

信息隐藏技术实验报告

第三次作业：图像信号常用处理方法

2113662 张丛

实验内容

离散傅里叶变换

原理

实验

离散小波变换

原理

实验

离散余弦变换

原理

实验

小结

实验内容

1. DFT（离散傅里叶变换）
2. DWT（离散小波变换）
3. DCT（离散余弦变换）

离散傅里叶变换

原理

- 离散傅立叶变换是一种将信号从时间域转换为频率域的技术。它通过将信号分解成一系列正弦和余弦函数的组合来完成这一转换。
- 在实践中，DFT 被广泛用于信号处理、频谱分析、滤波器设计等方面。它在数字通信、音频处理、图像处理等领域中都有应用。

正变换

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

$$u = 0, 1, \dots, M-1; \quad v = 0, 1, \dots, N-1$$

反变换

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

$$x = 0, 1, \dots, M-1; \quad y = 0, 1, \dots, N-1$$

实验

代码如下：

% 读取图像

```
originalImage = imread('cat.jpg');
```

% 转换为灰度图像

```
grayImage = rgb2gray(originalImage);
```

```
figure;
```

```
imshow(grayImage);
```

% 二值化处理

```
threshold = graythresh(grayImage);
```

```
binaryImage = imbinarize(grayImage, threshold);
```

% 对二值化后的图像应用二维快速傅里叶变换 `fft2`，并使用 `fftshift` 函数将零频分量移到频谱中心

```
fftImage = calculateFFT(binaryImage);
```

% 显示变换后的幅度谱

```
showAmplitudeSpectrum(fftImage);
```

% 绘制幅度谱的三维能量分布图

```
plot3DEnergyDistribution(fftImage);
```

% 计算二维快速傅里叶变换并移动零频分量到频谱中心

```
function fftImage = calculateFFT(image)
```

```
    fftImage = fftshift(fft2(image));
```

```
end
```

% 显示变换后的幅度谱

```
function showAmplitudeSpectrum(fftImage)
```

```
    amplitudeSpectrum = abs(fftImage);
```

```
    figure;
```

```
    imshow(log(1 + amplitudeSpectrum), []);
```

