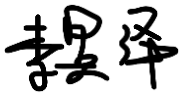
李昊泽 U202211174 人工智能 2204 班



- **结合授课内容和自身的理解，请尝试阐述傅里叶变换与傅里叶系数的物理含义，可以上网查阅相关的资料。**

傅里叶变换的物理含义

1. **频率分解**：傅里叶变换将一个信号（如声音、图像等）分解为一系列不同频率的正弦波（或余弦波）的叠加。这些正弦波（或余弦波）的幅度和相位信息由傅里叶变换的复数结果给出。
2. **频谱分析**：通过傅里叶变换，我们可以得到信号的频谱，即信号中各种频率成分的分布情况。频谱分析是信号处理中的一个重要环节，它有助于我们了解信号的频率特性，如哪些频率成分较强，哪些较弱，以及是否存在特定的频率模式等。
3. **滤波**：基于傅里叶变换的滤波技术（如低通滤波、高通滤波、带通滤波等）允许我们根据信号的频率特性对其进行处理。通过修改频谱中的某些频率成分，我们可以实现信号的平滑、去噪、增强等目的。
4. **压缩与编码**：在信号压缩和编码领域，傅里叶变换也发挥着重要作用。通过去除频谱中的冗余信息或低能量成分，我们可以实现信号的有效压缩。同时，基于傅里叶变换的编码方法（如 JPEG 图像压缩）也广泛应用于多媒体领域。

傅里叶系数的物理含义

1. **幅度**：傅里叶系数的模（即复数的绝对值）表示对应频率成分在信号中的幅度大小。幅度越大，说明该频率成分在信号中越显著。
2. **相位**：傅里叶系数的辐角（即复数的相位角）表示对应频率成分与参考信号（如单位正弦波）之间的相位差。相位信息对于理解信号的波形形状和时域特性至关重要。
3. **能量分布**：傅里叶系数的平方（即幅度的平方）表示对应频率成分在信号中的能量分布。通过计算频谱中各频率成分的能量分布，我们可以进一步了解信号的能量特性。

- **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）图像转换为灰度图

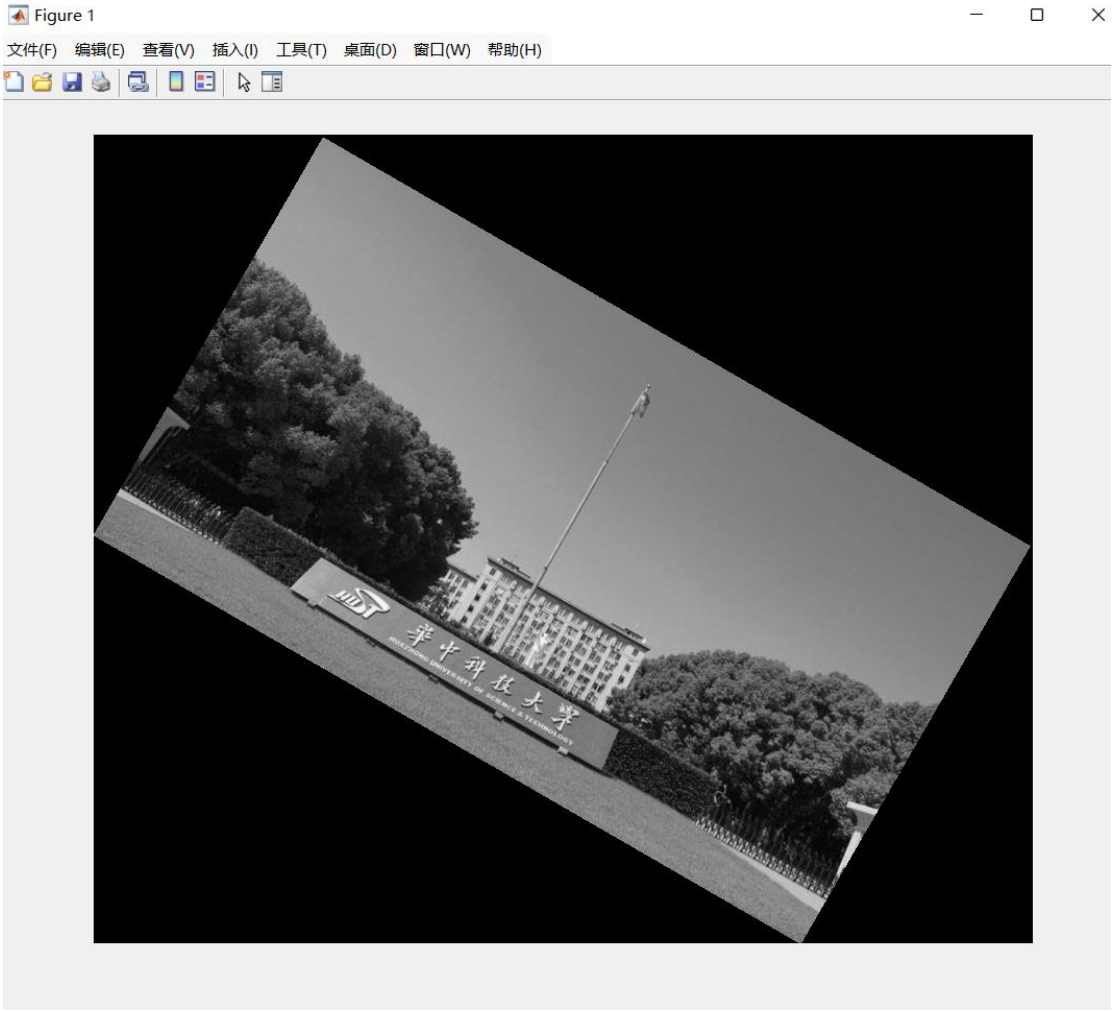
```
gray_img = zeros(size(test, 1), size(test, 2));
for i = 1:size(test, 1)
for j = 1:size(test, 2)
r = test(i, j, 1);
g = test(i, j, 2);
b = test(i, j, 3);
gray_img(i, j) = 0.299 * r + 0.587 * g + 0.114 * b;
end
end
gray_img=uint8(gray_img);
figure;
imshow(gray_img);
title('Image1');
```

