第四次课堂作业——频率域滤波

学号： 201983160037

姓名： 强盛周

班级： 19信计嵌入1班

邮箱： qshengz@foxmail.com

课程名称： 数字图像处理II

授课教师： 陈允杰教授

源代码仓库地址

github: https://github.com/Alephant6/Digital_imaging_processing

gitee: https://gitee.com/qiang-shengzhou/Digital_imaging_processing

一、作业要求

实现以下内容

- 理想低通滤波器
- 布特沃思低通滤波器
- 高斯低通滤波器

二、理论介绍

1. 理想低通滤波器

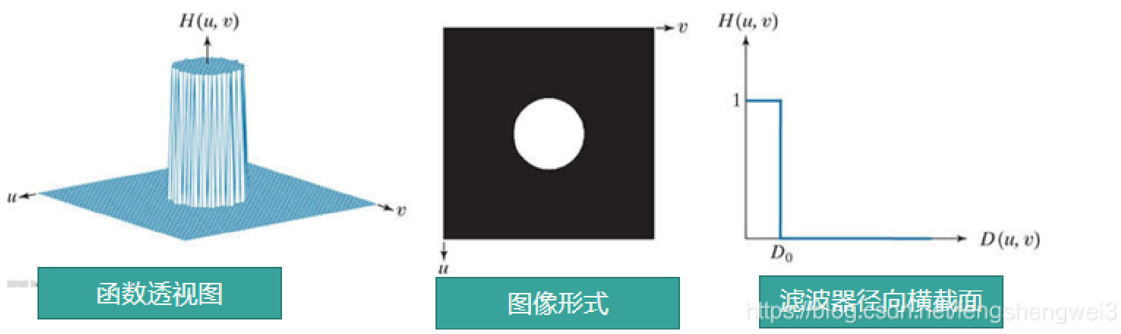
1.1. 定义

定义：

D 是一个正常数，D(u,v)是频率域的点(u,v)到频率矩形中心的距离

$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)\le D_0\\ 0 & \text{if } D(u,v)>D_0\end{cases}$$

$$D(u,v)=\left[(u-P/2)^2+(v-Q/2)^2\right]^{1/2}$$



说明：理想表明在半径为D0的圆内，所有频率无衰减的通过，而在圆外则完全被衰减，它是关于原点径向对称的，也就是说定义一个径向截面，然后旋转360°就可以得到一个理想低通滤波器

1.2. 示例

移除不同比例的高频分量的结果

1.3. 计算给定不同的半径值所过滤掉的功率大小

（1）首先计算总功率
（2）计算半径D0内的所有（u,v)对应的功率之和
（3）除以总功率*100 即可得到百分比

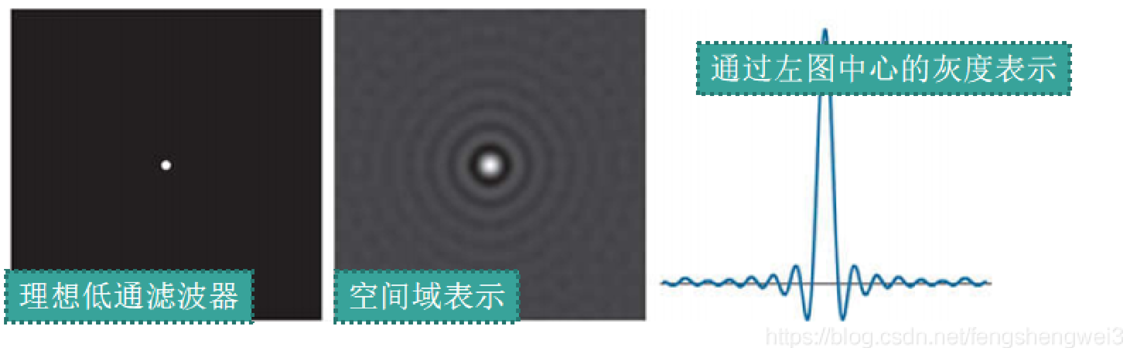
$$P_T = \sum_{u=0}^{P-1} \sum_{v=0}^{Q-1} P(u,v)$$

$$\alpha = 100\left[\sum_u \sum_v P(u,v)/P_T\right]$$

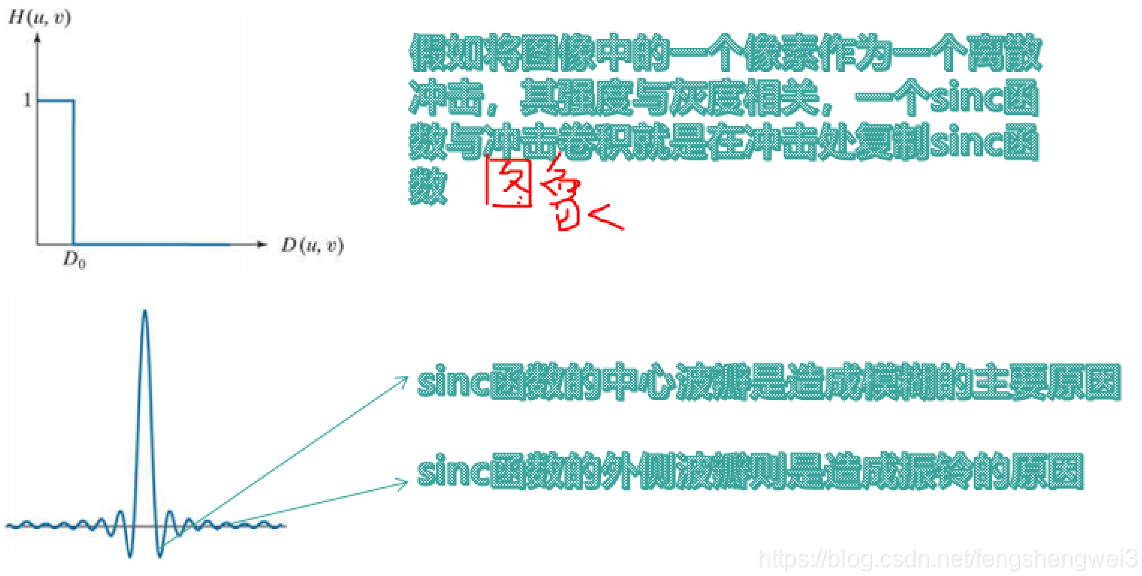
https://blog.csdn.net/fengshengwei3

1.4. 理想低通滤波器的振铃效应

观察下面三幅图，边缘处都有波纹一样的效果，称为"振铃"效应，举个例子，敲锣的时候，会有"翁翁翁"的响声一样 过滤掉的频率越多，振铃效应越明显



说明：由于理想低通滤波器的频率域剖面图类似于盒状滤波器，因此其空间域滤波器有sinc函数形状，空间域的滤波可以用卷积表示。



结论： sinc 函数的展开度与H(u，v)滤波函数的半径成反比，D0越大，sinc函数就会趋近于一个和图像卷积是根本不会发生模糊的冲击，低通滤波的目标是找到没有振铃或振铃效应很小的滤波器

