電子電路設計_補充教材

Medical Image Processing by using Matlab

Hao-Li Liu

Department of Electrical Engineering, Chang-Gung University

2014, Fall

1

Outline

- Fundamental Matlab usage
- Image Filtering by Using Matlab
- Fourier Transform by Using Matlab
- Reconstructing an Image from Projection Data by using Matlab

2

Image Filtering by Using Matlab

3

Data types and conversions

- Data type
  - int8, uint8, int16, uint16, double
- Data conversion
  - ind2gray, gray2ind, rgb2gray, gray2rgb, rgb2ind, ind2rgb.
  - How to use? Type help XXX!

4

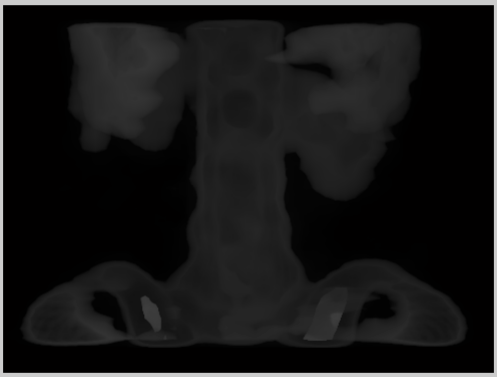
Example: ind2gray

- Usage: Convert indexed image (0-128) to intensity image (0-1)
- Code example
  - load trees;
  - I = ind2gray(X,map);
  - Figure; imshow(X,map);
  - Figure; imshow(I);

5

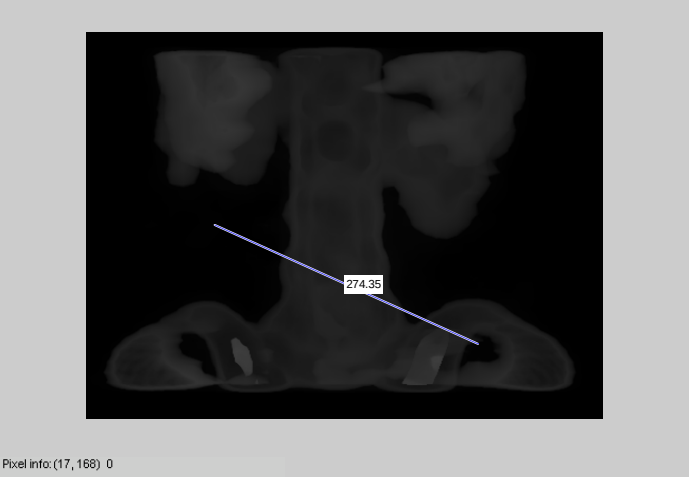
Image import and show

- >> w=imread('spine.tif');
- >> imshow(w); %w is a uint8 format



6

- >> imdistline
- >> impixelinfo



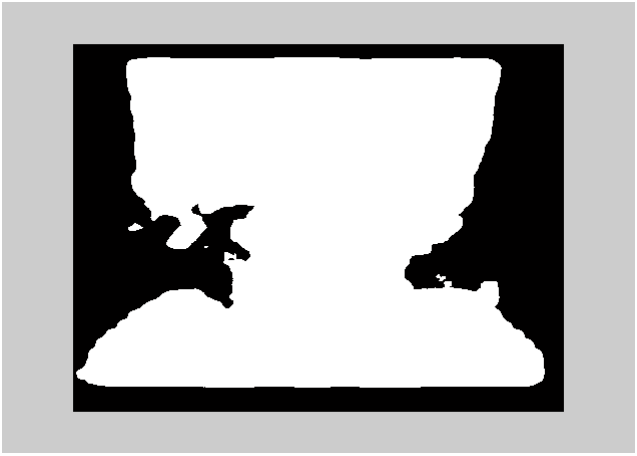
7

- imfinfo('spine.tif')

```
Command Window
>> imfinfo('spine.tif')

ans =

                    Filename: 'C:\MATLAB7\toolbox\images\imdemos\spine.tif'
                FileModDate: '04-Dec-2000 13:57:52'
                    FileSize: 69292
                      Format: 'tif'
                FormatVersion: []
                       Width: 490
                      Height: 367
                    BitDepth: 8
                   ColorType: 'indexed'
             FormatSignature: [73 73 42 0]
                   ByteOrder: 'little-endian'
              NewSubfileType: 0
               BitsPerSample: 8
                 Compression: 'PackBits'
    PhotometricInterpretation: 'RGB Palette'
                 StripOffsets: [23x1 double]
              SamplesPerPixel: 1
                 RowsPerStrip: 16
              StripByteCounts: [23x1 double]
```

