南　　开　　大　　学

网络空间安全学院

信息隐藏技术实验报告

实验 3：图像信号的常用处理方法上机实验

姓名：2113203 付政烨

年级：2021 级

专业：信息安全、法学

指导教师：李朝晖

2024 年 3 月 25 日

目录

一、 实验内容

在 matlab 中调试完成课堂上的例题，练习使用常用的图像信号处理方法。
要求：编程实现，提交实验报告。

1. DFT
2. DWT
3. DCT

二、 实验目的

了解和掌握图像信号处理基础，动手实际操作傅里叶变换，小波变换和余弦变换，并且了解和掌握峰值信噪比 PSNR。

三、 实验原理

数字图像处理的常用方法包括：

1. 图像变换：由于图像阵列很大，直接在空间域中进行处理需要涉及大量计算。因此，常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，这不仅可以减少计算量，而且可以获得更有效的处理效果。
2. 图像编码压缩：图像编码压缩技术可以减少描述图像所需的数据量，以节省图像传输和处理的时间以及减少存储容量的需求。压缩可以在不失真的前提下进行，也可以在允许的失真条件下进行。
3. 图像增强和复原：图像增强和复原旨在提高图像质量，例如去除噪声和提高图像清晰度。图像增强不考虑图像降质的原因，突出图像中感兴趣的部分。
4. 图像分割：图像分割是将图像中有意义的特征部分提取出来，这些特征包括图像中的边缘、区域等，是进行图像识别、分析和理解的基础。
5. 图像描述：图像描述是图像识别和理解的必要前提。不同的图像可采用不同的描述方法，如基于几何特性的描述、二维形状描述等。
6. 图像分类（识别）：图像分类属于模式识别的范畴，主要包括图像的预处理、分割、特征提取和判决分类等步骤。

对图像的常用处理方法还包括二维傅里叶变换，离散余弦变换和二维小波变换等。

傅里叶变换是众多科学领域，特别是信号处理、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的整体变换，要么完全在时域，要么完全在频域进行分析处理，无法给出信号的频谱如何随时间变化的规律。而有些信号，例如语音信号，它具有很强的时变性，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，或者说是不合适的。傅里叶变换无法针对性的分析相应时间区域内信号的频率特征。可以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。傅里叶定义为:

$$F(\omega) = F[f(t)] = \int f(t)e^{-i\omega t}dt$$

小波变换继承和发展了短时傅里叶变换局部化的思想，同时又客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问题某些方面的特征，能对时间（空间）频率的局部化分析，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。

离散余弦变换 (DCT for Discrete Cosine Transform) 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 (DFT for Discrete Fourier Transform)，但是只使用实数。离散余弦变换相当于一个长度大概是它两倍的离散傅里叶变换，这个离散傅里叶变换是对一个实偶函数进行的 (因为一个实偶函数的傅里叶变换仍然是一个实偶函数)，在有些变形里面需要将输入或者输出的位置移动半个单位 (DCT 有 8 种标准类型，其中 4 种是常见的)。离散余弦变换，尤其是它的第二种类型，经常被信号处理和图像处理使用，用于对信号和图像 (包括静止图像和运动图像) 进行有损数据压缩。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。尽管直接使用公式进行变换需要进行 $O(n^2)$ 次操作，但是和快速傅里叶变换类似，我们有复杂度为 $O(n\log(n))$ 的快速算法，这就是常常被称做蝶形变换的一种分解算法。另外一种方法是通过快速傅里叶变换来计算 DCT，这时候需要 $O(n)$ 的预操作和后操作。

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:

$$PSNR = 10\log_{10}\left(\frac{(2^n-1)^2}{MSE}\right)$$

其中，MSE 是原图像与处理图像之间的均方误差。

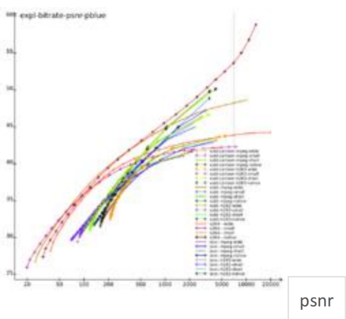

图 1

Peak 就是指 8bits 表示法的最大值 255。MSE 指 Mean Square Error，I_n 指原始影像第 n 个 pixel 值，P_n 指经处理后的影像第 n 个 pixel 值。PSNR 的单位为 dB。所以 PSNR 值越大，就代表失真越少。

四、 实验过程

(一) 离散傅里叶变换（DFT）

1. 原理简述

离散傅里叶变换（DFT）是一种用于分析离散信号频率特性的数学工具。在数字信号处理中，DFT 是非常重要的，因为它允许我们将时域中的信号转换到频域中。这个转换过程使得分析和处理信号变得更加简便，尤其是在处理信号的频率成分时。DFT 的核心思想是将一个时域的离散信号分解成一组频率不同的复数正弦波。这些复数正弦波的频率是固定且离散的，它们相互正交，代表着原始信号中的不同频率成分。

数学上，DFT 将一个长度为 N 的复数或实数序列 $x[n]$ 转换成另一个长度为 N 的复数序列 $X[k]$，通过以下公式：

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-i2\pi kn/N}$$

其中，n 是时间索引，k 是频率索引，$X[k]$ 是转换后的复数表示，包含了频率 k 的幅度和相位信息。

DFT 的一个常见应用是在频谱分析，它可以用来检测信号中的频率成分以及它们的强度。通过分析一个信号的 DFT，可以了解到该信号中存在哪些频率成分，这在音频处理、图像处理、通信系统等领域有着广泛的应用。

