Activity 2: Properties and Applications of the 2D Fourier Transform (Part 2 of 2)

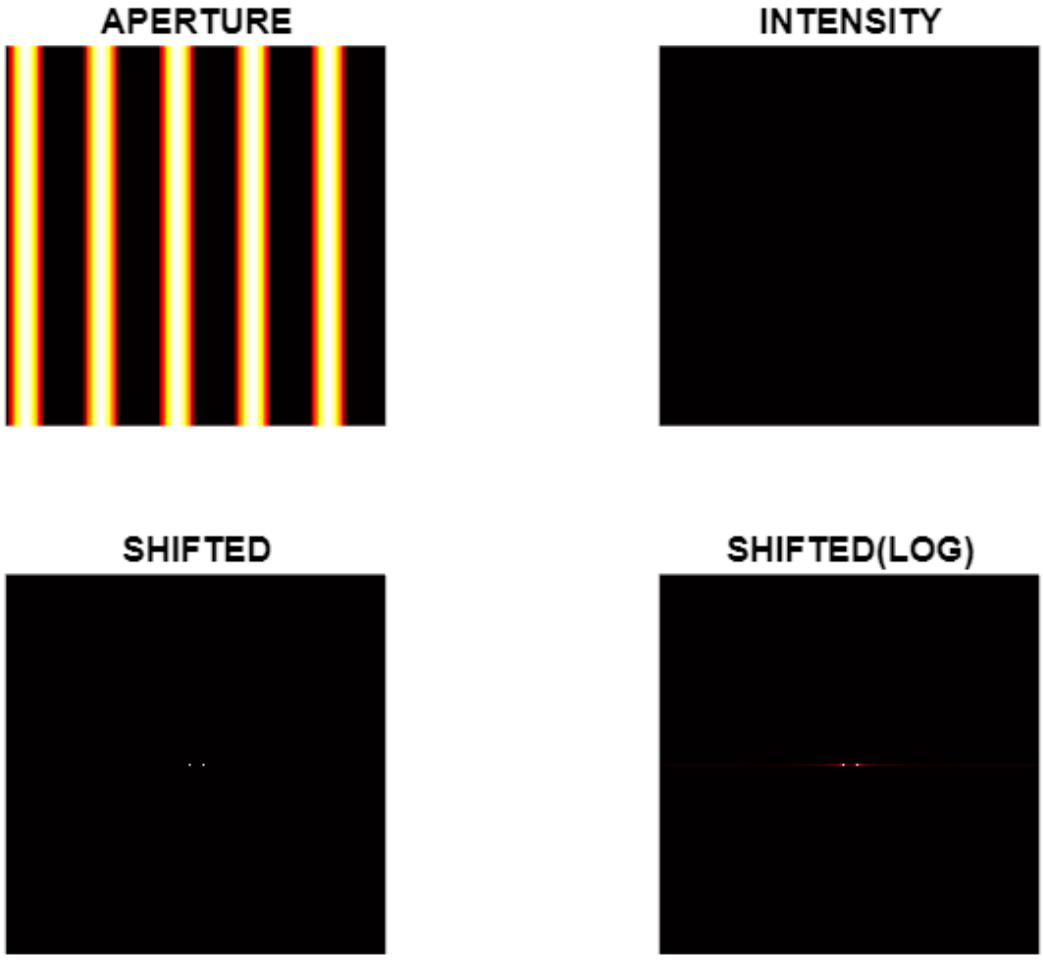
by Johnenn R. Manalang

Activity 2.2.1. Rotation Property of the FT

Varying Space (Vertical):

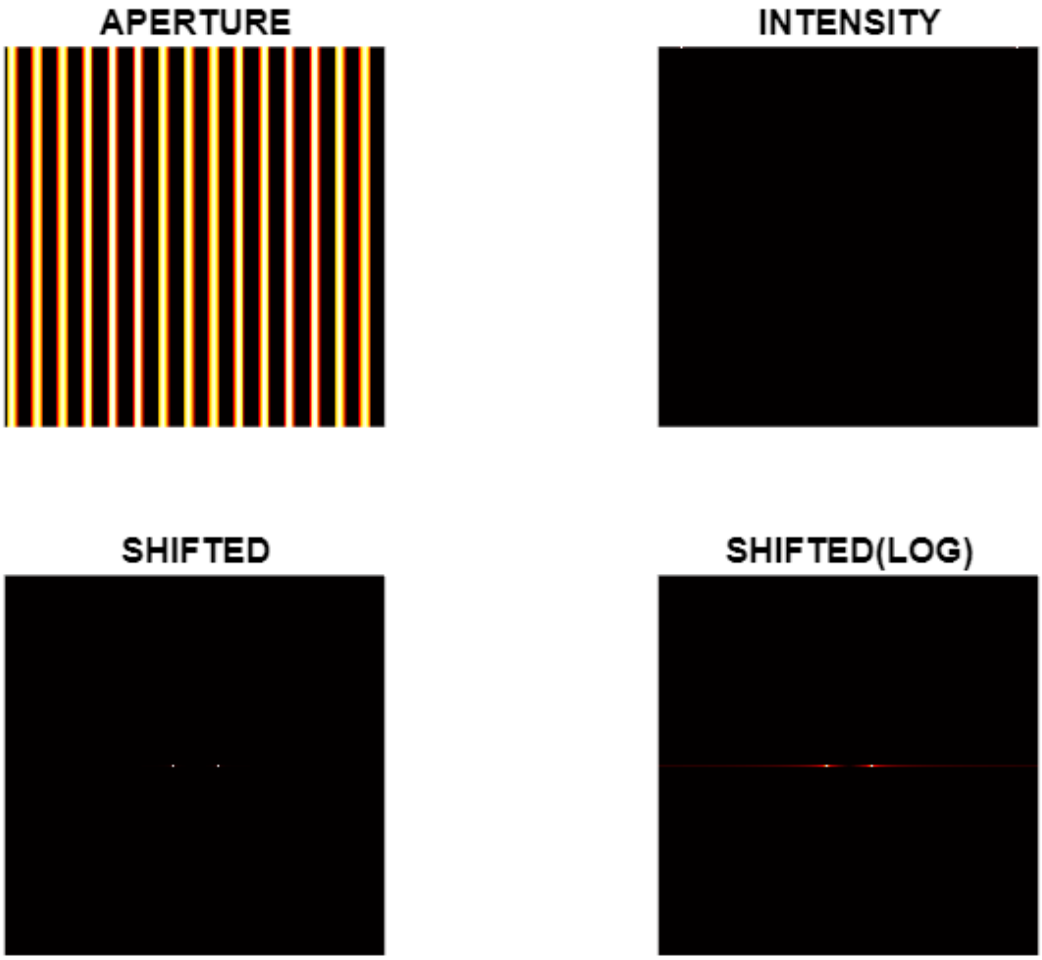
Sinusoid along the x-axis (f = 5)

```
FFT(Sinusoid(5,0));
```



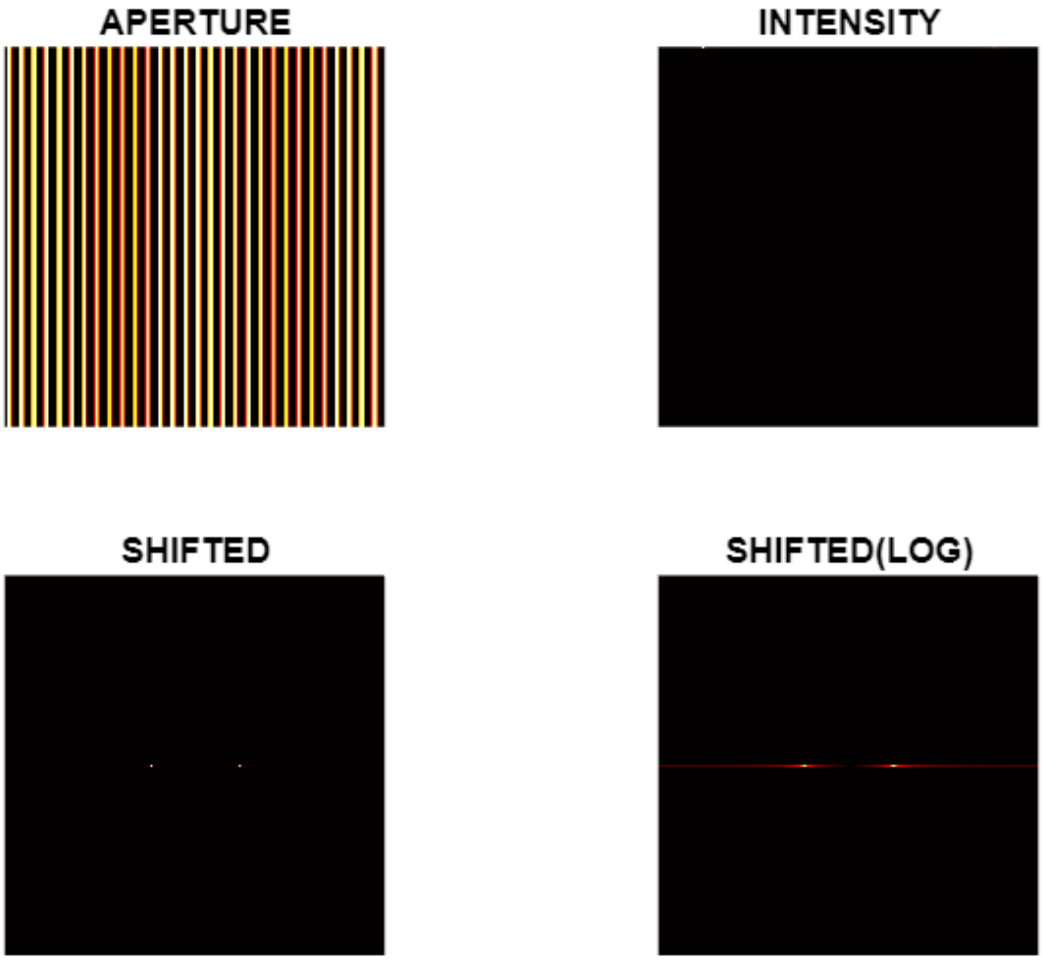
Sinusoid along the x-axis (f = 15)

```
FFT(Sinusoid(15,0));
```