（2）顺时针旋转 30 度实验代码

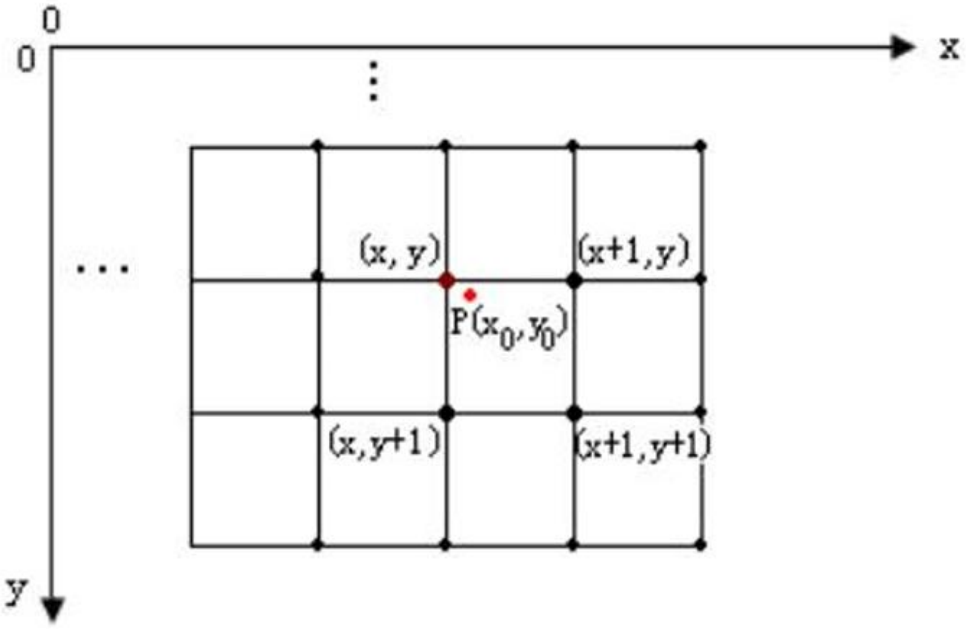
```
angle=30;
RGB=imread('D:\实验图像.bmp');
I = rgb2gray(RGB);%将图像灰度化
imshow(I)
[h,w,d]=size(I);
theta=angle/180*pi;
M=[cos(theta) -sin(theta) 0;sin(theta) cos(theta) 0;0 0 1];
leftup=[1 1 1]*M;
rightup=[1 w 1]*M;
leftdown=[h 1 1]*M;
rightdown=[h w 1]*M;
cos_val = cos(theta);
sin_val = sin(theta);

w2=round(abs(cos_val)*w+h*abs(sin_val));%新图宽度
h2=round(abs(cos_val)*h+w*abs(sin_val));%新图高度
img_rotate=uint8(zeros(h2,w2,3));
dy=abs(min([leftup(1) rightup(1) leftdown(1) rightdown(1)]));
```

```
dx=abs(min([leftup(2) rightup(2) leftdown(2) rightdown(2)]));
for i=1-dy:h2-dy
    for j=1-dx:w2-dx
        pix=[i j 1]/M;
        yy=pix(1)-floor(pix(1));
        xx=pix(2)-floor(pix(2));
        if pix(1)>=1 && pix(2)>=1 && pix(1) <= h && pix(2) <= w
            x11=floor(pix(1));          %四个相邻的点
            y11=floor(pix(2));
            x12=floor(pix(1));
            y12=ceil(pix(2));
            x21=ceil(pix(1));
            y21=floor(pix(2));
            x22=ceil(pix(1));
            y22=ceil(pix(2));

            value_leftup=(1-xx)*(1-yy);                 %计算临近四个点在双线性插值中
的权重
            value_rightup=xx*(1-yy);
            value_leftdown=(1-xx)*yy;
            value_rightdown=xx*yy;

            img_rotate(i+dy,j+dx,:)=value_leftup*I(x11,y11,:)+value_rightup*I(x12,y12,:)+
value_leftdown*I(x21,y21,:)+value_rightdown*I(x22,y22,:);
        end
    end
end

imshow(uint8(img_rotate))
```