2. 代码实现

```
% 清空命令窗口和内存
clc;
clear all;
close all;

b = imread("student.png");
b = rgb2gray(b);

figure(1);
% 使用imbinarize函数进行图像二值化（注意：im2bw在MATLAB R2022a已不再使用）
I = imbinarize(b);
imshow(b);
title("(a) 原图像");
saveas(gcf, '原图像.jpg');

figure(2);
fa = fft2(I);
ffa = fftshift(fa);
imshow(ffa, [200 225]);
title("(b) 幅度谱");
saveas(gcf, '幅度谱.jpg');

figure(3);
l = mesh(abs(ffa));
title("(c) 幅度谱的能量分布");
```

```
saveas(gcf, '幅度谱能量分布.jpg');
```

代码首先读取了名为”student.png” 的彩色图像，并将其转换为灰度图像。接着，代码对灰度图像进行了二值化处理，并进行了二维傅里叶变换，得到了幅度谱。幅度谱经过调整后被显示在一个新的图形窗口中。最后，代码绘制了幅度谱的能量分布网格曲面图，并将其保存为图片文件。整个过程分为四个步骤：读取图像并转换为灰度图像、显示并保存原始图像、计算并显示幅度谱并保存、绘制并保存幅度谱的能量分布网格曲面图。

3. 结果分析

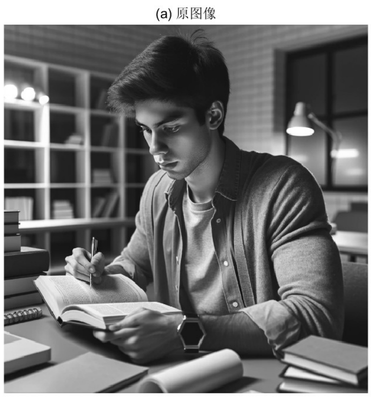

图 2: 原图像

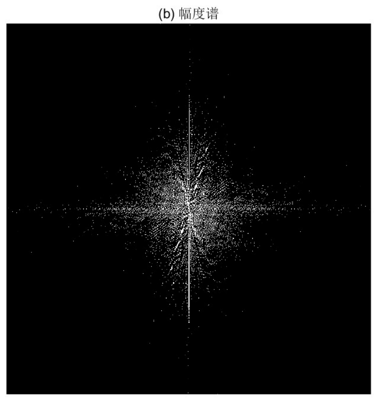

图 3: 幅度谱

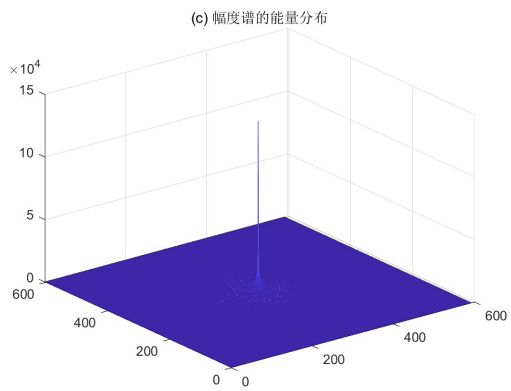

图 4: 幅度谱能量分布

- **原图像（图 2）**: 这是离散傅里叶变换处理流程的起点。这是进行傅里叶变换之前的原始图像，通常会先转换为灰度图，然后再进行进一步的处理。
- **幅度谱（图 3）**: 显示了原图像离散傅里叶变换的结果。傅里叶变换将图像从空间域转换到频率域，幅度谱展示了不同频率成分的幅度。在这张图中，亮点表示高幅度的频率成分，而暗区域表示幅度较低的频率成分。幅度谱中心的亮点表示低频信息，通常与图像的主要结构相关，而四周的亮点与图像的细节和噪声有关。

- **幅度谱的能量分布（图 4）**：这张图以三维形式展示了幅度谱的能量分布。它提供了一个直观的视角，显示了不同频率成分的能量大小。能量集中的区域（如图中的尖峰）表示那些在图像中占主导地位的频率成分。这种表示方法有助于理解图像的频率特性，例如辨别图像中主要的形状和边缘信息。

从图像中可以观察到，渐进式的灰度变化通常对应于低频信号，这反映了图像中颜色渐变或平滑区域的特征。相对地，灰度变化剧烈的区域，尤其是图像的边缘或轮廓，与高频信号相似，表明这些区域在空间上的快速变化。通常，在图像的频率分布中，大量的能量集中在低频区域，这代表了图像的主要视觉内容和整体结构。而高频区域，虽然只包含了小部分的能量，却承载了图像的细节信息，如纹理和噪声。在频域表示中，这种能量的分布倾向于在灰度变化缓慢的地方形成宽泛的能量峰，而在高频细节处形成尖锐的能量集中，尤其是在频域的中心区域。

（二） 离散小波变换（DWT）

1. 原理简述

离散小波变换（DWT）是一种数学转换，用于分析具有不同分辨率的数据。与离散傅里叶变换（DFT）相比，DWT 提供了时间和频率的局部化信息，使其在处理具有时变特性的信号时特别有用。DWT 通过应用一系列滤波器和下采样操作，将信号分解为不同频率带的近似和细节信息。

