

对于周期信号，其傅里叶变换在数学上等价于傅里叶级数展开，而傅里叶系数就是这个级数展开中的系数。而对于非周期信号，傅里叶变换通常指的是傅里叶变换的连续形式，或者是其离散形式，即离散傅里叶变换（DFT），后者是快速傅里叶变换（FFT）的基础。以下是两者的对比：

1. **定义：**

傅里叶变换：是一种数学变换，用于将时间域或空间域的信号转换到频率域，以分析信号的频率成分。

傅里叶系数：是周期信号在进行傅里叶级数展开时，对应于不同频率成分的系数。

2. **应用：**

傅里叶变换：主要应用于非周期信号的频谱分析，揭示信号的频率组成和能量分布。

傅里叶系数：用于周期信号的频谱分析，通过系数可以重建信号的周期性部分。

3. **形式：**

傅里叶变换：可以是连续的（连续时间信号的傅里叶变换）或离散的（离散时间信号的离散傅里叶变换）。

傅里叶系数：是离散的数值，对应于周期信号的特定频率成分。

4. **物理意义：**

傅里叶变换：显示了信号在频率域的分布，有助于理解信号的频率特性和能量分布。

傅里叶系数：描述了周期信号中各个频率成分的幅度和相位，是信号周期性特征的量化表示。

5. 系数：

傅里叶变换：对于非周期信号，频谱通常由连续函数表示，而不是具体的系数。

傅里叶系数：是周期信号分解后的离散数值，代表了信号中各个正弦波和余弦波的相对强度和相位。

6. 重建信号：

傅里叶变换：通过叠加无限多个频率成分来重建非周期信号。

傅里叶系数：通过有限或无限个系数来重建周期信号。

7. 应用领域：

傅里叶变换：广泛应用于信号处理、图像分析、通信系统等领域。

傅里叶系数：常用于电力系统分析、声学、音频处理等领域。

8. 计算：

傅里叶变换：通常需要使用数值方法，如快速傅里叶变换（FFT）来计算。

傅里叶系数：可以通过解析方法（对于简单周期信号）或数值方法（对于复杂周期信号）来计算。

9. 信号类型：

傅里叶变换：适用于所有类型的信号，无论是周期的还是非周期的。

傅里叶系数：仅适用于周期信号，因为它们描述了信号的周期性特征。

用 MATLAB 实现旋转和放大

```
img = imread('实验图像.bmp');
gray_img = rgb2gray(img);

% 获取图像尺寸
[rows, cols] = size(gray_img);

% 计算旋转中心
cx = cols / 2;
cy = rows / 2;

% 旋转角度（逆时针为正，顺时针为负）
theta = 30; % 顺时针旋转30度

% 计算旋转矩阵
cos_theta = cosd(theta);
sin_theta = sind(theta);
```