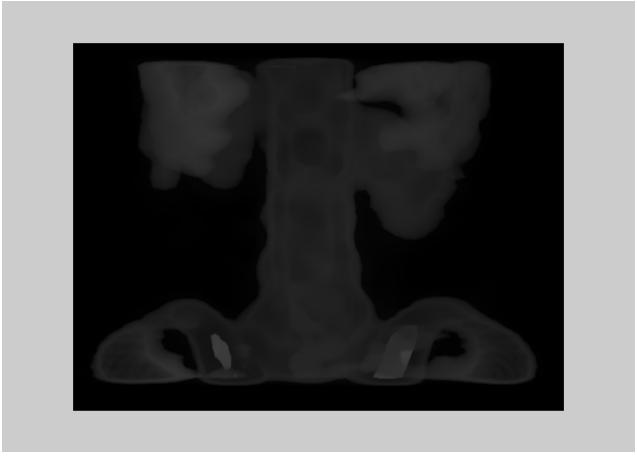
8

- >> w1 = double(w);
- >> figure;
- >> imshow(w1)



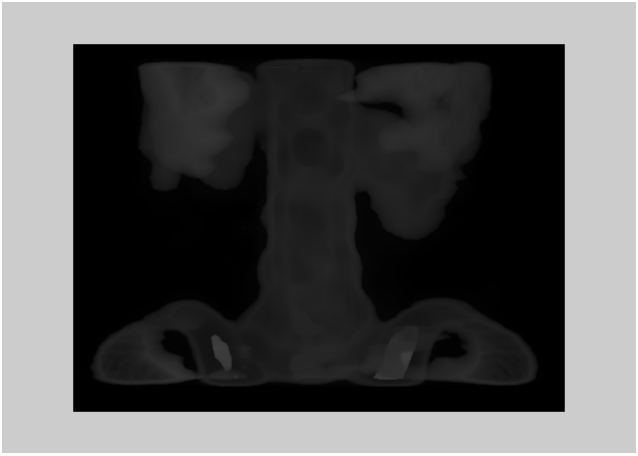
9

- Correct the image scale (0 ~1):
  - >> figure;
  - >> imshow(w1/256)



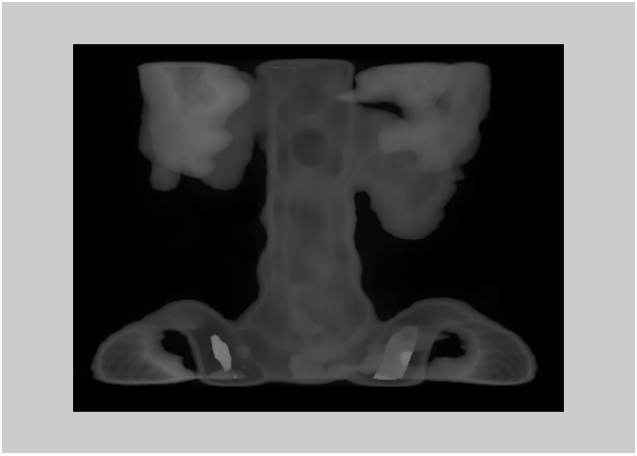
10

- Or use “im2double”:
  - >> w2 = im2double(w);
  - >> figure;
  - >> imshow(w2)