2. 布特沃斯低通滤波器[BLPF]

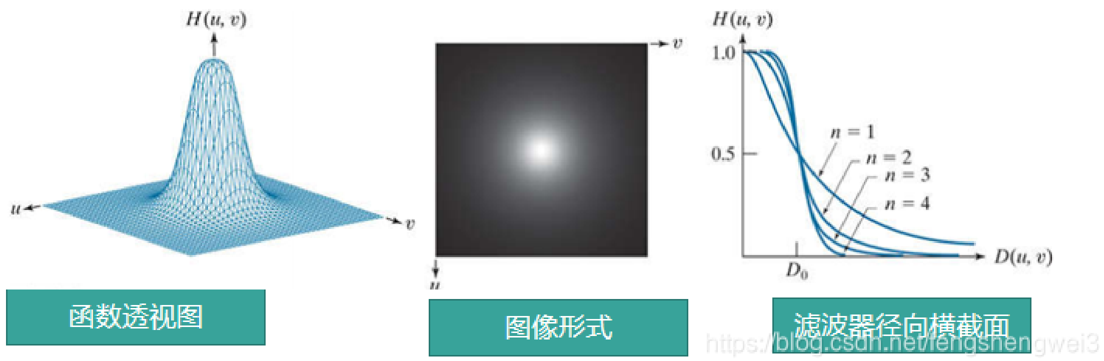
2.1. 定义

定义：

D 是一个正常数，D(u,v)是频率域的点（u,v)到频率矩形中心的距离
D0是截至频率

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

$$D(u,v) = \left[(u - P/2)^2 + (v - Q/2)^2\right]^{1/2}$$

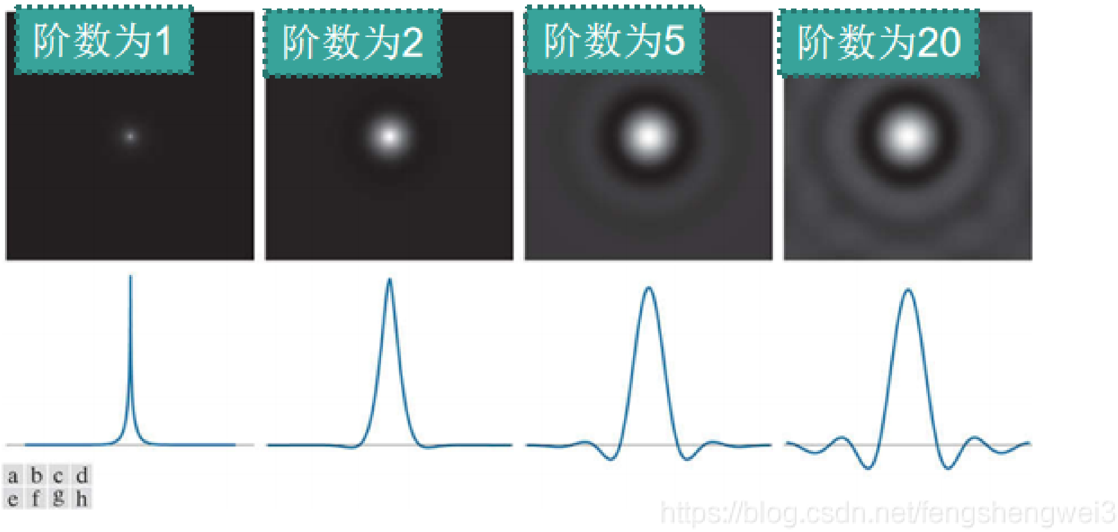


2.2. 如何定义截止频率?

