

ECE 565 – Image Processing and Computer Vision

Thomas BOOT¹

¹IIT, Illinois Institute of Technology, 3300 S Federal St, Chicago, IL 60616

tboot@hawk.iit.edu

1. 2D Fourier in polar coordinates

1.1 Representation of Fourier in polar

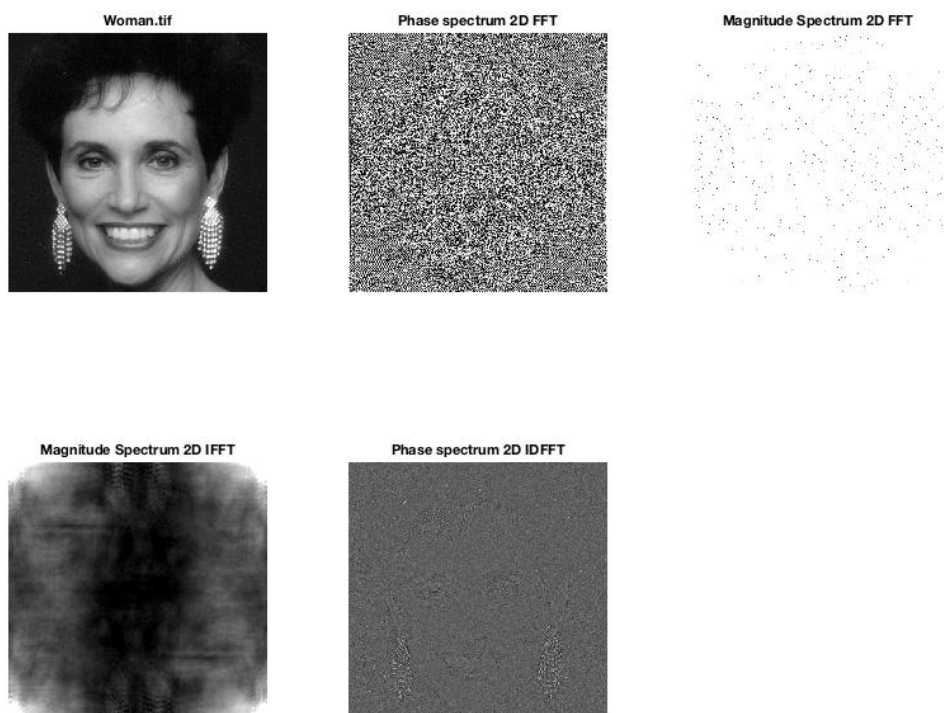
The 2D Fourier transform is represented in polar form as : $F(u,v) = |F(u,v)|e^{j\Phi(u,v)}$

Where, $|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}$ is the magnitude spectrum and $\Phi(u,v)$ is the phase angle.

The phase angle carries the information about the location of the image elements. The magnitude spectrum carries the information about contrast and intensity transitions.

1.2 Magnitude spectrum and phase angle of the file woman.tif – Inverse transform of phase and magnitude term

```
X = imread('woman.tif');
F = fft2(double(X));
pha1 = exp(1i*angle(F));
mag1 = abs(F);
mag2 = ifft2(mag1);
pha2 = ifft2(pha1);
figure,
set(gcf, 'Color', 'white');
subplot 231, imshow(X), title('Woman.tif')
subplot 232, imshow(real(pha1), []), title('Phase spectrum 2D FFT')
subplot 233, imshow(uint8(mag1)), title('Magnitude Spectrum 2D FFT')
subplot 234, imshow(uint8(mag2)), title('Magnitude Spectrum 2D IFFT')
subplot 235, imshow(pha2, []), title('Phase spectrum 2D IDFFT')
```