- **一级小波分解**：一级小波分解是 DWT 的基本操作，它将原始信号分解成两个部分：近似系数（低频部分）和细节系数（高频部分）。这是通过将原始信号先通过低通滤波器，再通过高通滤波器实现的，两者都后跟一个下采样步骤（通常是去除每两个样本中的一个）。近似系数反映了信号的主要趋势，而细节系数包含了信号的高频变化信息。
- **二级小波分解**：二级小波分解进一步细化了一级小波分解的结果。在这一级别，我们只对一级分解得到的近似系数进行再分解，得到第二级的近似系数和细节系数。这样做的目的是在更细的时间尺度上分析信号的低频部分，同时保持高频部分的信息不变。通过重复这个过程，我们可以得到多级小波分解，每一级都提供关于信号不同时间尺度的更多信息。

小波变换特别适合于分析具有不同时间尺度特性的信号，比如瞬态信号或是具有断点或尖峰的信号。它在图像压缩、特征提取、降噪和多分辨率分析等多个领域都有广泛应用。

2. 一级小波分解

(1) 代码实现

```
clc; % 清除命令窗口
clear all; % 清除当前工作空间中的所有变量
close all; % 关闭所有图形窗口

b = imread("student.png"); % 读入图像，像素值存储在变量b中
a = im2bw(b); % 将图像转换为二值图像（黑白图像）

nbcol = size(a, 1); % 获取图像的行数

% 进行离散小波变换
[ca1, ch1, cv1, cd1] = dwt2(a, 'db4');
```

```

% 对各个小波系数进行尺度系数矩阵转换
cod_ca1 = wcodemat(ca1, nbcol);
cod_ch1 = wcodemat(ch1, nbcol);
cod_cv1 = wcodemat(cv1, nbcol);
cod_cd1 = wcodemat(cd1, nbcol);

image ( [ cod_ca1 , cod_ch1 ; cod_cv1 , cod_cd1 ] );

% 保存最终的图片
saveas(gcf, '小波变换结果.png');
```

以上代码首先使用 MATLAB 的 imread 函数读取了名为”student.png” 的学生照片，并将其储存在变量 b 中。接下来，通过 im2bw 函数将彩色图像转换为黑白二值图像，以便进行后续处理。然后，利用 dwt2 函数对该二值图像进行二维离散小波变换，采用了 Daubechies-4 小波基，得到了近似系数 (ca1)、水平细节系数 (ch1)、垂直细节系数 (cv1) 和对角细节系数 (cd1)。

随后，通过 size 函数获取二值图像的行数，用于后续可视化处理。利用 wcodemat 函数将每个小波系数矩阵转换为可视化的矩阵，以便更清晰地展示各个系数的特征。最后，利用 image 函数将转换后的四个系数矩阵拼接成一张新的图像，将其在 MATLAB 中显示出来。

(2) 结果分析

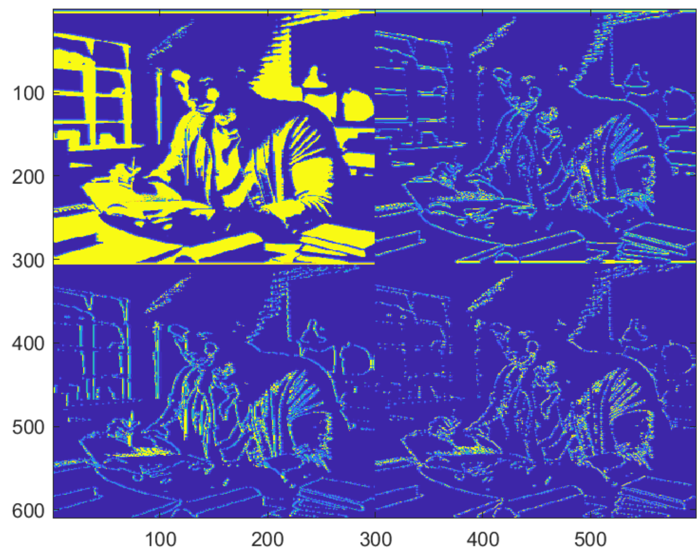

图 5: 一级小波分解结果

这张图片展示的是一级离散小波变换（DWT）的结果，使用了 Daubechies-4（db4）作为小波基。在这种变换中，原始图像被分解为四个不同的部分：近似系数、水平细节、垂直细节和对角线细节。

- **近似系数（左上角）**：这部分是经过低通滤波器处理的图像，它保留了原始图像的大部分重要信息，但以较低的分辨率呈现。近似系数表示图像中的低频分量，通常包含图像的主要结构信息。

- **水平细节（右上角）**：这部分是经过高通滤波器处理后只在水平方向上保留细节的结果。水平细节系数揭示了图像中水平边缘的位置，对应于图像中水平方向的高频分量。
- **垂直细节（左下角）**：这部分与水平细节相似，但是它反映的是图像中垂直方向上的细节，也就是垂直边缘的位置，代表图像中垂直方向的高频分量。
- **对角细节（右下角）**：这部分显示了图像中的对角线方向的细节，它反映了图像中对角线方向的高频分量，通常与图像中的对角线边缘或是角点有关。