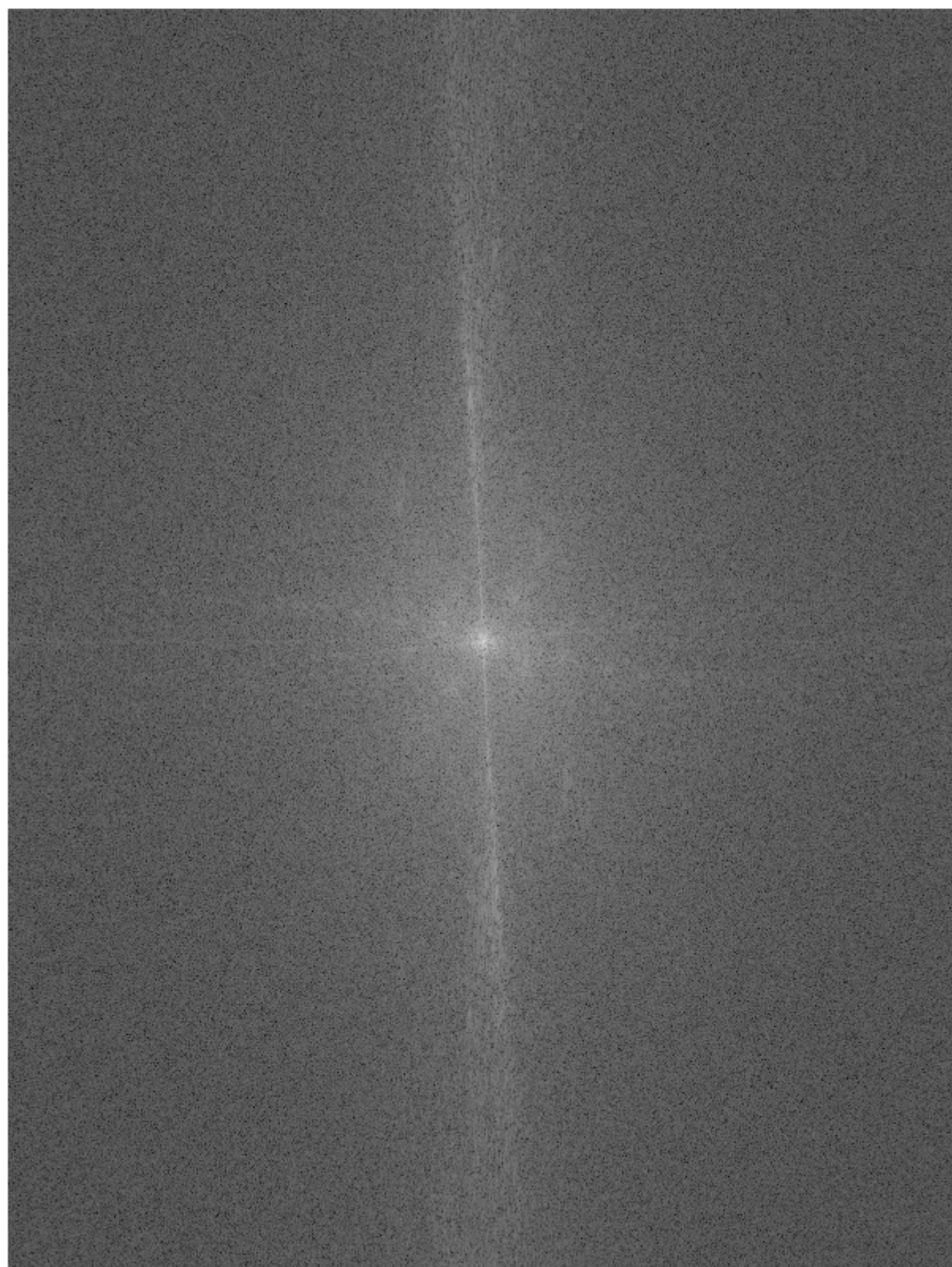
```
end
```

% 绘制幅度谱的三维能量分布图

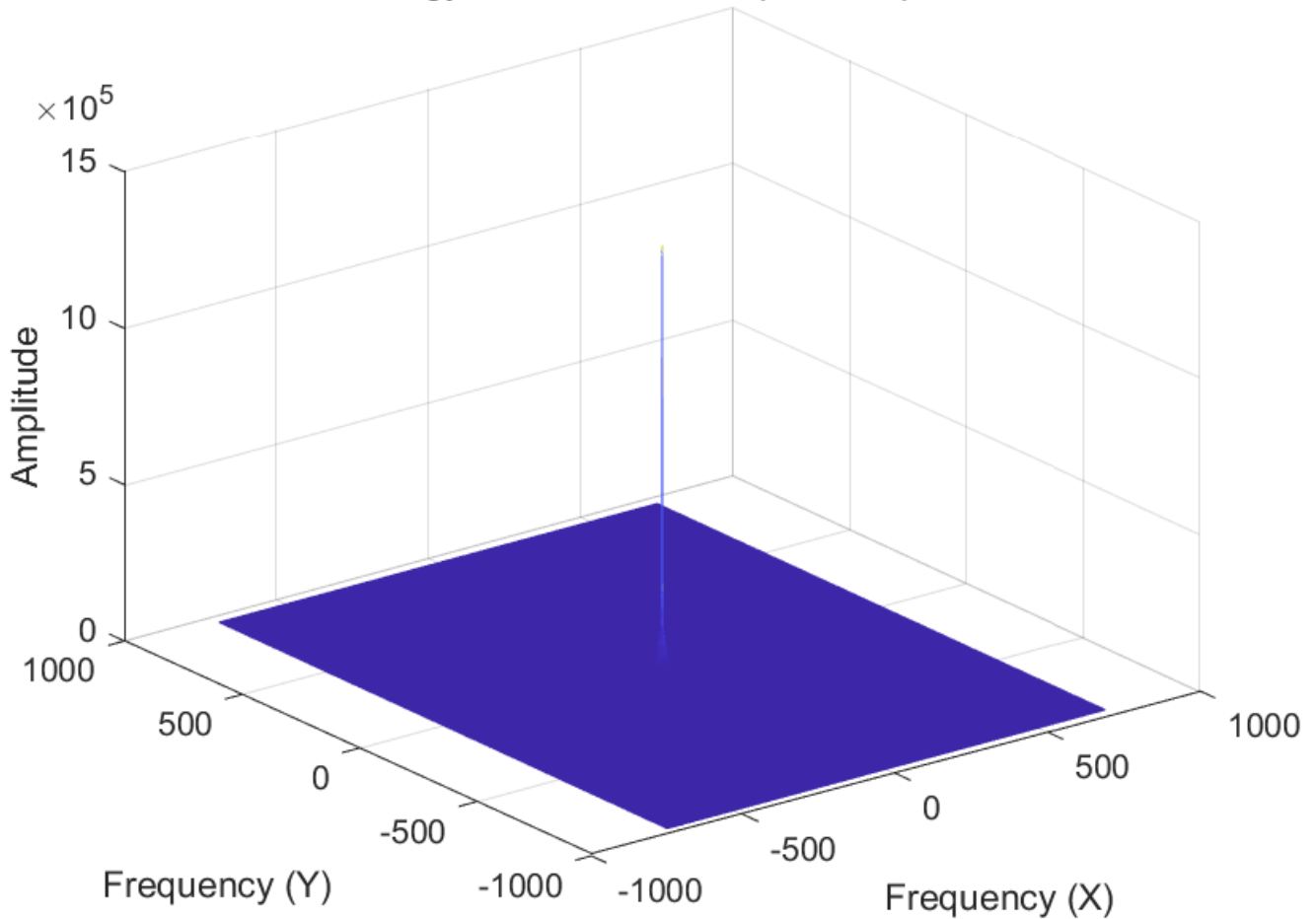
```
function plot3DEnergyDistribution(fftImage)
    amplitudeSpectrum = abs(fftImage);
    [M, N] = size(amplitudeSpectrum);
    [X, Y] = meshgrid(-N/2:N/2-1, -M/2:M/2-1);
    figure;
    mesh(X, Y, amplitudeSpectrum);
    xlabel('Frequency (X)');
    ylabel('Frequency (Y)');
    zlabel('Amplitude');
    title('3D Energy Distribution of Amplitude Spectrum');
end
```

结果如下:





3D Energy Distribution of Amplitude Spectrum



离散小波变换

原理

- 离散小波变换是一种将信号分解成不同尺度和频率的小波基函数的技术。与傅里叶变换不同，小波变换能够提供更好的局部化信息。
- DWT 在图像压缩、信号去噪、特征提取等领域中被广泛应用。它的局部化特性使得它在处理非平稳信号时效果更好。

实验

一级小波变换：

```
% 读取图像
b = imread("cat.jpg");

% 转换为灰度图像
a = rgb2gray(b);

% 获取图像的行数
nbc01 = size(a, 1);

% 进行一级二维小波变换
[ca1, ch1, cv1, cd1] = dwt2(a, 'db4');
```