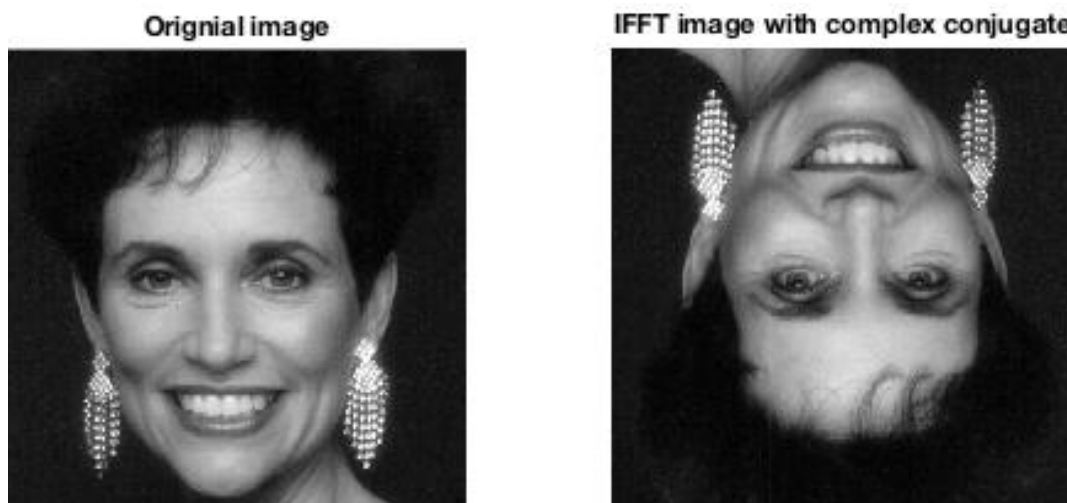


The magnitude spectrum of the IFFT shows that the maximum contrast and intensity is located at the center of the image, compared to the left and right side of the image. We can also see structure within the magnitude spectrum.

The phase spectrum of the IFFT represents the information about the location of the woman. Indeed we can see the edges of the face of the woman and her earrings, this is the phase image of “woman.tif”.

1.3 Image reconstruction using magnitude spectrum and complex conjugate of the phase angle

```
X = imread('woman.tif');
F = fft2(double(X));
pha1 = exp(1i*angle(F));
mag1 = abs(F);
image = mag1.*(conj(pha1));
image2 = ifft2(image);
figure,
set(gcf,'Color','white')
subplot 121, imshow(X),title('Original image')
subplot 122, imshow(uint8(image2)), title('IFFT image with complex conjugate')
```



The reconstructed image with the complex conjugate of the phase component is a 180° rotated image.

2. Edge detection combined with smoothing and thresholding

2.1 Spatial filtering – Code

```
function imageOut = spatialfiltering(image,mask)

[x,y]= size(image);
% Padding 'replicate'
image_p = zeros(x+2,y+2);

image_p(2:end-1,2:end-1) = image;
image_p(1,2:end-1) = image(1,:);
image_p(x+2,2:end-1) = image(x,:);
image_p(2:end-1,1) = image(:,1);
image_p(2:end-1,y+2) = image(:,y);
```

```

% filtering
imageOut = zeros(x,y);

for i=2:x-1
    for j=2:y-1
        imageOut(i,j) = sum(sum((mask.*image_p(i-1:i+1,j-1:j+1))));
    end
end
end