使$H(u，v)$下降为其最大值的某个百分比的点可以作为截止频率

2.3. 振铃效应说明

说明： 巴特沃斯滤波器没有明显的截止频率，它是平滑过渡的，所以一阶情况下不会产生振铃效应 二阶也不会有明显的振铃效应，但是更高阶的振铃效应明显

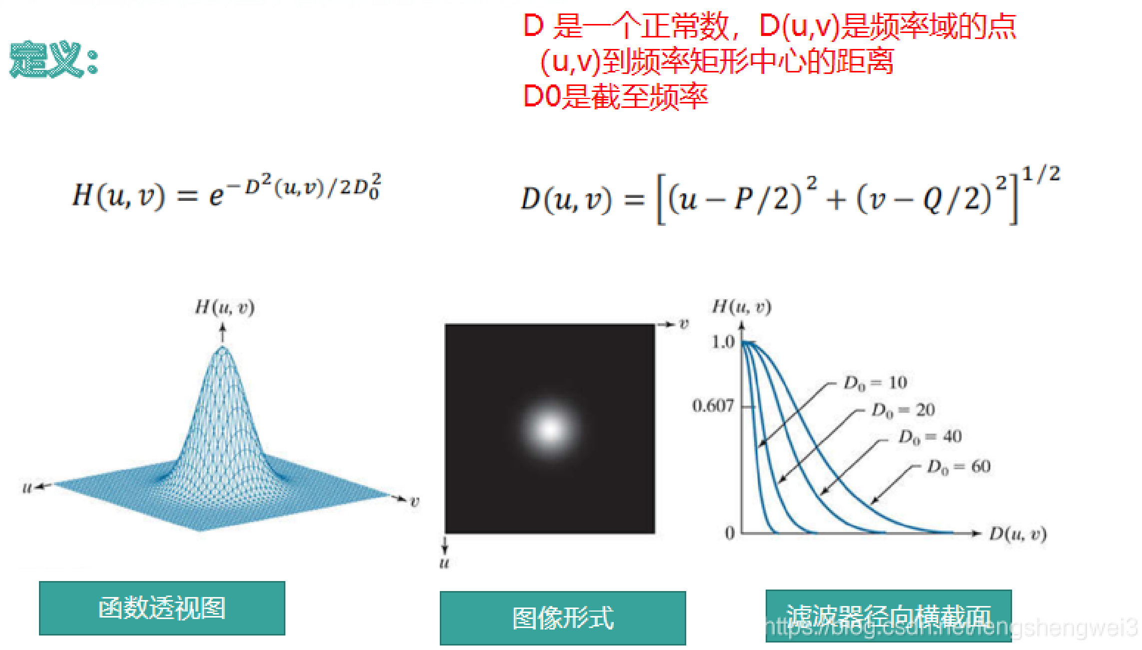


2.4. 举例

不同截止频率对应的滤波结果

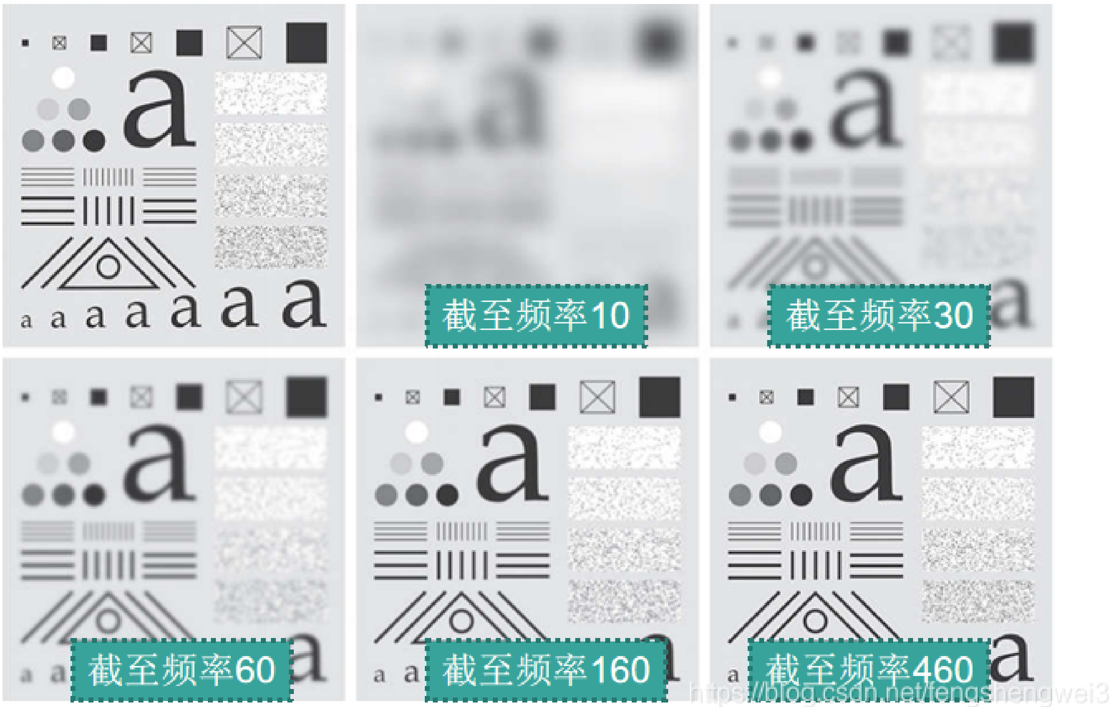
3. 高斯低通滤波器

3.1. 定义



3.2. 示例

不同截止频率对应的滤波结果如下所示



4. 三种低通滤波的区别与联系

Ideal	Gaussian	Butterworth
$H(u,v)=\begin{cases}1 & \text{if } D(u,v)\le D_0\\ 0 & \text{if } D(u,v)>D_0\end{cases}$	$H(u,v)=e^{-D^2(u,v)/2D_0^2}$	$H(u,v)=\frac{1}{1+[D(u,v)/D_0]^{2n}}$
简单直观	没有振铃效应	可以严格控制截止频率在低频和高频之间的过渡
有振铃现象，不适应与正常的应用中	平滑效果比二阶BLPF差	会产生轻微的振铃效应

https://blog.csdn.net/fengshengwei3

三、实验以及结果

1. 导入库

In [1]:
```
# @Author: Alephant——QSZ
import numpy as np
import cv2
import imageio
import matplotlib.pyplot as plt
from math import sqrt
from mpl_toolkits.mplot3d import Axes3D

eps = np.finfo(float).eps
```

2. 绘图

2.1. 绘制三维透视图

In [2]:
```
def drawPerspective(handleax,input_matrix,title=None,cmap = "gray"):
    handleax.set_title(title)
    handleax.set_zlabel('Z')  # 坐标轴
    handleax.set_ylabel('Y')
    handleax.set_xlabel('X')
    x,y = input_matrix.shape
    X = np.arange(0,x,1)
    Y = np.arange(0,y,1)
    # 由于图像x,y坐标和 meshgrid出来是互反的
    # 这里需要调转一下
    # 否则会出现mismatch的现象
    X,Y = np.meshgrid(Y, X)
    handleax.plot_surface(Y, X, input_matrix, cmap=cmap)
#     handleax.plot_wireframe(Y, X, input_matrix, cmap=cmap)
```

2.2. 绘制三维透视图

In [3]:
```
def drawPerspective(handleax,input_matrix,title=None,cmap = "gray"):
    handleax.set_title(title)
    handleax.set_zlabel('Z')  # 坐标轴
    handleax.set_ylabel('Y')
    handleax.set_xlabel('X')
    x,y = input_matrix.shape
    X = np.arange(0,x,1)
    Y = np.arange(0,y,1)
    # 由于图像x,y坐标和 meshgrid出来是互反的
    # 这里需要调转一下
    # 否则会出现mismatch的现象
    X,Y = np.meshgrid(Y, X)
```

```
    handleax.plot_surface(Y, X, input_matrix, cmap=cmap)
#     handleax.plot_wireframe(Y, X, input_matrix, cmap=cmap)
```

2.3. 绘制平面图

In [4]:
```
def drawPanel(handleax,input_matrix,title=None,cmap = "gray"):
    handleax.set_title(title)
    handleax.set_ylabel('Y')
    handleax.set_xlabel('X')
    handleax.imshow(input_matrix,cmap = cmap)
```

2.4. 绘制曲线图

In [5]:
```
def drawCurv(handleax,functions,labels,filter_d0,title=None,cmap = "gray"):
    # 绘制从0到 3D_0的函数剖面图
    handleax.set_title(title)
    handleax.set_xlabel("$D(u,v)$")
    handleax.set_ylabel("$H(u,v)$")
    # 标出D_0点
    # handleax.annotate(r"$D_0$", xy = (filter_d0,0) , weight='heavy')

    for func,lab in zip(functions,labels):
        # 对每一对func和label绘图  (针对需要画多条线的情况)
        X = np.arange(0,3*filter_d0+1,0.1)
        Y = func(X)
        handleax.plot(X,Y,label = lab)
    handleax.set_xticks([0,filter_d0])
    handleax.set_xticklabels(["$0$","$D_0$"])
    handleax.legend()
```

3. 频率与转空间域

In [6]:
```
def frequencyToSpatial(input_matrix):
    # 这里不太明白为什么shift与否最后都需要添加一个fftshift来得到想要的空间域图像
    shift_input_matrix = np.fft.ifftshift(input_matrix)
#     shift_input_matrix = input_matrix
    spatial_img = np.abs(np.fft.ifft2(shift_input_matrix))
    spatial_img = np.fft.fftshift(spatial_img)
    return spatial_img
```