（3）顺时针旋转 30°实验结果



（4）最近邻插值将图像放大两倍

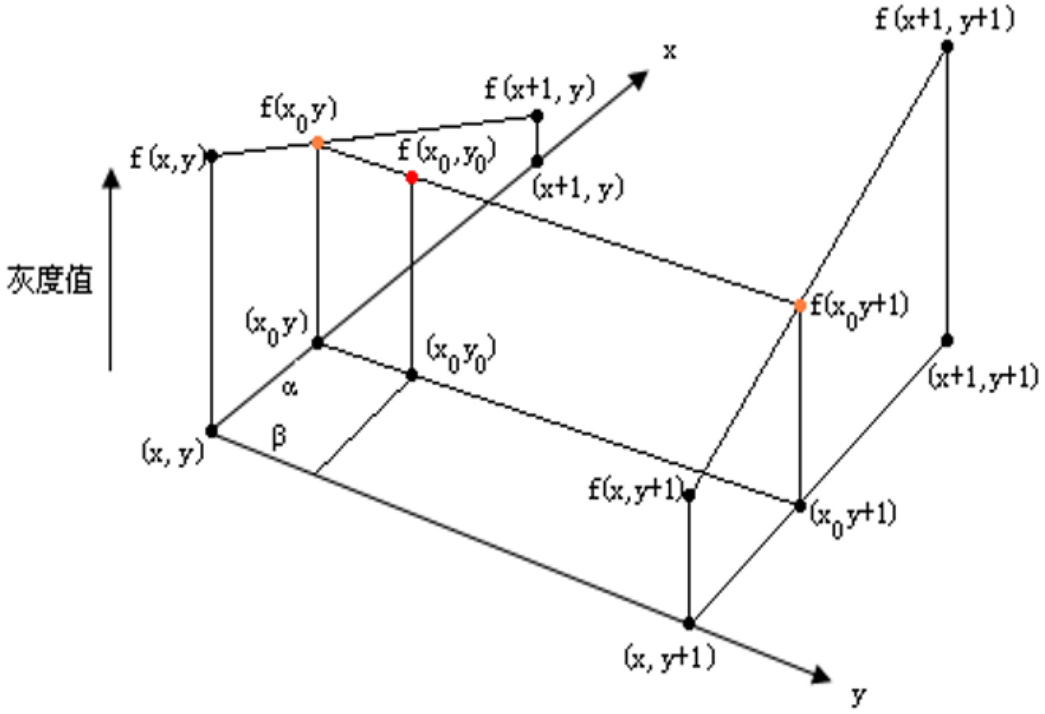



```
clear all
RGB=imread('D:\实验图像.bmp');
```

```
src= rgb2gray(RGB);
ratio = 2;
[row, col, color] = size(src);
row = round(ratio * row);
col = round(ratio * col);
dst = zeros(row, col, color, class(src));

for i = 1 : row
    for j = 1 : col
        x = round(i / ratio);
        y = round(j / ratio);
        if x == 0
            x = x + 1;
        end
        if y == 0
            y = y + 1;
        end
        dst(i, j, : ) = src(x, y, :);
    end
end

figure
imshow(src)
figure
imshow(dst)
>>
```