3. 二级小波分解

(1) 代码实现

```
% 清除内存和命令窗口
clc;
clear all;
close all;
b = imread("student.png");
a = im2bw(b);

nbcol = 512; % 用于第二级小波分解的颜色编码深度
nbc = 256;   % 用于第一级小波分解的颜色编码深度

% 进行第一级离散小波变换
[ca1, ch1, cv1, cd1] = dwt2(a, 'db4');
% 进行第二级离散小波变换
[ca2, ch2, cv2, cd2] = dwt2(ca1, 'db4');

% 将小波系数转换为可视化格式
cod_ca1 = wcodemat(ca1, nbc);
cod_ch1 = wcodemat(ch1, nbc);
cod_cv1 = wcodemat(cv1, nbc);
cod_cd1 = wcodemat(cd1, nbc);
cod_ca2 = wcodemat(ca2, nbcol);
cod_ch2 = wcodemat(ch2, nbcol);
cod_cv2 = wcodemat(cv2, nbcol);
cod_cd2 = wcodemat(cd2, nbcol);

% 将第二级变换结果组合，并调整至第一级分解大小
tt = [cod_ca2, cod_ch2; cod_cv2, cod_cd2];
tt = imresize(tt, size(ca1));

image([tt, cod_ch1; cod_cv1, cod_cd1]);
saveas(gcf, '小波变换结果.jpg');
```


上述代码是对的图像进行两级离散小波变换处理，然后将处理结果进行可视化并保存。首先，代码清除了命令窗口、删除了所有变量，并关闭了所有打开的图形窗口以准备环境。然后，它读取图像文件，并将这个彩色图像转换为二值图像，这是进行小波变换的前提。接着，代码执行了两次离散小波变换：第一次变换后得到近似系数、水平细节、垂直细节和对角细节四部分；第二次变换则对第一次的近似系数部分进行，也生成四个相应的系数。这些系数通过 wcodemat 函

数转换为可视化的格式，以便更直观地展示变换的效果。之后，第二级变换的结果被组合并调整大小以匹配第一级变换结果的大小。最终，第一级和第二级变换的结果被组合并显示在一张图像中，这张图像同时展示了图像的不同频率和方向的细节。

(2) 结果分析

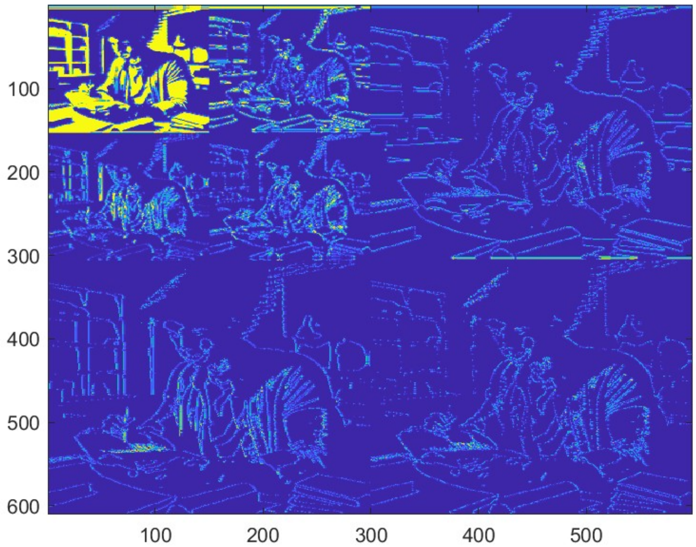

图 6: 二级小波分解结果

图片展示了经过两级离散小波变换的图像的可视化结果。离散小波变换是一种数学处理手段，常用于图像处理领域，可以将图像分解为不同的频率和方向的细节，这对于图像分析和压缩非常有用。在这个过程中，图像首先被转换为二值图像，这是为了准备进行小波变换。

图像处理开始于一级离散小波变换，该变换产生四个部分：近似系数（ca1）、水平细节（ch1）、垂直细节（cv1）、和对角细节（cd1）。近似系数代表了图像的概貌或者说是较低频的部分，而其他三个细节系数分别捕捉了图像的水平、垂直和对角方向的高频信息。接下来，对近似系数进行了第二级离散小波变换，同样生成了四个新的系数：近似系数（ca2）、水平细节（ch2）、垂直细节（cv2）和对角细节（cd2），这一级的变换聚焦于更高层次的数据抽象。为了将这些数学上的变换结果转换成可视化的格式，代码使用了 wcodemat 函数，它可以根据给定的颜色编码深度，将小波系数转换为图像格式。在这个例子中，第一级使用了 256 色编码，而第二级使用了更精细的 512 色编码。转换后，第二级变换的结果被组合在一起，并调整大小以匹配第一级变换的结果。最终的可视化图像在左上角展示了第二级变换的近似系数，右上角是第二级变换的水平细节，左下角是垂直细节，右下角是对角细节。整个图像通过调整这些细节部分的大小，将它们拼接在一起，与第一级变换的结果相结合，展现了原始图像在不同分解级别上的详细信息。

（三） 离散余弦变换（DCT）

1. 原理简述

离散余弦变换（Discrete Cosine Transform，简称 DCT）是一种与离散傅里叶变换（DFT）相似的技术，但主要用于信号和图像处理领域中的数据压缩。与 DFT 相比，DCT 更适合于处理实数信号，特别是在处理具有高度相关性数据时，如自然图像和声音。因此，DCT 在 JPEG