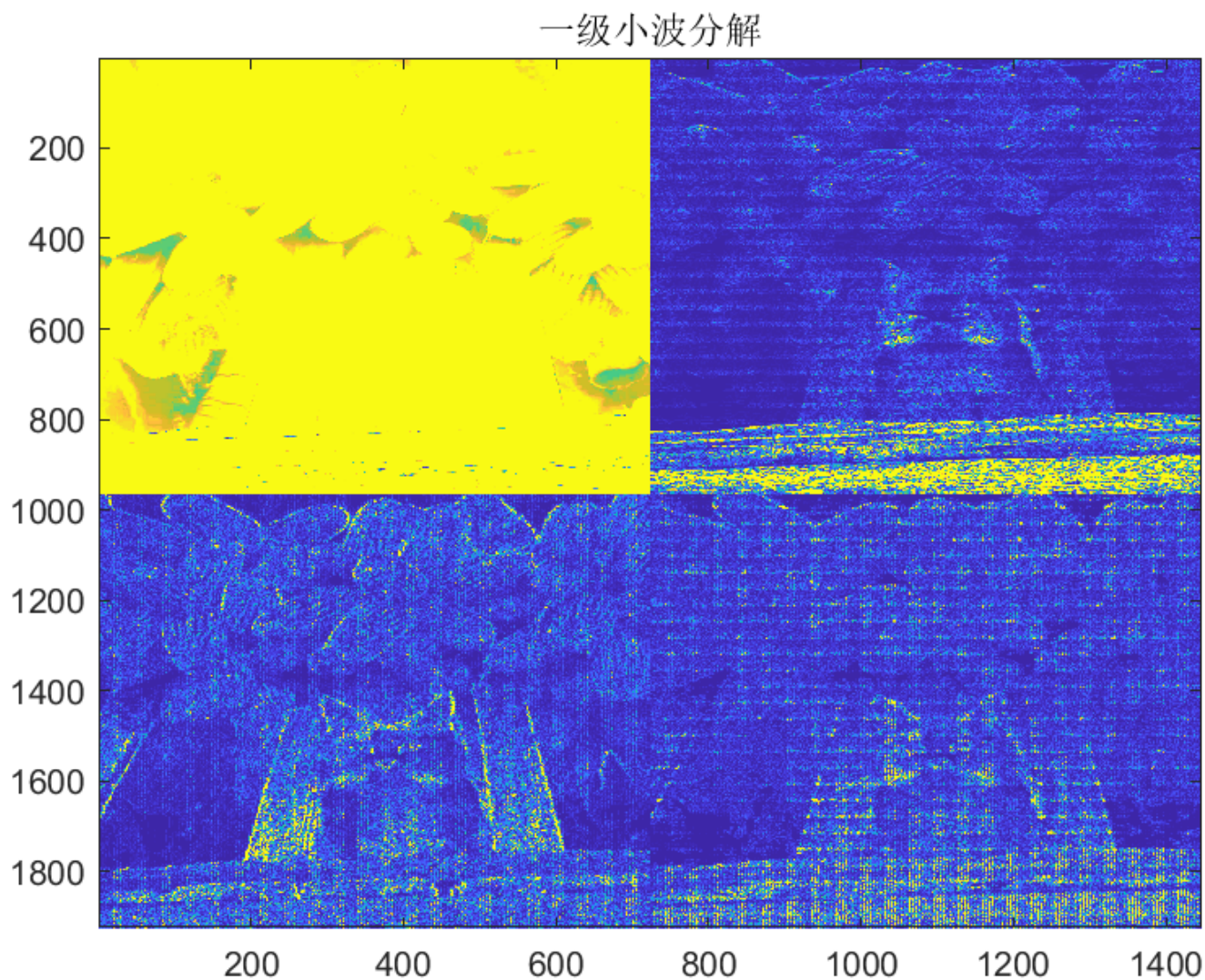


```

% 对小波系数进行可视化处理
cod_ca1 = wcodemat(ca1, nbcol);
cod_ch1 = wcodemat(ch1, nbcol);
cod_cv1 = wcodemat(cv1, nbcol);
cod_cd1 = wcodemat(cd1, nbcol);

% 显示一级小波分解结果
figure;
image([cod_ca1, cod_ch1; cod_cv1, cod_cd1]);
title('一级小波分解');

```



二级小波变换:

```

% 读取图像
b = imread("111.jpg");

% 转换为灰度图像
a = rgb2gray(b);

% 获取图像的行数
nbc = size(a, 1);

% 进行一级二维小波变换