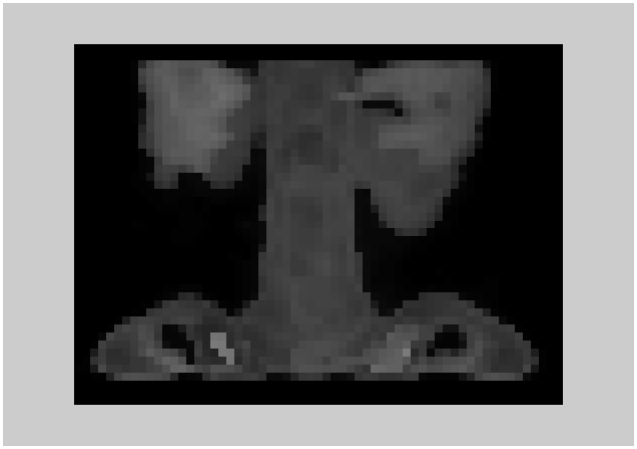
11

- The simplest way to Increase image contrast:
  - >> w = w*2;
  - >> imshow(w)



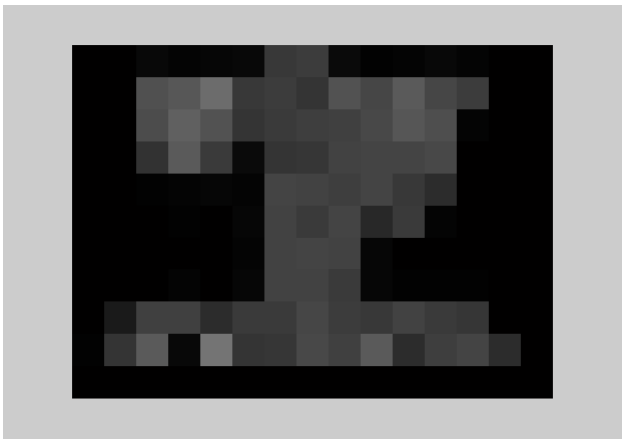
12

- Change the image resolution:
  - >> w3 = imresize(imresize(w,1/8),8);
  - >> imshow(w3)



13

  - >> w3 = imresize(imresize(w,1/32),32);
  - >> imshow(w3)

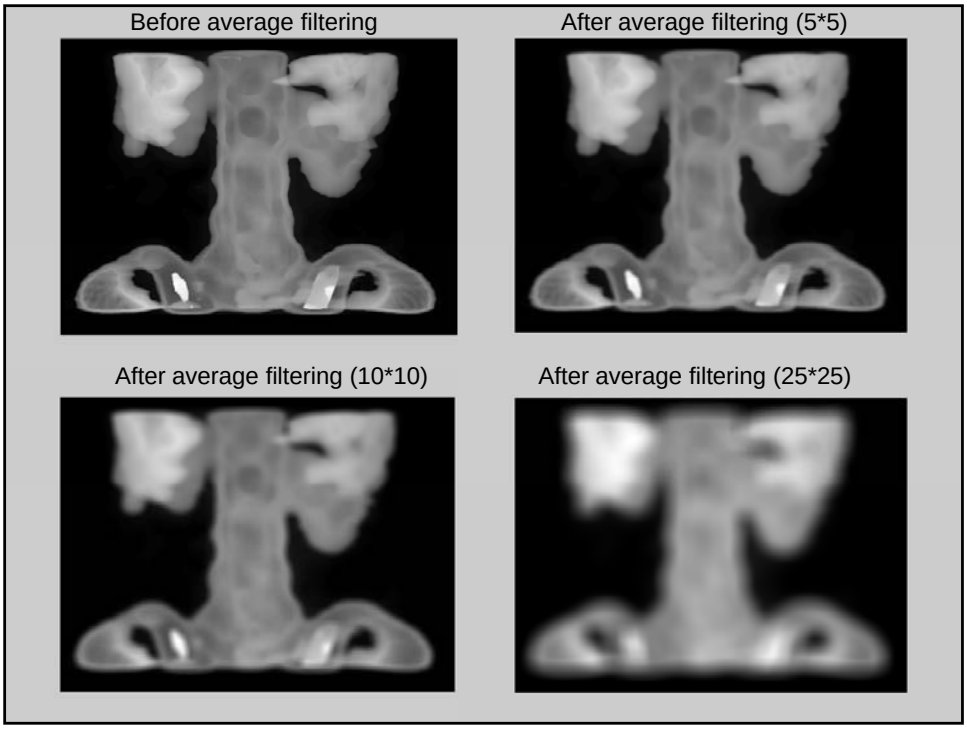


14

Image filtering

- Average filtering
  - >> w1 = double(w);
  - >> figure;
  - >> imshow(w1/max(max(w1)));
  - >> f1= fspecial('average',10); % 2-D filter;
  - >> w5 = filter2(f1,w);
  - >> figure;
  - >> imshow(w5/max(max(w5)));

