请尝试阐述傅里叶变换与傅里叶系数的物理含义：

傅里叶变换：傅里叶变换是一种数学变换，将一个时间域（或空间域）的信号转换为频率域表示。傅里叶变换允许我们理解和分析信号中包含的不同频率成分。通过将复杂的信号分解为简单的正弦波叠加，我们可以识别出信号的频谱特征，比如主频、带宽等。

傅里叶系数：在周期函数的傅里叶级数展开中，傅里叶系数是用来描述信号在各个频率上的权重或幅度。傅里叶系数表明了信号中每个特定频率成分的贡献大小。它们反映了信号的周期性特征以及各个频率成分在整体信号中的重要性。

编写程序（建议 Matlab）对以上图像（自行转换为灰度图）展开（1）顺时针旋转 30 度；（2）基于最近邻和双线性插值将图像分别放大 2 倍和 4 倍，并形成实验报告:

1. 转换为灰度:

```
function grayImg = customGrayConversion(colorImage)
    [height, width, ~] = size(colorImage);
    grayImg = zeros(height, width, 'like', colorImage);
    for y = 1:height
        for x = 1:width
            r = colorImage(y, x, 1);
            g = colorImage(y, x, 2);
            b = colorImage(y, x, 3);
            % 使用加权平均法转换为灰度
            grayImg(y, x) = 0.2989 * r + 0.5870 * g + 0.1140 * b;
        end
    end
end
```

2. 顺时针旋转:

```
% 图像顺时针旋转 30 度
function rotatedImage = rotateImage(image, angle)
    [height, width] = size(image);
    centerX = width / 2;
    centerY = height / 2;
    rotatedImage = zeros(height, width, 'like', image);
    radianAngle = deg2rad(angle);
    for y = 1:height
        for x = 1:width
            newX = round((cos(radianAngle) * (x - centerX) - 
sin(radianAngle) * (y - centerY)) + centerX);
            newY = round((sin(radianAngle) * (x - centerX) + 
cos(radianAngle) * (y - centerY)) + centerY);
            if newX > 0 && newX <= width && newY > 0 && newY <= height
                rotatedImage(newY, newX) = image(y, x);
            end
        end
    end
```

```
end
```

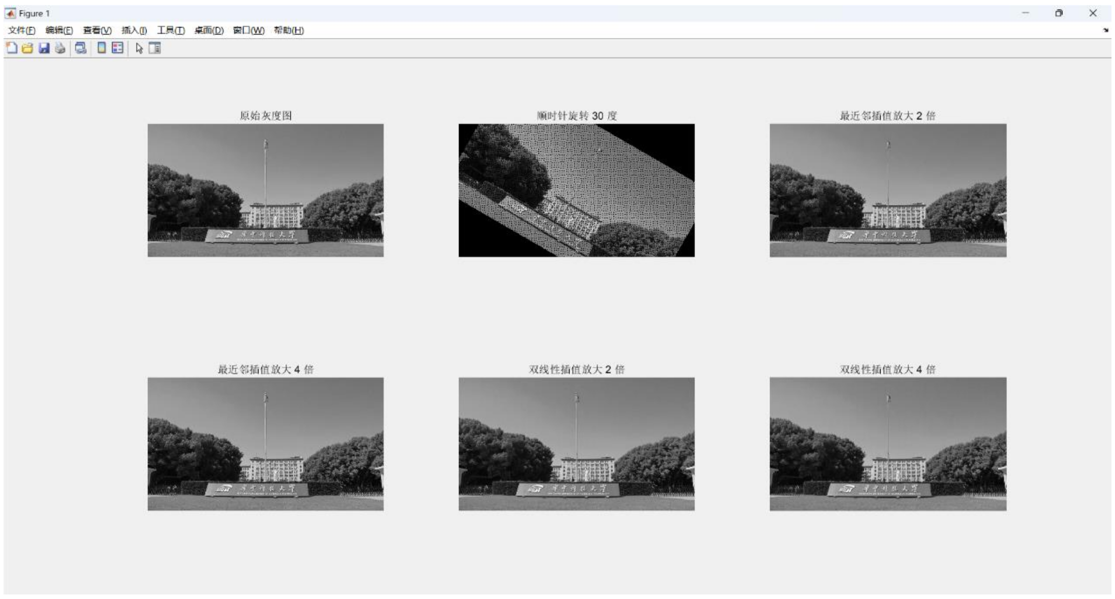
3. 最近邻插值放大：

```
% 最近邻插值放大图像
function enlargedImage = nearestInterpolation(image, scaleFactor)
    [height, width] = size(image);
    newHeight = round(height * scaleFactor);
    newWidth = round(width * scaleFactor);
    enlargedImage = zeros(height, width, 'like', image);
    for y = 1:newHeight
        for x = 1:newWidth
            oldY = round(y / scaleFactor);
            oldX = round(x / scaleFactor);
            if oldY > 0 && oldY <= height && oldX > 0 && oldX <= width
                enlargedImage(y, x) = image(oldY, oldX);
            end
        end
    end
end
```

