波分解式可知：

$$f_j(t)=\sum_{k\in Z}\langle f(t),\varphi_{j,k}(t)\rangle\varphi_{j,k}(t)=\sum_{k\in Z}c_{j,k}\varphi_{j,k}(t) \tag{5-77}$$

由于 $V_j=V_{j+1}+W_{j+1}$，所以，$f_j(t)=f_{j+1}(t)+w_{j+1}(t)$，而

$$f_{j+1}(t)=\sum_{k\in Z}\langle f(t),\varphi_{j+1,k}(t)\rangle\varphi_{j+1,k}(t)=\sum_{k\in Z}c_{j+1,k}\varphi_{j+1,k}(t)$$

$$w_{j+1}(t)=\sum_{k\in Z}\langle f(t),\psi_{j+1,k}(t)\rangle\psi_{j+1,k}(t)=\sum_{k\in Z}d_{j+1,k}\psi_{j+1,k}(t)$$

$$\begin{aligned}
f_{j+1}(t)+w_{j+1}(t)&=\sum_{k\in Z}c_{j+1,k}\varphi_{j+1,k}(t)+\sum_{k\in Z}d_{j+1,k}\psi_{j+1,k}(t)\\
&=\sum_{k\in Z}c_{j+1,k}2^{-(j+1)/2}\varphi(2^{-(j+1)}t-k)+\sum_{k\in Z}d_{j+1,k}2^{-(j+1)/2}\psi(2^{-(j+1)}t-k)\\
&=\sum_{k\in Z}c_{j+1,k}2^{-(j+1)/2}\sum_{n\in Z}h_n\sqrt{2}\varphi[2(2^{-(j+1)}t-k)-n]+\\
&\quad\sum_{k\in Z}d_{j+1,k}2^{-(j+1)/2}\sum_{n\in Z}g_n\sqrt{2}\varphi[2(2^{-(j+1)}t-k)-n]\\
&=\sum_{k\in Z}c_{j+1,k}2^{-j/2}\sum_{n\in Z}h_n\varphi[2^{-j}t-(2k+n)]+\\
&\quad\sum_{k\in Z}d_{j+1,k}2^{-j/2}\sum_{n\in Z}g_n\varphi[2^{-j}t-(2k+n)]\\
&=\sum_{k\in Z}c_{j+1,k}2^{-j/2}\sum_{m\in Z}h_{m-2k}\varphi[2^{-j}t-m]+\\
&\quad\sum_{k\in Z}d_{j+1,k}2^{-j/2}\sum_{m\in Z}g_{m-2k}\varphi[2^{-j}t-m]\\
&=\sum_{k\in Z}c_{j+1,k}\sum_{m\in Z}h_{m-2k}\varphi_{j,m}(t)+\sum_{k\in Z}d_{j+1,k}\sum_{m\in Z}g_{m-2k}\varphi_{j,m}(t)\\
&=\sum_{m\in Z}\Big(\sum_{k\in Z}c_{j+1,k}h_{m-2k}+\sum_{k\in Z}d_{j+1,k}g_{m-2k}\Big)\varphi_{j,m}(t)
\end{aligned} \tag{5-78}$$

由式(5-77)和式(5-78)可得 Mallat 小波重构算法。

$$c_{j,k}=\sum_{k\in Z}c_{j+1,k}h_{n-2k}+\sum_{k\in Z}d_{j+1,k}g_{n-2k} \tag{5-79}$$

其重构过程如图 5-27 所示。

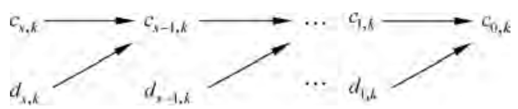


图 5-27　小波重构算法示意图

比较式(5-79)和式(5-76)发现，式(5-76)卷积式中 k 对所有的 n 值做卷积运算，而式(5-79)卷积式中是 n 对 k 的偶数序列 $2k$ 做卷积运算，从而造成 $c_{j+1,k}$、$d_{j+1,k}$ 的取值个数比 h_{n-2k}、g_{n-2k} 的取值个数多出一倍，可将 $(2k+1)$ 对应的 $c_{j+1,k}$、$d_{j+1,k}$ 当作 0 值来处理，即在两个数值之间插入一个 0，这一过程称为插值抽样，抽样率为 2。所以，分辨率 $j+1$ 的近似分量 $c_{j+1,k}$ 和细节分量 $d_{j+1,k}$ 重构分辨率 j 级近似分量 $c_{j,k}$ 的重构方法可以用图 5-28 所示的滤波过程来表示。

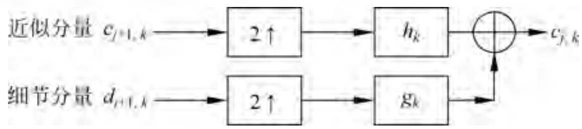


图 5-28 $c_{j+1,k}$ 和 $d_{j+1,k}$ 重构 $c_{j,k}$（2↑代表插值抽样，抽样率为 2）

6. 正交小波变换的实现

MATLAB 提供了一系列的函数实现小波分解、重构等功能。

（1）L＝wmaxlev(S,'wname')：计算尺寸为 S 的信号使用 wname 指定的小波分解时的最大分解级数。

（2）wfilters 函数：计算小波对应的滤波器。

① [LO_D,HI_D,LO_R,HI_R]＝wfilters('wname')：计算 wname 指定的正交或双正交小波对应的分解及重构滤波器。

② [F1,F2]＝wfilters('wname','type')：根据 type 的不同，返回不同的滤波器。type 为 d，返回分解滤波器 LO_D 和 HI_D；type 为 r，返回重构滤波器 LO_R 和 HI_R；type 为 l，返回低通滤波器 LO_D 和 LO_R；type 为 h，返回高通滤波器 HI_D 和 HI_R。

（3）dwt 函数：一级一维离散小波变换。

① [CA,CD]＝dwt(X,'wname')：对向量 X 进行 wname 指定的小波分解，计算近似系数向量 CA 和细节系数向量 CD。

② [CA,CD]＝dwt(X,Lo_D,Hi_D)：使用给定的分解滤波器实现向量 X 的小波分解。

（4）idwt 函数：一级一维小波逆变换。

① X＝idwt(CA,CD,'wname')：基于近似系数向量 CA 和细节系数向量 CD，使用 wname 指定的小波实现近似系数向量 X 的一级重构。

② X＝idwt(CA,CD,Lo_R,Hi_R)：给定重构滤波器实现一级近似系数向量 X 的计算。

（5）wavedec 函数：实现多级一维小波分解。

① [C,L]＝wavedec(X,N,'wname')：使用 wname 指定的小波实现信号 X 在 N 级上的小波分解。N 为正整数，可使用 wmaxlev 函数计算，确保小波系数不受边界效应的影响；若不考虑边界效应，可设置 N≤fix(log2(length(X)))。输出向量 C 为小波的分解系数，L 为各级系数的长度。