15



- A 5*5 averaging filter:
  - f2= fspecial('average',5);

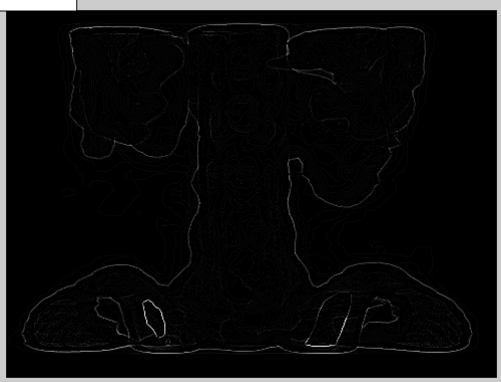
```
>> f2

f2 =

    0.0400    0.0400    0.0400    0.0400    0.0400
    0.0400    0.0400    0.0400    0.0400    0.0400
    0.0400    0.0400    0.0400    0.0400    0.0400
    0.0400    0.0400    0.0400    0.0400    0.0400
    0.0400    0.0400    0.0400    0.0400    0.0400
```

17

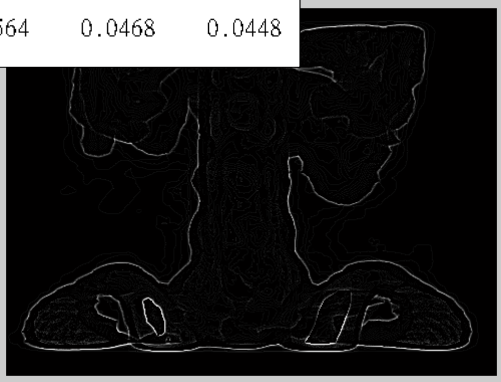
- Average filter is a high-pass filter
- A example of high-pass filter:
  - Laplacian filter
  - Log filter

18

```
>> f= fspecial('laplacian')

f =

    0.1667    0.6667    0.1667
    0.6667   -3.3333    0.6667
    0.1667    0.6667    0.1667
```

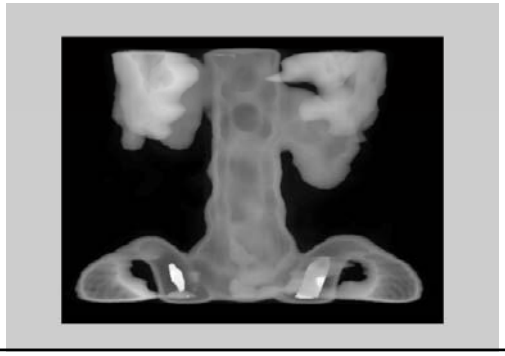


```
>> f= fspecial('log')

f =

    0.0448    0.0468    0.0564    0.0468    0.0448
    0.0468    0.3167    0.7146    0.3167    0.0468
    0.0564    0.7146   -4.9048    0.7146    0.0564
    0.0468    0.3167    0.7146    0.3167    0.0468
    0.0448    0.0468    0.0564    0.0468    0.0448
```



Gaussian filter

- >> f3 = fspecial('gaussian',[5,5]);
- >> w7= filter2(f3,w);
- >> imshow(w7/max(max(w7)))



21

- >> f1 = fspecial('gaussian',[5,5],0.5);
- >> f2 = fspecial('gaussian',[5,5],2);
- >> f3 = fspecial('gaussian',[11,11],1);
- >> f4 = fspecial('gaussian',[11,11],5);
- >> w1= filter2(f1,w);
- >> w2= filter2(f2,w);
- >> w3= filter2(f3,w);
- >> w4= filter2(f4,w);
- >> figure;imshow(w1/max(max(w1)))
- >> figure;imshow(w2/max(max(w2)))
- >> figure;imshow(w3/max(max(w3)))
- >> figure;imshow(w4/max(max(w4)))

