



南开大学
Nankai University

南 开 大 学

网 络 空 间 安 全 学 院

信息隐藏技术实验三

图像信号的常用处理方法

穆禹宸 2012026

年级：2020 级

专业：信息安全、法学双学位班

指导教师：李朝晖

2023 年 3 月 14 日

目录

一、 实验目的	1
二、 实验原理	1
(一) 图像增强和复原	1
(二) 图像编码压缩	1
(三) 图像变换	1
(四) 图像描述	1
(五) 图像分类 (识别)	1
(六) 图像分割	2
(七) 离散傅里叶变换	2
(八) 离散小波变换	2
(九) 离散余弦变换	2
(十) 峰值信噪比	2
三、 实验过程	3
(一) 离散傅里叶变换 (DFT)	3
(二) 离散小波变换 (DWT)	6
1. 一级小波分解	6
2. 二级小波分解	7
(三) 离散小波变换 (DCT)	9
四、 实验心得体会	12

一、 实验目的

了解和掌握图像信号处理基础，动手实际操作二维离散傅里叶变换，离散小波变换和离散余弦变换。

二、 实验原理

数字图像处理常用方法。

（一） 图像增强和复原

图像增强和复原的目的是为了提高图像的质量，如去除噪声，提高图像的清晰度等。图像增强不考虑图像降质的原因，突出图像中所感兴趣的部分。如强化图像高频分量，可使图像中物体轮廓清晰，细节明显；如强化低频分量可减少图像中噪声影响。图像复原要求对图像降质的原因有一定的了解，一般讲应根据降质过程建立“降质模型”，再采用某种滤波方法，恢复或重建原来的图像。

（二） 图像编码压缩

图像编码压缩技术可减少描述图像的数据量（即比特数），以便节省图像传输、处理时间和减少所占用的存储器容量。压缩可以在不失真的前提下获得，也可以在允许的失真条件下进行。编码是压缩技术中最重要的方法，它在图像处理技术中是发展最早且比较成熟的技术。

（三） 图像变换

由于图像阵列很大，直接在空间域中进行处理，涉及计算量很大。因此，往往采用各种图像变换的方法，如傅立叶变换、沃尔什变换、离散余弦变换等间接处理技术，将空间域的处理转换为变换域处理，不仅可减少计算量，而且可获得更有效的处理（如傅立叶变换可在频域中进行数字滤波处理）。目前新兴研究的小波变换在时域和频域中都具有良好的局部化特性，它在图像处理中也有着广泛而有效的应用。

（四） 图像描述

图像描述是图像识别和理解的必要前提。作为最简单的二值图像可采用其几何特性描述物体的特性，一般图像的描述方法采用二维形状描述，它有边界描述和区域描述两类方法。对于特殊的纹理图像可采用二维纹理特征描述。随着图像处理研究的深入发展，已经开始进行三维物体描述的研究，提出了体积描述、表面描述、广义圆柱体描述等方法。

（五） 图像分类（识别）

图像分类（识别）属于模式识别的范畴，其主要内容是图像经过某些预处理（增强、复原、压缩）后，进行图像分割和特征提取，从而进行判决分类。图像分类常采用经典的模式识别方法，有统计模式分类和句法（结构）模式分类，近年来新发展起来的模糊模式识别和人工神经网络模式分类在图像识别中也越来越受到重视。

(六) 图像分割

图像分割是数字图像处理中的关键技术之一。图像分割是将图像中有意义的特征部分提取出来,其有意义的特征有图像中的边缘、区域等,这是进一步进行图像识别、分析和理解的基础。虽然目前已研究出不少边缘提取、区域分割的方法,但还没有一种普遍适用于各种图像的有效方法。因此,对图像分割的研究还在不断深入之中,是目前图像处理中研究的热点之一。

(七) 离散傅里叶变换

傅里叶变换是信号处理、图像处理等领域中重要的应用工具之一。傅里叶变换是一种信号在时域或在频域的整体变换,无法给出信号的频谱如何随时间变化的规律。

傅里叶的定义如下所示:

$$F(\omega) = \int_{-\infty}^{\infty} f(x)e^{-i\omega x} dx$$

而语音信号等具有很强的时变性,在一段时间内呈现出周期性信号的特点,而在另一段时间内呈现出随机信号的特点,或者呈现出两个混合的特性。对于频谱随时间变化的确定性信号以及非平稳随机信号,利用傅里叶变换分析方法有很大的局限性。傅里叶变换无法针对性地分析相应时间区域内信号的频率特征,需要用一个窗函数与时间信号相乘积,当该窗函数的时宽足够窄,使取出的信号可以被看成是平稳信号时,就可以对乘积信号进行傅里叶变换,从而反映该时宽中的信号频谱变化规律。

