

作业7：理解频域图像处理

作业内容：

(1) 对附件中的两幅图像分别进行DFT，以 $\log(1+\text{abs}(f))$ 形式显示信号频谱。从空域观察：两张图片中并没有明显的水平和垂直边缘，但从图像频谱显示其包含有强烈的水平与垂直分量，请分析其原因，提出相应的解决办法并进行验证。

(2) 选一张灰度图像，然后顺序进行下列处理操作：

A.对 (x,y) 位置上的像素值乘以 $(-1)^{x+y}$ B.计算图像二维DFT C.对二维DFT的值取共轭 D.对共轭后的频谱做IDFT运算 E.对IDFT的结果取实部 F.对实部乘以 $(-1)^{x+y}$

```
In [9]: import cv2
import numpy as np
import matplotlib.pyplot as plt

def show_frequency_spectrum(image_path, use_hamming=False):
    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    rows, cols = img.shape

    #使用汉明窗
    if use_hamming:
        hamming_window = np.outer(np.hamming(rows), np.hamming(cols))
        img = img * hamming_window

    # 计算DFT
    dft = cv2.dft(np.float32(img), flags=cv2.DFT_COMPLEX_OUTPUT)
    dft_shift = np.fft.fftshift(dft)

    # 计算频谱:  $\log(1 + \text{abs}(f))$ 
    magnitude_spectrum = 20 * np.log(1 + cv2.magnitude(dft_shift[:, :, 0], dft_shift[:, :, 1]))

    return img, magnitude_spectrum

# 显示原始频谱和加窗后频谱
def display_results(image_path):
    # 原始图像的频谱
    img_orig, magnitude_spectrum_orig = show_frequency_spectrum(image_path, use_hamming=False)
    # 加窗后的图像频谱
    img_hamming, magnitude_spectrum_hamming = show_frequency_spectrum(image_path, use_hamming=True)

    # 显示结果对比
    plt.figure(figsize=(12, 6))

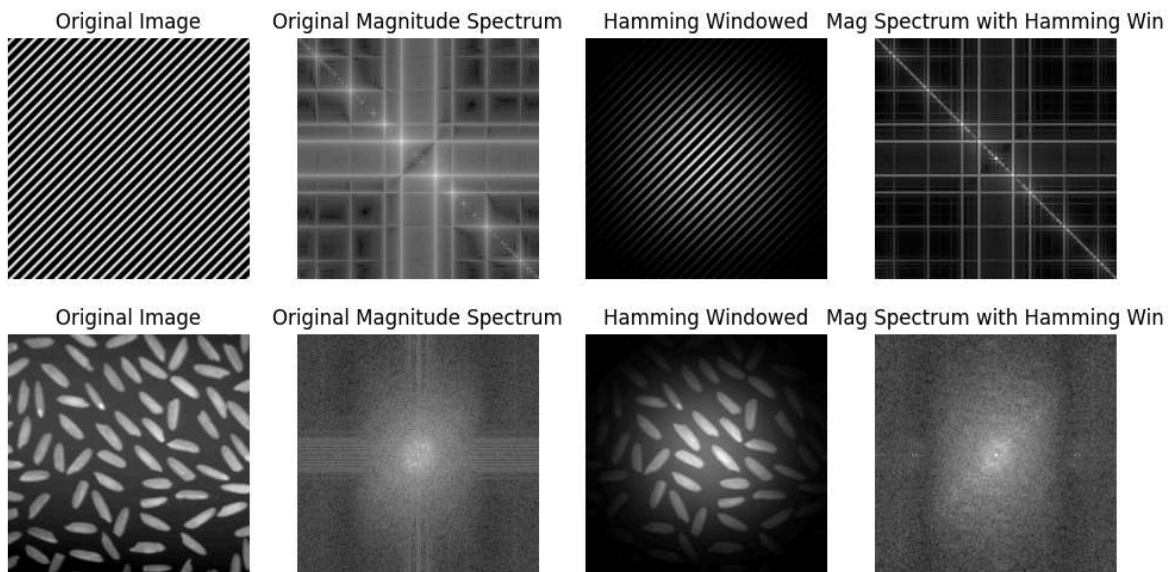
    # 原始图像及其频谱
    plt.subplot(141), plt.imshow(img_orig, cmap='gray')
    plt.title('Original Image'), plt.axis('off')
    plt.subplot(142), plt.imshow(magnitude_spectrum_orig, cmap='gray')
    plt.title('Original Magnitude Spectrum'), plt.axis('off')

    # 加窗图像及其频谱
    plt.subplot(143), plt.imshow(img_hamming, cmap='gray')
    plt.title('Hamming Windowed'), plt.axis('off')
    plt.subplot(144), plt.imshow(magnitude_spectrum_hamming, cmap='gray')
```

```
plt.title('Mag Spectrum with Hamming Win'), plt.axis('off')

plt.show()

# 对两幅图像分别应用上述函数
display_results('line.png')
display_results('rice.png')
```