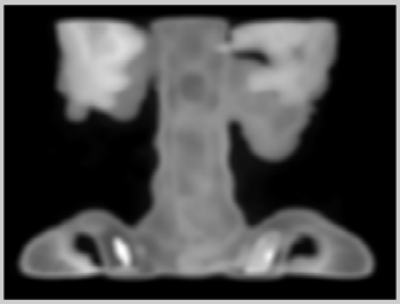
22

5×5, σ = 0.5

5×5, σ = 2

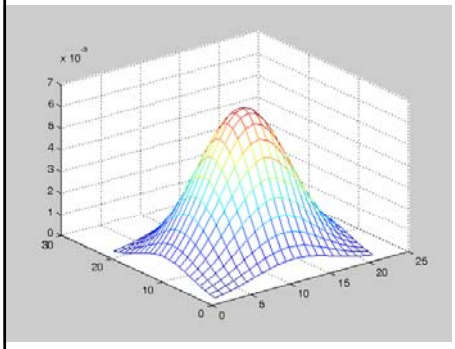
11×11, σ = 1

11×11, σ = 5

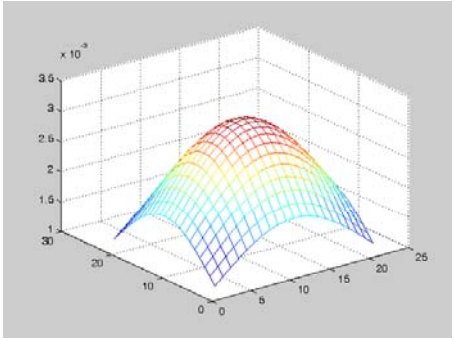


Gaussian filter with different σ

21×21, σ = 5



21×21, σ = 10



24

Fourier Transform by Using Matlab

25

Fourier Transform

- “fft” and “ifft” represents the Fourier and inverse Fourier transform

```
>> a=  [1 2 3 4 5 6];
>> fft(a)'

ans =

  21.0000
  -3.0000 - 5.1962i
  -3.0000 - 1.7321i
  -3.0000
  -3.0000 + 1.7321i
  -3.0000 + 5.1962i
```

26

- The fft(ifft(a)) mean the original number itself:

```
>> b =fft(a);
>> ifft(b)

ans =

    1.0000    2.0000    3.0000    4.0000    5.0000    6.0000
```

27

Shifting property

```
>> x = [2 3 4 5 6 7 8 1];
>> x1 = (-1).^[0:7].*x;
>> fft(x)'

ans =

  36.0000
  -9.6569 - 4.0000i
  -4.0000 + 4.0000i
   1.6569 + 4.0000i
   4.0000
   1.6569 - 4.0000i
  -4.0000 - 4.0000i
  -9.6569 + 4.0000i
```

```
>> fft(x1)'

ans =

   4.0000
   1.6569 - 4.0000i
  -4.0000 - 4.0000i
  -9.6569 + 4.0000i
  36.0000
  -9.6569 - 4.0000i
  -4.0000 + 4.0000i
   1.6569 + 4.0000i
```

28

2-D Fourier Transform

- fft2.m : 2-D Fourier transform
- ifft2.m: 2-D inverse Fourier transform
- fftshift.m: automatically shift

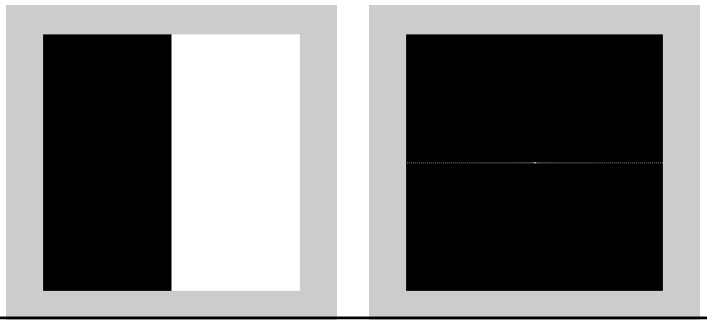
29

Create an useful subroutine for fft plot:

```
function[] =  fftshow(f,type)

if nargin<2,
    type ='log';
end;

if (type =='log'),
    f1 = log(1+abs(f));
    fm = max(f1(:));
    imshow(im2uint8(f1/fm));
elseif (type =='abs'),
    fa = abs(f);
    fm = max(fa(:));
    imshow(fa/fm);
else,
    error('TYPE must be abs or log.');
end;
```