(八) 离散小波变换

小波变换继承和发展了短时傅里叶变换局部化的思想,同时又克服了窗口大小不随频率变化等缺点,能够提供一个随频率改变的“时间-频率”窗口,是进行信号时频分析和处理的理想工具。它的主要特点是,通过变换,能够充分突出信号某些方面的特征,能对时间(空间)频率进行局部化分析,通过伸缩平移运算对信号(函数)逐步进行多尺度细化,最终达到高频处时间细分。

(九) 离散余弦变换

离散余弦变换是与傅里叶变换相关的一种变换,它类似于离散傅里叶变换,但是只使用实数。离散余弦变换,尤其是它的第二种类型,经常用于信号处理和图像处理,对信号和图像进行有损数据压缩(使用分解得到的近似分量,代替原来信号)。这是由于离散余弦变换具有很强的“能量集中”特性:大多数的自然信号(包括声音和图像)的能量都集中在离散余弦变换后的低频部分。

(十) 峰值信噪比

峰值信噪比(PSNR)一种评价图像的客观标准。它具有局限性,PSNR是“Peak Signal to Noise Ratio”的缩写。psnr一般是用于最大值信号和背景噪音之间的一个工程项目。通常在经过影像压缩之后,输出的影像都会在某种程度与原始影像不同。为了衡量经过处理后的影像品质,我们通常会参考PSNR值来衡量某个处理程序能否令人满意。它是原图像与被处理图像之间的均方误差相对于 $(2^n - 1)^2$ 的对数值(信号最大值的平方, n 是每个采样值的比特数),它的单位是dB。MATLAB用法的公式如下:

$$PSNR = 10 * \log_{10}((2^n - 1)^2 / MSE)$$

三、 实验过程

我的程序修改了参考程序之中的代码，具体如下：

在 matlab2018 及以后中建议用 `imbinarize` 来将图片转换为二值图，其参数必须为灰度图。

在 matlab2016 中，只有 `im2bw` 函数，其参数可以是灰度图或 RGB 图。

因此，由于我使用的是 matlab r2022a，因此只能使用 `imbinarize` 函数。也是本次实验所遇到的一个小问题和解决方案，提前在此说明。

(一) 离散傅里叶变换 (DFT)