图像压缩、MPEG 视频压缩以及其他多媒体应用中被广泛使用。DCT 的主要思想是将一个信号或图像分解成一系列的余弦函数，这些函数具有不同的频率。通过这种分解，DCT 能够将信号的能量集中到变换的一小部分系数上，这使得压缩更加高效。

DCT 有几种不同的变体，最常用的是二维 DCT（DCT-II），它是 JPEG 标准的核心部分。在数学上，对于一个 $N \times M$ 的图像或信号矩阵，二维 DCT-II 定义如下：

$$F(u,v) = \frac{1}{\sqrt{NM}} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) \cos\left[\frac{\pi u(2x+1)}{2N}\right] \cos\left[\frac{\pi v(2y+1)}{2M}\right]$$

其中，$f(x,y)$ 是原始图像或信号矩阵，$F(u,v)$ 是 DCT 变换后的矩阵，$C(u)$ 和 $C(v)$ 是归一化系数，当 u 或 v 为 0 时取值为 $\sqrt{1/N}$ 或 $\sqrt{1/M}$，否则取值为 $\sqrt{2/N}$ 或 $\sqrt{2/M}$。

DCT 的优势在于其能够有效地表示信号中的频率成分，同时在许多实际应用中能够实现数据的压缩。例如，在 JPEG 图像压缩中，图像首先被分割成小块（通常是 8×8 像素的块），然后对每个块应用 DCT，最后通过量化和编码步骤来实现压缩。由于许多高频成分通常接近零，它们可以被有效地移除，从而达到压缩数据的目的，而对图像质量的影响却很小。

2. 代码实现

```
clc;
clear all;
close all;

b = imread("student.png"); % 读入图像，像素值在b中
b = rgb2gray(b); % 转换为灰度图像

figure(1);
imshow(b);
title('(a)原图像');
saveas(gcf, 'original_image.png'); % 保存原图像结果

I = im2bw(b);
figure(2);
c = dct2(I); % 进行离散余弦变换
imshow(c);
title('(b)DCT变换系数');
saveas(gcf, 'dct_coefficients.png'); % 保存DCT变换系数图像结果

figure(3);
mesh(c); % 画网格曲面图
title('(c)DCT变换系数（立体视图）');
saveas(gcf, 'dct_coefficients_mesh.png'); % 保存DCT变换系数网格曲面图结果
```

代码首先读取名为”student.png” 的图像文件,并将其转换为灰度图像。这一步是通过 rgb2gray() 函数实现的，将彩色图像转换为灰度图像，这样处理后的图像只有一个灰度通道，简化了后续的处理步骤。接下来，程序将灰度图像进行二值化处理，使用 im2bw() 函数将图像二值化，生成一个二值图像。然后，程序对这个二值图像进行离散余弦变换（DCT），使用 dct2() 函数实现。DCT 是一种常用的变换技术，用于在图像处理中提取图像的频域信息。随后，程序绘制了 DCT

变换后的系数矩阵的网格曲面图，通过 mesh() 函数实现。这个网格曲面图能够直观地展示 DCT 变换后的系数矩阵的结构和分布情况。

3. 结果分析

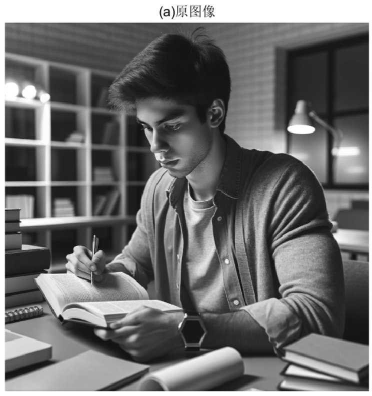

图 7: 原图像

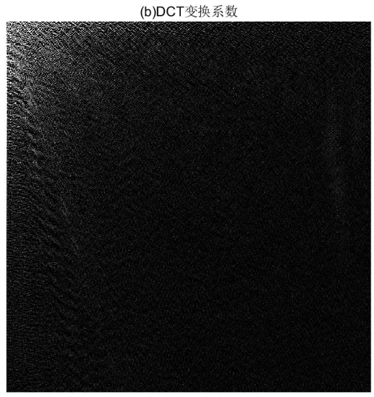

图 8: DCT 变换系数

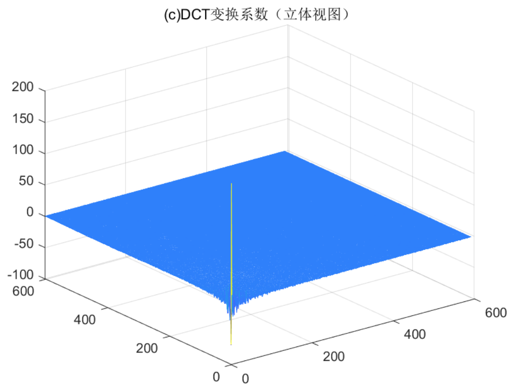

图 9: DCT 变换系数（立体视图）

- **原图像（图 7）**（略）
- **DCT 变换系数（图 8）**：图片几乎全部为黑色，但整个图中散布着白色和灰色的细节。这是对原图的二值版本应用 DCT 后的结果。DCT 将图像从空间域转换到频率域，允许分析图像的频率内容。白色和灰色部分代表图像中较高频率的区域，通常对应于边缘和细节。在 MATLAB 代码中，通过对从灰度图像创建的二值图像应用 dct2 函数来实现这一点。
- **DCT 变换系数（立体视图）（图 8）**：该图是一个三维图，显示了 DCT 系数的幅度。垂直轴（高度）表示 DCT 系数的幅度，而水平平面代表它们在频率域中的位置。图中的峰值指示图像中最显著的频率成分，通常对应于图像内容中的显著强度变化（如边缘）。