4. 双线性插值放大

```
% 双线性插值放大图像
function enlargedImage = bilinearInterpolation(image, scaleFactor)
    [height, width] = size(image);
    newHeight = round(height * scaleFactor);
    newWidth = round(width * scaleFactor);
    enlargedImage = zeros(height, width, 'like', image);
    for y = 1:newHeight
        for x = 1:newWidth
            oldY = y / scaleFactor;
            oldX = x / scaleFactor;
            y1 = floor(oldY);
            y2 = y1 + 1;
            x1 = floor(oldX);
            x2 = x1 + 1;
            if y1 > 0 && y1 <= height && y2 > 0 && y2 <= height && x1 > 0 &&
x1 <= width && x2 > 0 && x2 <= width
                q11 = image(y1, x1);
                q12 = image(y2, x1);
                q21 = image(y1, x2);
                q22 = image(y2, x2);
                fy1 = q11 * (x2 - oldX) + q21 * (oldX - x1);
                fy2 = q12 * (x2 - oldX) + q22 * (oldX - x1);
                enlargedImage(y, x) = fy1 * (y2 - oldY) + fy2 * (oldY - y1);
            end
        end
    end
end
```

运行结果:



编写程序（建议 Matlab）对以上图像（自行转换为灰度图）展开傅里叶变换，提取傅里叶变换图像（将频率原点移至图像中心），并形成实验报告:

开始使用嵌套循环实现 FFT 效率很低，根本跑不出来，后来改为行列分别计算，将嵌套改为并列，效率提高很多，大概十几秒跑出结果，但是都不如直接调用 matlab 内置函数，一秒内就可以跑出结果。

代码:

```
% 读取图像并转换为灰度图
I = imread('$RDLCZLX.bmp');
I = rgb2gray(I);
I = im2double(I);
[x, y] = size(I);

% 定义复数常量
com = 0 + 1i;

% 对每一列进行 DFT
Ax = zeros(x, y);
for m = 1:y
    for n = 1:x
        sumCol = 0;
```

```
            for u = 1:x
                sumCol = sumCol + I(u, m) * exp(-com * 2 * pi * ((n - 1) * (u - 1)/x));
            end
            Ax(n, m) = sumCol;
        end
    end

    % 对每一行进行 DFT
    ans = zeros(x, y);
    for k = 1:x
        for l = 1:y
            sumRow = 0;
            for v = 1:y
                sumRow = sumRow + Ax(k, v) * exp(-com * 2 * pi * ((l - 1) * (v - 1)/y));
            end
            ans(k, l) = sumRow;
        end
    end

    % 移频
    F = fftshift(ans);
    F = abs(F);
    F = log(F + 1);

    % 显示结果
    figure;
    imshow(F, []);
```

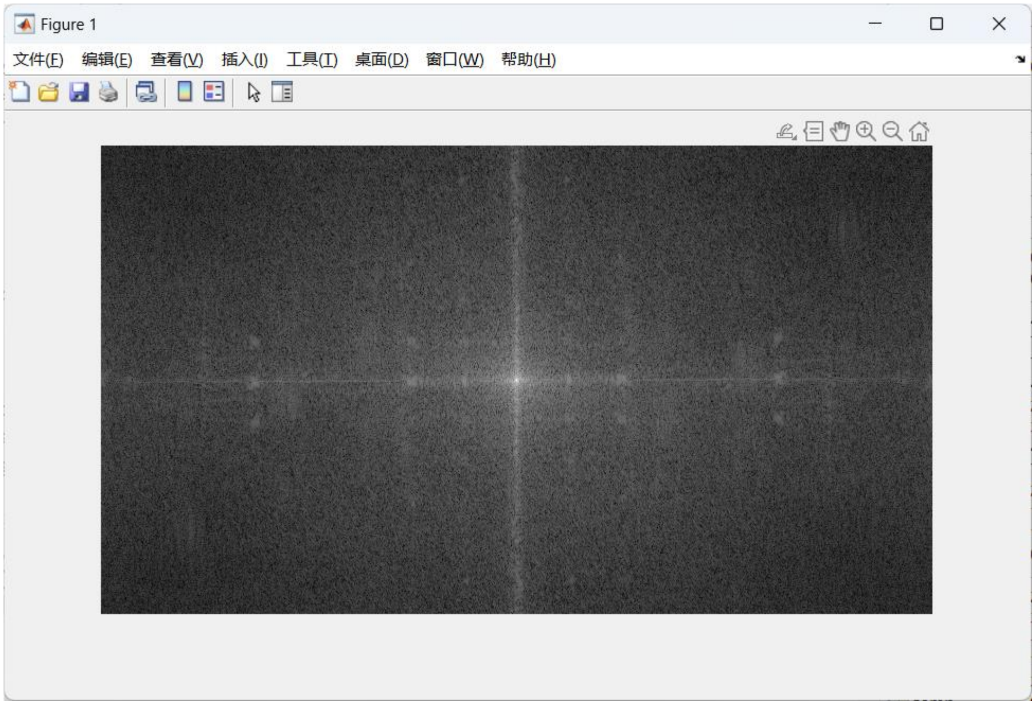



Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)