

第 5 章 图像频域增强

本章学习目标：

- (1) 从物理意义的视角理解傅里叶变换的实质。
- (2) 掌握二维快速傅里叶变换的实现方法。
- (3) 在理解图像频域增强原理的基础上,掌握常用低通、高通、同态滤波器的各自特点、实现方法及应用场景。

为了有效、快速地对图像进行处理和分析,常常需要将原定义在图像空间(空域)的图像以某种形式转换到另外的空间,并利用这些空间的特有性质方便地进行加工,最后再转换回图像空间(空域),以得到所需要的效果。这些转换称为图像变换技术。在解决某个图像处理问题时,会在不同的空间来回切换。掌握图像变换技术,就可以在不同的空间下,利用不同空间的优势将问题解决。

目前研究的图像变换基本上都是正交变换。正交变换可以减少图像数据的相关性,获取图像的整体特点,有利于用较少的数据表示原始图像。这对图像的分析、存储以及传输都非常有意义。主要的正交变换有离散傅里叶变换,离散余弦变换、K-L 变换、沃尔什-哈达玛变换及小波变换。本章重点介绍离散傅里叶变换及其在图像增强领域的应用。

5.1 离散傅里叶变换基础

傅里叶分析最初是用于热过程解析分析的工具,其思想方法仍然具有典型的还原论和分析主义的特征。“任意”的函数都可通过傅里叶变换表示为正弦函数的线性组合形式,而正弦函数在物理意义上是被充分研究、相对简单的函数类。在生活中,大到天体观测、小到 MP3 播放器上的音频频谱,没有傅里叶变换都无法实现。

1. 什么是频域

人们看到的世界都以时间贯穿,例如股票的走势、人的身高、汽车的轨迹都会随着时间发生改变。这种以时间作为参照观察动态世界的方法称为时域分析。世间万物都在随着时间不停地改变、永不静止。用另一个方法观察世界时,就会惊奇地发现,世界是永恒不变的,这个静止的世界称为频域。

例如,一段音乐从时域角度看,是一个随着时间变化的振动;但是从频域角度看,却是由不同的音符组成、永恒不变的乐谱,如图 5-1 所示。

2. 傅里叶变换

1807 年,法国数学家傅里叶提出,“任何”周期函数都可以看作不同频率、不同振幅、不同相位正弦波的叠加。可以说,傅里叶变换从复杂信号中将频率分离出来。在时域无法解决的问题转换到频域后便可迎刃而解。在两百多年的时间里,傅里叶变换在通信、自动控制、信号处理、图像处理等多个领域得到广泛的应用。

从时域的角度观察,信号是由一系列不同频率、不同振幅和不同相位的正弦函数叠加而成,表现为振幅随时间的动态变化;从频域角度观察信号,表现为这些正弦函数的频率与振幅、频率与相位的静态

