**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，

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 |  |  |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |

图 5.Haar变换

1. Page 71(章毓晋) 3.21: 设有一组64\*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\*64的图像，协方差矩阵为（64\*64）\*（64\*64）大小

又因为协方差矩阵为单位矩阵，因此特征值

由此计算均方误差为：

1. 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

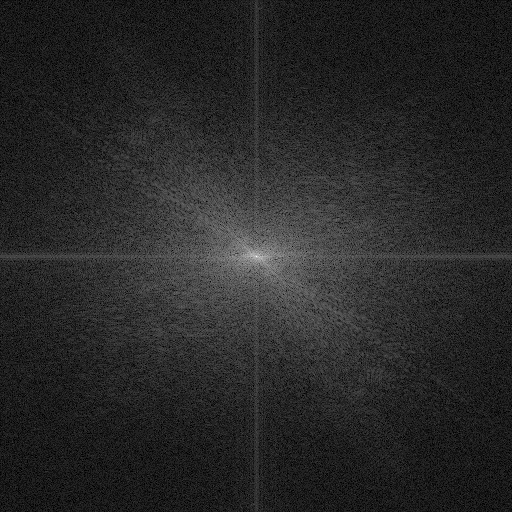


图 7.离散Fourier变换



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