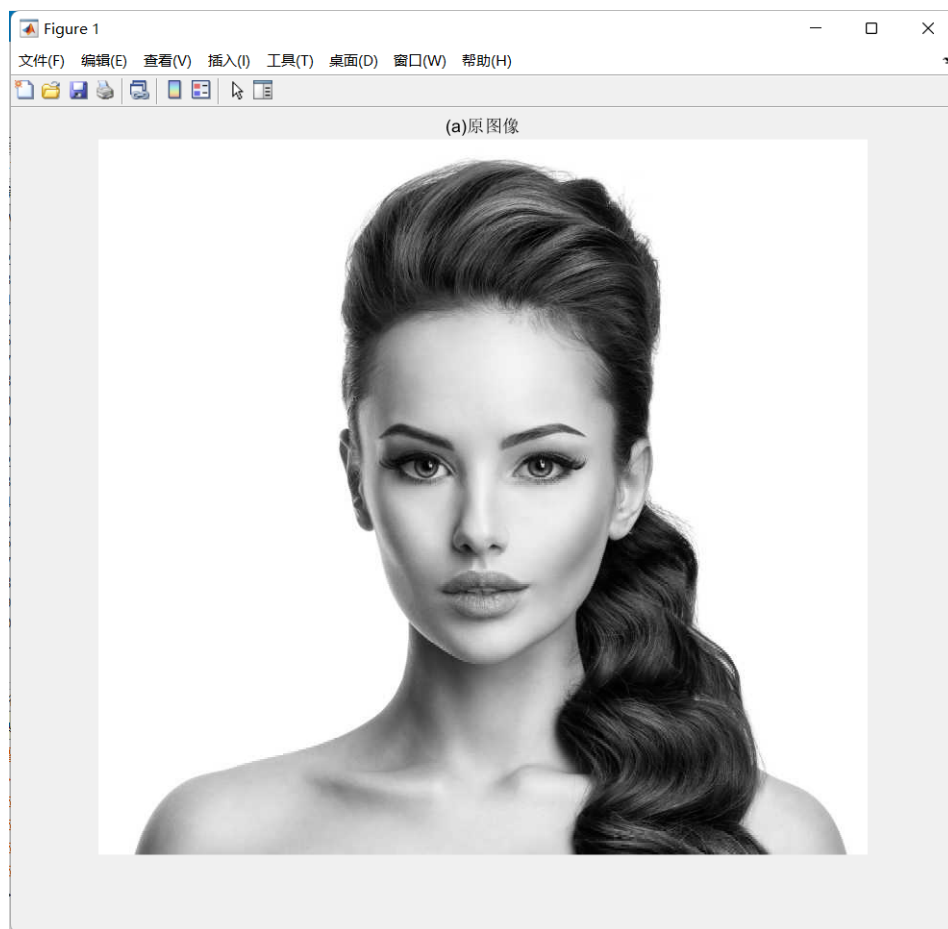
本次实验以及以后，我采取了自己在网络上找到的图片进行实验，具体代码如下：

DFT 代码

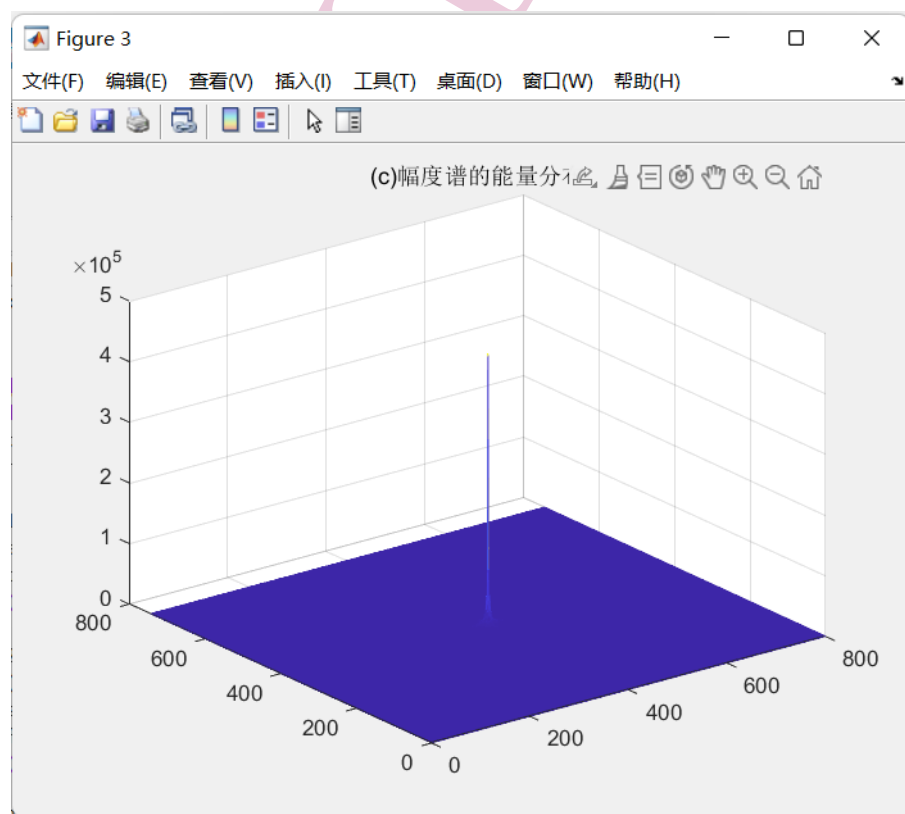
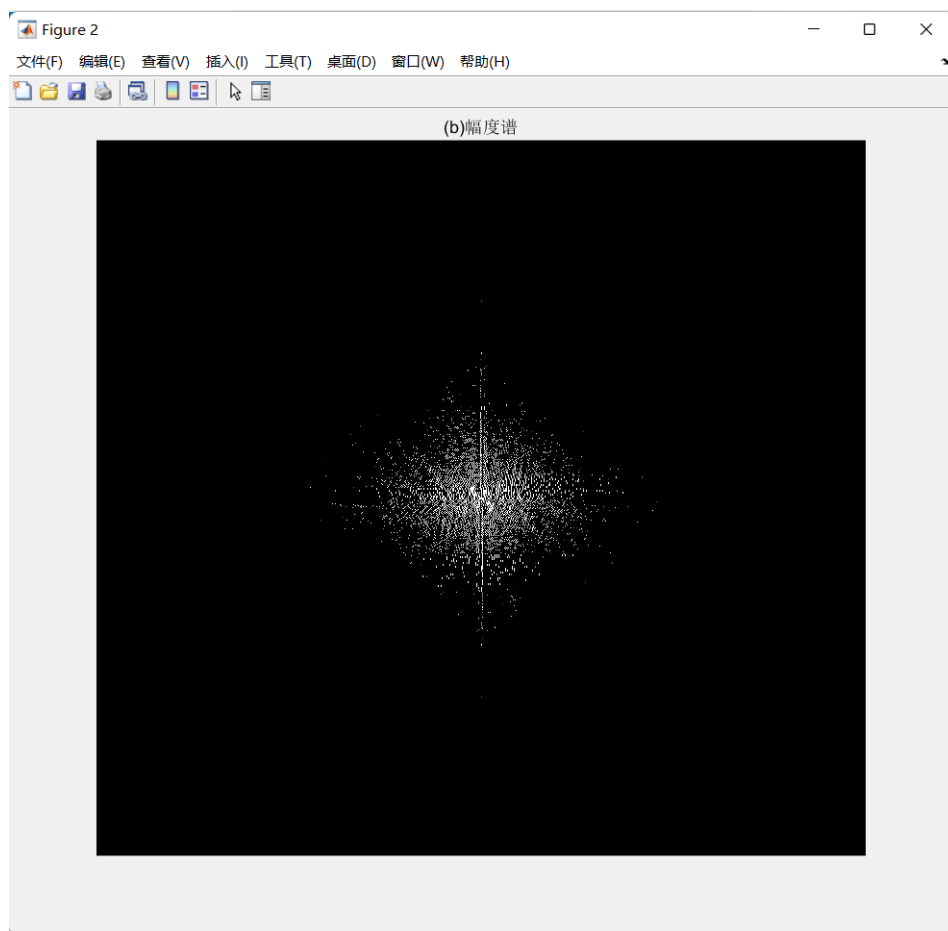
```
1 % Clear Memory and Command window
2 clc;
3 clear all;
4 close all;
5 b=imread("lady.jpg");%读入图像，像素值在b中
6 b=rgb2gray(b);%转换为灰度图像
7
8 figure(1);
9 I=imbinarize(b);%此处im2bw在matlab2022a之中已经不能使用
10 imshow(b);
11 title("(a)原图像");
12
13 figure(2);
14 fa=fft2(I);%使用fft函数进行快速傅里叶变换
15 ffa=fftshift(fa);%fftshift函数调整fft函数的输出顺序，将零频位置移到频谱的中心
16 imshow(ffa,[200,225]);%显示灰度在200-255之间的像
17 title("(b)幅度谱");
18
19 figure(3);
20 l=mesh(abs(ffa));%画网格曲面图
21 title("(c)幅度谱的能量分布");
```