② [C,L]＝wavedec(X,N,Lo_D,Hi_D)：使用指定的低通分解滤波器 Lo_D 和高通分解滤波器 Hi_D 实现信号 X 在 N 级上的小波分解。

（6）detcoef 函数：提取一维小波分解的细节系数。

① D＝detcoef(C,L,N)：从小波分解结构[C,L]中提取第 N 级细节系数。N 为整数，N≥1 且 N≤length(L)－2。如果 N 为整数向量，则提取各元素所示级别的细节系数。

② D＝detcoef(C,L)：从小波分解结构[C,L]中提取最高级细节系数。

（7）appcoef 函数：提取一维小波分解的近似系数。

① A＝appcoef(C,L,'wname',N)：从小波分解结构[C,L]中提取第 N 级近似系数。N

为整数,N≥1 且 N≤length(L)-2。

② A=appcoef(C,L,'wname'):提取第 length(L)-2 级近似系数。

(8) wrcoef 函数:由一维小波系数进行单支重构。

① X=wrcoef('type',C,L,'wname',N):基于小波分解结构[C,L]在 N 级计算重构系数向量。type 取 a,重构近似向量;取 b,重构细节向量;N 为整数,且 N≤length(L)-2。

② X=wrcoef('type',C,L,Lo_R,Hi_R,N):基于重构低通滤波器 Lo_R 和重构高通滤波器 Hi_R 实现重构系数向量计算。

(9) waverec 函数:多级一维小波重构。

① X=waverec(C,L,'wname'):基于多级小波分解结构[C,L] 重构信号 X。

② X=waverec(C,L,Lo_R,Hi_R):给定重构滤波器实现重构系数向量计算。

(10) upcoef 函数:一维小波系数直接重构。

① Y=upcoef(O,X,'wname',N):计算向量 X 向上 N 步的重构系数。N 为正整数。若 O 取'a',近似系数被重构;若 O 取'd',细节系数被重构。

② Y=upcoef(O,X,'wname',N,L):重构的同时取出结果中长度为 L 的中间部分。

③ Y=upcoef(O,X,Lo_R,Hi_R,N)或 Y=upcoef(O,X,Lo_R,Hi_R,N,L):使用重构滤波器实现直接重构。

【例 5-21】 装载 sumsin 信号,使用 dwt 和 idwt 函数实现一级一维小波分解与重构。sumsin 是 3 个正弦波的叠加,即 sumsin(t)=sin(3t)+sin(0.3t)+sin(0.03t)。

程序如下:

```
clear,clc,close all;
load sumsin;                          %装载 sumsin 信号
signal = sumsin;
[CA,CD] = dwt(signal,'db4');          %基于 db4 小波实现一级一维小波分解
RecS = idwt(CA,CD,'db4');             %重构
subplot(221),plot(signal),title('原始信号');
subplot(222),plot(CA),title('近似系数');
subplot(223),plot(CD),title('细节系数');
subplot(224),plot(RecS),title('重构信号');
```

程序运行结果如图 5-29 所示。

【例 5-22】 装载 sumsin 信号,实现多级一维小波分解与重构。

程序如下:

```
clear,clc,close all;
load sumsin;
signal = sumsin;
[C,L] = wavedec(signal,2,'db4');      %基于 db4 小波实现 2 级一维小波分解
[CD1,CD2] = detcoef(C,L,[1 2]);       %提取细节系数
CA2 = appcoef(C,L,'db4',2);           %提取近似系数
[Lo_R,Hi_R] = wfilters('db4','r');    %计算重构滤波器系数
Recs1 = waverec(C,L,Lo_R,Hi_R);       %采用重构滤波器实现信号重构
```

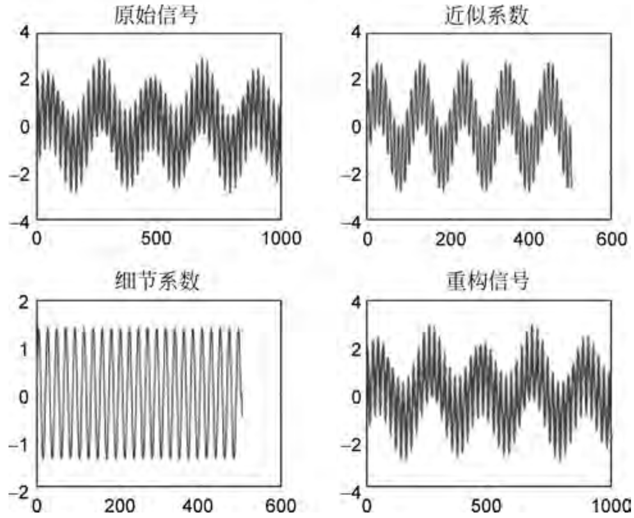


图 5-29　一级一维小波分解与重构

```
Recs2 = wrcoef('a',C,L,'db4',2);          % 基于近似系数重构信号
subplot(231),plot(signal),title('原始信号');
subplot(234),plot(CA2),title('二级近似系数');
subplot(232),plot(CD2),title('二级细节系数');
subplot(235),plot(CD1),title('一级细节系数');
subplot(233),plot(Recs2),title('近似系数重构');
subplot(236),plot(Recs1),title('重构信号');
```