Sinusoid along the x-axis (f = 30)

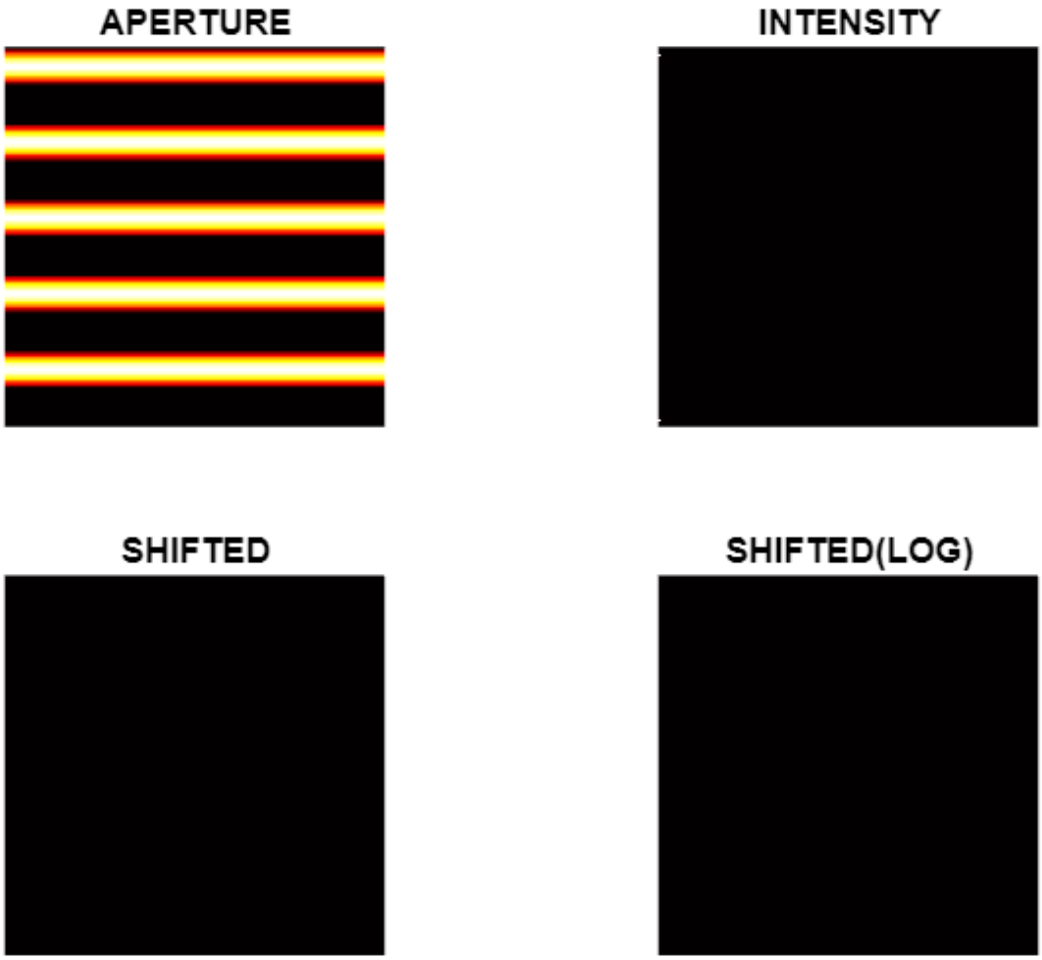
```
FFT(Sinusoid(30,0));
```



Varying Space (Horizontal):

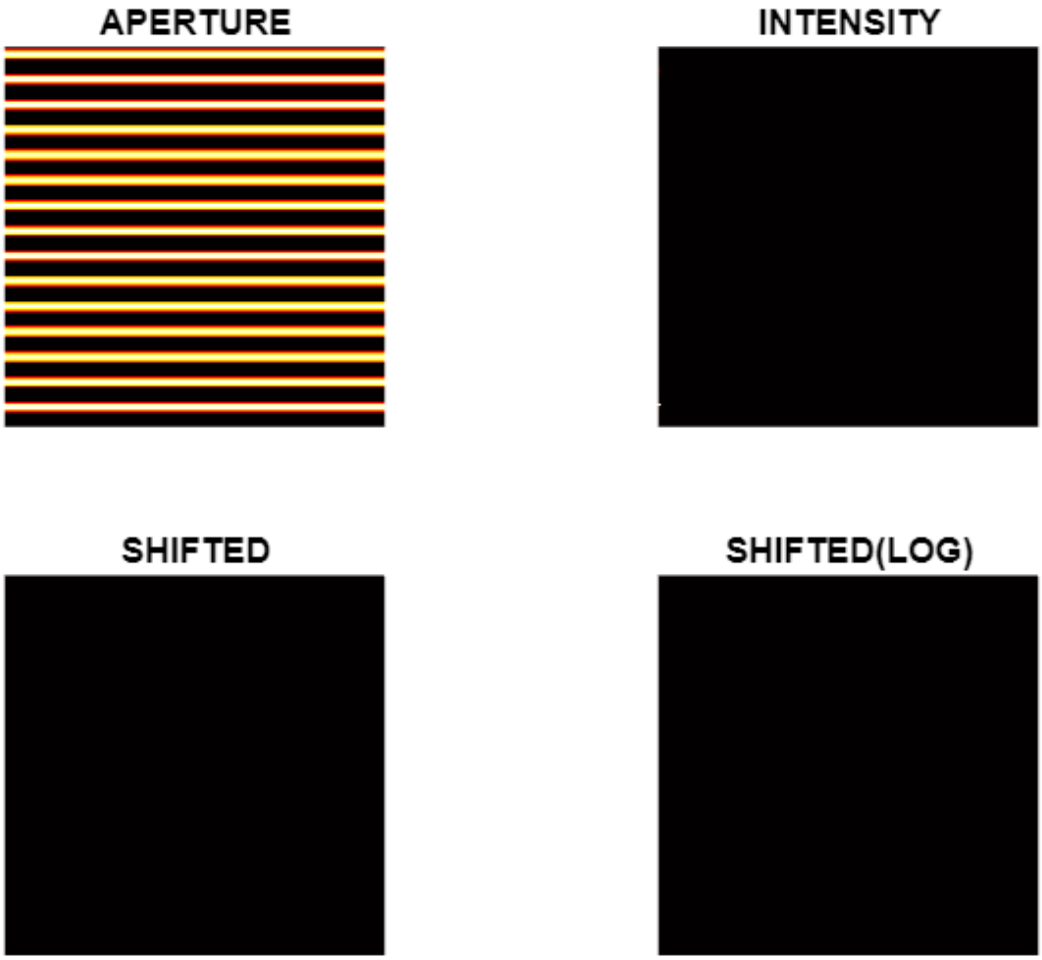
Sinusoid along the y-axis (f = 5)

```
FFT(Sinusoid(5,pi/2));
```



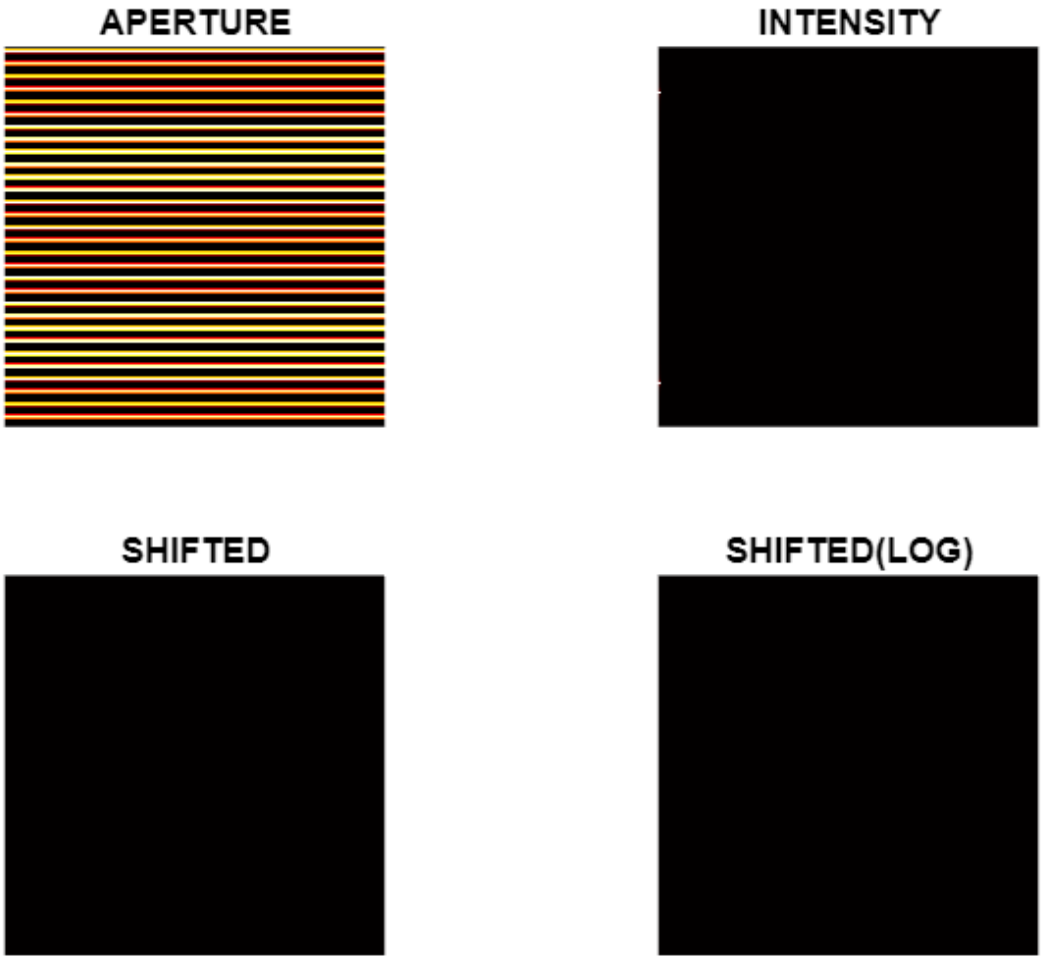
Sinusoid along the y-axis (f = 15)

```
FFT(Sinusoid(15,pi/2));
```