```

2.2 Sobel gradient mask - Smoothing and thresholding of kidney.tif – Code

```

image = imread('kidney.tif');

Sx = [-1 0 1; -2 0 2; -1 0 1];
Sy = [-1 -2 -1; 0 0 0; 1 2 1];

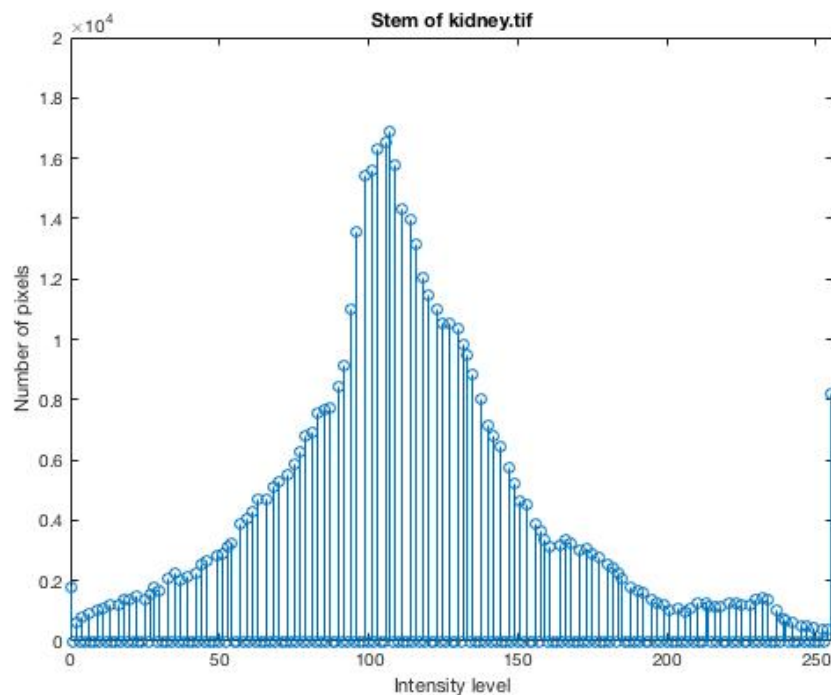
Gx = spatialfiltering(image,Sx);
Gy = spatialfiltering (image,Sy);
M = abs(Gx) + abs(Gy);

% Histogram stem image
[counts,x ]=imhist(image);
figure,set(gcf,'Color','white'),stem(x,counts);
title('Stem of kidney.tif')
xlabel('Intensity level'), ylabel('Number of pixels');
axis([0 256 0 20000])

M_binary = M > 100;

figure,set(gcf,'Color','white')
subplot 221, imshow(Gx,[]); title('gx');
subplot 222,

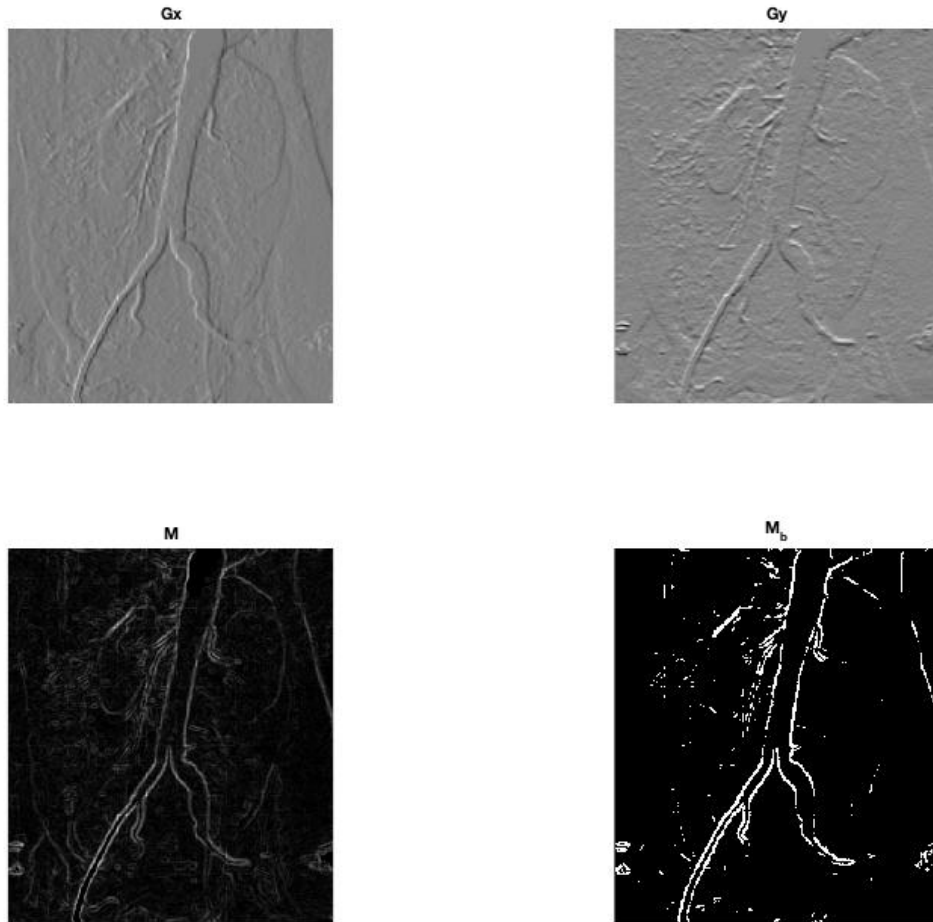
```



```

imshow(Gy,[]); title('gy');
subplot 223, imshow(M,[]); title('M');
subplot 224, imshow(M_binary,[]); title('M_binary');

```



Images of the kidney after performing spatial filtering with the Sobel mask (Gx and Gy) and M is the magnitude of the image computed regarding Gx and Gy. The threshold has been chosen after several trials regarding the stem histogram of the original image.