30

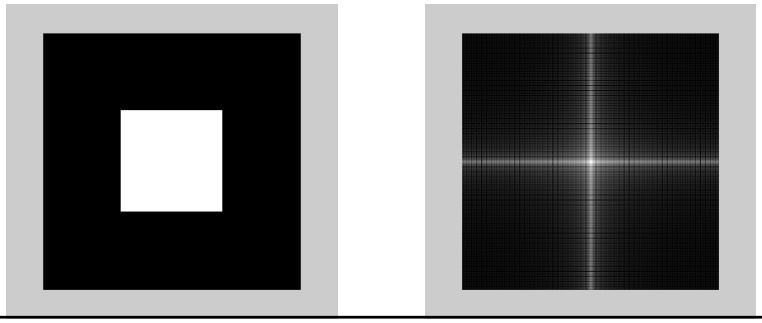
– >> a = [zeros(256,128), ones(256,128)];
– >> af = fftshift(fft2(a));
– >> figure;imshow(a)
– >> figure;fftshow(af)



31

FFT for a rectangular function

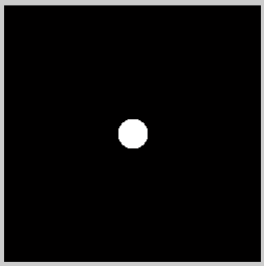
– >> a = zeros(256,256);
– >> a(78:178,78:178) = 1;
– >> af = fftshift(fft2(a));
– >> figure;imshow(a)
– >> figure;fftshow(af,'log')

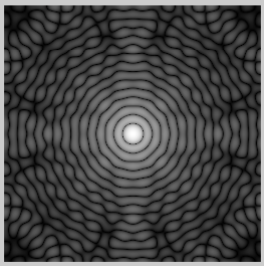


32

FFT for a circular function

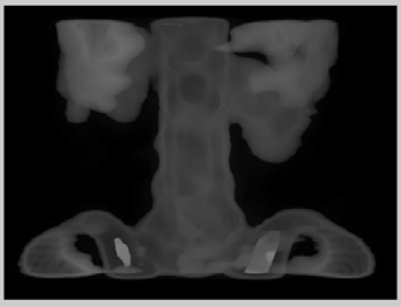
- >> [x,y]=meshgrid(-128:127,-128:127);
- >> z = sqrt(x.^2+y.^2);
- >> c = (z<15);
- >> cf = fftshift(fft2(c));
- >> figure;imshow(c)
- >> figure;fftshow(cf,'log')

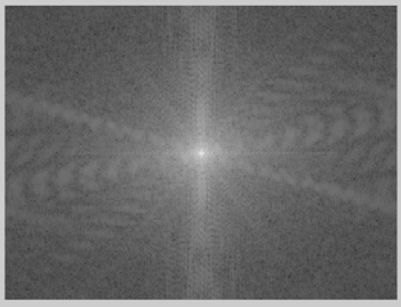




33

- Original image and its Fourier transform:





34

An example for low-pass filter

– >> w=imread('cameraman.tif');
– >> f = fftshift(fft2(w));
– >> cf = c.*f;
– >> cfi = ifft2(cf);
– >> figure;imshow(cfi/max(max(cfi)))
– >> figure;fftshow(cf,'log')

35

Original image

fftshow(cf, 'log')

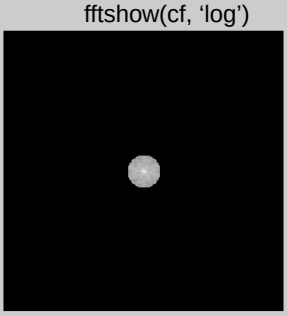


Image after filtering
→ Ringing effect can be found !

