

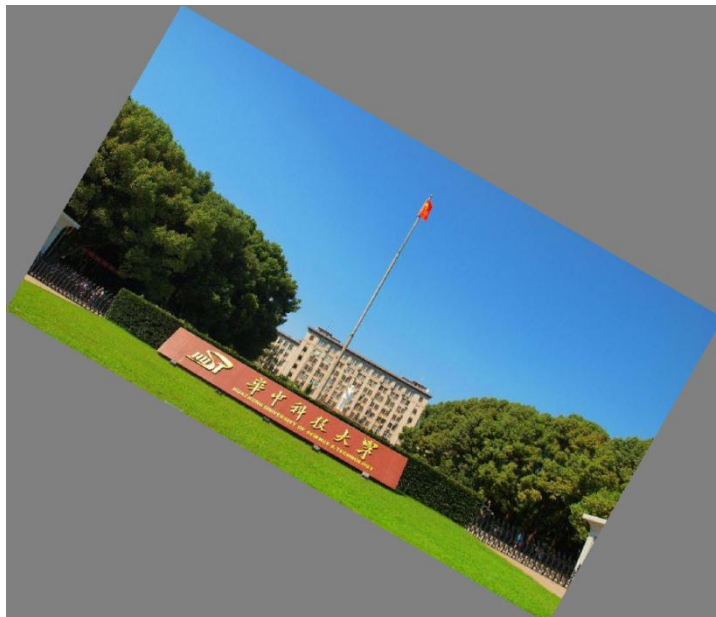
傅里叶级数：所有周期信号都可以分解为不同频率的各次谐波分量。

傅里叶变换：非周期信号可以看作不同频率的余弦分量叠加，其中频率分量可以从 0 到无穷大任意频率，而不是像傅里叶级数一样由离散的谐波分量组成。

实现图片的旋转

```
im = imread('D:\MATLAB代码\实验图像.bmp');
a = 30 / 180 * pi;
R = [cos(a), sin(a); -sin(a), cos(a)];% 求出旋转矩阵
R = R'; % 求出旋转矩阵的逆矩阵进行逆向查找
% 计算原图大小
sz = size(im);
h = sz(1);%高度
w = sz(2);%宽度
ch = sz(3);%通道数
c1 = [h;w] / 2;
% 计算显示完整图像需要的画布大小
hh = floor(w*sin(a)+h*cos(a))+1;
ww = floor(w*cos(a)+h*sin(a))+1;
c2 = [hh;ww] / 2;
% 初始化目标画布
im2 = uint8(ones(hh,ww,3)*128);
for k = 1:ch
    for i = 1:hh
        for j = 1:ww
            p = [i;j];
            pp = round(R*(p-c2)+c1);
            % 逆向进行像素查找
            if (pp(1) >= 1 && pp(1) <= h && pp(2) >= 1 && pp(2) <= w)
                im2(i, j, k) = im(pp(1), pp(2), k);
            end
        end
    end
end
figure;
imshow(im2);% 显示图像
```

结果如下



实现图片的放大

```

% 加载原始图像
im = imread('D:\MATLAB代码\实验图像.bmp');

% 定义放大比例
scale_factor = 2;

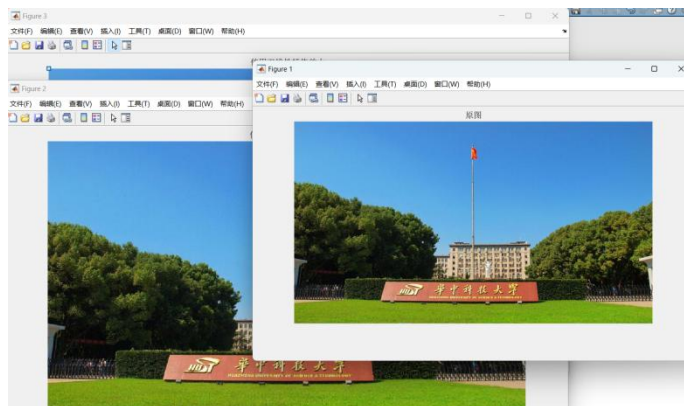
% 使用最近邻插值放大图像
bigger_im_nearest = imresize(im, scale_factor, 'nearest');

% 使用双线性插值放大图像
bigger_im_bicubic = imresize(im, [size(im,1)*scale_factor, size(im,2)*scale_factor], 'bicubic');

% 显示原图和放大后的图像
figure;
imshow(im);
title('原图');
figure;
imshow(bigger_im_nearest);
title('使用最近邻插值放大');
figure;
imshow(bigger_im_bicubic);
title('使用双线性插值放大');

```

可通过改变放大比例来实现 2 倍放大与 4 倍放大
结果如下



对图像提取傅里叶变换图像，并将频率原点移动到图像中心

```

im = imread('D:\MATLAB代码\实验图像.bmp');
im = rgb2gray(im);

im_FFT = fft2(im);
im_FFT_Shift = fftshift(im_FFT);
im_AM = log(abs(im_FFT_Shift));
im_Phase = log(angle(im_FFT_Shift)*180/pi);
figure;
subplot(141), imshow(im), title('原图');
subplot(142), imshow(im_FFT_Shift, []), title('傅里叶变换');
subplot(143), imshow(im_AM, []), title('图像幅度');
subplot(144), imshow(im_Phase, []), title('图像相位');

```

结果如下

