Frequency domain Image Processing

Outline

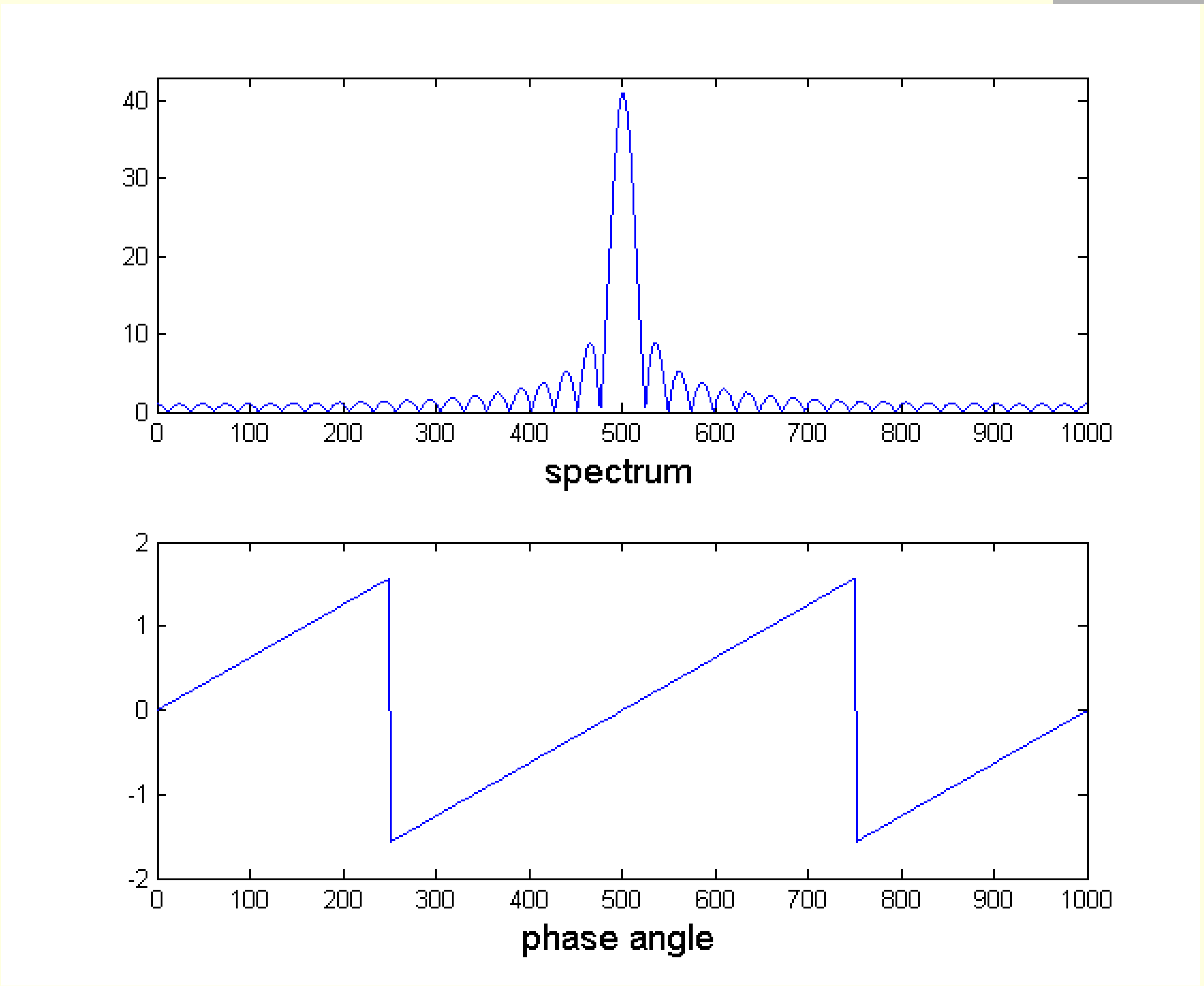
- **Matlab preliminaries**
- **Matlab function design**
- **Shifting frequency component**
- **Low pass filtering design**

Matlab Preliminaries

- Basic commands
  - 2d Fourier transform: F = fft2(f, P, Q);
    - P, Q is for padding, i.e., place the M by N input image f at the center of a larger P by Q matrix.
  - Demonstrating 2d signal(matrix): imshow(f)
  - Absoluate value: abs(f)
    - return spectrum of f if it is complex
  - Move origin of FT to the center of the period: fftshift(F)
    - the same for 1d/2d signals
  - Real or imaginary part of complex signal**:** real(f); imag(f);