36

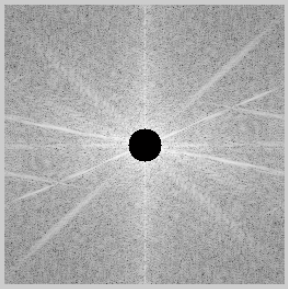
An example for high-pass filter

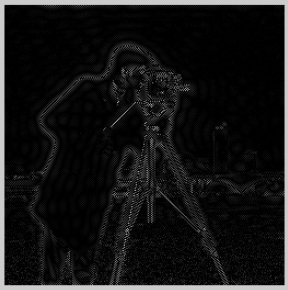
- >> [x,y]=meshgrid(-128:127,-128:127);
- >> z = sqrt(x.^2+y.^2);
- >> c = (z>15);
- >> f = fftshift(fft2(w));
- >> cf = c.*f;
- >> cfi = ifft2(cf);
- >> figure;fftshow(cf,'log')
- >> figure;imshow(cfi/max(max(cfi)))

37

Original image

fftshow(cf, 'log')





38

Exercise

- Create a 2-D Gaussian function to erase the ringing effect caused by a rigid low-pass filter in the “caremaman.tif” image.

39

Reconstructing an Image from Projection Data

(Parallel Beam)

40

Create phantom

- A shape-and-logan phantom (left in your homework)

41

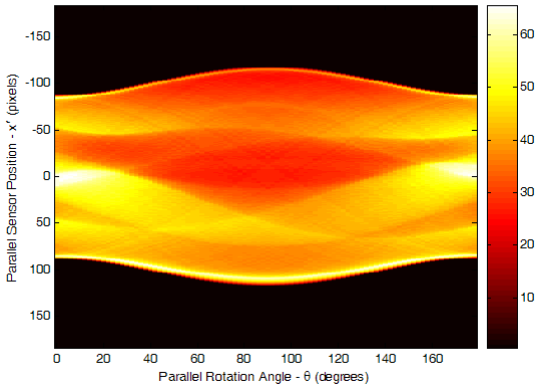
Radon transform

– >> theta1 = 0:10:170;
– >> [R1,xp] = radon(P,theta1);
– >> theta2 = 0:5:175;
– >> [R2,xp] = radon(P,theta2);
– >> theta3 = 0:2:178;
– >> [R3,xp] = radon(P,theta3);

42

Plot the radon transform

- Figure; imagesc(theta3,xp,R3);
- colormap(hot)
- colorbar
- xlabel('Parallel Rotation Angle - \theta (degrees)');
- ylabel('Parallel Sensor Position - x\prime (pixels)');



43

Image reconstruction using "iradon.m"

- >> dtheta1 = theta1(2) - theta1(1);
- >> I1 = iradon(R1,dtheta1,256);
- >> figure; imshow(I1);

44

– >> output_size = 256;
– >> dtheta2 = theta2(2) - theta2(1);
– >> I2 = iradon(R2,dtheta2,output_size);
– >> figure, imshow(I2);

45

– >> dtheta3 = theta3(2) - theta3(1);
– >> I3 = iradon(R3,dtheta3,output_size);
– >> figure, imshow(I3);

46

How to create matrix variable?

- Meshgrid.m
- Zeros.m
- Ones.m
- Special functions
  - Peaks.m
- Others:
  - See “demo”…

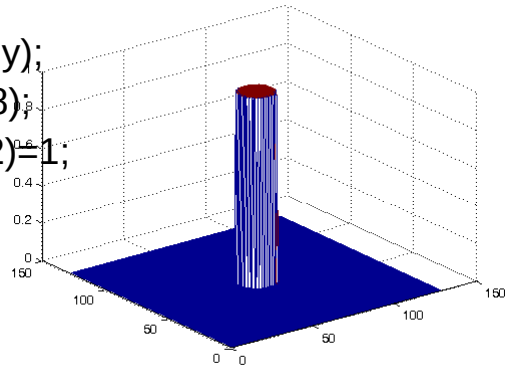
```
>> [X,Y] = meshgrid(-2:.2:2, -2:.2:2);

>> Z = X .* exp(-X.^2 - Y.^2);

>> mesh(Z)
>> zeros(5)

ans =

     0     0     0     0     0
```

47

How to create circle?

- Example:

```
>> x = -64:63;
>> y = -64:63;
>> [X,Y]=meshgrid(x,y);
>> Z = zeros(128,128);
>> Z(X.^2+Y.^2<10^2)=1;
>> mesh(Z)
```



48