代码逻辑整体上比较简单，首先使用 `clc`、`clear all` 和 `close all` 指令来清空内存和命令窗口。**注意，这是良好的代码习惯。**然后读入一张名为“lady.jpg”的图像，并将其转换为灰度图像。接着使用 `imshow` 函数在第一个窗口中显示原图像。在第二个窗口中，使用 `fft2` 函数进行快速傅里叶变换，并使用 `fftshift` 函数调整输出顺序，将零频位置移到频谱的中心。最后使用 `imshow` 函数在第二个窗口中显示幅度谱，其中灰度在 200-225 之间的像素被显示出来。在第三个窗口中，使用 `mesh` 函数画出幅度谱的能量分布的网格曲面图。

原始图像如下所示：



处理后的图像如下所示:



图像中，灰度变化缓慢的区域与低频信号近似；灰度变化大的边缘区域与高频信号近似。能量主要集中在灰度变化缓慢的低频部分，小部分能量集中在中心。

(二) 离散小波变换 (DWT)

1. 一级小波分解

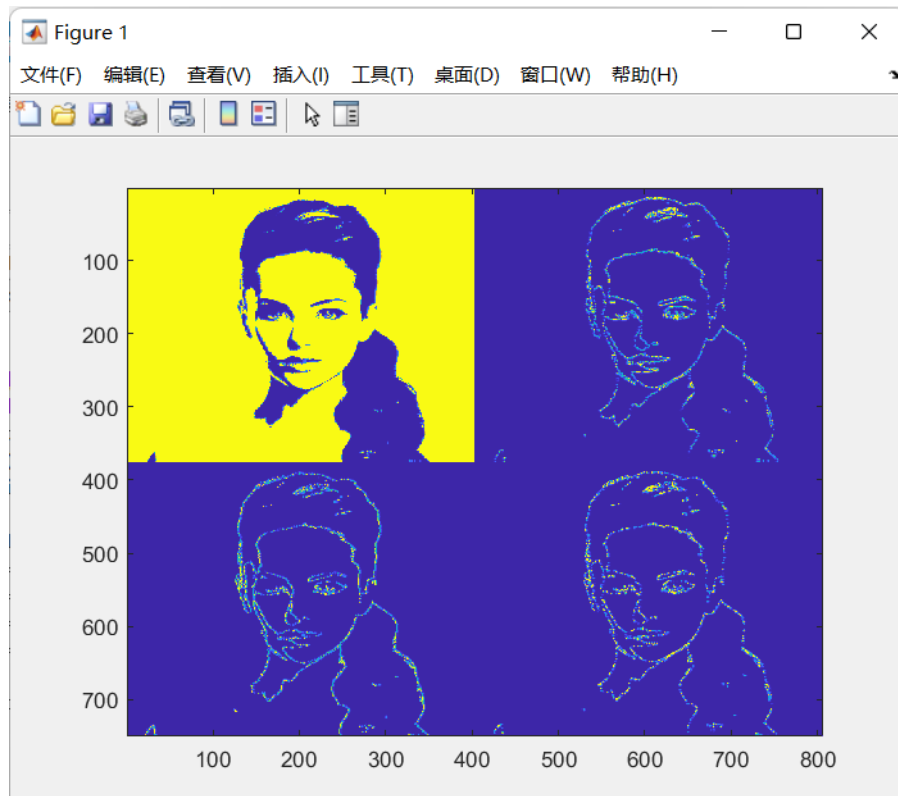
小波基采用 Daubechies-4 小波

一级小波分解代码

```
1 % Clear Memory and Command window
2 clc;
3 clear all;
4 close all;
5 b=imread('lady.jpg');%读入图像，像素值在b中
6 a=im2bw(b);
7 nbcot=size(a,1);
8
9 [ca1,ch1,cv1,cd1]=dwt2(a,'db4');
10 cod_ca1=wcodemat(ca1,nbcot);
11 cod_ch1=wcodemat(ch1,nbcot);
12 cod_cv1=wcodemat(cv1,nbcot);
13 cod_cd1=wcodemat(cd1,nbcot);
14
15 image([cod_ca1,cod_ch1;cod_cv1,cod_cd1]);
```

这段代码是用于进行小波变换的。首先使用 `clc`、`clear all` 和 `close all` 指令来清空内存和命令窗口。**注意，这是良好的代码习惯。**然后读入“lady.jpg”图像，并将其转换为二值图像。接着使用 `size` 函数获取二值图像的行数，存储在变量 `nbcot` 中。然后使用 `dwt2` 函数进行二维小波变换，使用‘db4’作为小波基函数。将小波变换后的系数分别存储在变量 `ca1`、`ch1`、`cv1` 和 `cd1` 中。接着使用 `wcodemat` 函数将小波系数矩阵进行可视化处理，存储在变量 `cod_ca1`、`cod_ch1`、`cod_cv1` 和 `cod_cd1` 中。最后使用 `image` 函数将四个可视化处理后的小波系数矩阵拼接成一个图像，并在 MATLAB 中显示出来。

得到的图像如下所示：



可以看到，左上角为分解得到的近似分量，右上角为水平方向细节分量，左下角为垂直方向细节分量，右下角为对角线方向细节分量。

2. 二级小波分解

小波基采用 Daubechies-4 小波

二级小波分解代码

```

1 % Clear Memory and Command window
2 clc;
3 clear all;
4 close all;
5 b=imread("lady.jpg");%读入图像，像素值在b中
6 a=im2bw(b);
7
8 nbcol=512;
9 nbc=256;
10
11 [ca1,ch1,cv1,cd1]=dwt2(a,'db4');
12 [ca2,ch2,cv2,cd2]=dwt2(ca1,'db4');
13
14 cod_ca1=wcodemat(ca1,nbc);
15 cod_ch1=wcodemat(ch1,nbc);
16 cod_cv1=wcodemat(cv1,nbc);
17 cod_cd1=wcodemat(cd1,nbc);
18