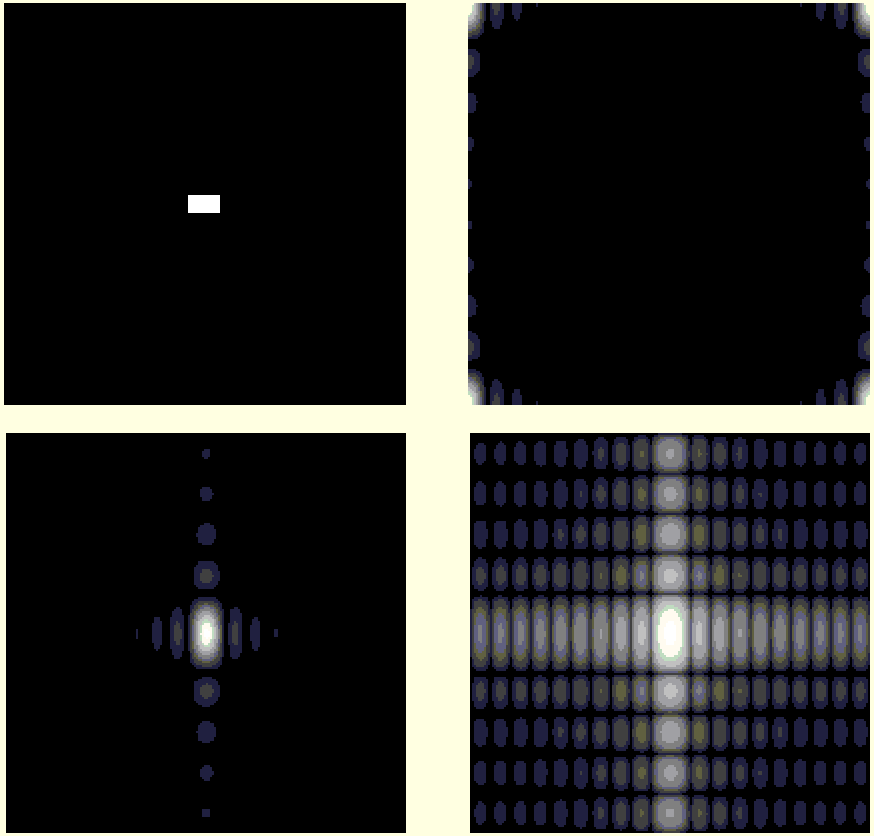
Examples

- Create a simple rectangular 1d signal and examine its Fourier Transform (spectrum and phase angle response).
  - M = 1000;
  - f = zeros(1, M);
  - l = 20;
  - f(M/2-l:M/2+l) = 1;
  - F = fft(f);
  - Fc = fftshift(F);
  - rFc = real(Fc);
  - iFc = imag(Fc);
  - Subplot(2,1,1),plot(abs(Fc));
  - Subplot(2,1,2),plot(atan(iFc./rFc));


40
30
20
10
0
0 100 200 300 400 500 600 700 800 900 1000
spectrum
2
1
0
-1
-2
0 100 200 300 400 500 600 700 800 900 1000
phase angle

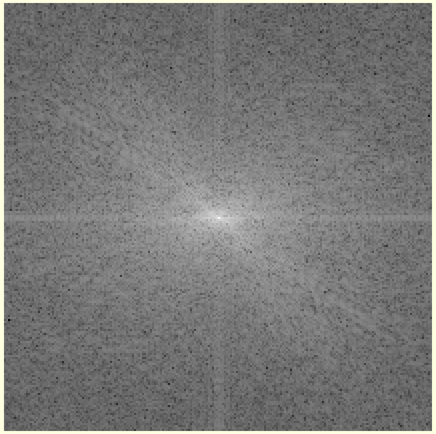
Examples

- Examine the fourier transform of a synthetic image
  - f1 = zeros(500,500);
  - f1(240:260,230:270) = 1;
  - F = fft2(f, 500,500);
  - subplot(2,2,1);imshow(f1,[]);
  - S = abs(F);
  - subplot(2,2,2); imshow(S,[]);
  - Fc = fftshift(F);
  - S1 = abs(Fc);
  - subplot(2,2,3); imshow(S1,[]);
  - S2 = log(1+S1);
  - subplot(2,2,4);imshow(S2,[]);