程序运行结果如图 5-30 所示。

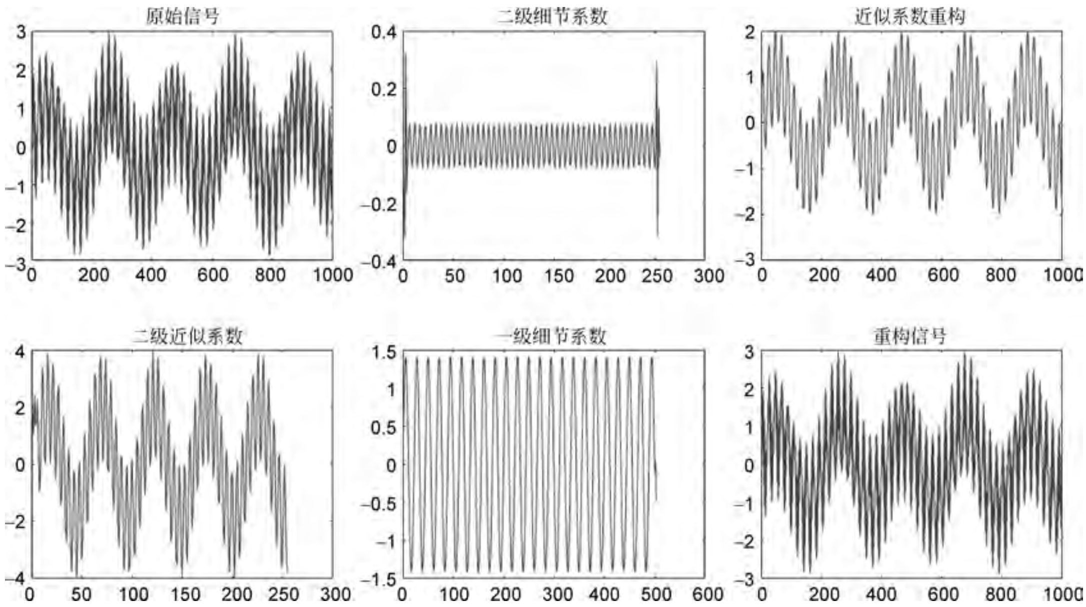


图 5-30　多尺度一维小波分解与重构

小波变换还有很多别的很有用的理论和特点，如小波包、多带小波、多小波等，因篇幅关系，不再深入分析。

5.5.3　二维小波变换

图像为二维信号，用二元函数 $f(x,y)\in L^2(R^2)$表示，可以对其进行二维小波变换和多分辨分析。

1. 二维多分辨分析

设$(\{V_j^1\}_{j\in Z},\varphi^1(t))$，$(\{V_j^2\}_{j\in Z},\varphi^2(t))$是 $L^2(R)$的两个多分辨分析，$\psi^1(t)$，$\psi^2(t)$分别是相应的正交小波函数，则 $\widetilde{V}_j$ 是 V_j^1 与 V_j^2 的张量积空间，如式(5-80)所示。

$$\widetilde{V}_j=V_j^1\otimes V_j^2=\{f^1(x)f^2(y)\mid f^1(x)\in V_j^1,f^2(y)\in V_j^2\} \tag{5-80}$$

$\{\varphi_{j,l}^1(x)\}_{l\in Z}$，$\{\varphi_{j,m}^2(y)\}_{m\in Z}$ 是 V_j^1 与 V_j^2 的标准正交基，则$\{\varphi_{j,l}^1(x)\varphi_{j,m}^2(y)\}_{l,m\in Z}$ 是 $\widetilde{V}_j$ 的标准正交基。

设 W_j^1 是 V_j^1 在 V_{j-1}^1 中的正交补，W_j^2 是 V_j^2 在 V_{j-1}^2 中的正交补，则

$$\begin{aligned}\widetilde{V}_{j-1}&=V_{j-1}^1\otimes V_{j-1}^2=(V_j^1\oplus W_j^1)\otimes(V_j^2\oplus W_j^2)\\&=(V_j^1\otimes V_j^2)\oplus(V_j^1\otimes W_j^2)\oplus(W_j^1\otimes V_j^2)\oplus(W_j^1\otimes W_j^2)\\&=\widetilde{V}_j\oplus\widetilde{W}_j^1\oplus\widetilde{W}_j^2\oplus\widetilde{W}_j^3\end{aligned} \tag{5-81}$$

其中，$\widetilde{W}_j^1$，$\widetilde{W}_j^2$，$\widetilde{W}_j^3$ 称为二维小波空间。它们的标准正交基依次为$\{\varphi_{j,l}^1(x)\psi_{j,m}^2(y)\}_{l,m\in Z}$，$\{\psi_{j,l}^1(x)\varphi_{j,m}^2(y)\}_{l,m\in Z}$ 和$\{\psi_{j,l}^1(x)\psi_{j,m}^2(y)\}_{l,m\in Z}$。

记

$$\begin{cases}\psi^1(x,y)=\varphi^1(x)\psi^2(y)\\\psi^2(x,y)=\psi^1(x)\varphi^2(y)\\\psi^3(x,y)=\psi^1(x)\psi^2(y)\\\varphi(x,y)=\varphi^1(x)\varphi^2(y)\end{cases} \tag{5-82}$$

则 $\varphi(x,y)$，$\psi^1(x,y)$，$\psi^2(x,y)$，$\psi^3(x,y)$的伸缩平移系分别构成 $\widetilde{V}_j$，$\widetilde{W}_j^1$，$\widetilde{W}_j^2$，$\widetilde{W}_j^3$ 的标准正交基。

由式(5-81)可知：

$$L^2(R^2)=\sum_{j=-\infty}^{+\infty}\widetilde{W}_j \tag{5-83}$$

其中，$\widetilde{W}_j=\widetilde{W}_j^1\oplus\widetilde{W}_j^2\oplus\widetilde{W}_j^3$。对于任何 $f(x,y)\in L^2(R^2)$，有：

$$f(x,y)=\sum_{j=-\infty}^{+\infty}w_j(x,y) \tag{5-84}$$

其中，$w_j(x,y)\in\widetilde{W}_j$。

因此,二维小波变换的重构公式为:

$$f(x,y)=\sum_{j=-\infty}^{+\infty}\sum_{l,m}\begin{bmatrix}\alpha_{l,m}^{j}\varphi_{j,l}^{1}(x)\psi_{j,m}^{2}(y)+\beta_{l,m}^{j}\psi_{j,l}^{1}(x)\varphi_{j,m}^{2}(y)\\+\gamma_{l,m}^{j}\psi_{j,l}^{1}(x)\psi_{j,m}^{2}(y)\end{bmatrix}\tag{5-85}$$

其中,$\alpha_{l,m}^{j}$、$\beta_{l,m}^{j}$、$\gamma_{l,m}^{j}$ 是 $f(x,y)$的二维离散小波变换,$\alpha_{l,m}^{j}=\iint\limits_{R^2}f(x,y)\varphi_{j,l}^{1^*}(x)\psi_{j,m}^{2^*}(y)\mathrm{d}x\mathrm{d}y$,$\beta_{l,m}^{j}=\iint\limits_{R^2}f(x,y)\psi_{j,l}^{1^*}(x)\varphi_{j,m}^{2^*}(y)\mathrm{d}x\mathrm{d}y$,$\gamma_{l,m}^{j}=\iint\limits_{R^2}f(x,y)\psi_{j,l}^{1^*}(x)\psi_{j,m}^{2^*}(y)\mathrm{d}x\mathrm{d}y$。

实际问题中,二元函数 $f(x,y)$只有有限分辨率,设 $f(x,y)\in\widetilde{V}_0$,因此,

$$f(x,y)=f_s(x,y)+w_s(x,y)+w_{s-1}(x,y)+\cdots+w_1(x,y)\tag{5-86}$$

$$\begin{aligned}f(x,y)=&\sum_{l,m}[\lambda_{l,m}^{s}\varphi_{s,l}^{1}(x)\varphi_{s,m}^{2}(y)]+\\&\sum_{j=1}^{s}\sum_{l,m}[\alpha_{l,m}^{j}\varphi_{j,l}^{1}(x)\psi_{j,m}^{2}(y)+\beta_{l,m}^{j}\psi_{j,l}^{1}(x)\varphi_{j,m}^{2}(y)+\gamma_{l,m}^{j}\psi_{j,l}^{1}(x)\psi_{j,m}^{2}(y)]\end{aligned}\tag{5-87}$$