```



```

[ca1, ch1, cv1, cd1] = dwt2(a, 'db4');

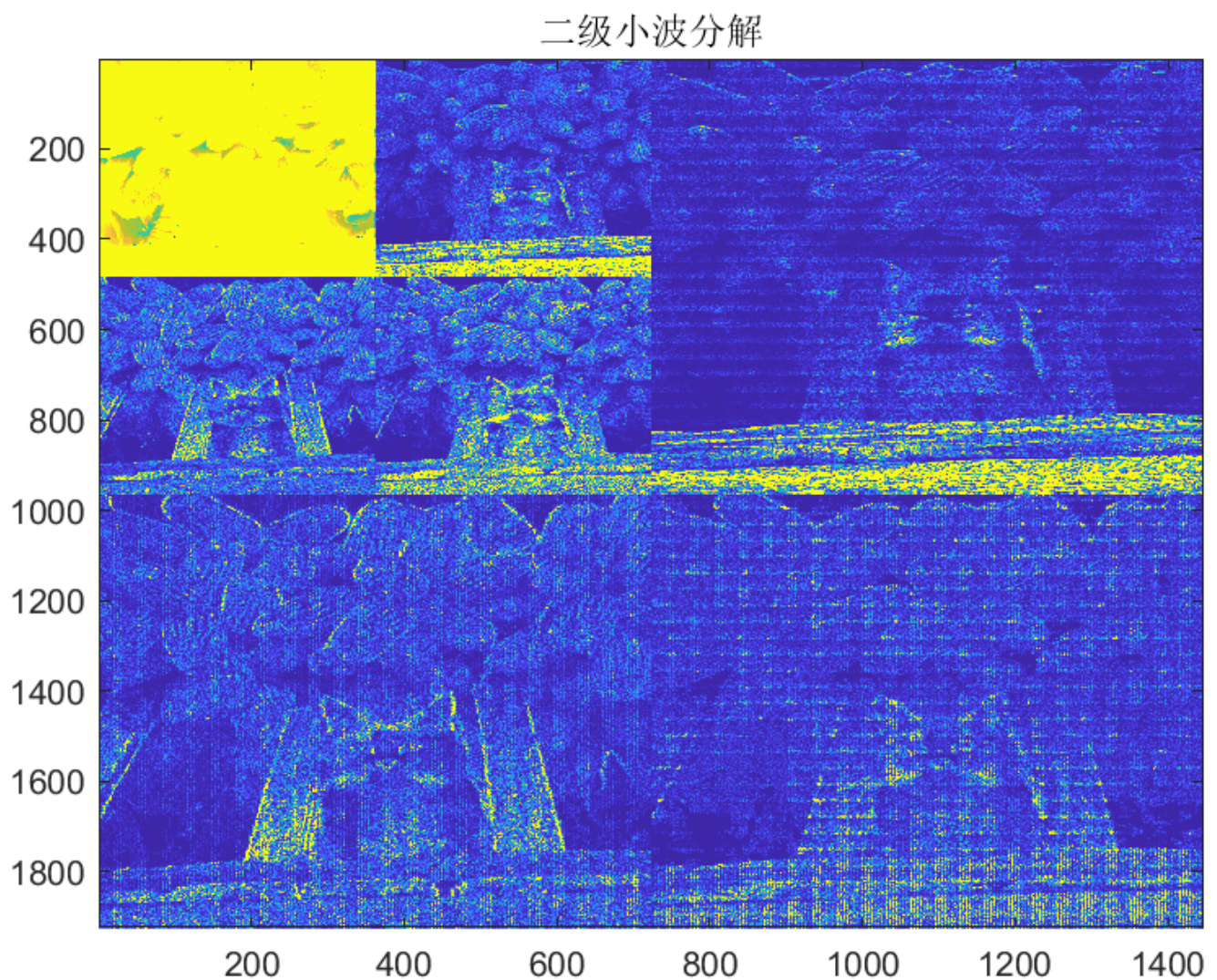
% 对第一级小波系数进行二级小波变换
[ca2, ch2, cv2, cd2] = dwt2(ca1, 'db4');

% 对小波系数进行可视化处理
cod_ca1 = wcodemat(ca1, nbc);
cod_ch1 = wcodemat(ch1, nbc);
cod_cv1 = wcodemat(cv1, nbc);
cod_cd1 = wcodemat(cd1, nbc);
cod_ca2 = wcodemat(ca2, nbc);
cod_ch2 = wcodemat(ch2, nbc);
cod_cv2 = wcodemat(cv2, nbc);
cod_cd2 = wcodemat(cd2, nbc);

% 对第二级近似系数进行尺寸调整以匹配第一级近似系数的大小
tt = imresize([cod_ca2, cod_ch2; cod_cv2, cod_cd2], size(ca1));

% 显示二级小波分解结果
figure;
image([tt, cod_ch1; cod_cv1, cod_cd1]);
title('二级小波分解');