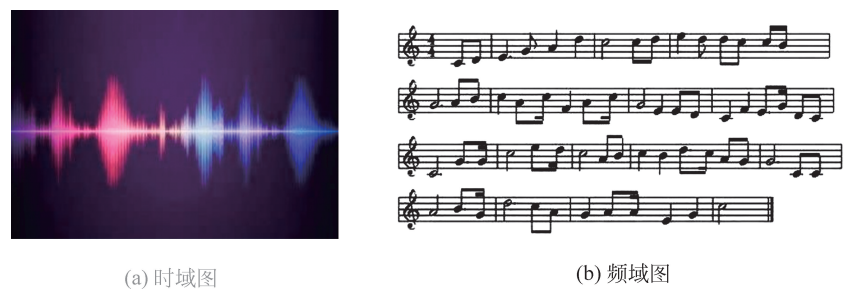


图 5-1 一段音乐的时域图与频域图

关系,如图 5-2 所示。

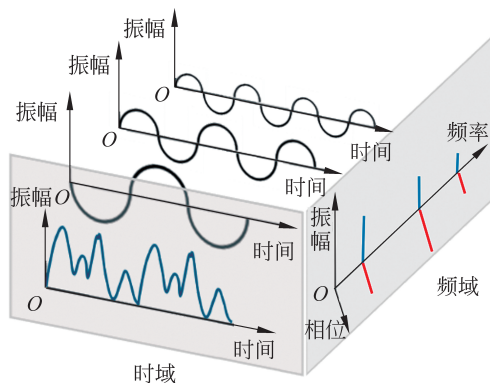


图 5-2 傅里叶变换

从广义上说,信号的某种特征量随信号的频率变化的关系称为信号的频谱,所绘制的图形称为信号的频谱图。其中,描述振幅与频率关系的图形称为振幅频谱图,描述相位与频率关系的图形称为相位频谱图。

下面以男声变女声、女声变男声的变声效果为例,初探傅里叶变换的应用场景。人发音是根据声带振动,推动气体造成标准气压差,进而让声波频率在空气中散播。而男声与女声的不同主要表现在其频率的高低。一般情况下,男声比较低沉,频率较低,女声则比较尖锐,频率较高,因此变声实际上就是在频域中对人的声音频率进行的调整处理,男声变女声需要强化突出尖锐的特点,即调高频率;女声变男声需要强化突出低沉的特点,即调低频率。

傅里叶变换能否推广到二维情形呢? 也就是说,类比于上述一维信号的傅里叶变换,二维图像信号的傅里叶变换是否也可以将一幅图像分解成若干个简单图像之和呢? 的确如此,经过二维傅里叶变换一幅图像被分解成了若干个不同频率、不同振幅、不同相位的正弦平面波之和,如图 5-3 所示。

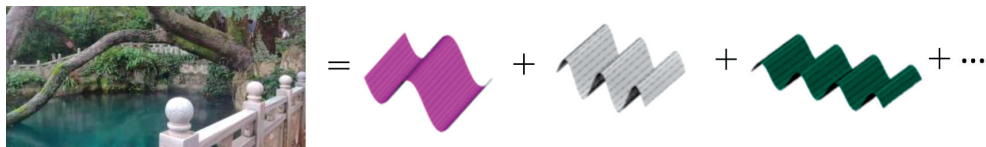


图 5-3 二维傅里叶变换示意图

3. 离散傅里叶变换

由于以计算机为代表的数字处理系统只能存储和处理有限长度的离散数字信号,由此演化出了一维和二维的离散傅里叶变换。对于本书讨论的数字图像,这种离散的二维信号需采用二维离散傅里叶

变换方可从空域切换到频域。

1) 二维频谱图及其物理意义

数字图像经过二维离散傅里叶变换得到了频域中与之对应的频谱图,如图 5-4 所示。居于频谱图中心的是图像的高频信息,四周是低频信息。在实际应用中,为了便于频域的滤波和频谱图的分析,常常在图像处理前进行频谱中心化处理,以使低频信息位于中心。

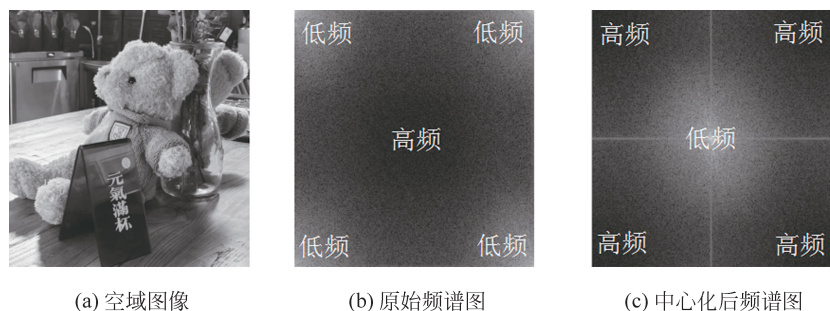


图 5-4 二维频谱图

这些低频和高频信息究竟对应于空域图像中的什么信息呢?下面从频域增强效果进行逆向分析。这里首先解释什么是低通滤波和高通滤波。顾名思义,低通滤波是对低频信息放行,对高频信息拦截;高通滤波是对高频信息放行,对低频信息拦截。结合图 5-5 所示的低通滤波和高通滤波效果可以发现,低通滤波效果等同于空域图像平滑处理,高通滤波效果等同于空域图像锐化处理中需增强的边缘轮廓信息。