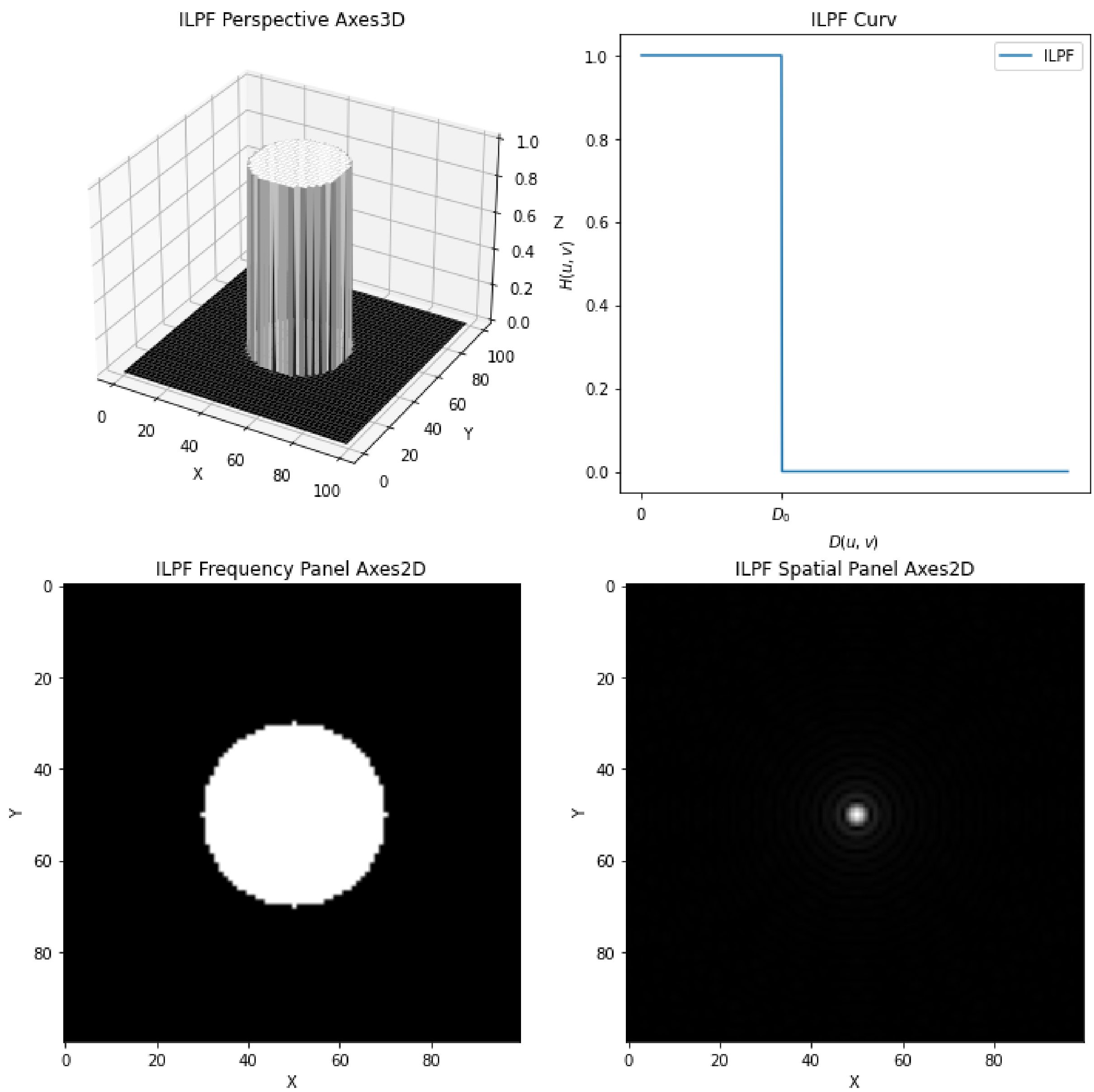
4. 理想滤波器

所谓"理想"是指无法通过硬件实现的硬截断

4.1. 理想低通滤波器 ILPF

In [7]:
```
# 获得指定大小的理想滤波器
def getIdealMask(mask_shape, filter_d0,hl_type):
    assert hl_type in ("lpf","hpf")
    rows,cols = mask_shape[0],mask_shape[1]
    crow = rows/2
    ccol = cols/2
    mask = np.zeros((rows,cols))
    for i in range(rows):
        for j in range(cols):
            dis = sqrt((i-crow)**2 + (j-ccol)**2)
            if hl_type == "lpf":
                if dis <= filter_d0:
                    mask[i,j] = 1
                else:
                    mask[i,j] = 0
            elif hl_type == "hpf":
                if dis <= filter_d0:
```

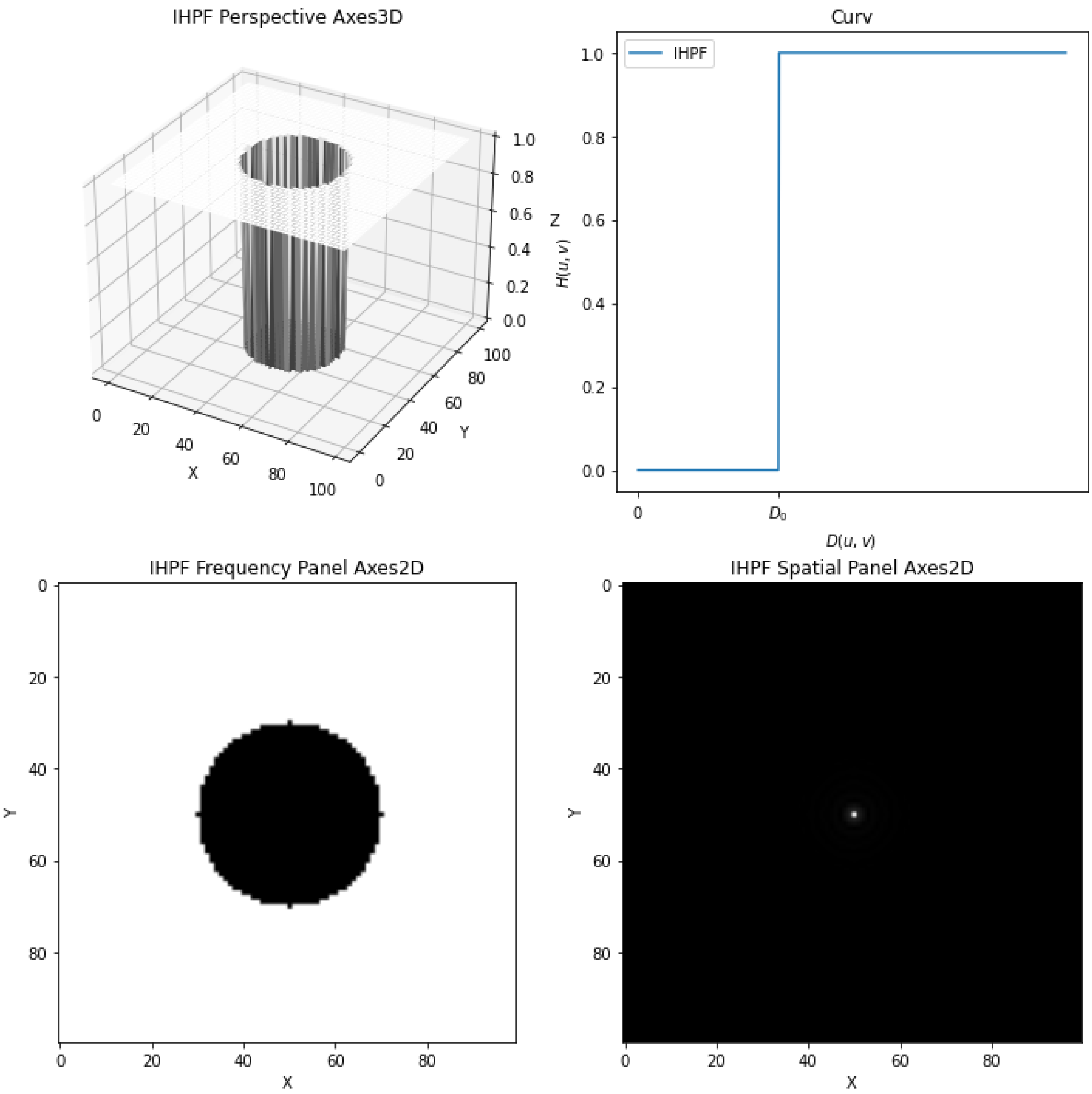
```python
                mask[i,j] = 0
            else:
                mask[i,j] = 1
    return mask

# 测试ILPF
# 参数设置
mask_shape = (100,100)
d = 20
filter_type = "lpf"
# 获得滤波器
myfilter = getIdealMask(mask_shape,d,filter_type)
# 绘图
plt.figure(figsize=(12,12))
ax1=plt.subplot(221,projection = "3d")
ax2=plt.subplot(222)
ax3=plt.subplot(223)
ax4=plt.subplot(224)
drawPerspective(ax1,myfilter,title = "ILPF Perspective Axes3D", cmap = "gray")
# 不想进行列表解析,需要调用frompyfunc构建np可以用的分段函数
ufunc1 = np.frompyfunc(lambda x: 0 if (x-d)>0 else 1, 1, 1)
drawCurv(ax2,[ufunc1],["ILPF"],d,title = "ILPF Curv")
drawPanel(ax3,myfilter,title = "ILPF Frequency Panel Axes2D")
spatial_myfilter = frequencyToSpatial(myfilter)
drawPanel(ax4,spatial_myfilter,title = "ILPF Spatial Panel Axes2D")
plt.show()
```