五、 扩展实验

（一） 快速傅里叶变换（FFT）

离散傅里叶变换（DFT）和快速傅里叶变换（FFT）皆为频域分析的工具，然而其应用场景有所差异。DFT 适于处理简单频域分析任务或信号长度较小的情况，其计算复杂度为 $O(N^2)$，对于大型信号或需要频繁进行频域分析的场景则性能较低。相反，FFT 通过将计算复杂度优化至 $O(N\log N)$ 的水平，显著提高了性能，使其更适用于处理大型信号或频繁进行频域分析的应用场景。因此，本次扩展实验旨在深入了解并熟悉 FFT 的实际运用。

1. 原理简介

快速傅里叶变换（Fast Fourier Transform，FFT）是一种非常重要的数学算法，用于将信号从时域转换到频域。简单来说，它将一个复杂的信号分解为多个简单的正弦波信号的组合。这些正弦波信号各自具有不同的频率、振幅和相位，FFT 帮助我们了解原始信号在不同频率上的成分有多强。在数学上，傅里叶变换是表示一个函数或信号为不同频率的正弦波和余弦波之和的过程。对于连续信号，这称为傅立叶积分。而对于离散信号（即数字信号），这个过程称为离散傅立叶变换（Discrete Fourier Transform, DFT）。

DFT 的数学表达式为：

$$X(k)=\sum_{n=0}^{N-1}x(n)\cdot e^{-j\frac{2\pi}{N}kn}$$

其中：$X(k)$ 是 DFT 的第 k 个元素，表示频率分量；$x(n)$ 是原始信号的第 n 个时间域样本；N 是信号的总样本数；$e^{-j\frac{2\pi}{N}kn}$ 是旋转因子，其中 j 是虚数单位。

直接计算 DFT 需要 $O(N^2)$ 次操作，这在处理大型数据集时会非常慢。FFT 是 DFT 的一种算法，可以在 $O(N\log N)$ 时间内完成同样的计算，大大加快了处理速度。FFT 算法的基本思想是分而治之。它将原始信号分成两部分：一部分包含所有偶数索引的样本，另一部分包含所有奇数索引的样本。通过递归地将这些子信号进一步拆分，并计算它们的 DFT，最后将结果组合起来以得到原始信号的 DFT。这种方法称为“蝴蝶算法”，因为数据流的图形类似于蝴蝶的形状。

2. 代码实现

```
% 清空命令窗口和内存
clc;
clear all;
close all;

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

% 显示原始灰度图像
figure(1);
imshow(b);
title("(a) 原图像");
saveas(gcf, '原图像.jpg');

% 对灰度图像进行二值化
```

```matlab
I = imbinarize(b);

% 对二值化后的图像进行FFT变换
fa = fft2(I);

% 将FFT结果中的直流分量移到频谱中心
ffa = fftshift(fa);

% 计算幅度谱并显示
figure(2);
magnitude_spectrum = abs(ffa);  % 计算幅度谱
imshow(log(magnitude_spectrum+1), []);  % 使用对数尺度增强显示效果
title("(b) 幅度谱");
saveas(gcf, '幅度谱.jpg');

% 使用mesh显示幅度谱的能量分布
figure(3);
mesh(log(magnitude_spectrum+1));  % 对幅度进行对数变换来更好地显示能量分布
title("(c) 幅度谱的能量分布");
saveas(gcf, '幅度谱能量分布.jpg');
```

代码首先读取”student.png” 的彩色图像，并使用 rgb2gray 函数将其转换为灰度图像，这是因为在频域分析中通常使用灰度图像。接着，对灰度图像进行二值化处理，将其转换为二值图像，这可以通过 imbinarize 函数实现。然后，对二值化后的图像执行快速傅里叶变换（FFT），使用 fft2 函数实现。FFT 变换将图像从时域转换到频域，得到图像在频域上的表示。为了更好地可视化频域表示，使用 fftshift 函数将 FFT 结果进行移动，以便将图像的低频成分移到频谱的中心位置，这有助于分析和理解频域信息。接下来，计算并显示频域表示的幅度谱，这可以通过计算 FFT 结果的绝对值来实现。为了更好地显示幅度谱的范围，采用对数尺度对幅度谱进行转换，以增强图像中较小幅度的细节。最后，使用 mesh 函数以三维网格的形式显示幅度谱的能量分布情况，这可以更直观地展示频域中不同频率成分的能量分布情况。通过这种方式，可以更深入地理解图像在频域中的特征和结构。