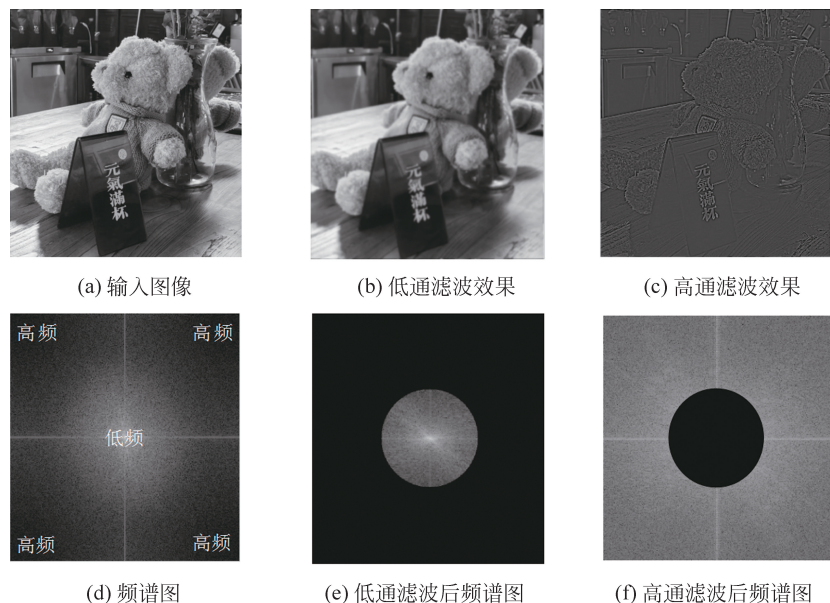


图 5-5 频域增强效果及对应的频谱图

综上所述,可以得出以下结论。

- (1) 频谱图的低频信息对应于空域图像中的平缓区域,即主体部分。
- (2) 频谱图的高频信息对应于空域图像中的突变区域,即边缘、噪声细节部分。

2) 二维离散傅里叶变换的 MATLAB 实现

离散傅里叶变换已经成为数字图像处理的一种重要手段,但是该方法计算量太大、速度太慢。1965

年,J. W. 库利和 T. W. 图基提出了一种改进方法——快速傅里叶变换(fast Fourier transform,FFT),极大地提高了傅里叶变换的速度,也正是 FFT 的出现才使得傅里叶变换得以广泛应用。下面,详细介绍二维快速傅里叶变换的 MATLAB 实现。

(1) MATLAB 相关内置函数。

① 二维快速傅里叶变换函数 fft2()。

语法格式如下:

```
J=fft2(I);
```

其中, J 为频谱图, I 为空域图像,且 J 的大小与 I 相同。

② 频谱中心化函数 fftshift()。

语法格式如下:

```
Z=fftshift(J);
```

其中, Z 为中心化后的频谱图, J 为频谱图。

③ 频谱中心化反变换函数 ifftshift()。

语法格式如下:

```
J1=ifftshift(Z);
```

其中, $J1$ 为反中心化频谱图, Z 为中心化后的频谱图。

④ 二维快速傅里叶反变换函数 ifft2()。

语法格式如下:

```
I1=ifft2(J1);
```

其中, $I1$ 为空域图像, $J1$ 为反中心化频谱图。

(2) 实现代码。下面是以灰度图像为例的实现代码:

```
clear all
[filename,pathname]=uigetfile('*. * ','选取一幅灰度图像');
str=[pathname filename];
src=im2double(imread(str));
subplot(2,2,1),imshow(src),title('输入图像');
J=fft2(src); %二维快速傅里叶变换
subplot(2,2,2),imshow(log(1+abs(J)),[]),title('原始频谱图');
Z=fftshift(J); %频谱中心化
subplot(2,2,3),imshow(log(1+abs(Z)),[]),title('中心化频谱图');
%对频谱 Z 做频域滤波处理(后续在这里添加相应代码)
J=ifftshift(Z); %频谱中心化反变换
result=real(ifft2(J)); %二维快速傅里叶反变换并取实部
subplot(2,2,4),imshow(result),title('输出图像');
```