Sinusoid along the y-axis (f = 30)

```
FFT(Sinusoid(30,pi/2));
```

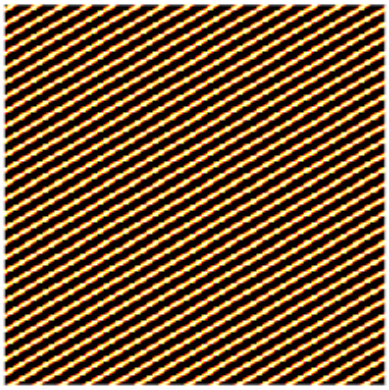


Varying degree of rotation:

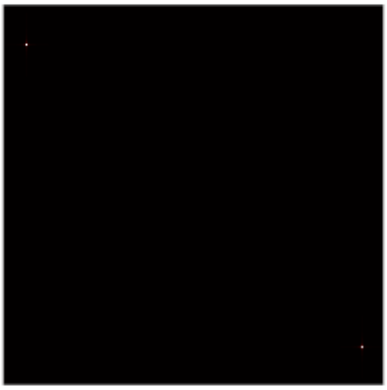
Changing degree of rotation (d = pi/3)

```
FFT(Sinusoid(30,pi/3));
```

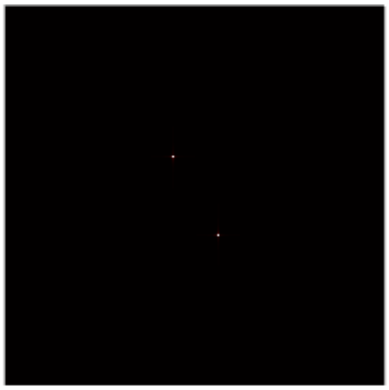
APERTURE



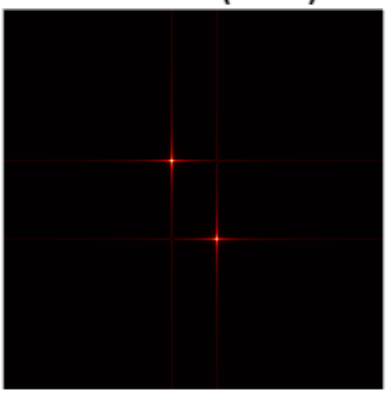
INTENSITY



SHIFTED

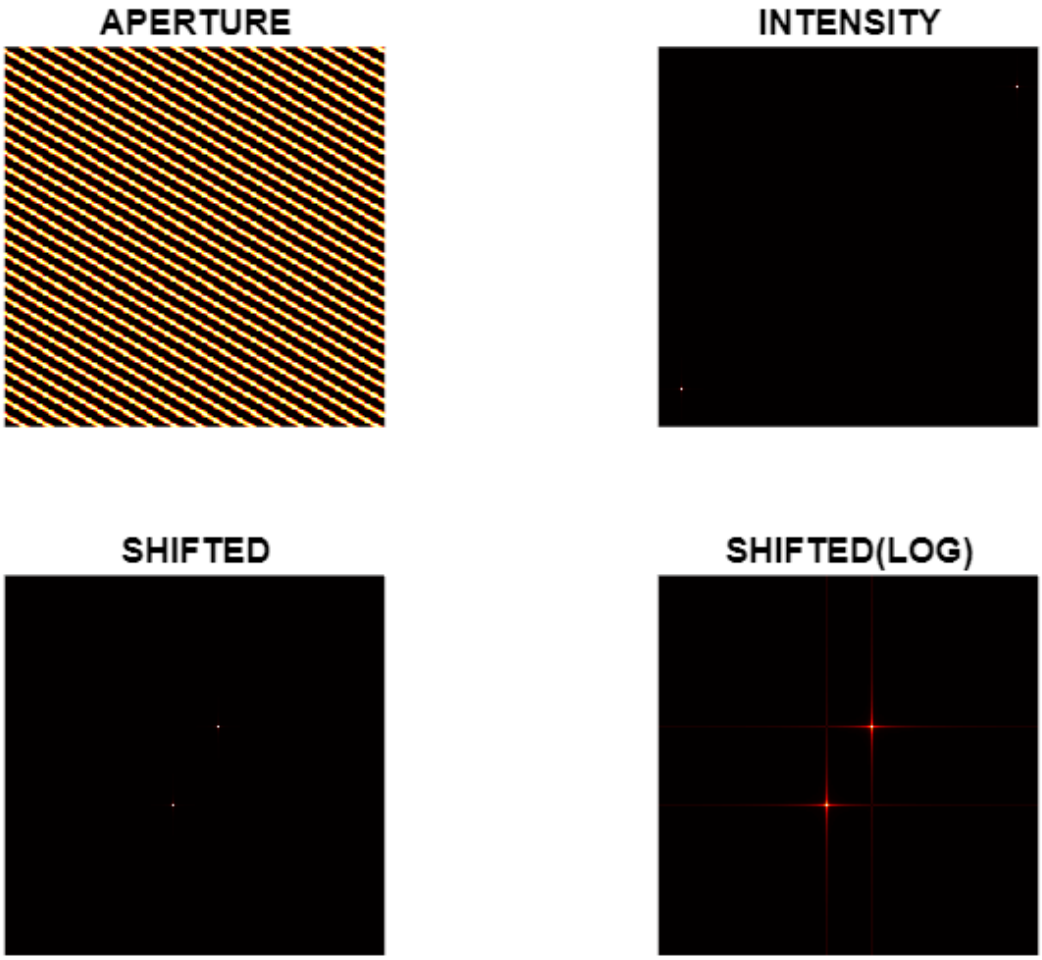


SHIFTED(LOG)



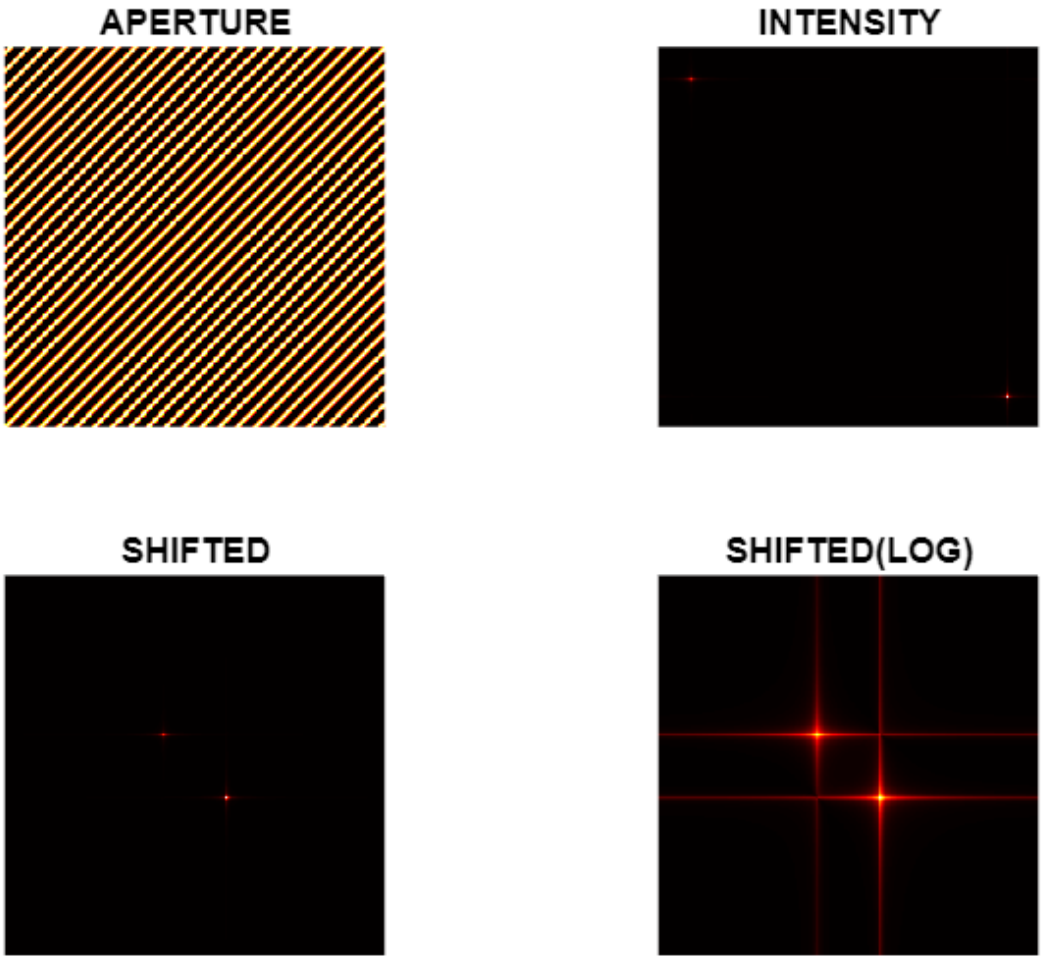
Changing degree of rotation (d = 2pi/3)

```
FFT(Sinusoid(30,2*pi/3));
```