（5）最近邻插值法图像放大两倍实验结果

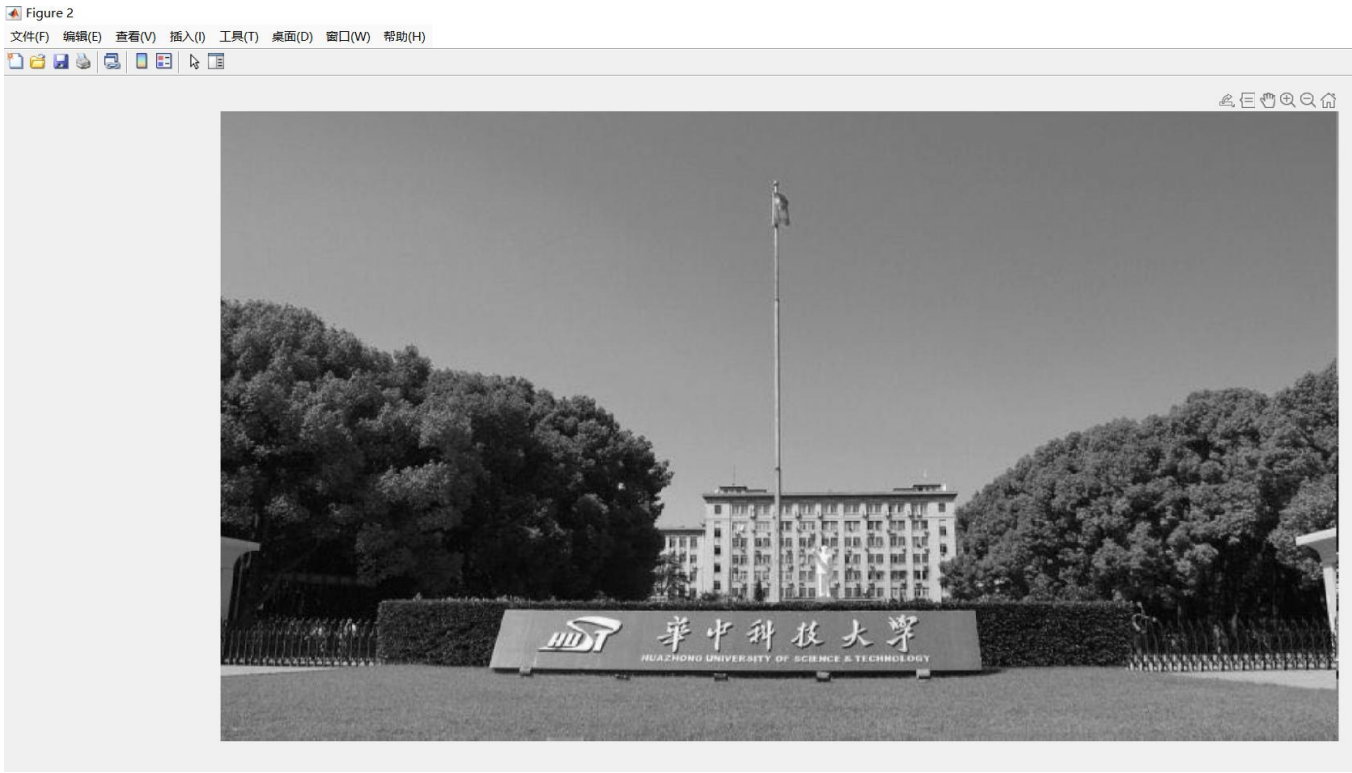
（6）双线性插值图像放大 4 倍




```
>> clear all
RGB=imread('D:\实验图像.bmp');
src= rgb2gray(RGB);
ratio = 4;
[row, col, color] = size(src);
row = round(ratio * row);
col = round(ratio * col);

dst = zeros(row, col, color, class(src));

for i = 1 : row
    for j = 1 : col
        x = i / ratio;
        y = j / ratio;
        x1 = ceil(x);            % 向上取整
        y1 = ceil(y);
        x2 = ceil(x) + 1;
        y2 = ceil(y) + 1;
        if x2 >= size(src, 1)            % 溢出检查
            x2 = x2 - 1;
        end
        if y2 >= size(src, 2)
            y2 = x2 - 1;
        end
        du = (x+1) - x1;
        dv = (y+1) - y1;
```

```
            dst(i, j, :) = (1-du)*(1-dv)*src(x1, y1, :) + (1-du)*dv*src(x1, y2, :) + du*(1-dv)*src(x2, y1, :) + du*dv*src(x2, y2, :);
        end
end

figure
imshow(src)
figure
imshow(dst)
```