【代码说明】

- 进行二维快速傅里叶变换前一定要调用 im2double() 函数将输入图像的数据类型由 uint8 转换为 double 类型,否则会因为 unit8 数据类型只能表示 0~255 的整数而出现数据截断,进而出现错误结果。

- 图像的傅里叶频谱的动态范围可宽达 $0 \sim 10^6$ ，直接显示的话显示设备的动态范围往往不能满足要求，此时需要使用对数变换将其压缩到合理范围，即

$$Z = \text{Log}(1 + \text{abs}(Z));$$

- 由于二维快速傅里叶反变换的计算结果是虚部极小或零的复数，而变换回空域的图像像素值为实数，因此需要舍去该复数的虚部。

(3) 实现效果，如图 5-6 所示。

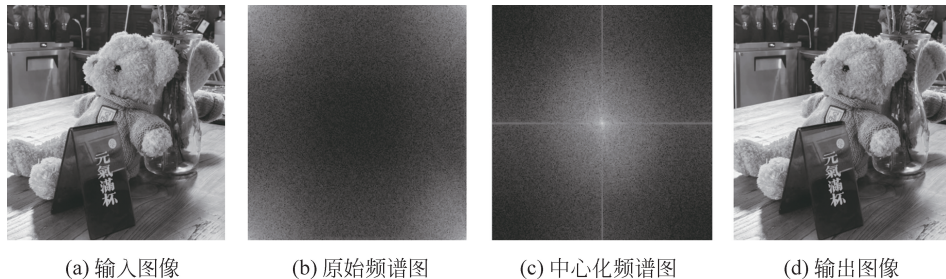


图 5-6 二维快速傅里叶变换的无损转换

5.2 频域增强原理

频域增强实质是通过滤除的频率和保留的频率不同，从而获得不同的增强效果。具体地说，滤除高频、保留低频可达到图像平滑效果；滤除低频、保留高频则可达到图像锐化效果。频域增强的实现流程如图 5-7 所示。具体如下。

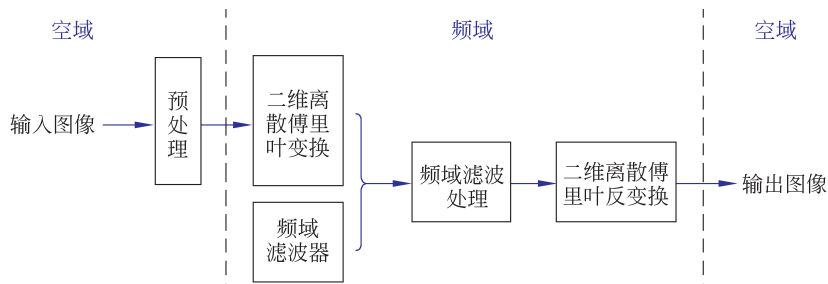


图 5-7 频域增强的实现流程

步骤 1：空域转换频域。预处理后的输入图像经过二维离散傅里叶变换后，从空域变换到频域，计算得到频谱图。

步骤 2：频域增强处理。将中心化频谱图与同尺寸的频域滤波器相乘实现图像的频域滤波处理。

从图 5-8 和图 5-9 所示的频域滤波处理过程可以看出，频域滤波处理实际上是通过中心化频谱图与频域滤波器（等同于二值蒙版）的“点乘”运算完成的。类似于图像乘法运算中的抠图，频域滤波方法也是通过“点乘”运算使得频谱上的某些频率成分得以保留，从而达到期望的增强效果。