```



离散余弦变换

原理

- 离散余弦变换是一种类似于傅里叶变换的变换技术，但它主要用于将信号从空间域转换为频率域，而不是时间域。
- DCT 在图像和音频压缩中经常被使用，特别是在 JPEG 图像压缩和 MP3 音频压缩中。它的优点之一它能够更好地集中信号的能量在较少的系数上，从而实现更高的压缩比。

实验

代码如下：

```
% 读取图像并转换为灰度图像
b = imread("cat.jpg");
b = rgb2gray(b);

% 显示原图像
figure;
imshow(b);
title('(a)原图像');

% 对灰度图像进行二值化处理
I = imbinarize(b);

% 进行离散余弦变换
c = dct2(I);

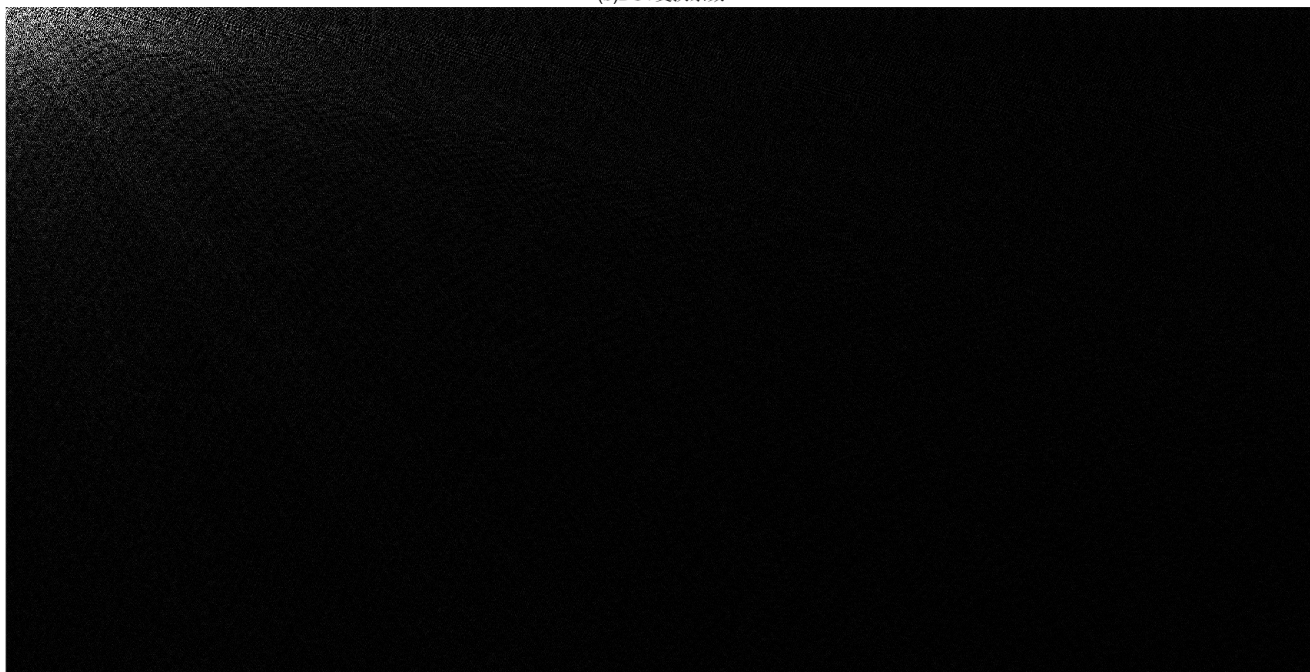
% 显示DCT变换系数
figure;
imshow(c, []);
title('(b)DCT变换系数');

% 绘制DCT变换系数的立体视图
figure;
mesh(c);
title('(c)DCT变换系数（立体视图）');
```

