

---

# Part 1: Discrete Fourier Transform

```
% a
% a step 1
rows = 512;
[x,y] = meshgrid(1:rows, 1:rows);

% a step 2
img = sin(0.2 * x) + sin(0.3 * x) + cos(0.4 * x) + sin(sqrt(x .* x +
    y .* y) * 0.15) + sin(sqrt(x .* x + y .* y) * 0.35);
figure;
imshow(img);
title("Original image");

% a step 3
fftImg = fftshift(fft2(img));
% Show the magnitude and phase of DFT for this image
phase = angle(fftImg);
amplitude = abs(fftImg);

figure;
imshow(log(amplitude),[]);
title("Log amplitude");

figure;
imshow(unwrap(phase), []);
title("Phase");

% a step 4
% Multiply the magnitude of DFT with 2
mul = ifftshift(amplitude) * 2;
% Calculate the inverse Discrete Fourier Transform
ifftImg = ifft2(mul);

figure;
imshow(ifftImg);
title("Magnitude multiply 2");

% Explain:
% After multiplying the magnitude of DFT with 2 and applying inverse
    IDF,
% the high frequency pattern increased compared with the original
    image.

% b
img = imread('./images/Cross.jpg');
img = im2double(img);

dft = fft2(img);
fftImg = fftshift(dft);

figure;
imshow(fftImg);
```

---

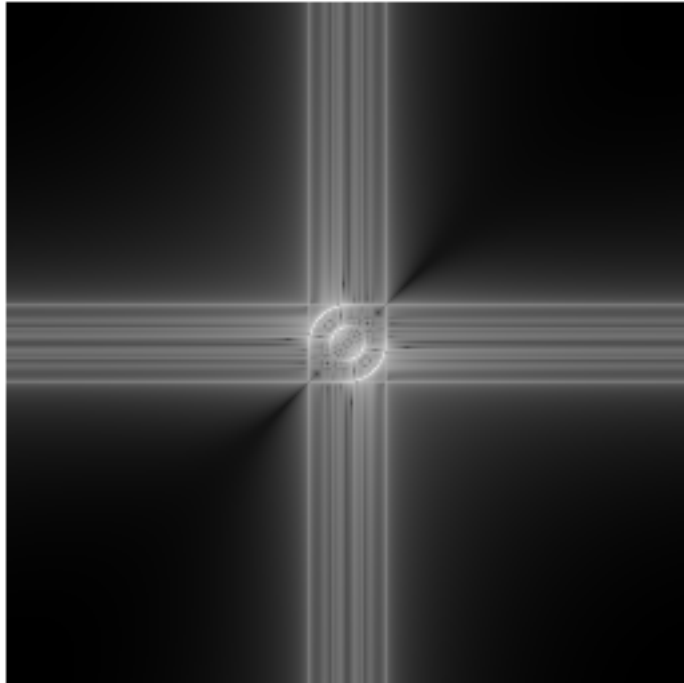
```
% Explain:  
% The diagonal line pattern in original image also exists in the DFT  
  image.
```

*Warning: Displaying real part of complex input.*



---

### Phase

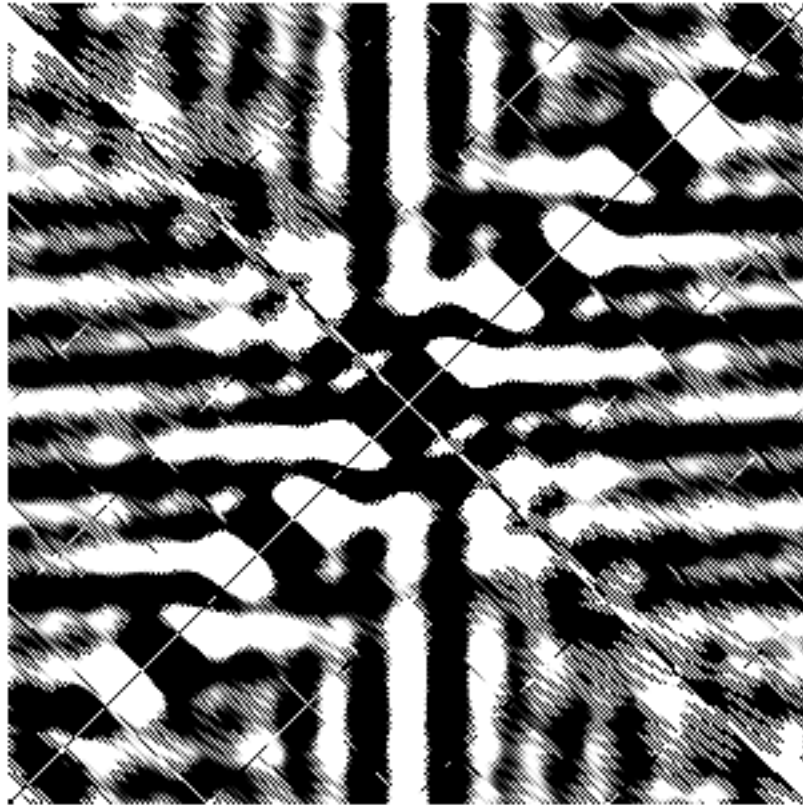




---

Magnitude multiply 2





*Published with MATLAB® R2019b*