3. Global thresholding

```
image = imread('noisy_fingerprint.tif');
% Step 1 : Estimate for the global threshold T
[counts, N]=imhist(image);
i = 1;
ms = sum(counts);
T(i)= round(sum(N.*counts)/ms);

% Step 2 : Mean Above T (MAT) and Mean Below T (MBT)
ms2 = sum(counts(1:T(i)));
MBT = sum(N(1:T(i)).*counts(1:T(i)))/ms2;
ms3 = sum(counts(T(i):end));
MAT = sum(N(T(i):end).*counts(T(i):end))/ms3;
i= i+1;
% new T
T(i) = round((MAT+MBT)/2);

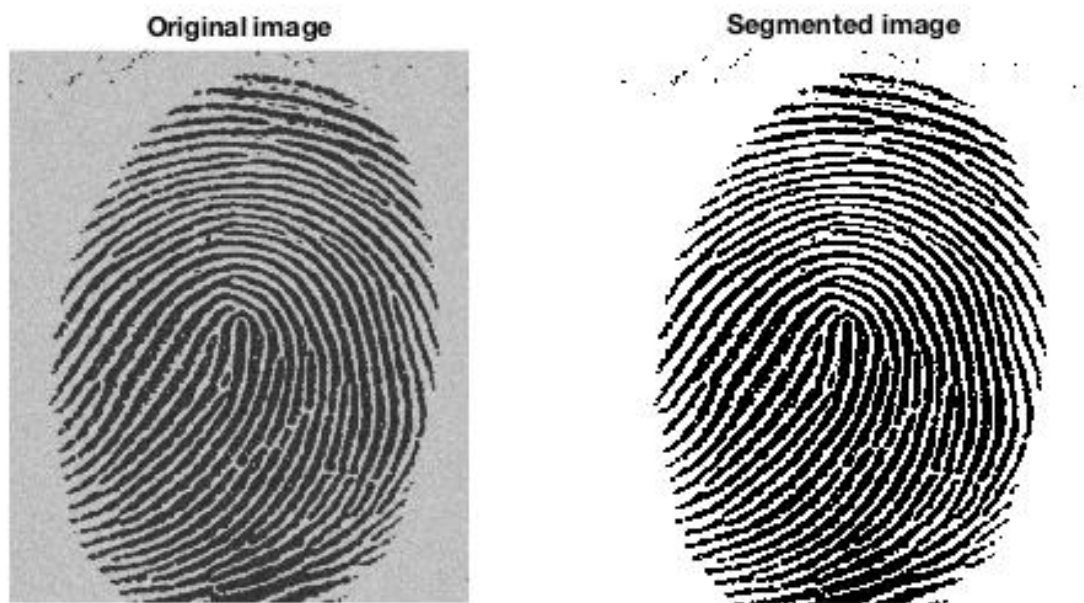
% Step 3 : Repeat step 2 if T(i) ~= T(i-1)
while abs(T(i)-T(i-1))>1
    ms2 = sum(counts(1:T(i)));
    MBT = sum(N(1:T(i)).*counts(1:T(i)))/ms2;
    ms3 = sum(counts(T(i):end));
```

```

MAT = sum(N(T(i):end).*counts(T(i):end))/ms3;
i= i+1;
% new T
T(i) = round((MAT+MBT)/2);
threshold = T(i);
end

% Step 4 : Normalize the threshold to the range [i 1]
level = (threshold - 1)/(N(end)-1);
BW = im2bw(image,level);
figure, set(gcf,'Color','white')
subplot 121, imshow(image), title('Original image')
subplot 122, imshow(BW), title('Segmented image')

```



4. Otsu's thresholding

```

image = imread('polymersomes.tif');
[M,N]= size(image);
h = imhist(image);
pi = h/(M*N);

for i=1:1:256
    if pi(i)~=0
        lt = i;
        break
    end
end

for i=256:-1:1
    if pi(i)~=0
        ht = i;
        break
    end
end

for k=1:256
    p1(k)= sum(pi(1:k));
    p2(k)= sum(pi(k+1:256));
end

% Mean intensity values
for k=1:256