4.2. 理想高通滤波器 IHPF

In [8]:

```
# 测试IHPF—理想高通滤波器
d = 20
filter_type = "hpf"
myfilter = getIdealMask(mask_shape,d,filter_type)
plt.figure(figsize=(12,12))
ax1=plt.subplot(221,projection = "3d")
ax2=plt.subplot(222)
ax3=plt.subplot(223)
ax4=plt.subplot(224)
drawPerspective(ax1,myfilter,title = "IHPF Perspective Axes3D", cmap = "gray")
# 不想进行列表解析,需要调用frompyfunc构建np可以用的分段函数
ufunc1 = np.frompyfunc(lambda x: 1 if (x-d)>0 else 0, 1, 1)
drawCurv(ax2,[ufunc1],["IHPF"],d,title = "Curv")
drawPanel(ax3,myfilter,title = "IHPF Frequency Panel Axes2D")
spatial_myfilter = frequencyToSpatial(myfilter)
drawPanel(ax4,spatial_myfilter,title = "IHPF Spatial Panel Axes2D")
plt.show()
```



5. 布特沃斯滤波器

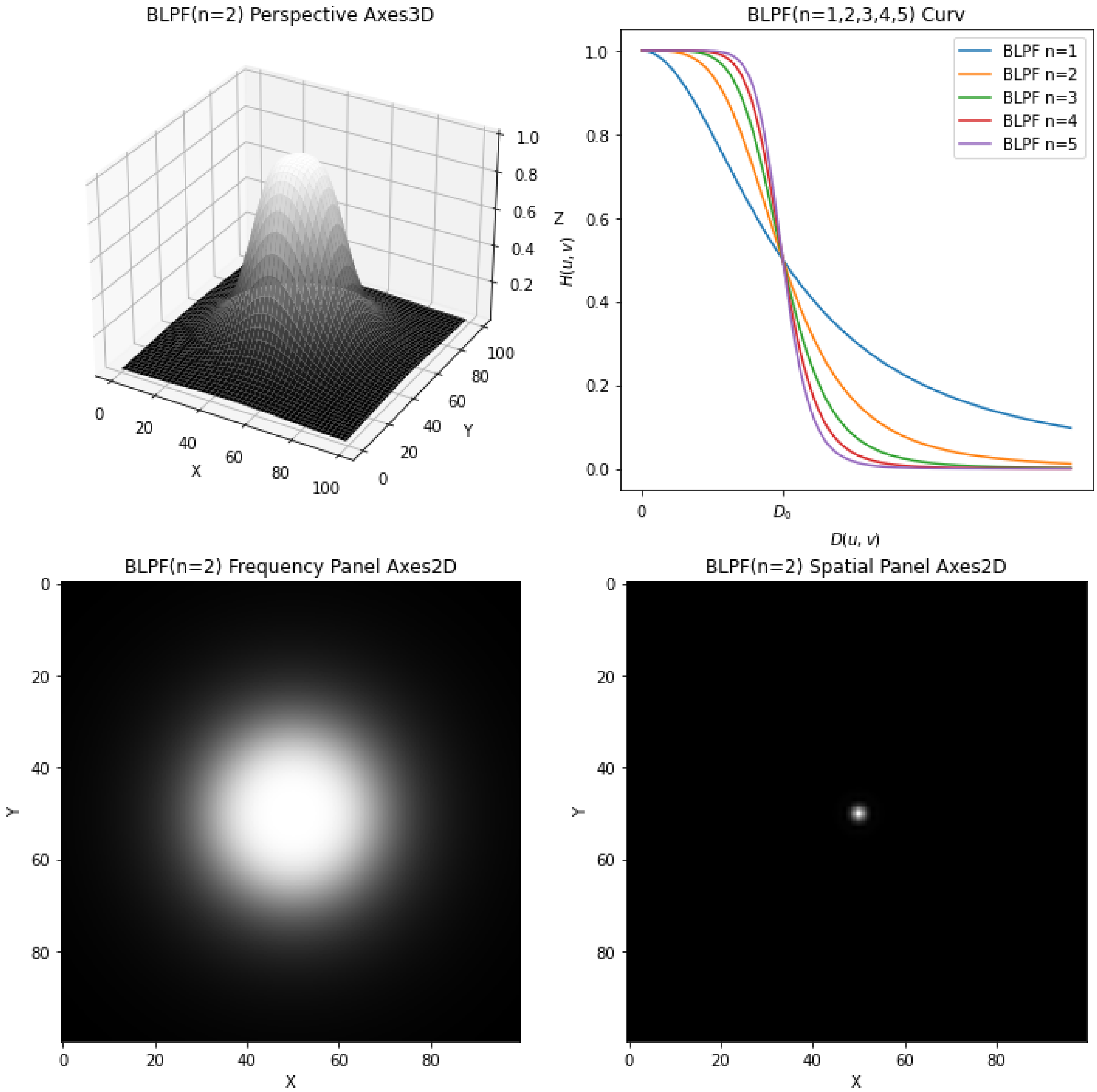
可通过硬件实现，可以通过阶数进行控制，一些资料中又称之为“巴特沃斯滤波器”