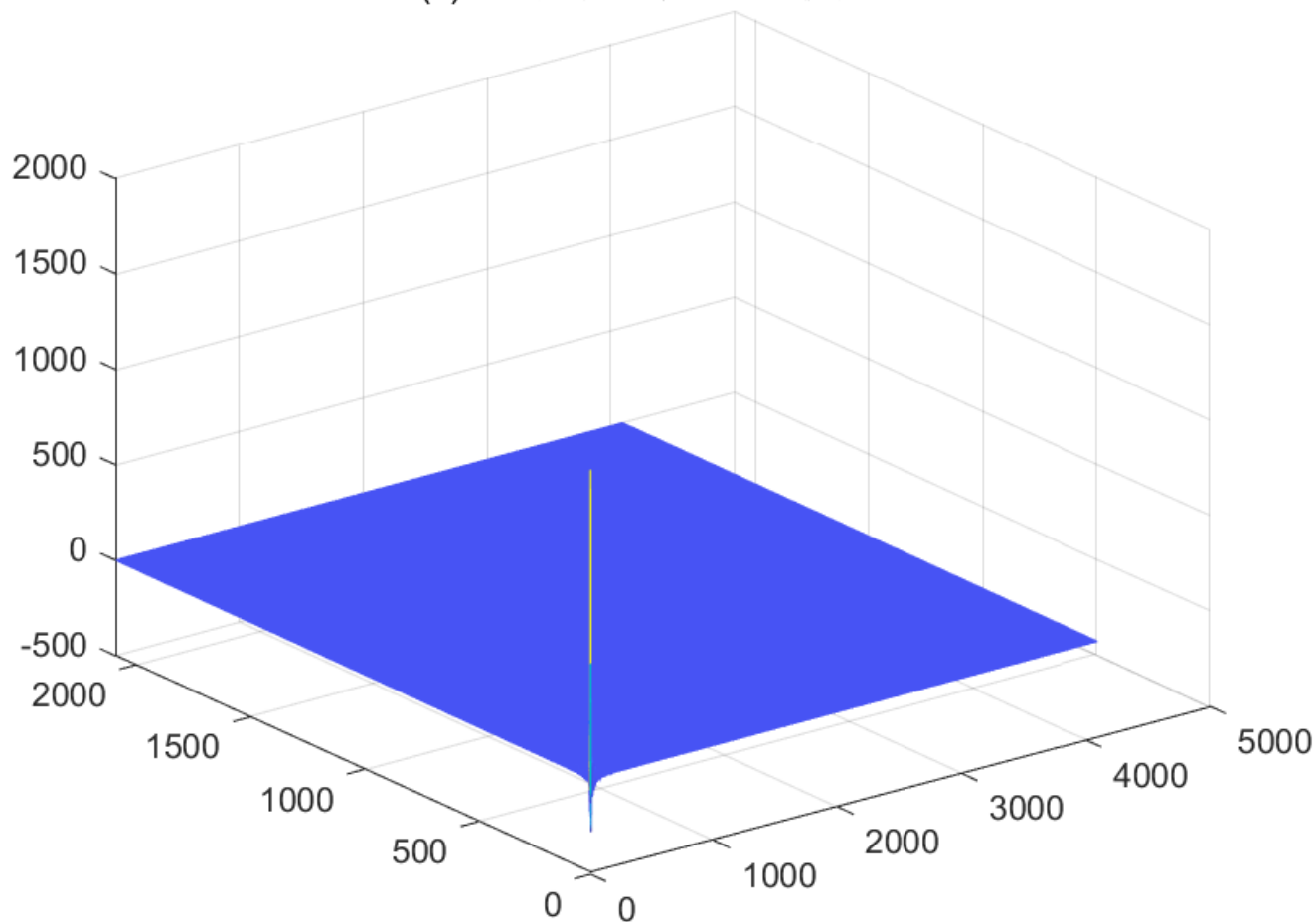
(a)原图像



(b)DCT变换系数



(c)DCT变换系数（立体视图）



小结

结合课上所学的方法，分别实现了 dft、dwt、dct 三种图像信号处理方式，且进一步熟悉了所学知识。

再次熟悉了matlab的使用。