```

```

    m1(k)=sum((k)*pi(1:k))/p1(k);
    m2(k)=sum((k)*pi(k+1:256))/p2(k);
end
% Cumulative mean (average intensity)
L = 1:1:256;
mg = sum(L*pi);

for k=1:256
    varG(k) = (k-mg^2)*pi(k);
end

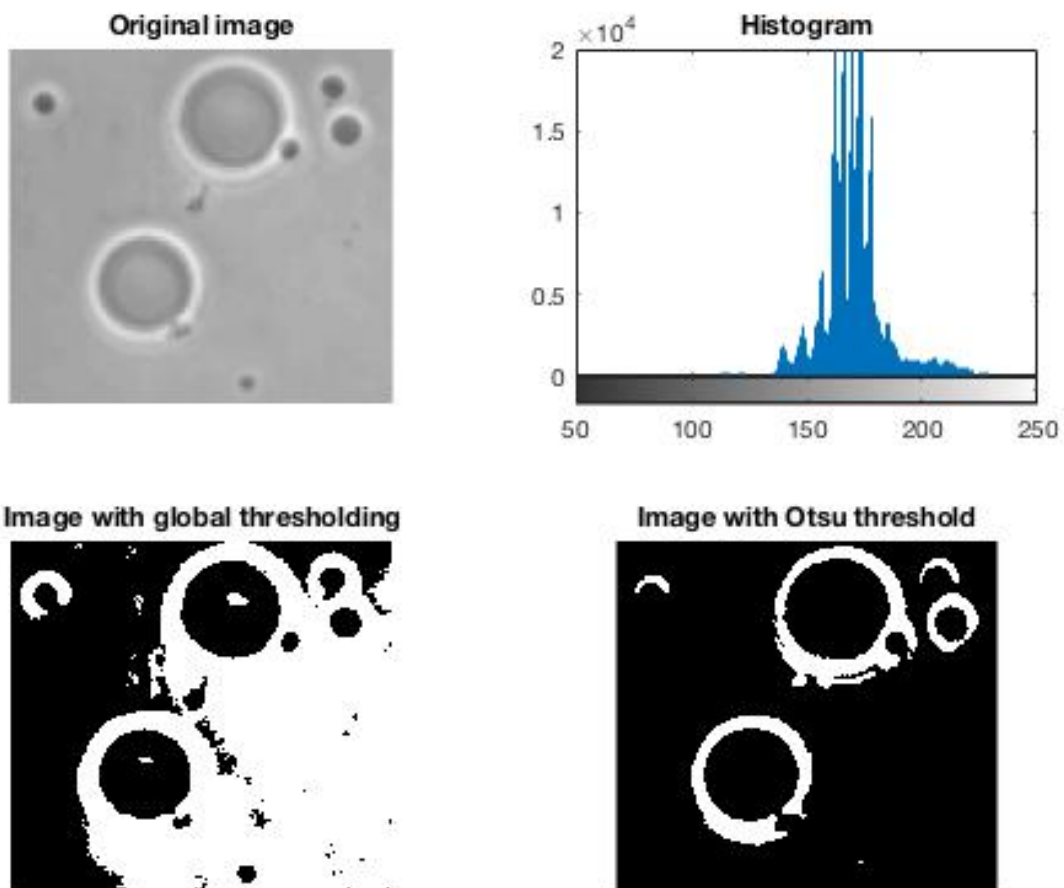
for k=1:256
    var(k) = p1(k)*(m1(k)-mg)^2 + p2(k)*(m2(k)-mg)^2;
end

%Separability

[a, T]=max(var(:));
T = T+1t;
g = zeros(size(image));
g(image >(T))=1;

% Global thresholding
image_gt = globalThresholding(image);
figure,
subplot 221, imshow(image), title('Original image')
subplot 222, imhist(image), title('Histogram'), axis([50 250 0 20000])
subplot 223, imshow(image_gt), title('Image with global thresholding')
subplot 224, imshow(g), title('Image with Otsu threshold')

```



In the program, the function `globalThresholding` corresponds to the thresholding algorithm in part 3.

Comparing, the Otsu's method versus the global thresholding, we see that the performance is enhanced as it is based on the histogram of the image. The position of the maximum variance is added to the lowest threshold of the normalized histogram in order to have the threshold for the separability.