结合频谱信息的物理意义，当进行低通滤波后，图像高频信息（图像的边缘、轮廓等突变信息）被滤除，图像低频信息（图像的主体信息）被保留下来，进而达到图像平滑的效果；当进行高通滤波后，图像低频信息（图像的主体信息）被滤除，图像高频信息（图像的边缘、轮廓等突变信息）被保留下来，进而达到图像锐化的效果。

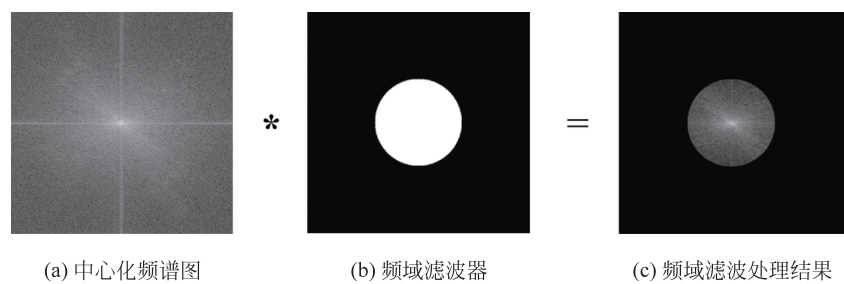


图 5-8 低通滤波处理过程



图 5-9 高通滤波处理过程

【贴士】 频域与空域的内在联系为

$$f * h = FH$$

其中, f 是空域图像, $*$ 是卷积运算, h 是空域中的模板, F 是 f 的傅里叶变换频谱图, H 是频域滤波器。

可见, 傅里叶变换将空域的卷积运算被简化为频域的点乘运算, 这样可以极大地减少计算时间开销, 提高计算速度。

步骤 3: 频域转换空域。对滤波后的频谱图进行二维离散傅里叶反变换从频域变换回空域, 最终得到增强后的输出图像。

5.3 低 通 滤 波

5.3.1 理想低通滤波器

1. 数学定义

理想低通滤波器的数学定义如下:

$$H(u, v) = \begin{cases} 1, & D(u, v) \leq D_0 \\ 0, & D(u, v) > D_0 \end{cases} \tag{5-1}$$

其中参数含义如下。

$D(u, v)$: 频谱图上的点 (u, v) 到频谱中心的欧几里得距离。公式如下:

$$D(u, v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$$

D_0 : 截断频率, 非负整数。 M 、 N 分别为频谱图的高、宽。

从图 5-10 中可知, 满足 $D(u, v) = D_0$ 的点的轨迹为圆。理想低通滤波器是截断圆外的所有频率,

保留圆上和圆内的所有频率。

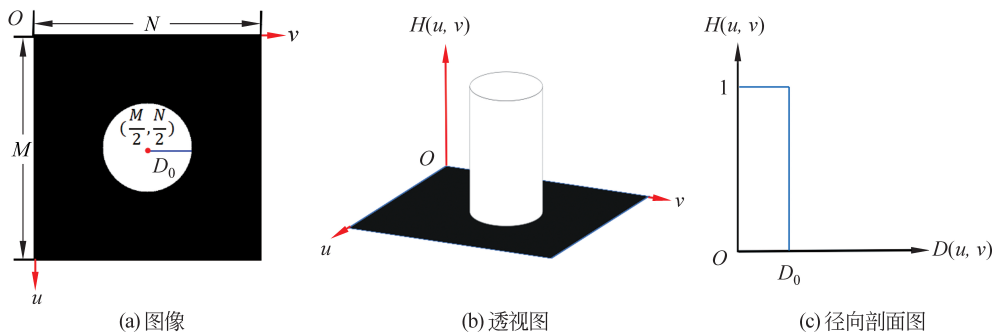


图 5-10 理想低通滤波器示意图

2. 实现代码

理想低通滤波器的实现代码如下：

```
clear all
[filename,pathname]=uigetfile('*. * ','选择图像');
str=[pathname filename];
src=im2double(imread(str));
subplot(1,2,1),imshow(src),title('输入图像');
%设置理想低通滤波器
M=size(src,1);N=size(src,2);
H=zeros(M,N);
D0=str2double(inputdlg('请输入截断频率 D0: ','理想低通滤波'));
for u=1:M
    for v=1:N
        D=sqrt((u-fix(M/2))^2+(v-fix(N/2))^2);
        if D<=D0
            H(u,v)=1;
        end
    end
end
%频域滤波处理
if ismatrix(src)
    %灰度图像
    J=fft2(src);
    Z=fftshift(J);
    G=Z.*H;
    J=ifftshift(G);
    result=real(ifft2(J));
else
    %彩色图像
    r=src(:,:,1);
    g=src(:,:,2);
    b=src(:,:,3);
    %红色通道的频域滤波处理
    Jr=fft2(r);
    Zr=fftshift(Jr);
```

```
Gr=Zr.*H;
Jr=ifftshift(Gr);
result_R=real(ifft2(Jr));
%绿色通道的频域滤波处理
Jg=fft2(g);
Zg=fftshift(Jg);
Gg=Zg.*H;
Jg=ifftshift(Gg);
result_G=real(ifft2(Jg));
%蓝色通道的频域滤波处理
Jb=fft2(b);
Zb=fftshift(Jb);
Gb=Zb.*H;
Jb=ifftshift(Gb);
result_B=real(ifft2(Jb));
%合成滤波处理后的三通道
result=cat(3,result_R,result_G,result_B);
end
subplot(1,2,2),imshow(result),title('理想低通滤波后的图像');
```

3. 实现效果

理想低通滤波效果如图 5-11 所示。



图 5-11 $D_0 = 50$ 的理想低通滤波效果

从图 5-11 的滤波效果上可以明显地看出,理想低通滤波器具有平滑图像的效果,但是也出现了很严重的波纹状“振铃”现象。“振铃”是指输出图像的边缘轮廓或噪声等突变处产生的振荡,就好像钟被敲击后产生的空气振荡。

究其原因,会发现“振铃”现象是由于理想低通滤波器在 D_0 处,即通过频率和滤除频率之间有陡直截断的“不连续性”造成的,并且截断频率 D_0 越小,“振铃”现象就越明显,如图 5-12 所示。

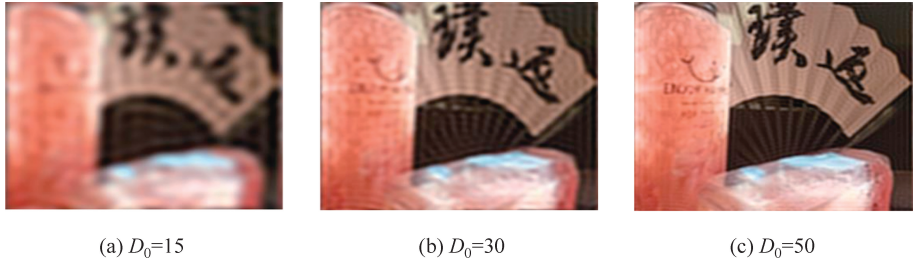


图 5-12 理想低通滤波器的“振铃”现象

5.3.2 巴特沃斯低通滤波器

1. 数学定义

巴特沃斯低通滤波器的数学定义如下：

$$H(u, v) = \frac{1}{1 + \left(\frac{D(u, v)}{D_0} \right)^{2n}} \quad (5-2)$$

其中参数含义如下。

n ：阶数。

$D(u, v)$ ：频谱图上的点 (u, v) 到频谱中心的欧几里得距离。公式如下：

$$D(u, v) = \sqrt{\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2}$$

D_0 ：截断频率，非负整数。 M 、 N 分别为频谱图的高、宽。

从图 5-13 可以看出，与理想低通滤波器不同，巴特沃斯低通滤波器在 D_0 处，即通过频率与被滤除的频率之间没有尖锐的不连续，过渡平坦，因此在一定程度上减弱了“振铃”现象。但是随着阶数 n 的增大，巴特沃斯低通滤波器形状越来越陡峭（图 5-13(c)），越来越趋近于理想低通滤波器，因此“振铃”现象会越来越明显。