其中,$\lambda_{l,m}^{s}=\iint\limits_{R^2}f(x,y)\varphi_{s,l}^{1^*}(x)\varphi_{s,m}^{2^*}(y)\mathrm{d}x\mathrm{d}y$。

式(5-87)中第一项 $f_s(x,y)$是 $f(x,y)$在尺度 s 下的近似;后三项称为 $f(x,y)$在不同尺度 j 下的细节。

2. 二维正交小波分解

由于 $\varphi^1(x)$,$\varphi^2(y)$,$\psi^1(x)$,$\psi^2(y)$满足双尺度方程:

$$\begin{cases}\varphi^1(x)=\sqrt{2}\sum\limits_{l}h_l^1\varphi^1(2x-l)\\\psi^1(x)=\sqrt{2}\sum\limits_{l}g_l^1\varphi^1(2x-l)\\\varphi^2(y)=\sqrt{2}\sum\limits_{m}h_m^2\varphi^2(2y-m)\\\psi^2(y)=\sqrt{2}\sum\limits_{m}g_m^2\varphi^2(2y-m)\end{cases}\tag{5-88}$$

将式(5-88)代入式(5-82),得

$$\begin{cases}\varphi(x,y)=\varphi^1(x)\varphi^2(y)=2\sum\limits_{l,m}h_l^1h_m^2\varphi^1(2x-l)\varphi^2(2y-m)=2\sum\limits_{l,m}h_l^1h_m^2\varphi(2x-l,2y-m)\\\psi^1(x,y)=\varphi^1(x)\psi^2(y)=2\sum\limits_{l,m}h_l^1g_m^2\varphi^1(2x-l)\varphi^2(2y-m)=2\sum\limits_{l,m}h_l^1g_m^2\varphi(2x-l,2y-m)\\\psi^2(x,y)=\psi^1(x)\varphi^2(y)=2\sum\limits_{l,m}g_l^1h_m^2\varphi^1(2x-l)\varphi^2(2y-m)=2\sum\limits_{l,m}g_l^1h_m^2\varphi(2x-l,2y-m)\\\psi^3(x,y)=\psi^1(x)\psi^2(y)=2\sum\limits_{l,m}g_l^1g_m^2\varphi^1(2x-l)\varphi^2(2y-m)=2\sum\limits_{l,m}g_l^1g_m^2\varphi(2x-l,2y-m)\end{cases}\tag{5-89}$$

则

$$
\begin{aligned}
\lambda_{l,m}^{j+1} &= \iint_{R^2} f(x,y)\varphi_{j+1,l}^{1^*}(x)\varphi_{j+1,m}^{2^*}(y)\,\mathrm{d}x\,\mathrm{d}y \\
&= \iint_{R^2} f(x,y)\{2^{-(j+1)/2}\varphi^{1^*}[2^{-(j+1)}x-l]\}\{2^{-(j+1)/2}\varphi^{2^*}[2^{-(j+1)}y-m]\}\,\mathrm{d}x\,\mathrm{d}y \\
&= \iint_{R^2} f(x,y)\left\{2^{-j/2}\sum_{n} h_n^{1^*}\varphi^{1^*}[2(2^{-(j+1)}x-l)-n]\right\} \\
&\quad \left\{2^{-j/2}\sum_{k} h_k^{2^*}\varphi^{2^*}[2(2^{-(j+1)}y-m)-k]\right\}\mathrm{d}x\,\mathrm{d}y \\
&= \iint_{R^2} f(x,y)\left\{2^{-j/2}\sum_{p} h_{p-2l}^{1^*}\varphi^{1^*}(2^{-j}x-p)\right\}\left\{2^{-j/2}\sum_{q} h_{q-2m}^{2^*}\varphi^{2^*}(2^{-j}y-q)\right\}\mathrm{d}x\,\mathrm{d}y \\
&= \sum_{p,q} h_{p-2l}^{1^*}h_{q-2m}^{2^*}\iint_{R^2} f(x,y)\varphi_{j,p}^{1^*}(x)\varphi_{j,q}^{2^*}(y)\,\mathrm{d}x\,\mathrm{d}y \\
&= \sum_{p,q} h_{p-2l}^{1^*}h_{q-2m}^{2^*}\lambda_{p,q}^{j}
\end{aligned}
\tag{5-90}
$$

同理，得

$$
\begin{cases}
\alpha_{l,m}^{j+1} = \sum\limits_{p,q} h_{p-2l}^{1^*}g_{q-2m}^{2^*}\lambda_{p,q}^{j} \\
\beta_{l,m}^{j+1} = \sum\limits_{p,q} g_{p-2l}^{1^*}h_{q-2m}^{2^*}\lambda_{p,q}^{j} \\
\gamma_{l,m}^{j+1} = \sum\limits_{p,q} g_{p-2l}^{1^*}g_{q-2m}^{2^*}\lambda_{p,q}^{j}
\end{cases}
\tag{5-91}
$$

由式(5-90)和式(5-91)可以看出，分辨率 j 的近似分量 $\lambda_{p,q}^{j}$ 分解为分辨率为 $j+1$ 的近似分量 $\lambda_{l,m}^{j+1}$ 和细节分量 $\alpha_{l,m}^{j+1}$,$\beta_{l,m}^{j+1}$,$\gamma_{l,m}^{j+1}$ 的分解方法可以用如图 5-31 所示的滤波过程来表示：首先对水平方向进行滤波，再对垂直方向进行滤波，得到 4 个不同的频带；若对近似分量 $\lambda_{l,m}^{j+1}$ 继续进行这样的滤波过程，即可得到如图 5-32 所示的塔形分解。

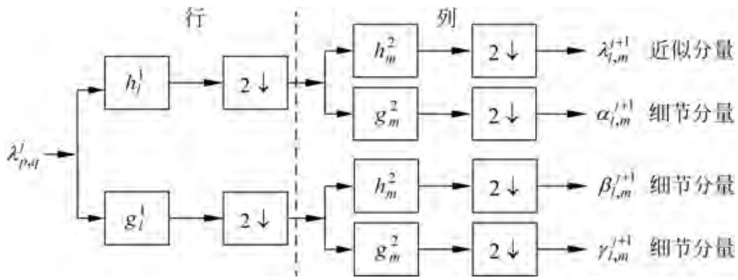


图 5-31　二维小波变换近似分量 $\lambda_{p,q}^{j}$ 分解为 $\lambda_{l,m}^{j+1}$ 和 $\alpha_{l,m}^{j+1}$、$\beta_{l,m}^{j+1}$、$\gamma_{l,m}^{j+1}$