3. 结果分析

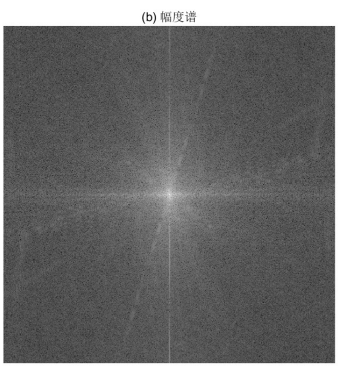

图 10: 原图像

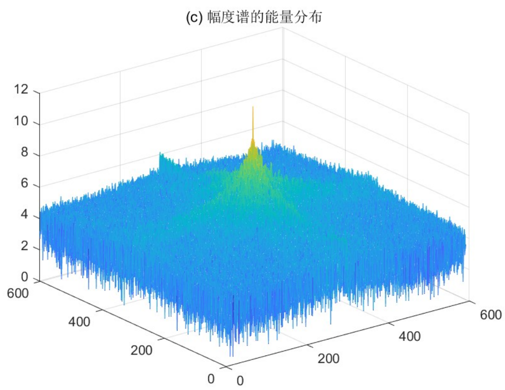

图 11: 幅度谱

- **原图像**（与之前原图片相同）
- **幅度谱（图 10）**：第二张图显示了图像傅里叶变换的幅度谱，表示了原始图像的频率成分。该过程包括将灰度图像转换为二值图像（将前景与背景分离，简化图像）、应用 FFT（将空间域图像转换为频域图像），然后应用对数尺度来增强谱细节的可见性。这张图片中的明亮中心指示了低频成分的高集中度，这对于具有大面积均匀区域和整体结构的图像是典型的。
- **幅度谱的能量分布（图 11）**：第三张图是表示幅度谱能量分布的 3D 网格图，有助于理解不同频率成分的能量（或强度）如何分布。这种分布中的峰值代表高频内容区域，对应于边缘、纹理和图案。

（二） 直方图均衡化

1. 原理简介

直方图均衡化是一种在图像处理中常用的对比度增强技术。它的目的是改善图像的对比度，使得图像的直方图（即图像亮度的分布）更加均匀，从而使图像的细节更加清晰。这种方法特别适合于背景和前景都很暗或者都很亮的图像。直方图均衡化的基本思想是将原图像的直方图转变为一个均匀分布的直方图，这样就会增加图像的对比度。在连续的情况下，这意味着转换函数（也称为累积分布函数，CDF）将图像的灰度级从原始范围映射到整个灰度级范围。

变换公式

设原始图像的灰度级为 $[0, L-1]$，L 为可能的灰度级数量（例如，对于 8 位图像，$L=256$）。设 $p_r(r_k)$ 为原始图像中灰度级 r_k 的概率密度函数（PDF），即 r_k 级灰度出现的概率。累积分布函数（CDF）定义为从最低灰度级到当前级别 r_k 的 $p_r(r)$ 的累积和，表示为 s_k：

$$s_k = \sum_{j=0}^{k} p_r(r_j) = CDF(r_k)$$

直方图均衡化中，新的灰度级 s_k 是通过将原始图像的 CDF 乘以最大灰度级 $L-1$ 并四舍五入到最近的整数来计算的：

$$s_k = \left\lfloor (L-1) \times \sum_{j=0}^{k} p_r(r_j) \right\rfloor = \lfloor (L-1) \times CDF(r_k) \rfloor$$

这里的 s_k 就是均衡化后的灰度级，所有原始图像中的 r_k 灰度值都被映射到这个新值。

实现过程:

1. 计算原始图像的直方图。
2. 计算直方图的累积分布函数（CDF）。
3. 使用上述变换公式创建一个映射，将旧的灰度级映射到新的灰度级。
4. 使用这个映射重新映射原始图像的每个像素值。

2. 代码实现

```
clc;
clear all;
close all;

```

```matlab
b = imread("student.png");
b = rgb2gray(b);

figure(1);
imshow(b);
title("(a) 原图像");
saveas(gcf, '原图像.jpg');

heq = histeq(b);
figure(2);
imshow(heq);
title("(b) 直方图均衡化后的图像");
saveas(gcf, '直方图均衡化后的图像.jpg');

figure(3);
subplot(2, 1, 1);
imhist(b);
title("(c) 原始图像的直方图");
subplot(2, 1, 2);
imhist(heq);
title("(d) 直方图均衡化后的图像的直方图");
saveas(gcf, '直方图对比.jpg');

I = imbinarize(heq);
fa = fft2(I);
ffa = fftshift(fa);

figure(4);
magnitude_spectrum = abs(ffa);  % 计算幅度谱
imshow(log(magnitude_spectrum + 1), []);  % 使用对数尺度增强显示效果
title("(e) 幅度谱（直方图均衡化后）");
saveas(gcf, '幅度谱（直方图均衡化后）.jpg');

figure(5);
mesh(log(magnitude_spectrum + 1));  % 对幅度进行对数变换来更好地显示能量分布
title("(f) 幅度谱的能量分布（直方图均衡化后）");
saveas(gcf, '幅度谱能量分布（直方图均衡化后）.jpg');
```


代码首先读取图像并将其转换为灰度图。接着，代码使用直方图均衡化处理这个灰度图像以增强其对比度，并展示及保存处理后的结果。为了对比直方图均衡化前后的效果，代码同时显示并保存了原始图像和均衡化后图像的直方图。之后，对直方图均衡化后的图像进行二值化处理，并对这个二值化后的图像执行快速傅立叶变换（FFT），将 FFT 的直流分量移到频谱的中心。最后，计算并显示了 FFT 结果的幅度谱，并使用 mesh 图展示了幅度谱的能量分布，这些过程也都是基于直方图均衡化后的图像进行的，从而可以对比观察直方图均衡化对图像处理步骤的影响。