Example

- Fourier transform of natural images
  - f = imread(‘lenna.jpg’);
  - subplot(1,2,1), imshow(f);
  - f = double(f);
  - F = fft2(f);
  - Fc = fftshift(F);
  - S = log(1+abs(Fc));
  - Subplot(1,2,2),imshow(S,[]);

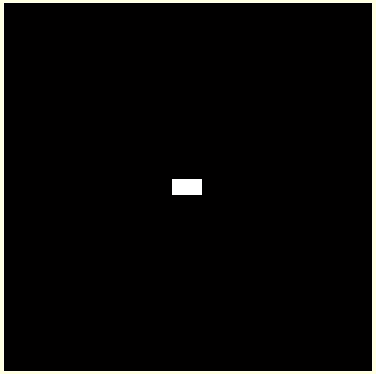
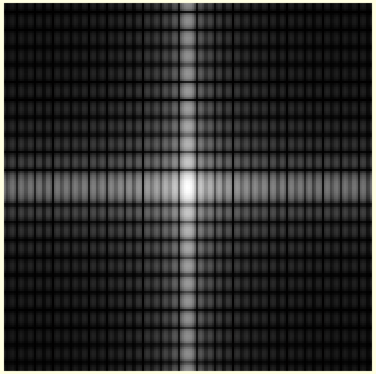
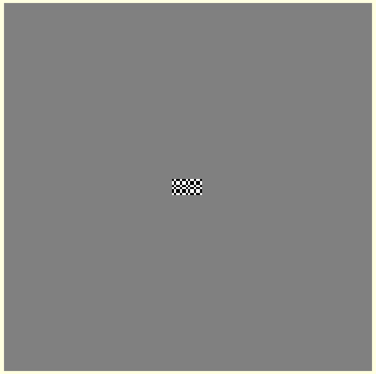
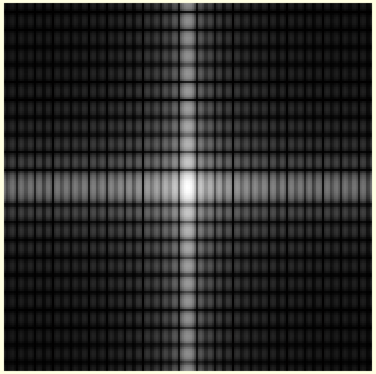


Matlab functions

- Suppose we want to define a matlab function f1 = shift(f), which multiplies the (i,j) pixel of f by (-1)^(i+j), which can be used to shift the frequency components to be visually clearer.
  - function f1 = shift(f);
  - [m,n] = size(f);
  - f1 = zeros(m,n);
  - for i = 1:m;
  -   for j = 1:n;
  -         f1(i,j) = f(i,j) * (-1)^(i+j);
  -   end;
  - end;

Example

- Move origin of FT to the center of the period
  - f = zeros(500,500);
  - f(240:260,230:270) = 1;
  - subplot(2,2,1);imshow(f,[]);
  - F = fftshift(fft2(f));
  - S = log(1+abs(F));
  - subplot(2,2,2);imshow(S,[]);
  - f1 = shift(f);
  - subplot(2,2,3);imshow(f1,[]);
  - F = fft2(f1);
  - S = log(1+abs(F));
  - subplot(2,2,4);imshow(S,[]);

Lowpass filtering (frequency domain)

- Low pass filtering can be achieved by masking away high frequency components of the given image in the frequency domain, and then transform back to the spatial domain.
  - Suppose we are given image f, with Fourier transform F
  - We have designed a low-pass filter in the frequency domain LPF
  - Then the filtered image can be represented by

$$\text{real}(\mathcal{F}^{-1}(F \text{ .* } LPF))$$

Example

- f = imread(‘lenna.jpg’);
- f = double(f);
- F = fftshift(fft2(f));
- [m,n] = size(f);
- sig = 10;
- H = Gaussian(m, n, sig);
- G = H.*F;
- g = abs(ifft2(G));
- Imshow(g,[]);

The 2d Gaussian function

```
function f = Gaussian(M, N, sig);
if(mod(M,2) == 0);
    cM = floor(M/2) + 0.5;
else;
    cM = floor(M/2) + 1;
end;
if(mod(N,2) == 0);
    cN = floor(N/2) + 0.5;
else;
    cN = floor(N/2) + 1;
end;

f = zeros(M,N);
for i = 1:M;
   for j = 1:N;
       dis = (i - cM)^2 + (j - cN)^2;
        f(i,j) = exp(-dis/2/sig^2);
   end;
end;
```

Gaussian lowpass filtering using different bandwidth