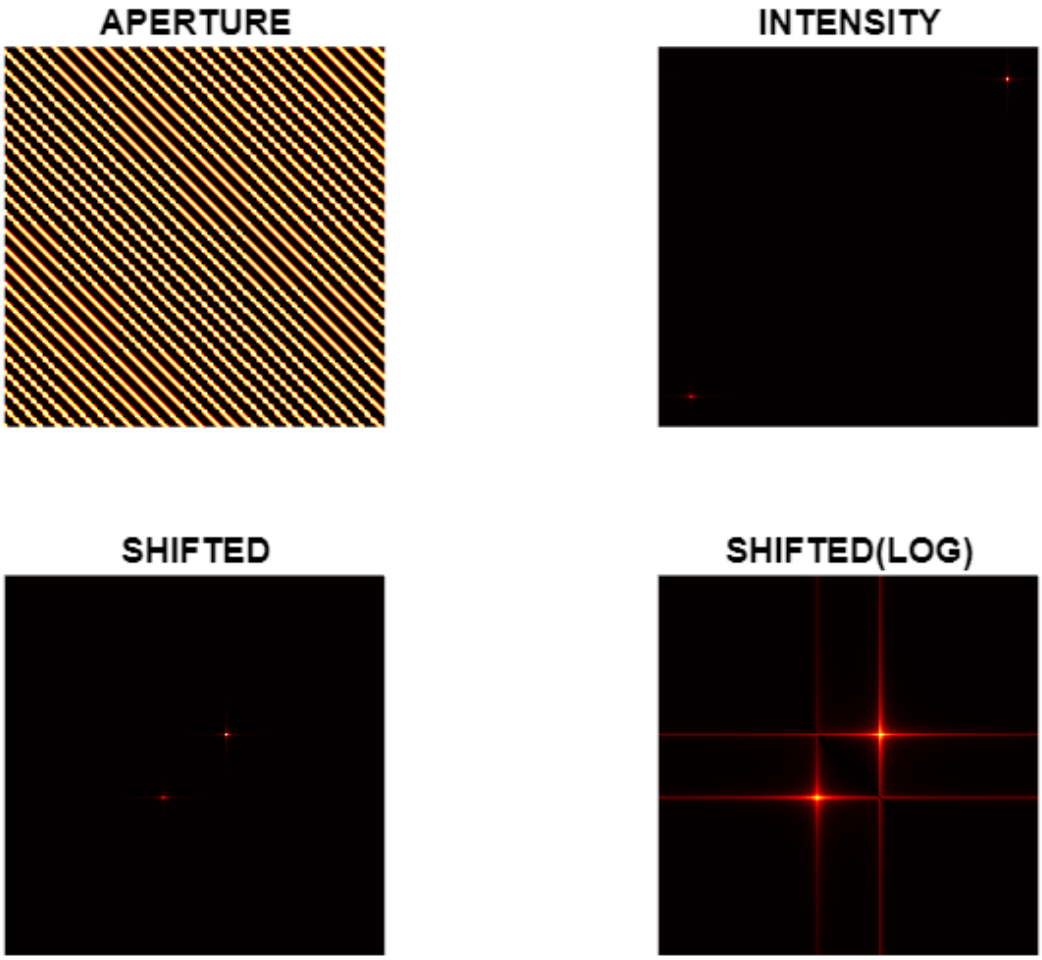
Changing degree of rotation (d = pi/4)

```
FFT(Sinusoid(30,pi/4));
```



Changing degree of rotation (d = 3pi/4)

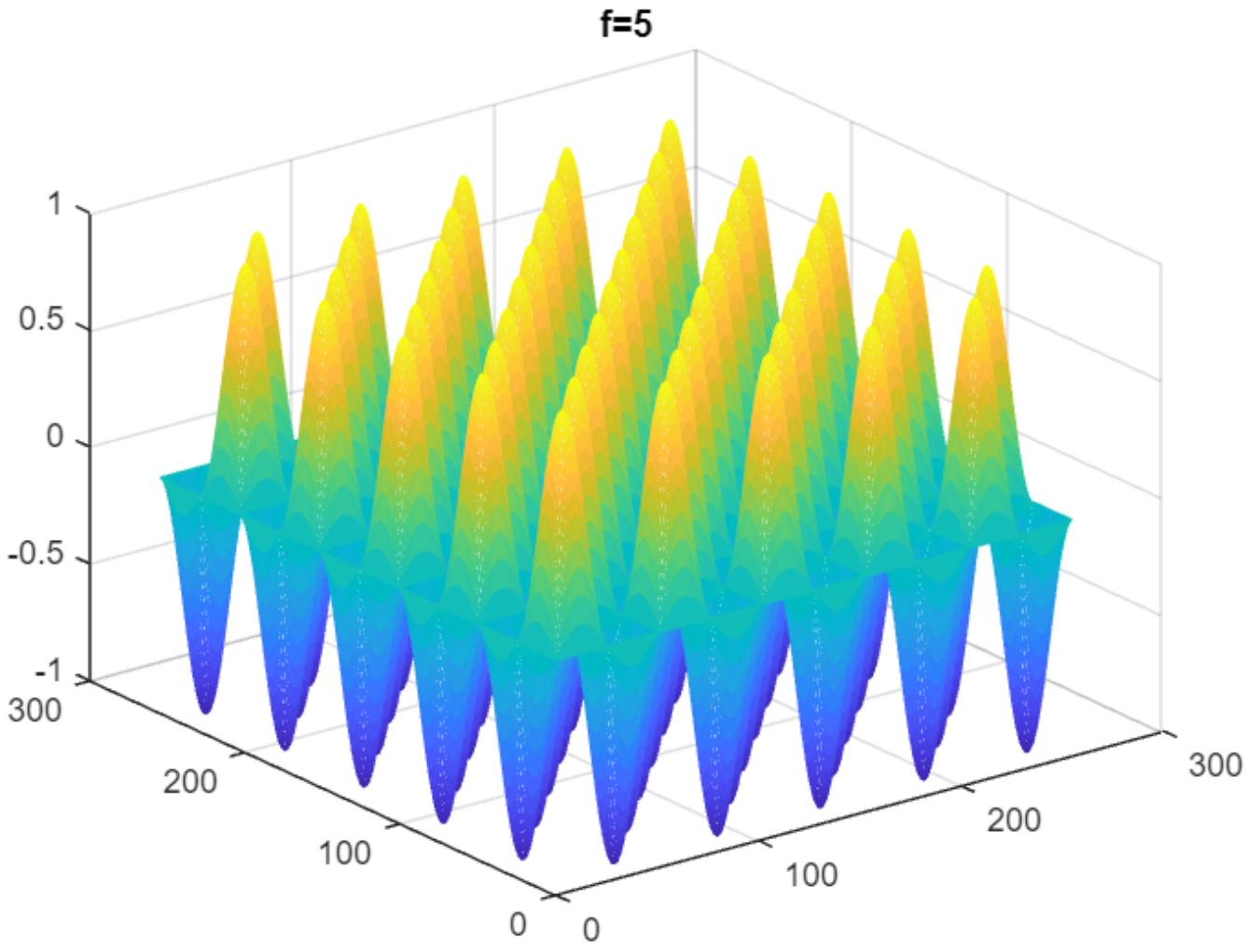
```
FFT(Sinusoid(30,3*pi/4));
```



Combinations of 2 to 3 sinusoids:

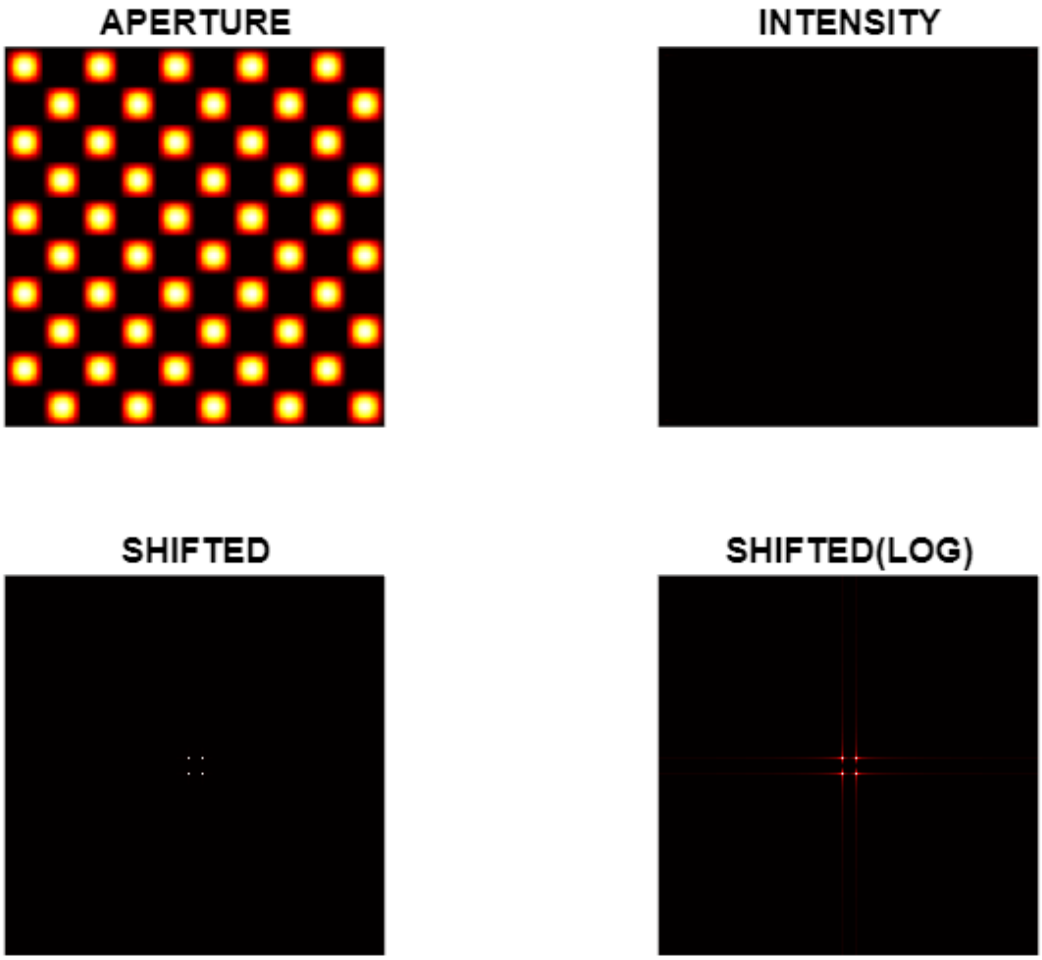
Here, we can see that the combination of two sinusoids along the x and along the y axis creates a figure resembling an egg carton.

```
close; clear;
mesh(Sinusoid(5,0).*Sinusoid(5,pi/2)); title('f=5')
```



Combination of sinusoids along x and y (f =5)

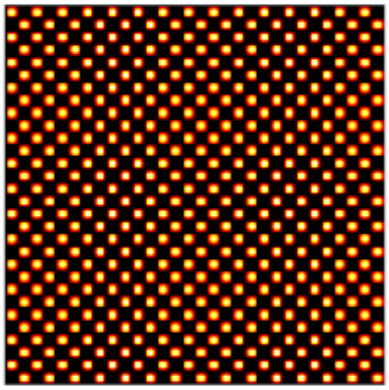
```
FFT(Sinusoid(5,0).*Sinusoid(5,pi/2));
```



