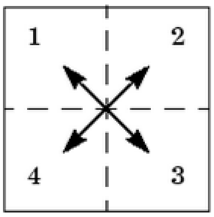
01418588 Practical Image Processing

Department of Computer Science
Kasetsart University

Lab 7 Periodic noise
Asst.Prof.Dr. Pakaket Wattuya

2D Discrete Fourier Transform

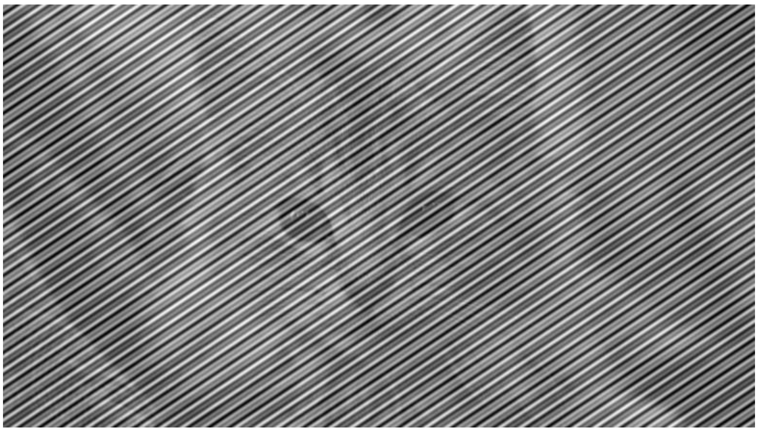
MATLAB functions	Example
`[M,N] = size(X)`	Return the sizes of each dimension of array X in a vector [M,N].
`Y = fftshift(X)`	Shift zero-frequency component to center of spectrum.
`X = ifftshift(Y)`	Inverse FFT shift
`imshow(I,[])`	displays the grayscale image I, scaling the display based on the range of pixel values in I



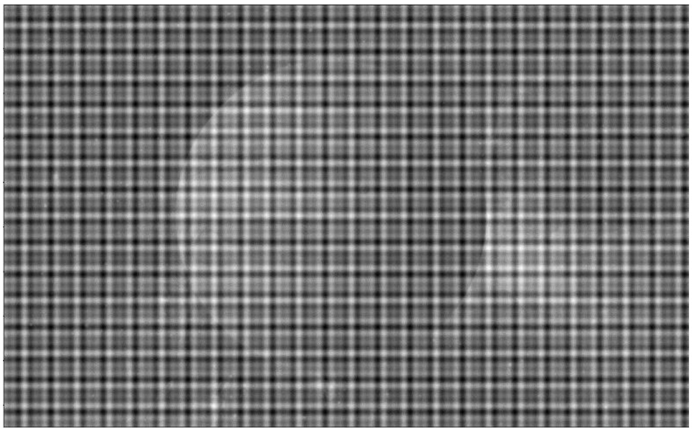
Periodic noise generation

$$f(x,y) = 127 + \left(A \cdot cos\left[\frac{2\pi(ux+vy)}{M}\right] + A \cdot \sin\left[\frac{2\pi(ux+vy)}{N}\right]\right)$$

where *u* and *v* are *x*-axis and *y*-axis spatial frequency parameters, respectively; *A* is an amplitude; *M* and *N* is a dimension of input image.



'freehdw_noisy.bmp': $u = 60$, $v = 40$, $A = 64$

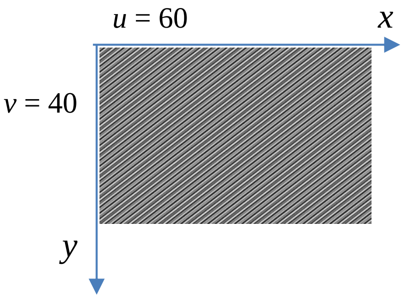


'jupitergray_noisy.bmp': $u = 40$, $v = 0$, $A = 64$
$u = 0$, $v = 40$, $A = 64$

READ INPUT :

```
imNoisy = imread(' freehdw_noisy.bmp');
fftR = fft2(imNoisy);
figure(1); imshow(log(abs(fftshift(fftR))),[]);
```

PERIODIC NOISE PATTERN



```
[M,N] = size(imNoisy);
for x=1:M
   for y=1:N
        fnoise(x,y) = 127 +(A*cos((2*pi*(u*x + v*y))/M)+
                              A*sin((2*pi*(u*x + v*y))/N));
    end
end

fft_noise = fft2(fnoise);
fft_noise_amplitude = log(abs(fft_noise));
figure(); imshow(fftshift(fft_noise_amplitude),[]);

amplitudeThreshold = □
brightSpikes = fft_noise_amplitude > amplitudeThreshold;
figure(); imshow(fftshift(brightSpikes),[]);
```