```

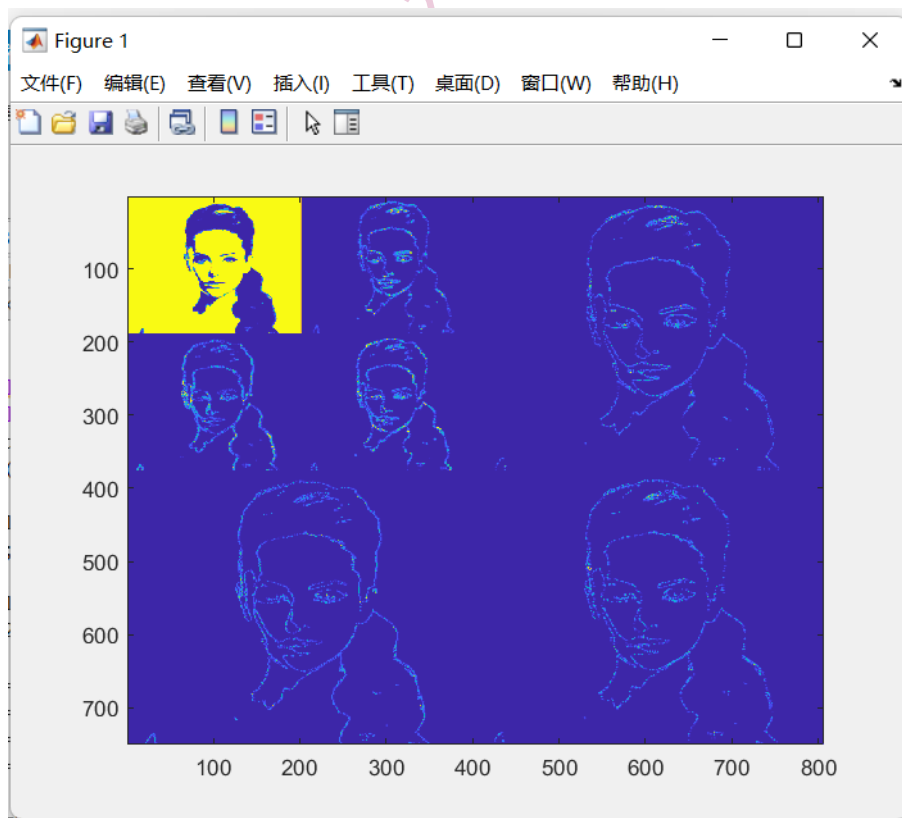
```

19 cod_ca2=wcodemat(ca2,nbcol);
20 cod_ch2=wcodemat(ch2,nbcol);
21 cod_cv2=wcodemat(cv2,nbcol);
22 cod_cd2=wcodemat(cd2,nbcol);
23
24 tt=[cod_ca2,cod_ch2;cod_cv2,cod_cd2];
25 tt=imresize(tt,size(ca1));
26
27 image([tt,cod_ch1,cod_cv1,cod_cd1]);

```

首先使用 `imread` 函数读入“lady.jpg”图像，并将其存储在变量 `b` 中。接着使用 `im2bw` 函数将图像转换为二值图像，并将其存储在变量 `a` 中。然后定义变量 `nbcol` 和 `nbc`，分别表示小波系数矩阵的列数。接着使用 `dwt2` 函数进行二维小波变换，使用‘`db4`’作为小波基函数。将小波变换后的系数分别存储在变量 `ca1`、`ch1`、`cv1` 和 `cd1` 中。然后再次使用 `dwt2` 函数对 `ca1` 进行小波变换，将小波变换后的系数分别存储在变量 `ca2`、`ch2`、`cv2` 和 `cd2` 中。接着使用 `wcodemat` 函数将小波系数矩阵进行可视化处理，存储在变量 `cod_ca1`、`cod_ch1`、`cod_cv1` 和 `cod_cd1` 中。然后使用 `wcodemat` 函数将小波系数矩阵进行可视化处理，存储在变量 `cod_ca2`、`cod_ch2`、`cod_cv2` 和 `cod_cd2` 中。接着将 `cod_ca2`、`cod_ch2`、`cod_cv2` 和 `cod_cd2` 拼接成一个图像，存储在变量 `tt` 中。然后使用 `imresize` 函数将 `tt` 的大小调整为 `ca1` 的大小。最后使用 `image` 函数将四个可视化处理后的小波系数矩阵拼接成一个图像，并在 MATLAB 中显示出来。

得到的图像如下所示：



代码流程与一级小波分解类似，它的目的在于让图片效果感官更好。为此需要使用 `imresize` 函数修改图片大小。注意到二级小波分解实际上就是对近似部分在此进行一次一级小波变换。

(三) 离散小波变换 (DCT)