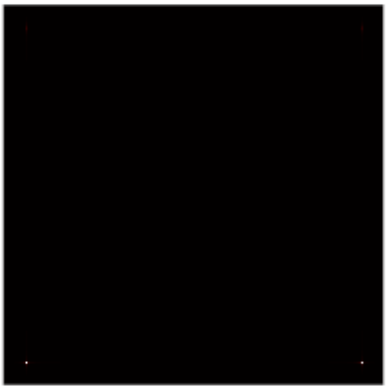
Combination of sinusoids along x and y (f =15)

```
FFT(Sinusoid(15,0).*Sinusoid(15,pi/2));
```

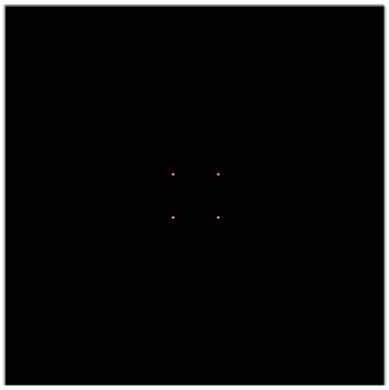
APERTURE



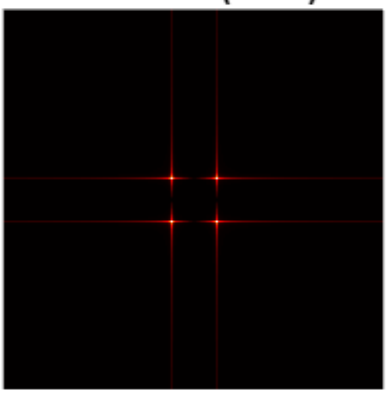
INTENSITY



SHIFTED

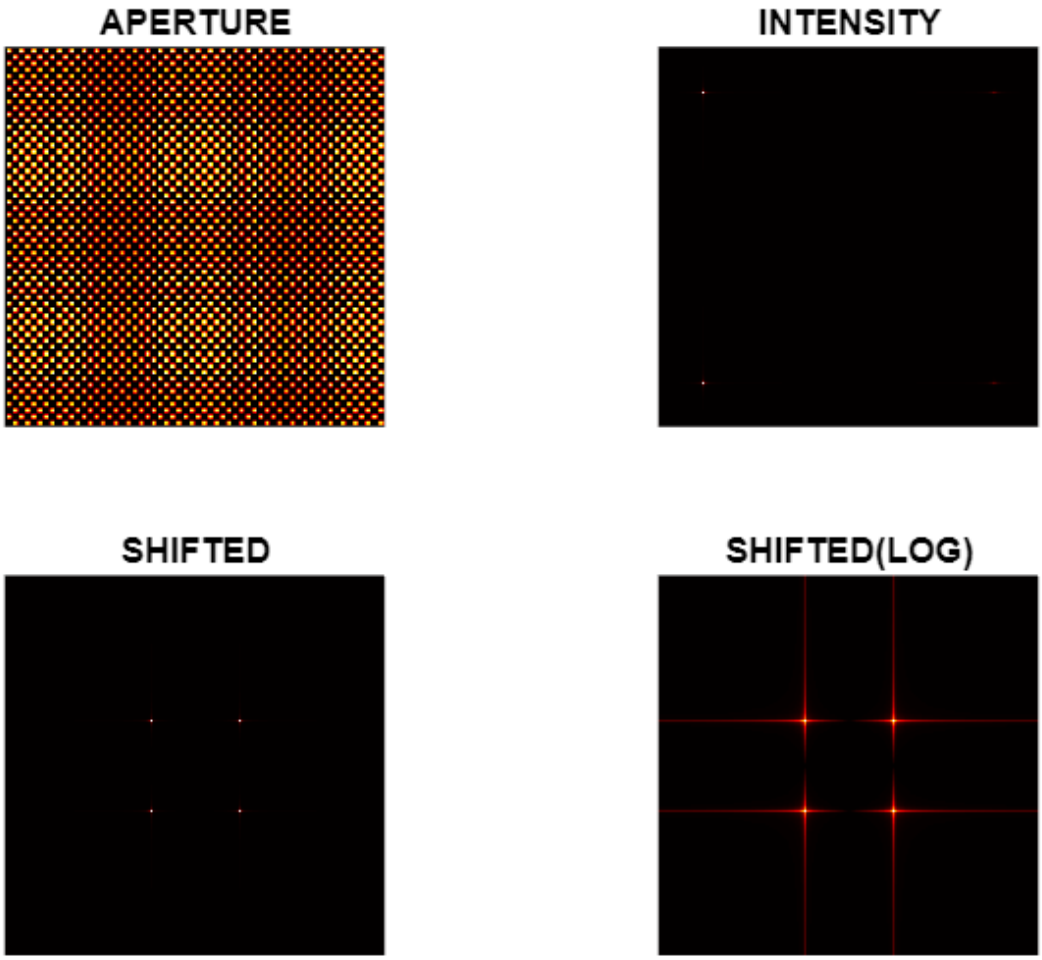


SHIFTED(LOG)



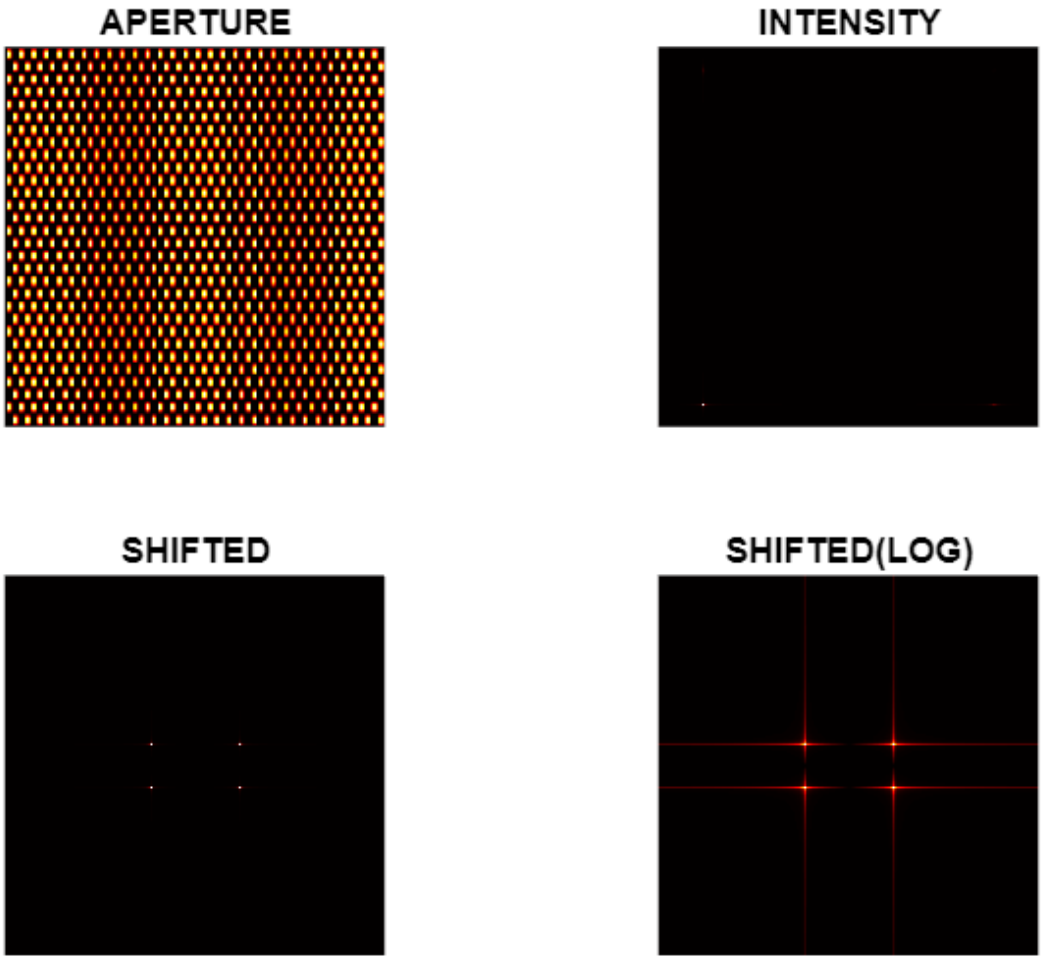
Combination of sinusoids along x and y (f =30)

```
FFT(Sinusoid(30,0).*Sinusoid(30,pi/2));
```



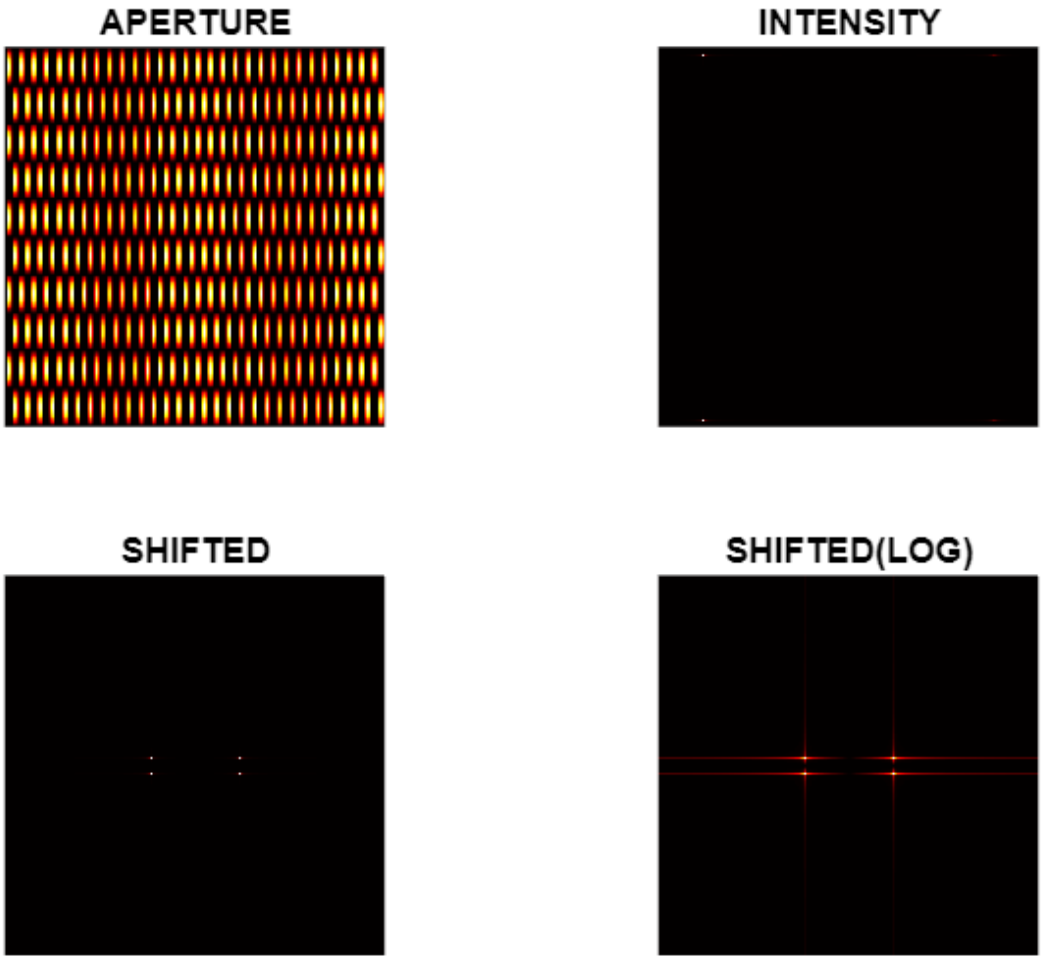
Combination of sinusoids along x (f=30) and y (f =15)

```
FFT(Sinusoid(30,0).*Sinusoid(15,pi/2));
```