原因分析

在DFT中，图像被视为离散周期信号。cv2.dft()函数假定图像边缘会与对侧边缘相连（左边与右边、上边与下边）。如果图像内容在边缘处没有平滑过渡，则边缘处的“突变”会在频谱上表现为高频分量，特别是在水平和垂直方向上，造成明显的高频分量。

为何**频谱图水平和垂直分量较强**：可以见[链接](#)，这与傅里叶变换频谱图的纵横“交错”性有关，空间域横向的变化体现在频域的横轴，空间域纵向的变化体现在频域的纵轴

解决

为了减少水平和垂直方向上的高频伪分量，可以使用汉明窗对图像边缘进行平滑过渡。这样能减少频谱中的高频干扰成分。

对图像乘以一个窗函数，相当于在频域引入卷积操作，减少边界的突变来减少频谱中的伪像。由图片也可以看到，汉明窗能使频谱在中心区域更加集中。

```
In [10]: ''' 对灰度图像进行第二个操作 '''

def process_image(image_path):
    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    rows, cols = img.shape

    #A 乘(-1)^(x+y) 进行中心化处理
    img_centered = img * np.array([(-1) ** (x + y) for x in range(rows) for y in range(cols)])

    #B 计算DFT
    dft = cv2.dft(np.float32(img_centered), flags=cv2.DFT_COMPLEX_OUTPUT)

    #C 取共轭
```

```

dft_conjugate = np.conj(dft)

#D IDFT
idft = cv2.idft(dft_conjugate, flags=cv2.DFT_SCALE | cv2.DFT_REAL_OUTPUT)

#E 取实部
idft_real = np.real(idft)

#F 还原频谱位置
img_processed = idft_real * np.array([(-1)**(x+y) for x in range(rows) for y in range(cols)])

# 显示原始图像和处理后的图像
plt.figure(figsize=(10, 5))
plt.subplot(121), plt.imshow(img, cmap='gray')
plt.title('Original Image'), plt.axis('off')
plt.subplot(122), plt.imshow(img_processed, cmap='gray')
plt.title('Processed Image'), plt.axis('off')
plt.show()

process_image('lenna.jpg')

```



对图像频谱乘以 $(-1)^{x+y}$ 是对图像进行中心化处理，即对图像的频谱进行平移操作，使得低频成分移到频谱中心。

$$f_c(x, y) = f(x, y) \cdot (-1)^{x+y}$$

$$F_c(u, v) = \mathcal{F}[f_c(x, y)]$$

接下来对 $F_c(u, v)$ 取共轭，因为傅里叶变换本身具有共轭对称性，取共轭不会改变幅度信息，只会反转相位，也不会体现在空间的图像上。因此逆变换后得到的 $g(x, y)$ 仍然是 $f_c(x, y)$ 的实数版本

$$g_f(x, y) = g(x, y) \cdot (-1)^{x+y} = \mathcal{F}^{-1}[\overline{F_c(u, v)}] \cdot (-1)^{x+y} g_f(x, y) = f(x, y)$$