（7）双线性插值图像放大 4 倍



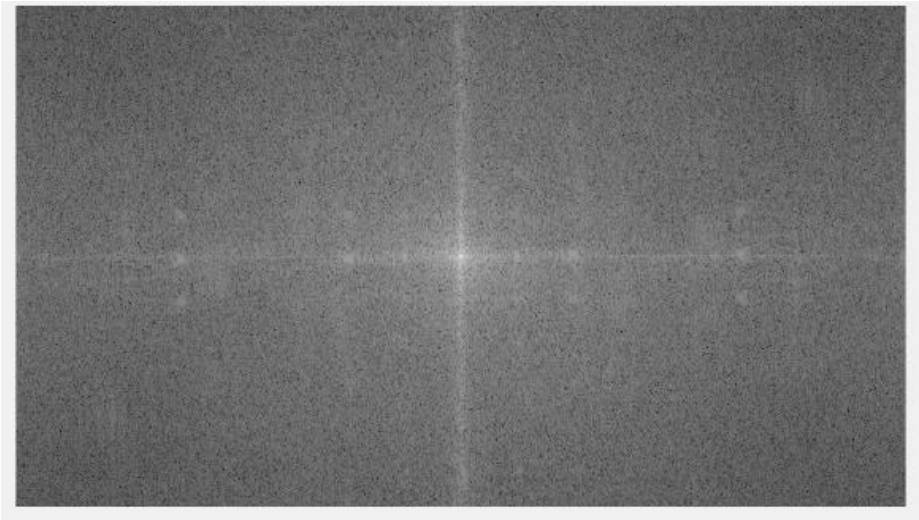
- **编写程序（建议 Matlab）对以上图像（自行转换为灰度图）展开傅里叶变换，提取傅里叶变换图像（将频率原点移至图像中心），并形成实验报告。**

（1）实验代码

```
function magnitude_spectrum = perform_dft(gray_img)
[M,N]=size(gray_img);
Wm = exp(-1i*2*pi/M);
Wn = exp(-1i*2*pi/N);
Em = zeros(M);
En = zeros(N); % E 是辅助计算矩阵
Gm = zeros(M)+Wm;
Gn = zeros(N)+Wn; % G 是计算时要用的矩阵
F = zeros(M,N); % F 是转换到频域的结果
E = zeros(M,N);
for row = 0:M-1
for col=0:M-1
```

```
Em(row+1,col+1) = row * col;
Gm(row+1,col+1) = Gm(row+1,col+1)^Em(row+1,col+1);
end
end
for row = 0:N-1
for col =0:N-1
En(row+1,col+1) = row * col;
Gn(row+1,col+1) = Gn(row+1,col+1)^En(row+1,col+1);
end
end
for row =1:M
%变换到图像中点
for col =1:N
E(row,col)=double(gray_img(row,col))*((-1)^(row+col));
end
end
F = real(Gm*E*Gn);
magnitude_spectrum = log(1 + abs(F));
figure;
imshow(magnitude_spectrum, []);
end
```

（2）实验结果



这幅图的频率成分较为复杂，从上图可以看出中心部分也就是低频成分较多，即广阔的天空，也有一些高频成分，即图片中的树木以及南一楼等。