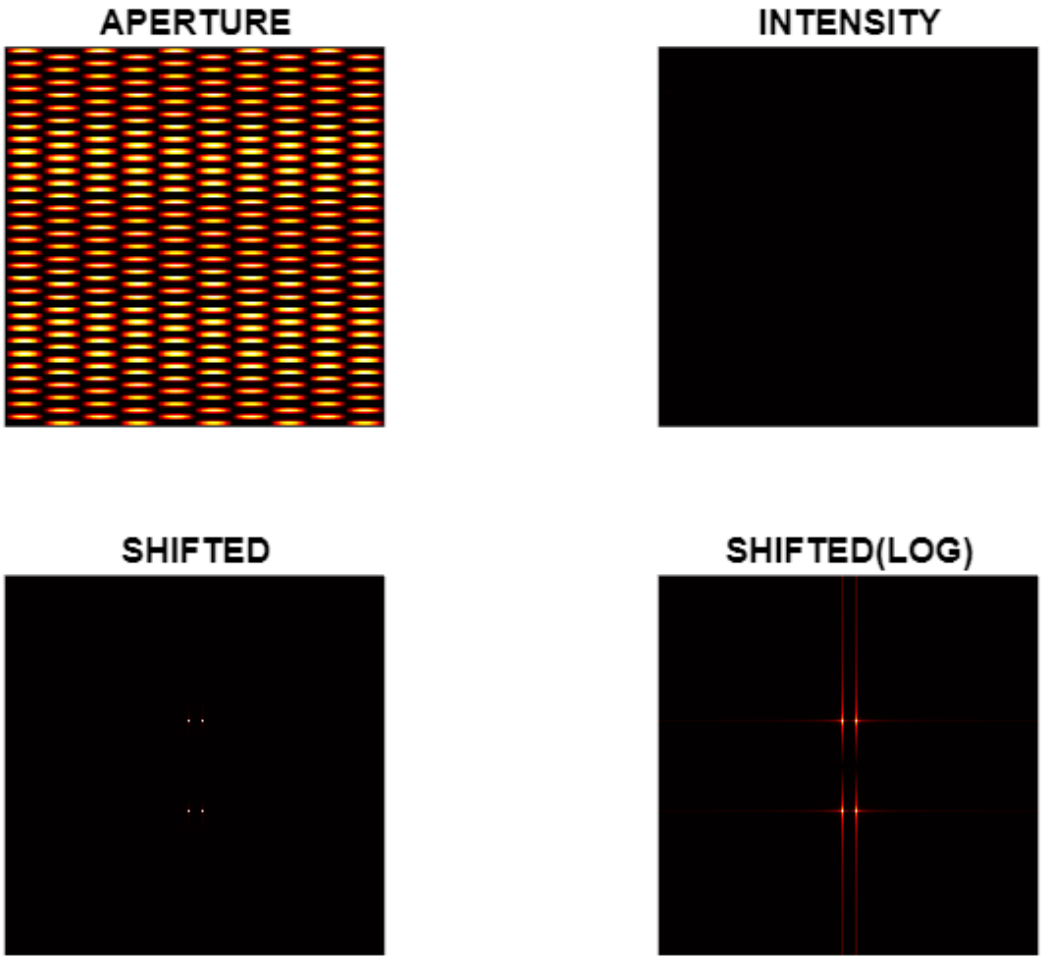
Combination of sinusoids along x (f=30) and y (f =5)

```
FFT(Sinusoid(30,0).*Sinusoid(5,pi/2));
```



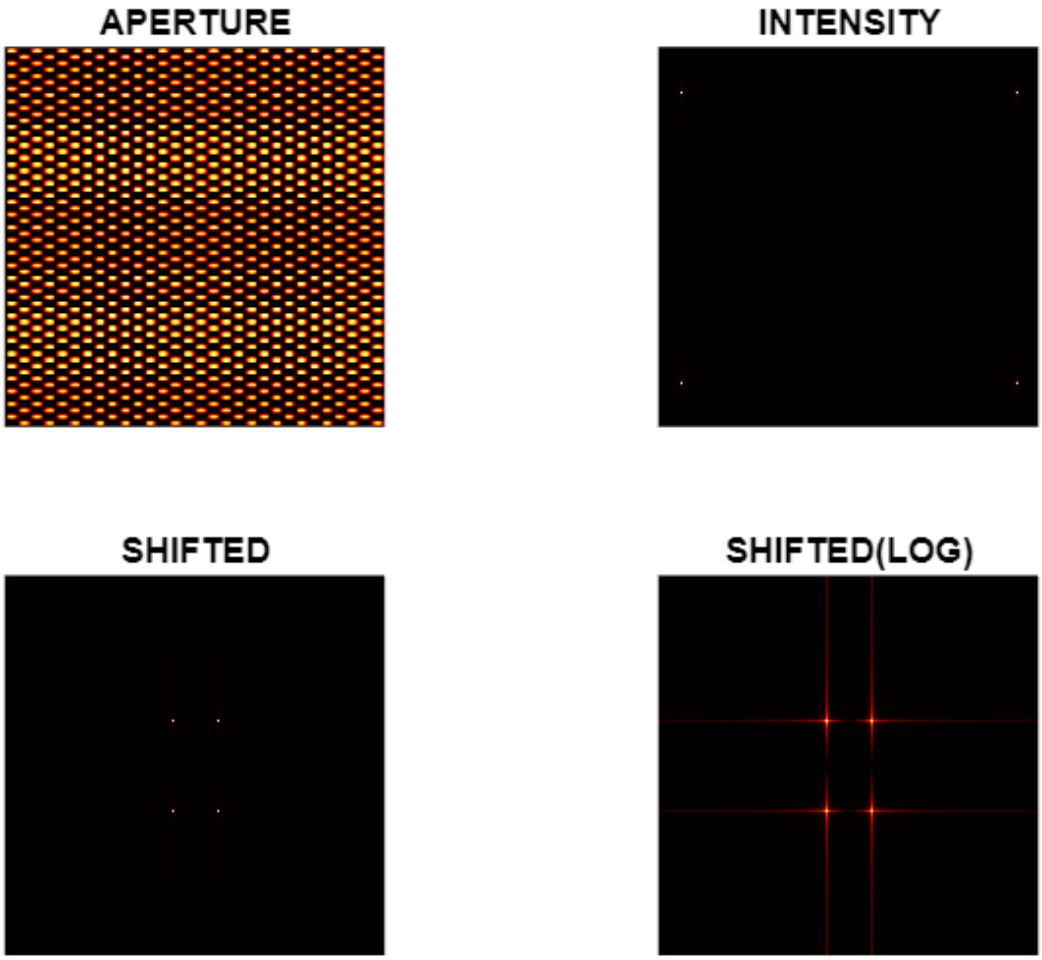
Combination of sinusoids along x (f=5) and y (f =30)

```
FFT(Sinusoid(5,0).*Sinusoid(30,pi/2));
```



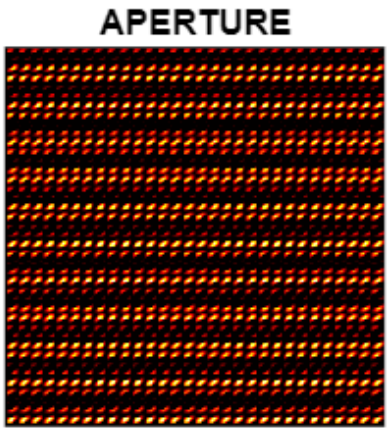
Combination of sinusoids along x (f=15) and y (f =30)

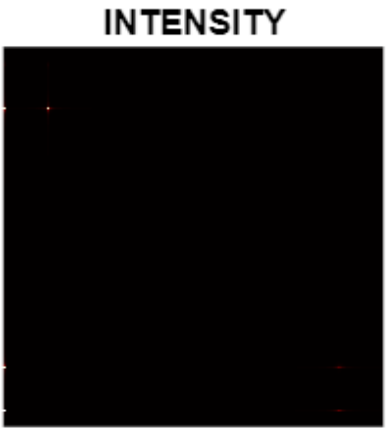
```
FFT(Sinusoid(15,0).*Sinusoid(30,pi/2));
```