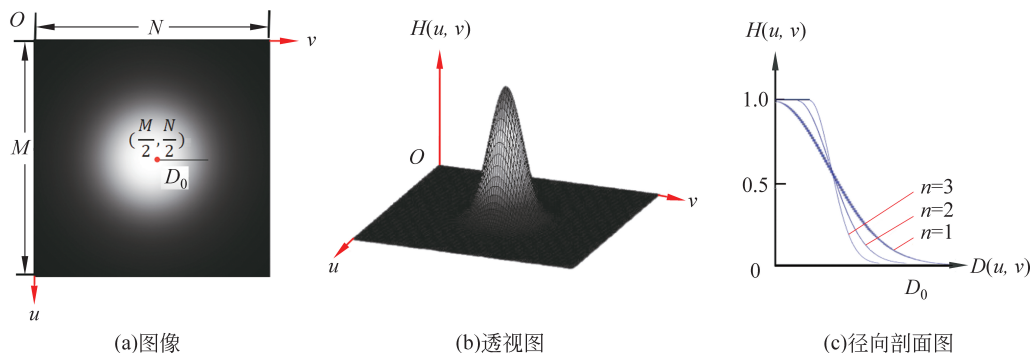


图 5-13 巴特沃斯低通滤波器示意图

2. 实现代码

巴特沃斯低通滤波器的实现代码如下：

```
clear all
[filename, pathname] = uigetfile('*. *', '选择图像');
str = [pathname filename];
src = im2double(imread(str));
subplot(1, 2, 1), imshow(src), title('输入图像');
% 设置巴特沃斯低通滤波器
M = size(src, 1); N = size(src, 2);
H = zeros(M, size(N));
n = str2double(inputdlg('请输入阶数:', '巴特沃斯低通滤波'));
D0 = str2double(inputdlg('请输入截断频率 D0:', '巴特沃斯低通滤波'));
for u = 1:M
    for v = 1:N
```

```

        D=sqrt((u-fix(M/2))^2+(v-fix(N/2))^2);
        H(u,v)=1/(1+(D/D0)^(2*n));
    end
end
%频域滤波处理
if ismatrix(src)
    %灰度图像
    J=fft2(src);
    Z=fftshift(J);
    G=Z.*H;
    J=ifftshift(G);
    result=real(ifft2(J));
else
    %彩色图像
    r=src(:,:,1);g=src(:,:,2);b=src(:,:,3);
    %红色通道的频域滤波处理
    Jr=fft2(r);
    Zr=fftshift(Jr);
    Gr=Zr.*H;
    Jr=ifftshift(Gr);
    result_R=real(ifft2(Jr));
    %绿色通道的频域滤波处理
    Jg=fft2(g);
    Zg=fftshift(Jg);
    Gg=Zg.*H;
    Jg=ifftshift(Gg);
    result_G=real(ifft2(Jg));
    %蓝色通道的频域滤波处理
    Jb=fft2(b);
    Zb=fftshift(Jb);
    Gb=Zb.*H;
    Jb=ifftshift(Gb);
    result_B=real(ifft2(Jb));
    %合成滤波处理后的三通道
    result=cat(3,result_R,result_G,result_B);
end
subplot(1,2,2),imshow(result),title('巴特沃斯低通滤波后的图像');

```

3. 实现效果

如图 5-14 所示,与同一截断频率 D_0 的理想低通滤波效果相比,巴特沃斯低通滤波的“振铃”现象确实有一定程度的减弱。



(a) 输入图像

(b) 理想低通滤波效果

(c) $n=2$ 时巴特沃斯低通滤波效果

图 5-14 $D_0=50$ 的低通滤波效果比较

当阶数 $n=1$ 时,巴特沃斯低通滤波是没有“振铃”现象的;当阶数 $n=2$ 时,巴特沃斯低通滤波处于有效平滑和可接受的振铃特征之间;当阶数 n 不断增大时,“振铃”现象就会越发明显,如图 5-15 所示。