5.1. 布特沃斯低通滤波器 BLPF

In [9]:

```
# 获得指定大小的布特沃斯滤波器
```

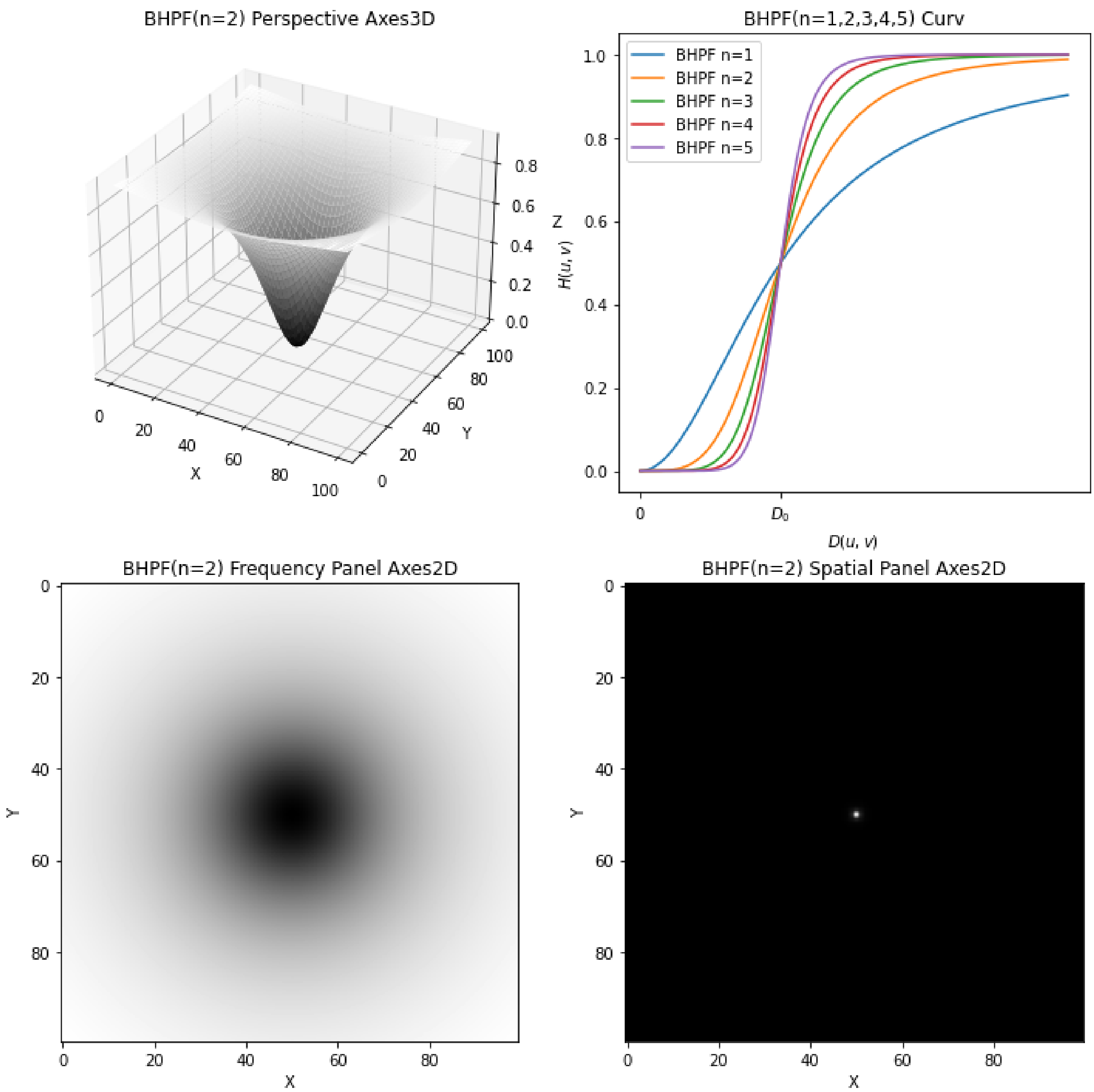
```
def getButterworthMask(mask_shape,filter_d0,hl_type,butter_n = 1):
    assert hl_type in ("lpf","hpf")
    rows,cols = mask_shape[0],mask_shape[1]
    crow = rows/2
    ccol = cols/2
    mask = np.zeros((rows,cols))
    for i in range(rows):
        for j in range(cols):
            dis = sqrt((i-crow)**2 + (j-ccol)**2)
            if hl_type == "lpf":
                mask[i,j] = 1.0/(1+(dis/filter_d0)**(2*butter_n))
            elif hl_type == "hpf":
                # 除以0情况特判一下
                if np.abs(dis)<eps:
                    mask[i,j] = 0
                else:
                    mask[i,j] = 1.0/(1+(filter_d0/dis)**(2*butter_n))

    return mask

# 测试BLPF
# 参数设置
mask_shape = (100,100)
d = 20
filter_type = "lpf"
# 获得滤波器
myfilter = getButterworthMask(mask_shape,d,filter_type, butter_n=2)
# 绘图
plt.figure(figsize=(12,12))
ax1=plt.subplot(221,projection = "3d")
ax2=plt.subplot(222)
ax3=plt.subplot(223)
ax4=plt.subplot(224)
drawPerspective(ax1,myfilter,title = "BLPF(n=2) Perspective Axes3D", cmap = "gray")
funcs = []
labels = []
for i in range(1,6):
    labels.append("BLPF "+"n="+str(i))
funcs.append(lambda x:1.0/(1+(x/d)**(2*1)))
funcs.append(lambda x:1.0/(1+(x/d)**(2*2)))
funcs.append(lambda x:1.0/(1+(x/d)**(2*3)))
funcs.append(lambda x:1.0/(1+(x/d)**(2*4)))
funcs.append(lambda x:1.0/(1+(x/d)**(2*5)))

drawCurv(ax2,funcs,labels,d,title = "BLPF(n=1,2,3,4,5) Curv")
drawPanel(ax3,myfilter,title = "BLPF(n=2) Frequency Panel Axes2D")
spatial_myfilter = frequencyToSpatial(myfilter)
drawPanel(ax4,spatial_myfilter,title = "BLPF(n=2) Spatial Panel Axes2D")
plt.show()
```