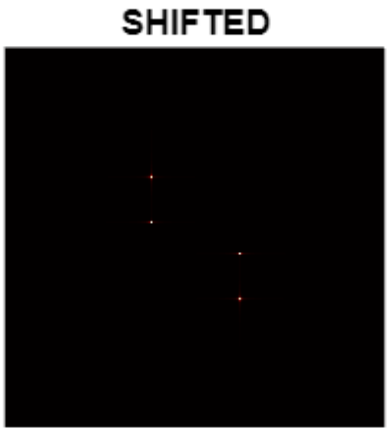


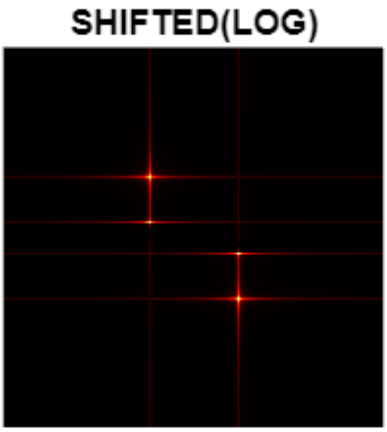
Several rotated sinusoids

```
FFT(Sinusoid(15,0).*Sinusoid(15,pi/2).*Sinusoid(30,pi/3));
```



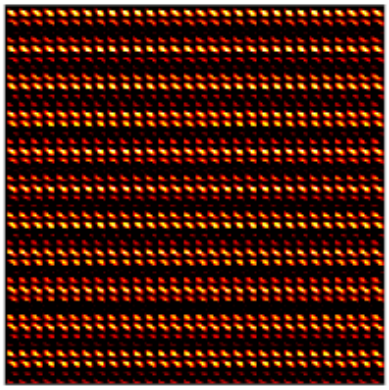




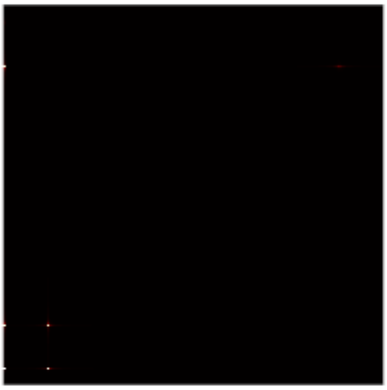



```
FFT(Sinusoid(15,0).*Sinusoid(15,pi/2).*Sinusoid(30,2*pi/3));
```

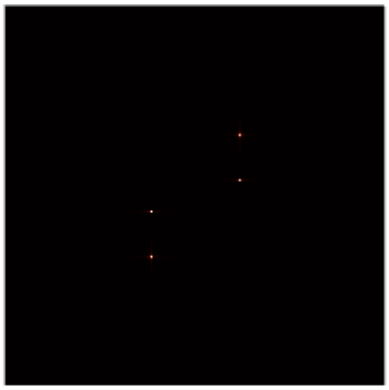
APERTURE



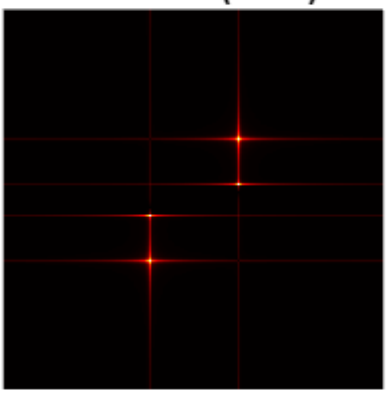
INTENSITY



SHIFTED

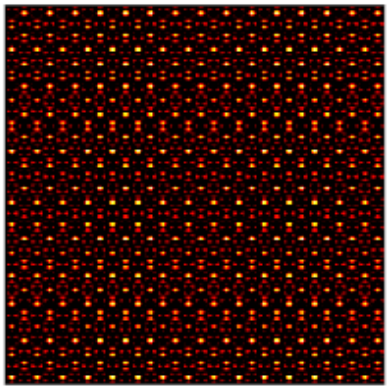


SHIFTED(LOG)

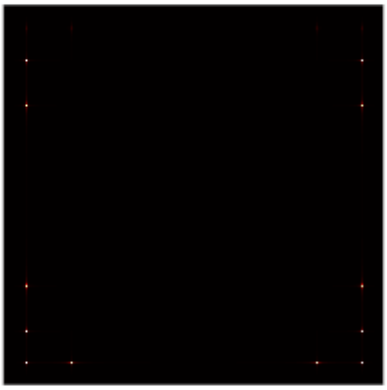


```
FFT(Sinusoid(15,0).*Sinusoid(15,pi/2).*Sinusoid(30,2*pi/3).*Sinusoid(30,pi/3));
```

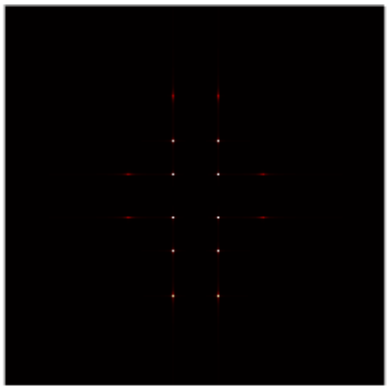
APERTURE



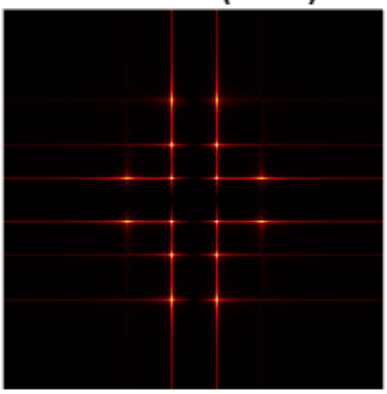
INTENSITY



SHIFTED

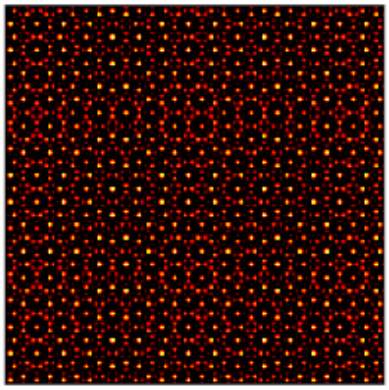


SHIFTED(LOG)

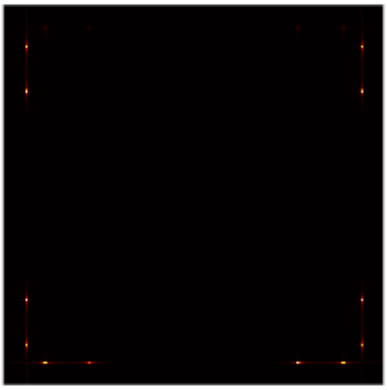


```
FFT(Sinusoid(15,0).*Sinusoid(15,pi/2).*Sinusoid(30,pi/4).*Sinusoid(30,3*pi/4));
```

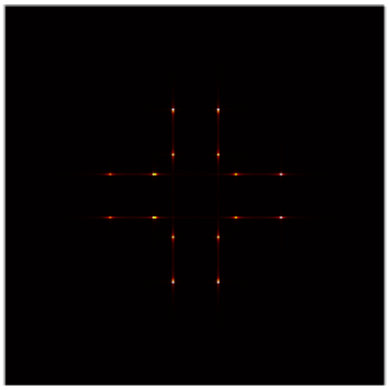
APERTURE



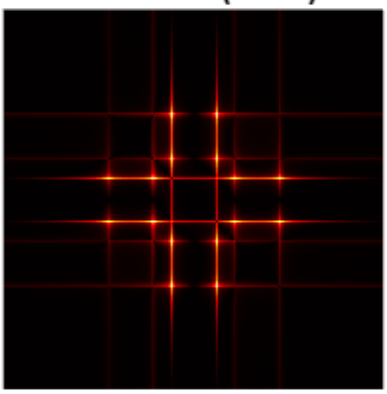
INTENSITY



SHIFTED

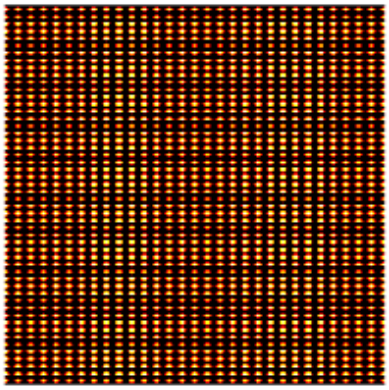


SHIFTED(LOG)



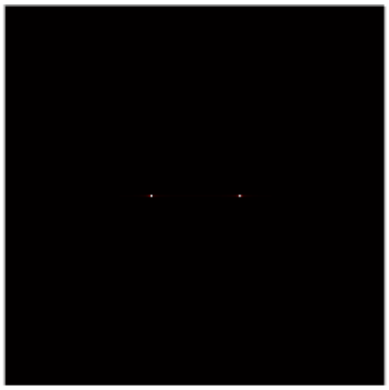
```
FFT(Sinusoid(30,pi/3).*Sinusoid(30,2*pi/3));
```

APERTURE

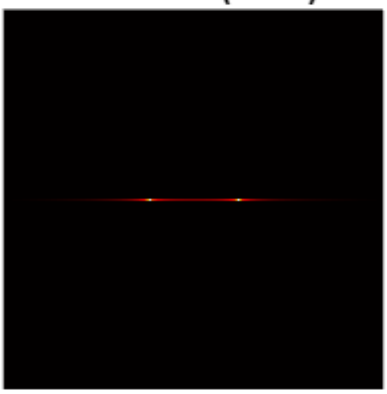


INTENSITY

SHIFTED

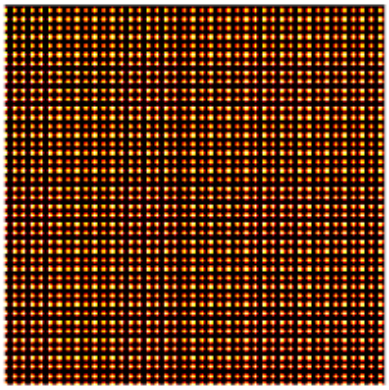


SHIFTED(LOG)



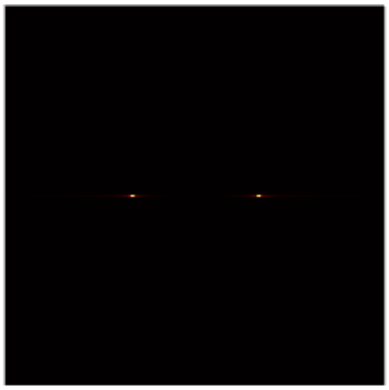
```
FFT(Sinusoid(30,pi/4).*Sinusoid(30,3*pi/4));
```

