

## 数字图像处理第三次作业

浮焕然 PB22061345

问题：

1. 计算下图的 DFT, DCT, Hadamard 变换和 Haar 变换

将二维变换为两次一维变换， $T = FAF^T$

$$F_{DFT} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$F_{DCT} = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.65 & 0.27 & -0.27 & -0.65 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.27 & -0.65 & 0.65 & -0.27 \end{bmatrix}$$

$$F_{Hadamard} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$F_{Haar} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

利用公式与变换矩阵计算可得结果如下：

0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0

图 1.原图

2	-1-1j	0	1-1j
0	0	0	0
0	0	0	0
0	0	0	0

图 2.DFT 变换

2	0	-2	0
0	0	0	0
0	0	0	0
0	0	0	0

图 3.DCT 变换

2	0	0	-2
0	0	0	0
0	0	0	0
0	0	0	0

图 4. Hadamard 变换

2	0	$-\sqrt{2}$	$-\sqrt{2}$
0	0	0	0
0	0	0	0
0	0	0	0

图 5.Haar 变换

2. Page 71(章毓晋) 3.21: 设有一组  $64 \times 64$  的图像,它们的协方差矩阵式单位矩阵.如果只使用一半的原始特征值计算重建图像,那么原始图像和重建图像间的均方误差是多少?

对于一组  $64 \times 64$  的图像,协方差矩阵为  $(64 \times 64) \times (64 \times 64)$  大小

又因为协方差矩阵为单位矩阵,因此特征值  $\lambda_j = 1$

由此计算均方误差为:

$$e_{ms} = \sum_{j=1}^{64 \times 64} \lambda_j - \sum_{j=1}^{64 \times 64 / 2} \lambda_j = 2048$$

3. 编程实现 lena.bmp 的离散 Fourier 变换和离散余弦变换,并显示频谱图像。  
代码见最后一页



图 6.lena 原图

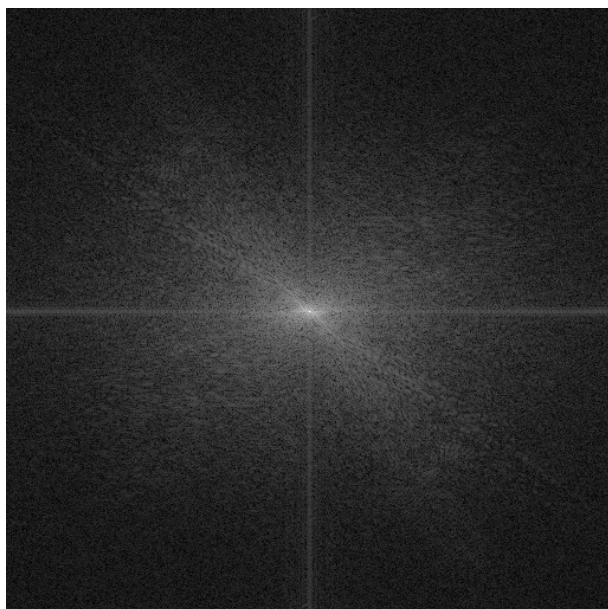


图 7.离散 Fourier 变换



图 8.离散余弦变换

第三题主要代码（省略保存图像）如下：

### 1.DFT 部分：

```
import numpy as np
import matplotlib.pyplot as plt

def create_dft_basis(N):
    """创建 DFT 基函数矩阵"""
    W = np.sqrt(1 / N) * np.exp(-2j * np.pi * np.arange(N)[:, None] * np.arange(N) / N)
    return W

def dft_2d(image):
    """对图像进行二维 DFT 变换"""
    M, N = image.shape
    W = create_dft_basis(N)
    # 对每一行应用 DFT
    dft_rows = np.dot(W, image)
    dft_cols = np.dot(dft_rows, W.T)
    return dft_cols

# 读取图像并转换为灰度
image_path = 'Lena.bmp'
image = plt.imread(image_path)
# 应用 DFT
dft_image = dft_2d(image)
dft_image_shifted = np.fft.fftshift(dft_image) # 将零频分量移到频谱中心
magnitude_spectrum = np.log(np.abs(dft_image_shifted) + 1) # 加 1 避免对数为负无穷
```

### 2.DCT 部分：

```
import numpy as np
import matplotlib.pyplot as plt

def create_dct_basis(N):
    """创建 DCT 基函数矩阵"""
    C = np.zeros((N, N))
    for u in range(N):
        for x in range(N):
            if u == 0:
                C[u, x] = 1 / np.sqrt(2 * N)
            else:
                C[u, x] = np.sqrt(2 / N) * np.cos(np.pi * (2 * x + 1) * u / (2 * N))
    return C

def dct_2d(image):
    """对图像进行二维 DCT 变换"""
    M, N = image.shape
    C = create_dct_basis(N)
```

```
dct_rows = np.dot(C, image)
dct_cols = np.dot(dct_rows, C.T)
return dct_cols

image_path = 'Lena.bmp'
image = plt.imread(image_path)
# 应用 DCT
dct_image = dct_2d(image)
dct_image = np.log(np.abs(dct_image) + 1) # 加 1 避免对数为负无穷
```