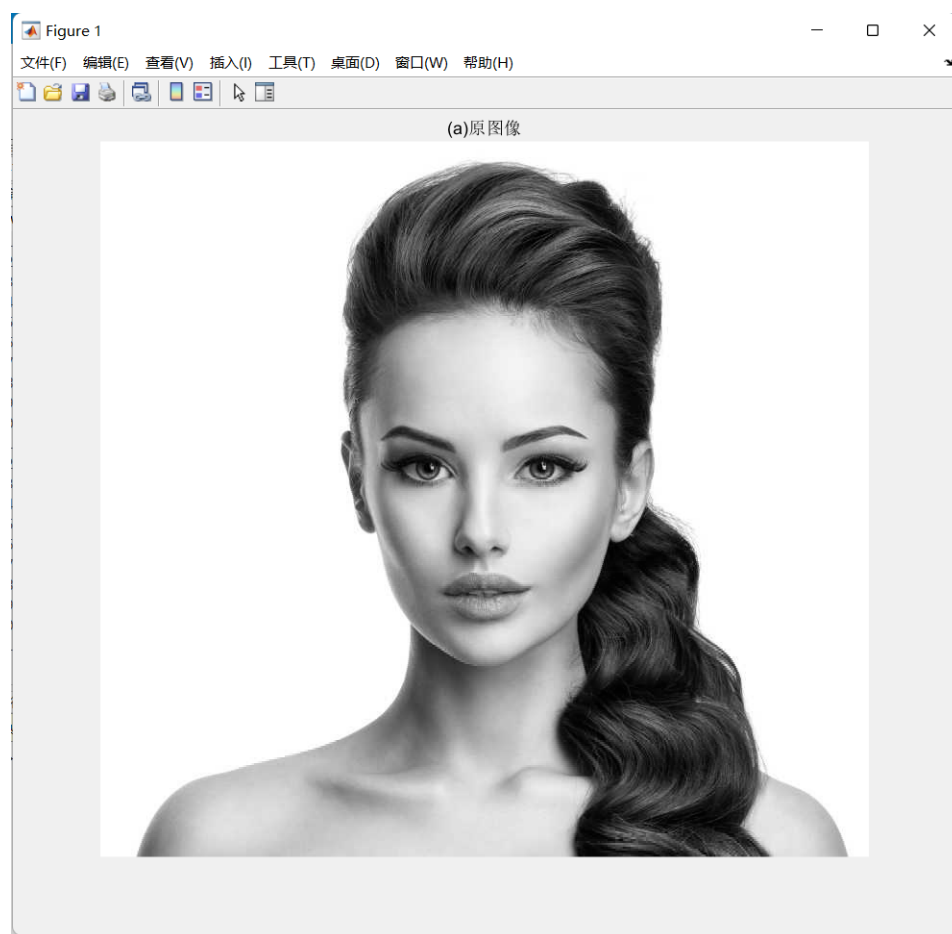
离散小波变换的过程简单，先将图片读取后转成灰度图，然后转为二值图，最后采用 DCT 变换绘制即可。代码如下所示：