读入后预处理：计算尺寸，中心，旋转矩阵

```
% 旋转图像
rotated_img = zeros(rows, cols, 'like', gray_img);
for x = 1:cols
    for y = 1:rows
        x_new = round(cos_theta * (x - cx) - sin_theta * (y - cy) + cx);
        y_new = round(sin_theta * (x - cx) + cos_theta * (y - cy) + cy);
        if x_new >= 1 && x_new <= cols && y_new >= 1 && y_new <= rows
            rotated_img(y_new, x_new) = gray_img(y, x);
        end
    end
end

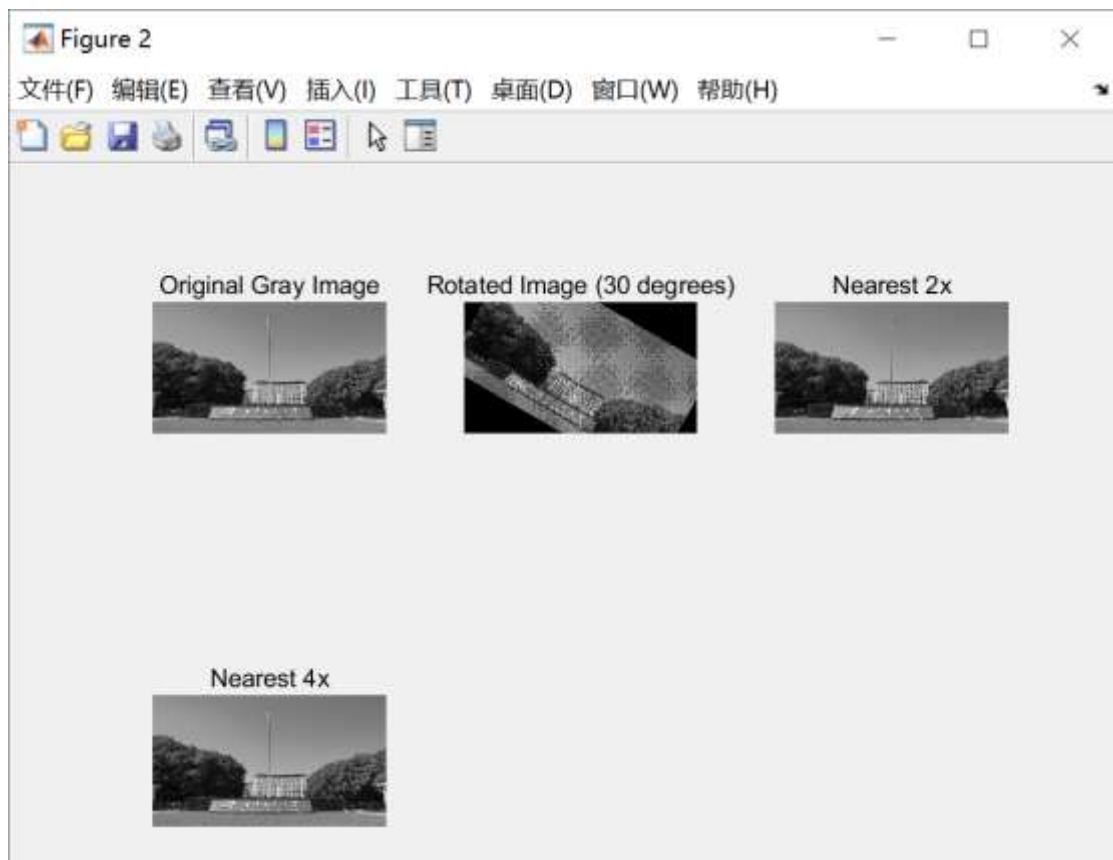
% 放大图像 - 最近邻插值
nearest_2x = zeros(2 * rows, 2 * cols, 'like', gray_img);
for i = 1:2 * rows
    for j = 1:2 * cols
        i_new = (i + 1) / 2;
        j_new = (j + 1) / 2;
        i_new = round(i_new);
        j_new = round(j_new);
        if i_new >= 1 && i_new <= rows && j_new >= 1 && j_new <= cols
            nearest_2x(i, j) = rotated_img(i_new, j_new);
        end
    end
end
```