5.2. 布特沃斯高通滤波器 BHPF

In [10]:
```
# 测试BHPF
d = 20
filter_type = "hpf"
myfilter = getButterworthMask(mask_shape,d,filter_type, butter_n=1)
plt.figure(figsize=(12,12))
ax1=plt.subplot(221,projection = "3d")
ax2=plt.subplot(222)
ax3=plt.subplot(223)
ax4=plt.subplot(224)
drawPerspective(ax1,myfilter,title = "BHPF(n=2) Perspective Axes3D", cmap = "gray")
funcs = []
labels = []
for i in range(1,6):
#     funcs.append(lambda x:1.0/(1+(d/x)**(2*i)))
#     ufunc = np.frompyfunc(lambda x: 0 if np.abs(x)<eps else 1.0/(1+(d/x)**(2*i)), 1, 1)
#     funcs.append(ufunc)
    labels.append("BHPF "+"n="+str(i))
ufunc1 = np.frompyfunc(lambda x: 0 if np.abs(x)<eps else 1.0/(1+(d/x)**(2*1)), 1, 1)
ufunc2 = np.frompyfunc(lambda x: 0 if np.abs(x)<eps else 1.0/(1+(d/x)**(2*2)), 1, 1)
ufunc3 = np.frompyfunc(lambda x: 0 if np.abs(x)<eps else 1.0/(1+(d/x)**(2*3)), 1, 1)
ufunc4 = np.frompyfunc(lambda x: 0 if np.abs(x)<eps else 1.0/(1+(d/x)**(2*4)), 1, 1)
ufunc5 = np.frompyfunc(lambda x: 0 if np.abs(x)<eps else 1.0/(1+(d/x)**(2*5)), 1, 1)
funcs.append(ufunc1)
funcs.append(ufunc2)
funcs.append(ufunc3)
```

```
funcs.append(ufunc4)
funcs.append(ufunc5)


drawCurv(ax2,funcs,labels,d,title = "BHPF(n=1,2,3,4,5) Curv")
drawPanel(ax3,myfilter,title = "BHPF(n=2) Frequency Panel Axes2D")
spatial_myfilter = frequencyToSpatial(myfilter)
drawPanel(ax4,spatial_myfilter,title = "BHPF(n=2) Spatial Panel Axes2D")
plt.show()
```

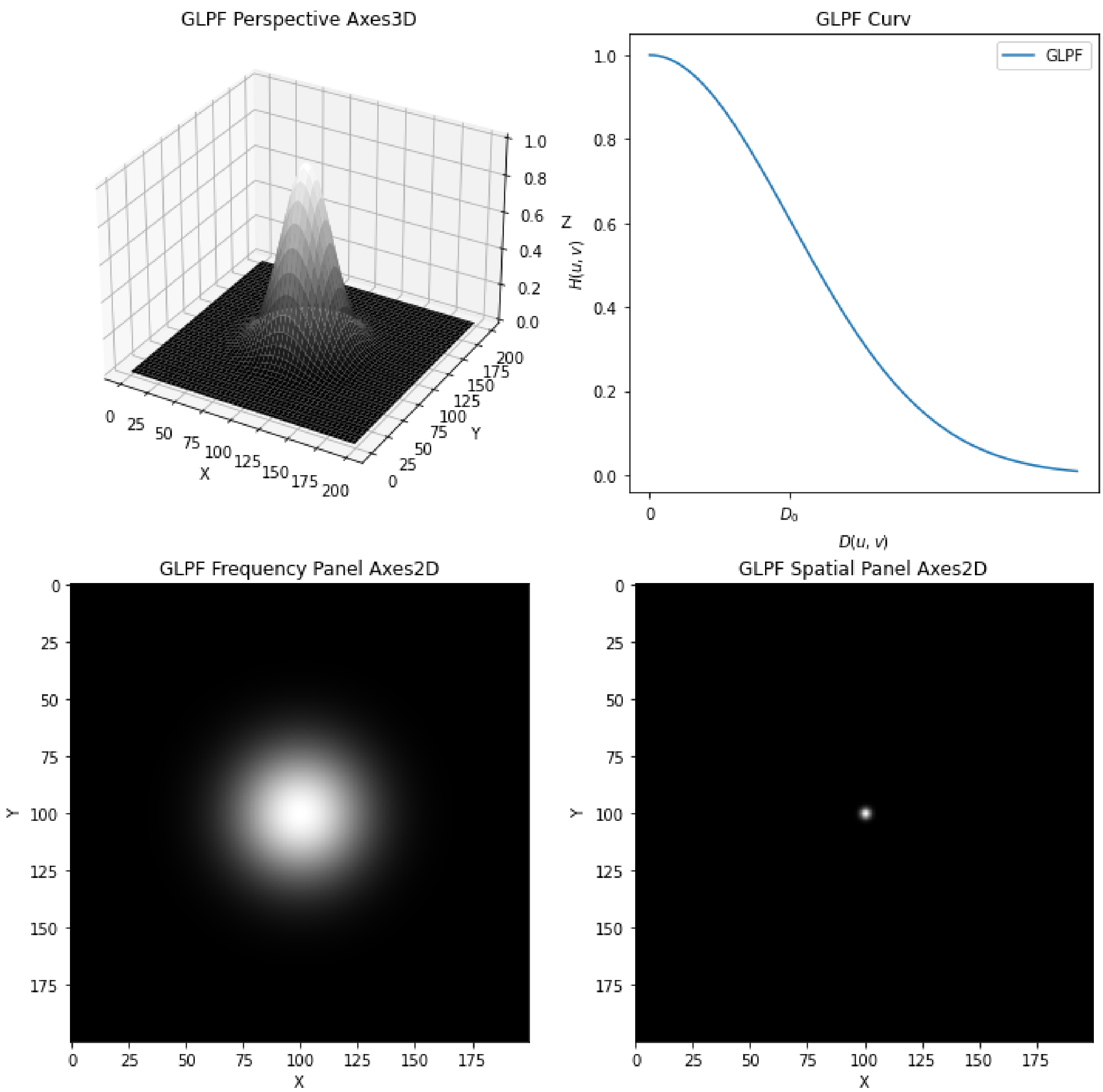


6. 高斯滤波器

6.1. 高斯低通滤波器 GLPF

In [11]:
```
# 获得指定大小的高斯滤波器
def getGaussianMask(mask_shape,filter_d0,hl_type):
    assert hl_type in ("lpf","hpf")
    rows,cols = mask_shape[0],mask_shape[1]
    crow = rows/2
    ccol = cols/2
    mask = np.zeros((rows,cols))
    for i in range(rows):
        for j in range(cols):
            dis = sqrt((i-crow)**2 + (j-ccol)**2)
            if hl_type == "hpf":
                mask[i,j] = 1-np.exp(-(dis**2) / (2*(filter_d0**2)))
```