离散小波变换

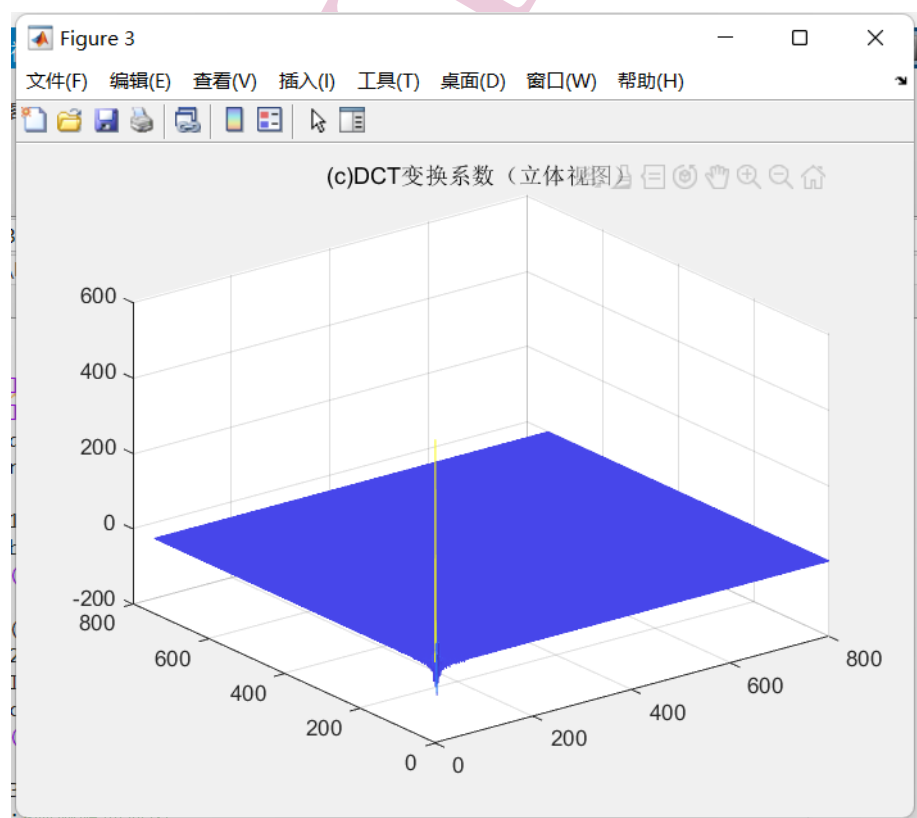
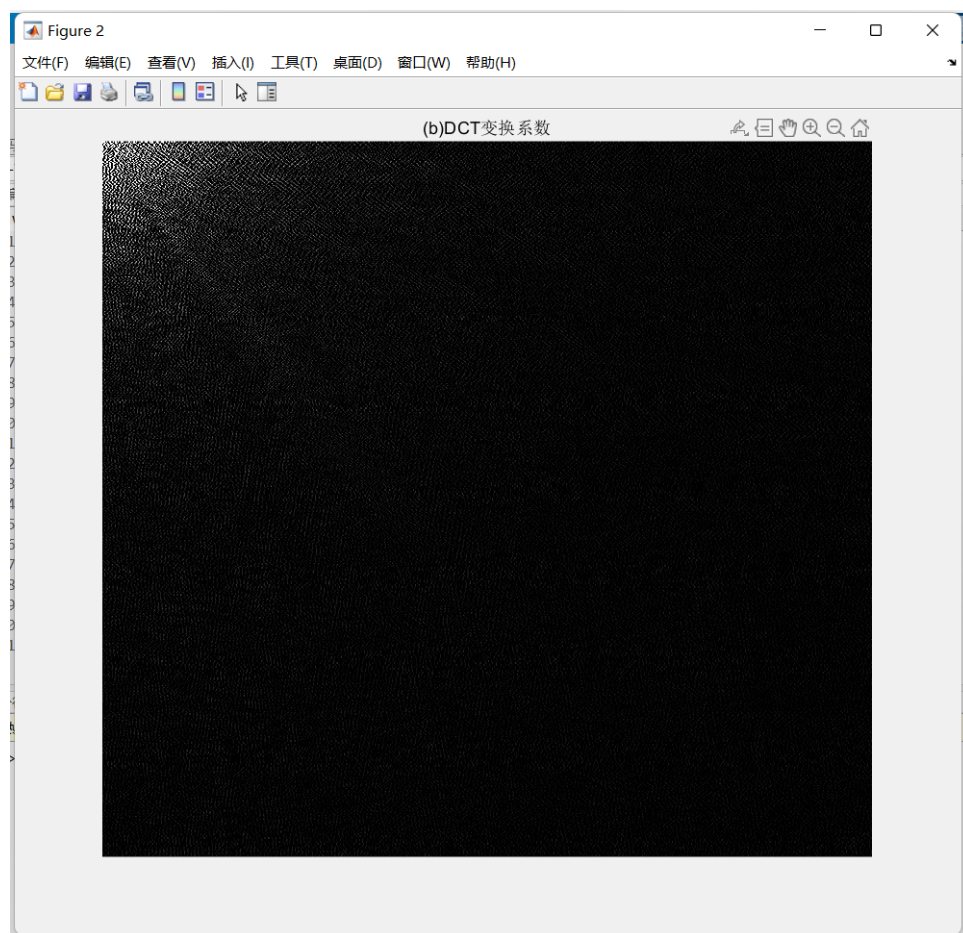
```
1 % Clear Memory and Command window
2 clc;
3 clear all;
4 close all;
5 b=imread("lady.jpg");%读入图像，像素值在b中
6 b=rgb2gray(b);%转换为灰度图像
7
8 figure(1);
9 imshow(b);
10 title('(a)原图像');
11
12 I=im2bw(b);
13 figure(2);
14 c=dct2(I);%进行离散余弦变换
15 imshow(c);
16 title('(b)DCT变换系数');
17
18 figure(3);
19 mesh(c);%画网格曲面图
20 title('(c)DCT变换系数 (立体视图)');
```

在读入图片后，将其存储在变量 `b` 中。然后使用 `rgb2gray` 函数将图像转换为灰度图像。接着使用 `imshow` 函数在第一个窗口中显示原图像。在第二个窗口中，使用 `im2bw` 函数将灰度图像转换为二值图像，并使用 `dct2` 函数进行离散余弦变换，将变换后的系数存储在变量 `c` 中。然后使用 `imshow` 函数在第二个窗口中显示变换后的系数。在第三个窗口中，使用 `mesh` 函数画出变换后的系数的网格曲面图，并使用 `title` 函数为图像添加标题。最后在 MATLAB 中显示出来。

原始图像如下所示：



处理后的图像如下所示:



可以看到, DCT 的系数主要集中在左上角, 代表直流、低频系数, 其余系数接近于零, 因此, DCT 可以广泛用于图像压缩算法中。

四、 实验心得体会

本次实验, 我通过课上所学的方法, 分别实现了 dft、dwt、dct 三种图像信号处理方式, 并且进一步熟悉了所学知识, 并初步掌握了 Matlab 对图像信号处理的方式。另外, 通过上机实验, 由于本次实验进行较为顺利, 我多次尝试了不同原始图像信号, 做了更多的探索, 在处理最初图像信号的基础上, 也进一步探索了不同处理方式的影响和开销, 尝试了不同参数和不同返回值。总而言之, 本次实验, 我收获很大!

NIJUB