若对一幅二维图像进行 3 层分解，可得图 5-32。其中，L 代表低频分量，H 代表高频分量；LH 代表垂直方向上的高频信息；HL 频带存放的是图像水平方向的高频信息；HH 频带存放图像在对角线方向的高频信息。

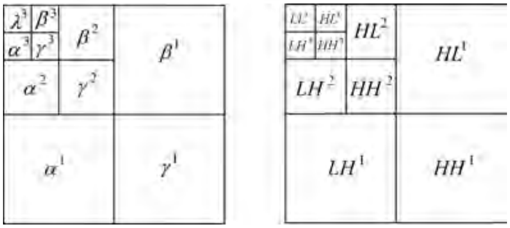


图 5-32　二维图像 3 层小波分解示意图

3. 二维正交小波重构

因为 $\widetilde{V}_j=\widetilde{V}_{j+1}\oplus\widetilde{W}^1_{j+1}\oplus\widetilde{W}^2_{j+1}\oplus\widetilde{W}^3_{j+1}$，所以，

$$f_j(x,y)=f_{j+1}(x,y)+w^1_{j+1}(x,y)+w^2_{j+1}(x,y)+w^3_{j+1}(x,y)$$

而

$$\begin{aligned}
f_{j+1}(x,y)&=\sum_{l,m\in Z}\langle f(x,y),\varphi_{j+1,l,m}(x,y)\rangle\varphi_{j+1,l,m}(x,y)=\sum_{l,m\in Z}\lambda^{j+1}_{l,m}\varphi^1_{j+1,l}(x)\varphi^2_{j+1,m}(y)\\
&=\sum_{l,m\in Z}\lambda^{j+1}_{l,m}[2^{-(j+1)/2}\varphi^1(2^{-(j+1)}x-l)][2^{-(j+1)/2}\varphi^2(2^{-(j+1)}y-m)]\\
&=\sum_{l,m\in Z}\lambda^{j+1}_{l,m}\left\{2^{-j/2}\sum_{n\in Z}h^1_n\varphi^1[2(2^{-(j+1)}x-l)-n]\right\}\\
&\quad\left\{2^{-j/2}\sum_{n\in Z}h^2_n\varphi^2[2(2^{-(j+1)}y-m)-n]\right\}\\
&=\sum_{l,m\in Z}\lambda^{j+1}_{l,m}\left[2^{-j/2}\sum_{p\in Z}h^1_{p-2l}\varphi^1\{2^{-j}x-p\}\right]\left[2^{-j/2}\sum_{q\in Z}h^2_{q-2m}\varphi^2(2^{-j}y-q)\right]\\
&=\sum_{l,m\in Z}\lambda^{j+1}_{l,m}\sum_{p,q\in Z}h^1_{p-2l}h^2_{q-2m}\varphi^1_{j,p}(x)\varphi^2_{j,q}(y)
\end{aligned}\tag{5-92}$$

同理，得

$$\begin{aligned}
w^1_{j+1}(x,y)&=\sum_{l,m\in Z}\langle f(x,y),\psi^1_{j+1,l,m}(x,y)\rangle\psi^1_{j+1,l,m}(x,y)=\sum_{l,m\in Z}\alpha^{j+1}_{l,m}\varphi^1_{j+1,l}(x)\psi^2_{j+1,m}(y)\\
&=\sum_{l,m\in Z}\alpha^{j+1}_{l,m}\sum_{p,q\in Z}h^1_{p-2l}g^2_{q-2m}\varphi^1_{j,p}(x)\varphi^2_{j,q}(y)
\end{aligned}\tag{5-93}$$

$$\begin{aligned}
w^2_{j+1}(x,y)&=\sum_{l,m\in Z}\langle f(x,y),\psi^2_{j+1,l,m}(x,y)\rangle\psi^2_{j+1,l,m}(x,y)=\sum_{l,m\in Z}\beta^{j+1}_{l,m}\psi^1_{j+1,l}(x)\varphi^2_{j+1,m}(y)\\
&=\sum_{l,m\in Z}\beta^{j+1}_{l,m}\sum_{p,q\in Z}g^1_{p-2l}h^2_{q-2m}\varphi^1_{j,p}(x)\varphi^2_{j,q}(y)
\end{aligned}\tag{5-94}$$

$$\begin{aligned}
w^3_{j+1}(x,y)&=\sum_{l,m\in Z}\langle f(x,y),\psi^3_{j+1,l,m}(x,y)\rangle\psi^3_{j+1,l,m}(x,y)=\sum_{l,m\in Z}\gamma^{j+1}_{l,m}\psi^1_{j+1,l}(x)\psi^2_{j+1,m}(y)\\
&=\sum_{l,m\in Z}\gamma^{j+1}_{l,m}\sum_{p,q\in Z}g^1_{p-2l}g^2_{q-2m}\varphi^1_{j,p}(x)\varphi^2_{j,q}(y)
\end{aligned}\tag{5-95}$$

所以，

$$f_{j+1}(x,y)+w^1_{j+1}(x,y)+w^2_{j+1}(x,y)+w^3_{j+1}(x,y)$$

$$=\sum_{p,q\in Z}\left\{\sum_{l,m\in Z}\left[\lambda_{l,m}^{j+1}h_{p-2l}^{1}h_{q-2m}^{2}+\alpha_{l,m}^{j+1}h_{p-2l}^{1}g_{q-2m}^{2}+\beta_{l,m}^{j+1}g_{p-2l}^{1}h_{q-2m}^{2}+\gamma_{l,m}^{j+1}g_{p-2l}^{1}g_{q-2m}^{2}\right]\right\}\varphi_{j,p}^{1}(x)\varphi_{j,q}^{2}(y)$$

而

$$f_j(x,y)=\sum_{p,q\in Z}\langle f(x,y),\varphi_{j,p,q}(x,y)\rangle\varphi_{j,p,q}(x,y)=\sum_{p,q\in Z}\lambda_{p,q}^{j}\varphi_{j,p}^{1}(x)\varphi_{j,q}^{2}(y)$$

因此

$$\lambda_{p,q}^{j}=\left\{\sum_{l,m\in Z}\left[\lambda_{l,m}^{j+1}h_{p-2l}^{1}h_{q-2m}^{2}+\alpha_{l,m}^{j+1}h_{p-2l}^{1}g_{q-2m}^{2}+\beta_{l,m}^{j+1}g_{p-2l}^{1}h_{q-2m}^{2}+\gamma_{l,m}^{j+1}g_{p-2l}^{1}g_{q-2m}^{2}\right]\right\} \tag{5-96}$$