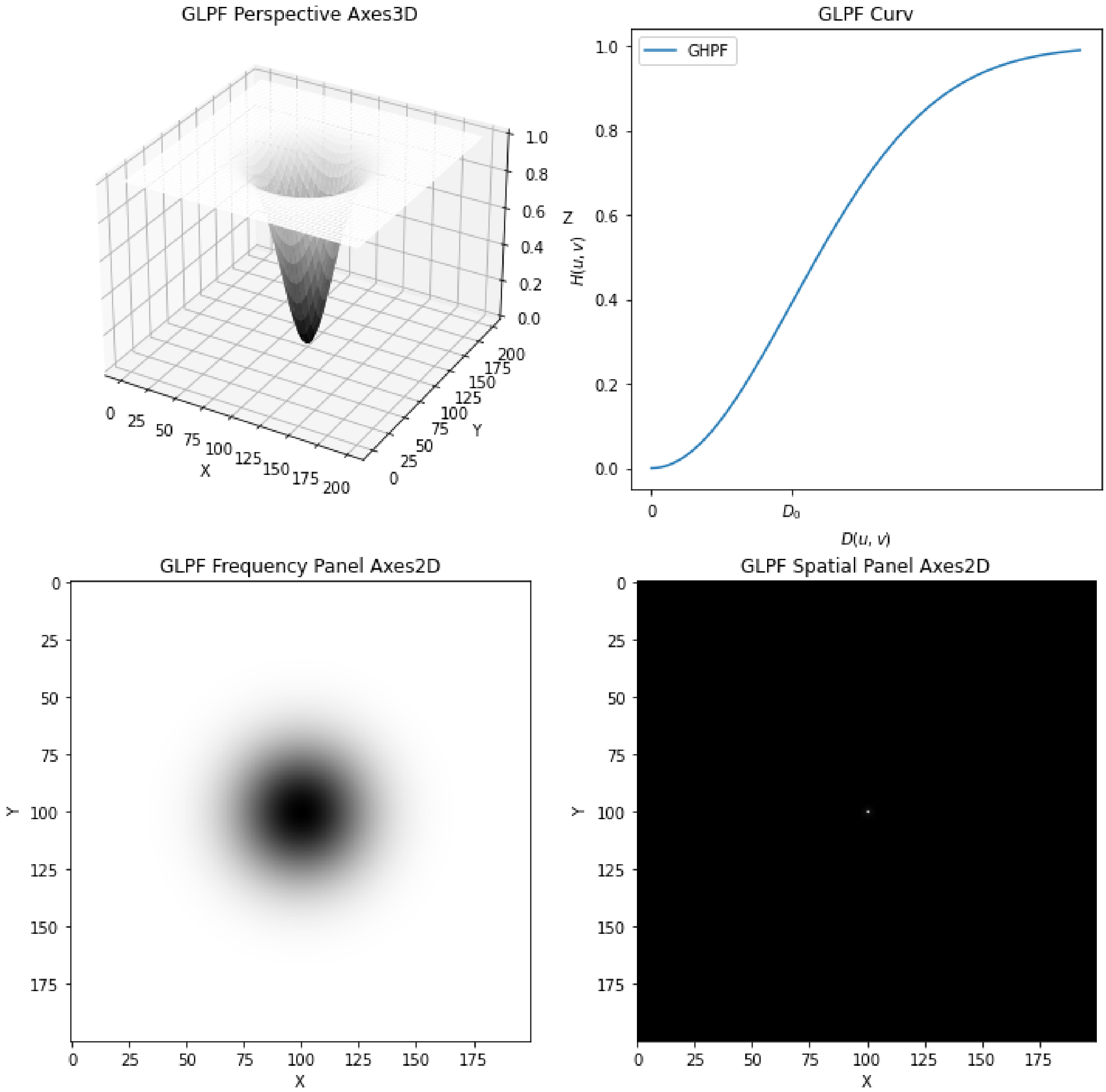
```
            elif hl_type == "lpf":
                mask[i,j] = np.exp(-(dis**2)/(2*(filter_d0**2)))
    return mask

# 测试GLPF
# 参数设置
mask_shape = (200,200)
d = 20
filter_type = "lpf"
# 获得滤波器
myfilter = getGaussianMask(mask_shape,d,filter_type)
# 绘图
plt.figure(figsize=(12,12))
ax1=plt.subplot(221,projection = "3d")
ax2=plt.subplot(222)
ax3=plt.subplot(223)
ax4=plt.subplot(224)
drawPerspective(ax1,myfilter,title = "GLPF Perspective Axes3D", cmap = "gray")
drawCurv(ax2,[lambda x:np.exp(-(x**2)/(2*(d**2)))],["GLPF"],d,title = "GLPF Curv")
drawPanel(ax3,myfilter,title = "GLPF Frequency Panel Axes2D")
spatial_myfilter = frequencyToSpatial(myfilter)
drawPanel(ax4,spatial_myfilter,title = "GLPF Spatial Panel Axes2D")
plt.show()
```



6.2. 高斯高通滤波器 GLPF

In [12]:
```
# 测试GHPF
d = 20
```

```
filter_type = "hpf"
myfilter = getGaussianMask(mask_shape,d,filter_type)
plt.figure(figsize=(12,12))
ax1=plt.subplot(221,projection = "3d")
ax2=plt.subplot(222)
ax3=plt.subplot(223)
ax4=plt.subplot(224)
drawPerspective(ax1,myfilter,title = "GLPF Perspective Axes3D", cmap = "gray")
drawCurv(ax2,[lambda x:1-np.exp(-(x**2)/(2*(d**2)))],["GHPF"],d,title = "GLPF Curv")
drawPanel(ax3,myfilter,title = "GLPF Frequency Panel Axes2D")
spatial_myfilter = frequencyToSpatial(myfilter)
drawPanel(ax4,spatial_myfilter,title = "GLPF Spatial Panel Axes2D")
plt.show()
```



四、参考资料

[1]"Low-pass filter," Wikipedia. Aug. 16, 2021. Accessed: Mar. 22, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Low-pass_filter&oldid=1039066300

[2]"Gaussian filter," Wikipedia. Mar. 14, 2022. Accessed: Mar. 22, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Gaussian_filter&oldid=1077064443

[3]"Frequency domain," Wikipedia. Feb. 11, 2022. Accessed: Mar. 22, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Frequency_domain&oldid=1071136237

[4]“Bessel filter,” Wikipedia. Mar. 11, 2022. Accessed: Mar. 22, 2022. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Bessel_filter&oldid=1076599449

[5]“数字图像处理——频率域平滑锐化图像常用滤波器 - Edward’s blog.” https://www.edwardzcn98yx.com/post/e371c683.html (accessed Mar. 22, 2022).

[6]“8.频率域平滑滤波_MYVision_嗯的博客-CSDN博客_频率域平滑滤波器.” https://blog.csdn.net/fengshengwei3/article/details/100029697 (accessed Mar. 22, 2022).