四倍放大同理

```

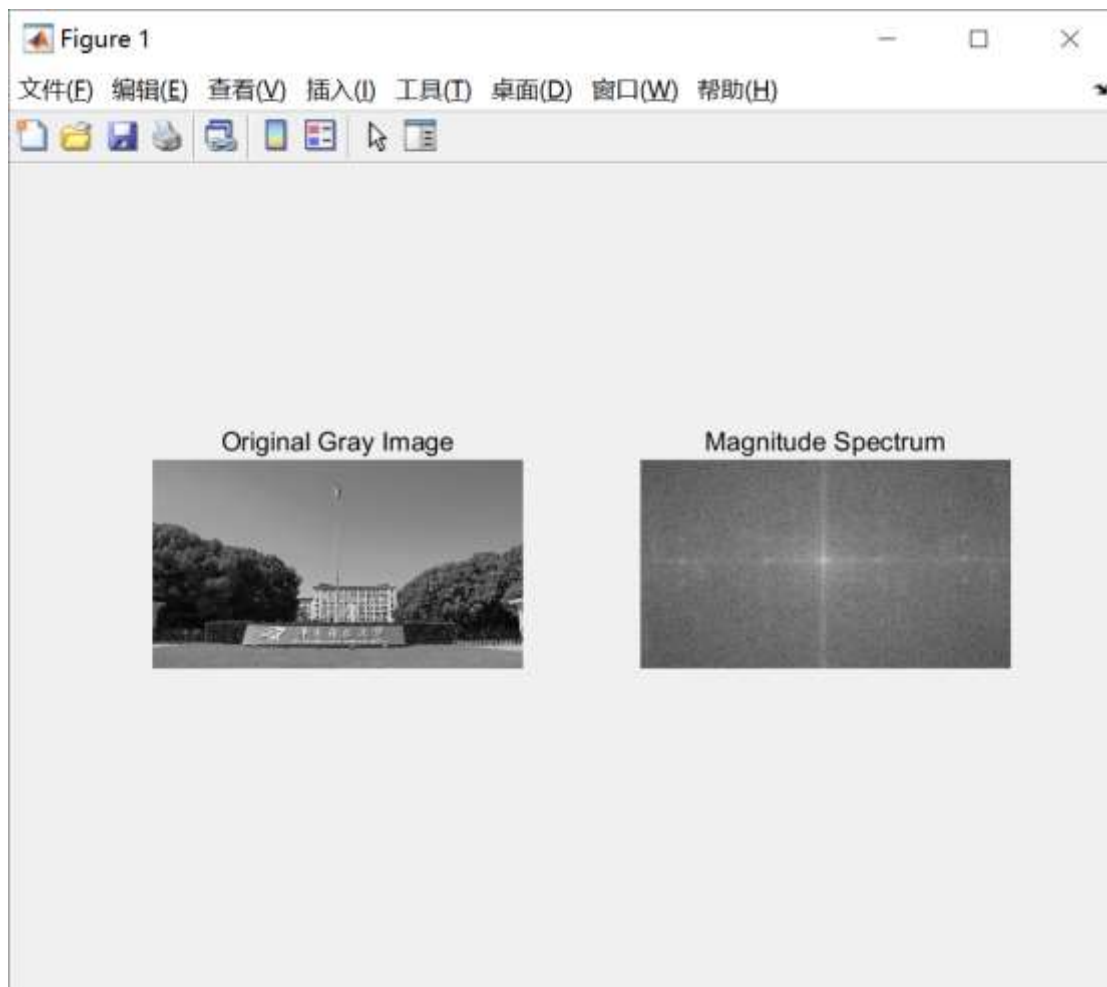
nearest_4x = zeros(4 * rows, 4 * cols, 'like', gray_img);
for i = 1:4 * rows
    for j = 1:4 * cols
        i_new = (i + 1) / 4;
        j_new = (j + 1) / 4;
        i_new = round(i_new);
        j_new = round(j_new);
        if i_new >= 1 && i_new <= rows && j_new >= 1 && j_new <= cols
            nearest_4x(i, j) = rotated_img(i_new, j_new);
        end
    end
end
end

```



傅里叶变换

效果如下：



```
img = imread('实验图像.bmp');  
gray_img = rgb2gray(img);  
% 计算傅里叶变换  
F = fft2(double(gray_img));  
  
% 将频率原点移至图像中心  
F_shifted = fftshift(F);  
  
% 计算幅度谱  
magnitude_spectrum = log(abs(F_shifted) + 1);
```

调用函数 FFT2 对灰度图傅里叶变换，再调用 FFTshift 移到中心

李金龙