式(5-96)所示的重构可以用图5-33所示的滤波过程来表示。

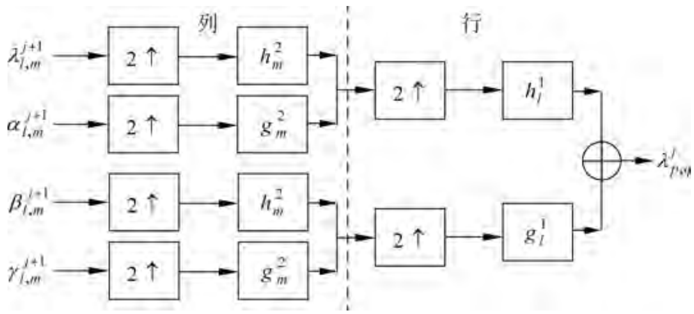


图5-33　二维多分辨分析的重构

4. 二维正交小波的实现

利用MATLAB小波工具箱可以实现对图像的小波分解与重构。在命令窗口输入：

```
>> wavemenu
```

打开小波分析工具箱，如图5-34(a)所示；选择Wavelet 2-D，如图5-34(b)所示。

在二维小波分析页面，从菜单中打开图像，在右侧选择Daubechies小波(N=4)进行分解与重构，如图5-35所示。可以尝试其他的分析和统计计算。

MATLAB提供了相应的二维小波函数，部分列举如下。

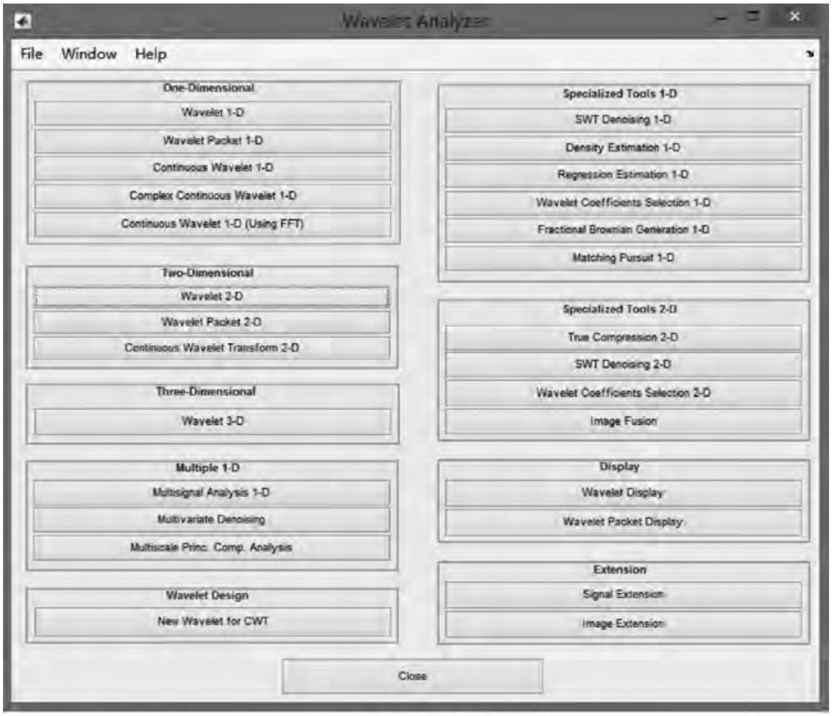
(1) wavefun2函数：计算尺度、小波函数。

① [S,W1,W2,W3,XYVAL]=wavefun2('wname',ITER)：返回wname指定的正交小波的尺度函数S与3个小波函数W1、W2、W3，ITER为迭代次数，XYVAL为取值点，支撑区间内有$2^{ITER}\times 2^{ITER}$个点。

② …=wavefun2(…,'plot')：计算并绘制函数。

(2) dwt2函数：实现一级二维离散小波变换。

① [CA,CH,CV,CD]=dwt2(X,'wname')：用wname指定的小波分解矩阵X为近似系数矩阵CA和细节系数矩阵CH、CV和CD。

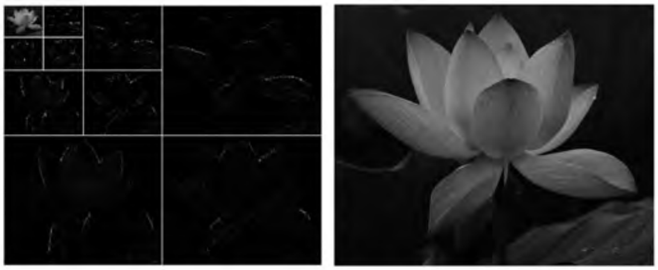


(a) 小波分析工具箱主页面

(b) 二维小波分析页面

图 5-34 MATLAB 中的小波分析工具

(a) 原图　　(b) 一级小波分解子带图　　(c) 二级小波分解子带图



(d) 三级小波分解子带图　　(e) 三级分解重构图

图 5-35　lotus 图像的小波分解与重构

② [CA,CH,CV,CD]=dwt2(X,Lo_D,Hi_D)：采用低通分解滤波器 Lo_D 和高通分解滤波器 Hi_D 分解矩阵 X。

(3) idwt2 函数：一级二维离散小波逆变换。

① X=idwt2(CA,CH,CV,CD,'wname')：基于近似系数矩阵 CA 和细节系数矩阵 CH、CV 和 CD,使用 wname 指定的小波实现近似系数矩阵 X 的一级重构。

② X=idwt2(CA,CH,CV,CD,Lo_R,Hi_R)：采用低通重构滤波器 Lo_R 和高通重构滤波器 Hi_R 重构矩阵 X。

(4) wavedec2 函数：多级二维小波分解。

① [C,S]=wavedec2(X,N,'wname')：使用 wname 指定的小波实现矩阵 X 的 N 级分解。

② [C,S]=wavedec2(X,N,Lo_D,Hi_D)：使用分解滤波器分解矩阵 X。

C=[A(N)|H(N)|V(N)|D(N)|H(N−1)|V(N−1)|D(N−1)|…|H(1)|V(1)|D(1)],A、H、V、D 分别为低频、水平高频、垂直高频、对角高频系数矩阵。N 为正整数。S(1,:)是级数 N 的低频系数长度；S(i,:)是级数 N−i+2 的高频系数长度,i=2,…,N+1；S(N+2,:)=size(X)。