3. 结果分析

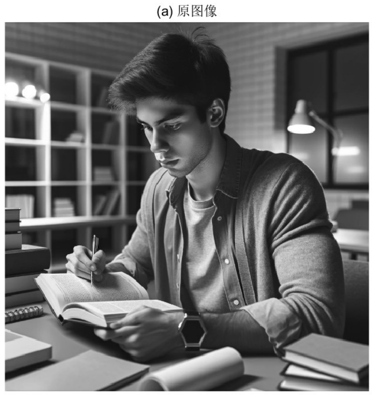

图 12: 原图像

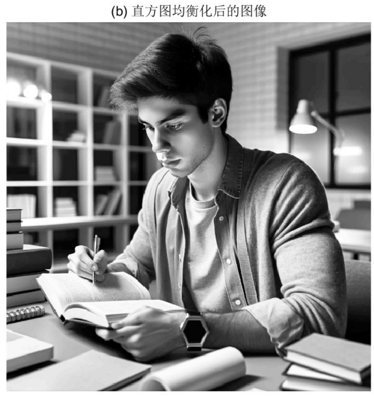

图 13: 直方图均衡化后的图像

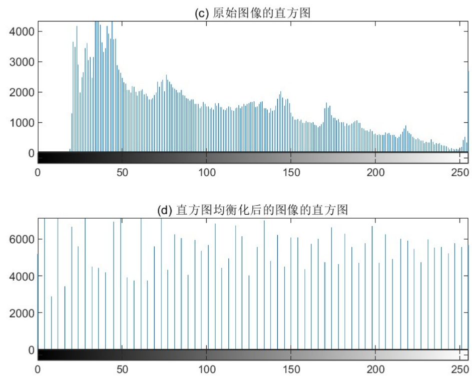

图 14: 直方图对比

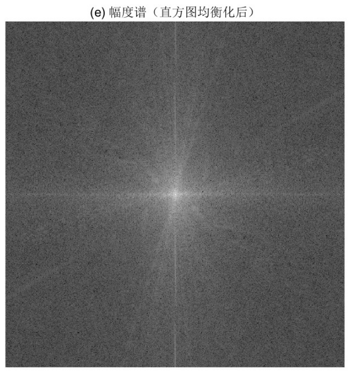

图 15: 幅度谱（直方图均衡化后）

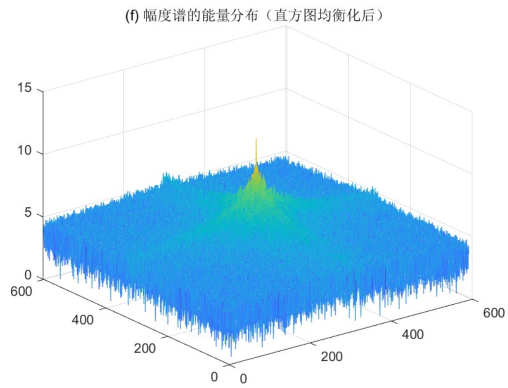

图 16: 幅度谱能量分布（直方图均衡化后）

- **原始图像与直方图均衡化后的图像（图 12/图 13）**：第一和第二张照片（图 12）和（图 13）分别是原图片和直方图均衡化过程的示例对象。与原始图像相比，均衡化后的图像预期会

显示出增强的对比度。

- **直方图均衡化前后（图 14）**：原始直方图（图 14-上）有集中的柱形图，表明特定亮度级别的集中，而均衡化后的直方图（图 14-下）跨越整个光谱，表明图像的亮度级别均匀分布，从而增加了图像的对比度。
- **幅度谱（图 15）**：展示了对直方图均衡化后的图像执行傅立叶变换后的幅度谱。分析将图像的空间域转换为频率域，揭示图像中存在的不同频率（或细节）。亮中心表明低频率（总体形状和边界），而周围区域代表高频率（细节）。
- **幅度谱能量分布（图 16）**：图 16 是幅度谱的三维表示，从不同角度提供了图像中频率分布的视图。图中的峰值对应于图像中普遍存在的频率或模式。

六、 实验心得体会

本次实验我完成了离散傅里叶变换（DFT）、离散小波变换（DWT）、离散余弦变换（DCT）、快速傅里叶变换（FFT）以及直方图均衡化的一系列实验后，我对数字信号处理有了更深刻的理解。这些实验不仅加深了我对于这些变换在理论上的认识，而且通过实际的编程应用，让我了解了它们在图像处理中的具体效果和应用场景。特别是，FFT 的优化算法对于加快大数据集的处理速度展现了巨大的优势，而 DWT 在提供时间和频率局部化信息方面的独特效用让我认识到了它在处理非平稳信号时的重要性。通过 DCT 实验，我深入理解了它在图像压缩中的应用，尤其是 JPEG 标准中的关键作用。直方图均衡化实验则让我体会到了图像对比度调整的重要性，并且了解了如何有效地增强图像质量。编程实践加深了我的理论知识，提高了我解决实际问题的能力，同时也让我意识到了理论与实践之间的差距。通过不断试验和调整，我学会了如何更好地应用这些技术来解决具体问题。这一系列的实验不仅让我掌握了重要的图像处理技术，还激发了我将这些技能应用于更广泛领域的兴趣和信心。