(5) waverec2 函数：多级二维小波重构。

① X=waverec2(C,S,'wname')：基于多级小波分解结构[C,S]重构矩阵 X。

② X=waverec2(C,S,Lo_R,Hi_R)：采用重构滤波器实现矩阵 X 重构。

(6) appcoef2 函数：提取二维小波分解的低频系数。

① A=appcoef2(C,S,'wname',N)：从小波分解结构[C,L]中提取第 N 级近似系数矩

阵 A,N 为正整数,且 N≤size(S,1)−2。

② A=appcoef2(C,S,'wname'):提取第 size(S,1)−2 级近似系数矩阵 A。

③ A=appcoef2(C,S,Lo_R,Hi_R)或者 A=appcoef2(C,S,Lo_R,Hi_R,N):基于分解滤波器提取近似系数 A。

(7) detcoef2 函数:提取二维小波分解的高频系数。

① D=detcoef2(O,C,S,N):从小波分解结构[C,L]中提取第 N 级细节系数矩阵 D。O 可取'h'、'v'或者'd',对应水平、垂直和对角高频系数;N≥1 且 N≤size(S,1)−2。

② [H,V,D]=detcoef2('all',C,S,N):提取第 N 级的水平 H、垂直 V、对角 D 细节系数矩阵。

(8) upcoef2 函数:二维小波系数的直接重构。

① Y=upcoef2(O,X,'wname',N,S):计算矩阵 X 的向上 N 步的重构系数,并提取长度为 S 的中间部分。若 O 取'a',则对近似系数重构;若 O 取'h'、'v'、'd',则对水平、垂直或对角细节系数重构。N 为正整数。

② Y=upcoef2(O,X,Lo_R,Hi_R,N,S):基于重构滤波器实现重构。

(9) Y=wcodemat(X,NBCODES,OPT,ABSOL):返回对数据矩阵 X 编码的矩阵 Y。NBCODES 为伪编码的最大值,即编码范围为 0～NBCODES−1,默认为 16;若 ABSOL 为 0,返回编码矩阵;若 ABSOL 非 0,返回数据矩阵的绝对值 abs(X);默认为 1。OPT 指定了编码方式,若为'row'或'r',按行编码;若为'col'或'c',按列编码;若为'mat'或'm',按整个矩阵编码;默认为'mat'。

【例 5-23】 对 cameraman 图像进行一级小波分解及重构。

程序如下:

```
clear,clc,close all;
Image = imread('cameraman.jpg');
grayI = rgb2gray(Image);
[ca1,ch1,cv1,cd1] = dwt2(grayI,'db4');                  %用 db4 小波对图像进行一级小波分解
DWTI = [wcodemat(ca1,256),wcodemat(ch1,256);wcodemat(cv1,256),wcodemat(cd1,256)];
                                                         %组成小波系数显示矩阵
result = idwt2(ca1,ch1,cv1,cd1,'db4');                  %一级重构
subplot(131),imshow(Image),title('原图');
subplot(132),imshow(DWTI/256),title('一级分解');         %显示一级分解后的近似和细节图像
subplot(133),imshow(result,[]),title('一级重构');        %重构图像显示
```

程序运行结果如图 5-36 所示。

【例 5-24】 对 cameraman 图像进行二级小波分解及重构。

程序如下:

```
clc,clear,close all;
Image = imread('cameraman.jpg');
grayI = rgb2gray(Image);
[C,S] = wavedec2(grayI,2,'db4');                        %用 db4 小波对图像进行二级小波分解
```

(a) 原图 (b) 一级小波分解子带图 (c) 一级分解重构图

图 5-36 cameraman 图像的一级分解及重构

```
siz = S(size(S,1),:);
CA2 = appcoef2(C,S,'db4',2);                          % 提取二级小波分解低频变换系数
[CH2,CV2,CD2] = detcoef2('all',C,S,2);                % 提取二级小波分解高频变换系数
[CH1,CV1,CD1] = detcoef2('all',C,S,1);                % 提取一级小波分解高频变换系数
CA1 = [wcodemat(CA2,256),wcodemat(CH2,256);wcodemat(CV2,256),wcodemat(CD2,256)];
k = S(2,1) * 2 - S(3,1);                              % 两级高频系数长度差
CH1 = padarray(CH1,[k k],1,'pre');
CV1 = padarray(CV1,[k k],1,'pre');
CD1 = padarray(CD1,[k k],1,'pre');          % 填充一级小波高频系数数组,使两级系数维数一致
DWTI = [CA1,wcodemat(CH1,256);wcodemat(CV1,256),wcodemat(CD1,256)];
RecA = upcoef2('a',CA2,'db4',2,siz);                  % 二级近似系数向上重构
RecV = upcoef2('v',CV2,'db4',2,siz);                  % 垂直细节系数向上重构
result = waverec2(C,S,'db4');                         % 二级重构
RecA = mat2gray(RecA);
RecV = mat2gray(RecV);
subplot(221),imshow(DWTI/256),title('二级分解');      % 显示二级分解后的近似和细节图像
subplot(222),imshow(RecA),title('二级近似系数重构');
subplot(223),imshow(RecV),title('垂直细节系数重构');
subplot(224),imshow(result,[]),title('二级重构');
```

程序运行结果如图 5-37 所示。

(a) 二级小波分解子带图 (b) 二级近似系数重构 (c) 垂直细节系数重构 (d) 二级分解重构

图 5-37 cameraman 图像的二级分解及重构

5.5.4 小波变换在图像处理中的应用

小波变换因其频率分解、多分辨分析等特性，应用于数字图像处理，可以出色地完成诸如图像滤波、图像增强、图像融合、图像压缩等多种处理，得到广泛的应用。

1. 基于小波变换的图像降噪

小波变换具有下述特点。

(1) 低熵性。图像变换后熵降低。

(2) 多分辨性。采用多分辨率的方法，可以非常好地刻画信号的非平稳特征，如边缘、尖峰、断点等，可在不同分辨率下根据信号和噪声分布的特点去噪。

(3) 小波变换可以灵活地选择不同的小波基。因此，小波去噪是小波变换在数字图像处理中的一个重要应用。

如前文所述，小波变换实际上是通过滤波器将图像信号分解为低频和高频的，噪声的大部分能量集中在高频部分，通过处理小波分解后的高频系数，实现噪声的降低。常见的基于小波变换的图像降噪方法有以下几种。

(1) 基于小波变换极大值原理的降噪方法。该方法根据信号与噪声在小波变换各尺度上不同的传播特性，剔除由噪声产生的模极大值点，用剩余的模极大值点恢复信号。

(2) 基于相关性的降噪方法。该方法对含噪声的信号进行变换后，计算相邻尺度间小波系数的相关性，根据相关性大小区别小波系数的类型，并进行取舍、重构。

(3) 基于阈值的降噪方法。该方法按一定的规则(或阈值化)将小波系数划分成两类：重要的、规则的小波系数和非重要的或受噪声干扰的小波系数，舍弃不重要的小波系数然后重构去噪后的图像。这种方法的关键是阈值的设计。常用的阈值函数有硬阈值和软阈值函数。硬阈值方法指的是设定阈值，小波系数绝对值大于阈值的保留，小于阈值的置0，可以很好地保留边缘等局部特征，但会出现振铃等失真现象；软阈值方法指将较小的小波系数置0，较大的小波系数按一定的函数计算，向0收缩，处理结果较硬阈值方法平滑，但因绝对值较大的小波系数减小，会导致损失部分高频信息，造成图像边缘的失真模糊。

【例 5-25】 基于小波变换对图像进行硬阈值、软阈值去噪。

程序如下：

```
clear,clc,close all;
Image = rgb2gray(imread('peppers.jpg'));
noiseI = imnoise(Image,'gaussian');            % 添加高斯噪声
[c,s] = wavedec2(noiseI,2,'sym5');             % 用 sym5 小波对图像进行二层小波分解
sigma = std(c);                                % 小波系数标准差
thresh = 2 * sigma;                            % 设定阈值
csize = size(c);
c(abs(c)< thresh) = 0;                         % 小波系数小于阈值则置 0
```

```
denoiseI1 = uint8(waverec2(c,s,'sym5'));
pos1 = find(c > thresh);       c(pos1) = c(pos1) - thresh;
pos2 = find(c < - thresh);     c(pos2) = c(pos2) + thresh;      % 大系数向 0 收缩
denoiseI2 = uint8(waverec2(c,s,'sym5'));
subplot(221),imshow(Image),title('原图像');
subplot(222),imshow(noiseI),title('高斯噪声图像');
subplot(223),imshow(denoiseI1),title('硬阈值降噪');
subplot(224),imshow(denoiseI2),title('软阈值降噪');
```

程序运行结果如图 5-38 所示。

(a) 高斯噪声图像　(b) 硬阈值降噪　(c) 软阈值降噪

图 5-38　基于小波变换的图像降噪

2. 基于小波变换的边缘检测

图像边缘是指在图像平面中灰度值发生跳变的点连接所成的曲线段，包含了图像的重要信息。找出图像的边缘称为边缘检测，是图像处理中的重要内容。二维小波变换能检测二维函数 $f(x,y)$的局部突变，因此是检测图像边缘的有力工具。

随着技术的发展，目前已经有很多新颖的基于小波变换的图像边缘检测技术和方法，如多尺度小波变换边缘提取算法、嵌入可信度的边缘检测方法、奇异点模极大值检测算法等。

3. 基于小波变换的图像压缩

小波变换特别适用于细节丰富、空间相关性差、冗余度低的图像数据压缩处理。同DCT 类似，小波变换后使图像能量集中在少部分的小波系数上，可以通过简单的量化方法，将较小能量的小波系数省去，保留能量较大的小波系数，从而达到压缩的目的。所以，可以采用直接阈值方法实现基于小波变换的图像压缩，压缩效果好坏关键在于阈值的选择。考虑到人眼视觉系统对高频分量反应不敏感，而对低频分量反应敏感，所以，可以给低频区分配相对高的码率，给高频区分配相对低的码率，以降低数据量，如基于小波树结构的矢量量化法、嵌入式零树小波编码等。JPEG 2000 压缩标准中采用基于小波变换的图像压缩技术。

4. 基于小波变换的图像增强

图像增强是指提高图像的对比度,以增加图像的视觉效果和可理解性,同时减少或抑制图像中的噪声,提高视觉质量。常用的图像增强技术可以分为基于空间域和基于变换域两种,前者直接对像素点进行运算,后者通过将图像进行正交变换,对变换域内的系数进行调整以达到提高输出图像对比度的目的。小波变换将图像分解为大小、位置和方向不同的分量,根据需要改变某些分量系数,从而使得感兴趣的分量放大,不需要的分量减小,以达到图像增强的目的。

5. 基于小波变换的图像融合

图像融合是指将同一对象的两幅或更多的图像合成在一幅图像中,以便比原来任何一幅图像更容易为人所理解。基于小波变换的图像融合是指将原图像进行小波分解,在小波域通过一定的融合算子融合小波系数,再重构生成融合的图像,如图 5-39 所示。小波变换可以将图像分解到不同的频率域,在不同的频率域运用不同的融合算法,得到合成图像的多分辨分解,从而在合成图像中保留原图像在不同频率域的显著特征。

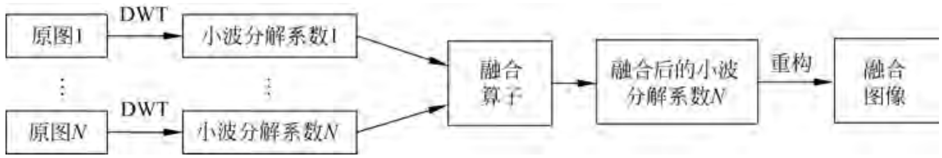


图 5-39　基于小波变换的图像融合过程

基于小波变换的图像融合的关键在于融合算法。例如,对于低频小波分解系数采用取平均的方法,高频分解系数的融合可以采用均值法、最大值法、基于区域的方法、基于边缘强度的方法等。

小波融合能够针对输入图像的不同特征来选择小波基及小波变换的级数,在融合时可以根据实际需要来引入双方的细节信息,表现出更强的针对性和实用性,融合效果更好。

【例 5-26】 采用 DWT 对图像进行融合。

程序如下:

```
Image1 = rgb2gray(imread('desert.jpg'));
Image2 = rgb2gray(imread('car.jpg'));
[ca1,ch1,cv1,cd1] = dwt2(Image1,'db4');              % 用 db4 小波对背景图进行一级小波分解
[ca2,ch2,cv2,cd2] = dwt2(Image2,'db4');              % 用 db4 小波对前景图进行一级小波分解
ca = (ca1 + ca2)/2;                                  % DWT 低频系数取平均融合
ch = max(ch1,ch2);cv = max(cv1,cv2);cd = max(cd1,cd2);   % DWT 高频系数取最大值融合
result = idwt2(ca,ch,cv,cd,'db4')/256;
imshow(result),title('图像融合');
```

程序运行结果如图 5-40 所示。

(a) 背景图　　(b) 前景图　　(c) DWT融合

图 5-40　综合实例结果图

以上是对小波变换在图像处理中的部分主要应用做了简要介绍，有兴趣的读者可以在学习过相关图像处理原理和概念后，结合小波变换的理论进行详细学习。

5.6　本章小结

本章主要介绍了图像的常见正交变换，包括 DFT、DCT、K-L 变换、Radon 变换以及小波变换，详细介绍了各种正交变换的原理及 MATLAB 实现。正交变换在不同的处理算法中经常用到，应熟悉其基本原理、变换特点、常用函数及处理效果，以便灵活应用。