APERTURE

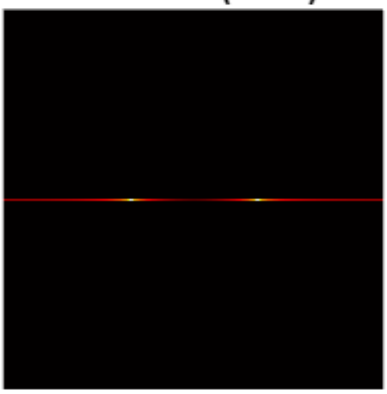


INTENSITY

SHIFTED

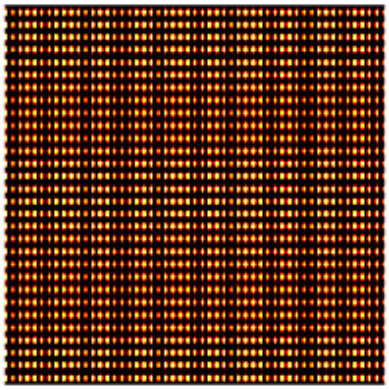


SHIFTED(LOG)



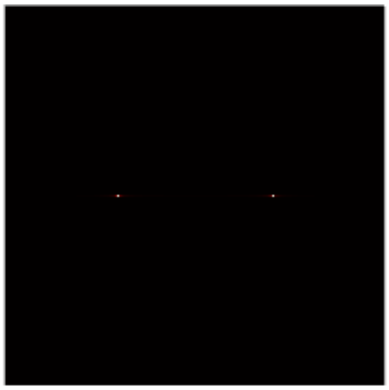
```
FFT(Sinusoid(30,pi/6).*Sinusoid(30,5*pi/6));
```

APERTURE

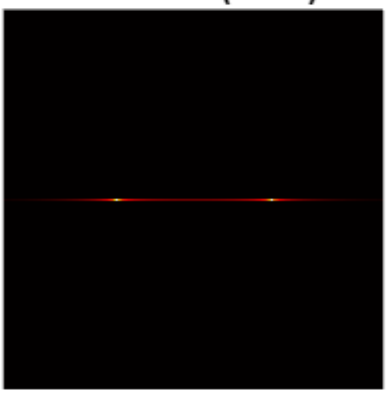


INTENSITY

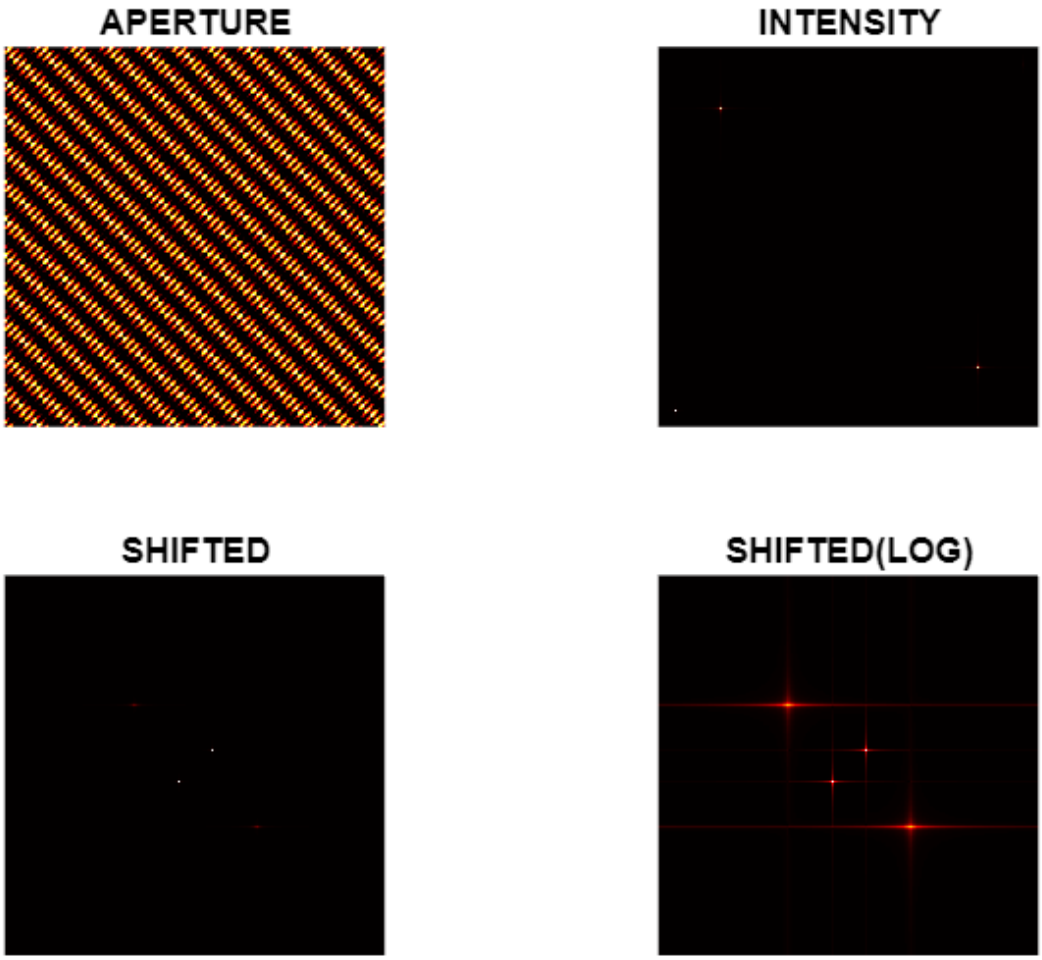
SHIFTED



SHIFTED(LOG)



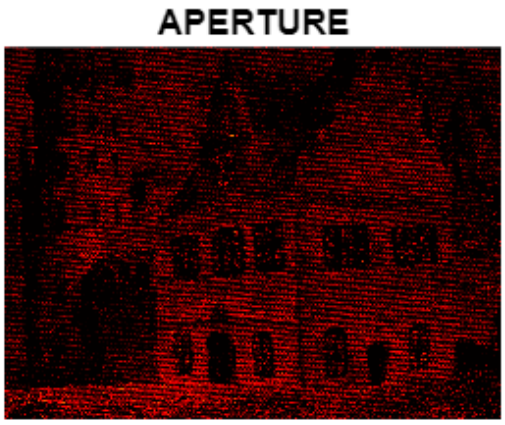
```
FFT(Sinusoid(30,pi/6).*Sinusoid(30,pi/3));
```

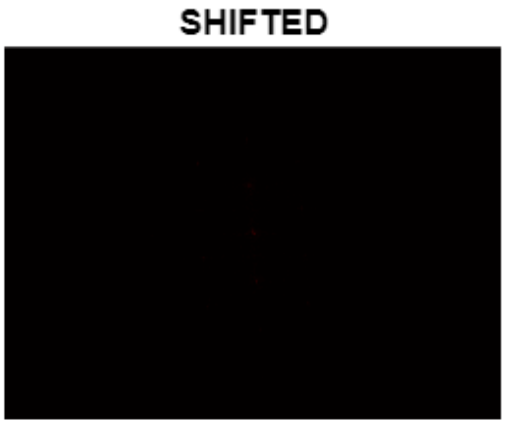


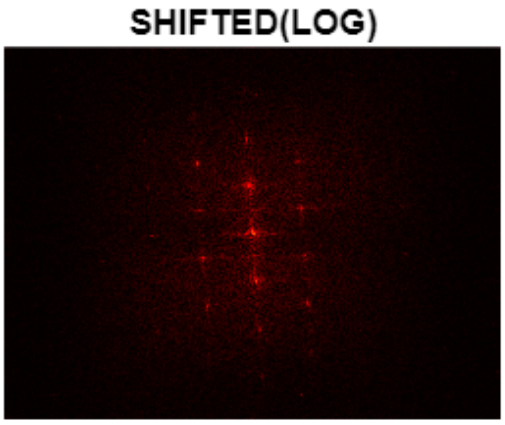
Activity 2.2.2. Application: Canvas Weave Modeling and Removal

I used the function meansubtract(image) then applied the FFT(aperture) function.

```
clear; close;
FFT(meansubtract('painting.jpg'));
```



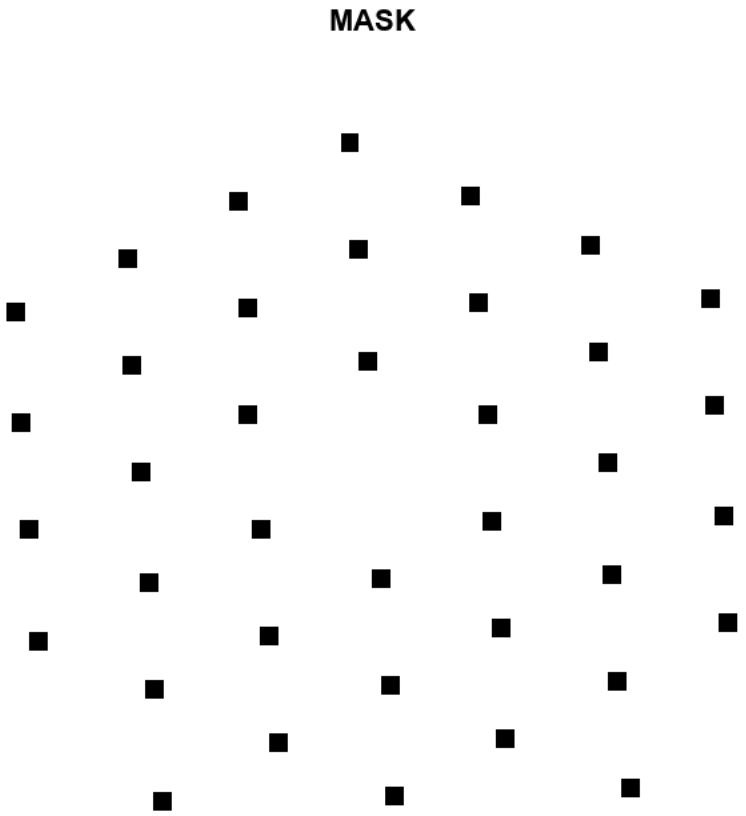




Creating the mask based on the intensity peaks found in the FFT shifted log.

```
close; clear;
%Mask
N = 960; M = 1280;
con = 0;
B = ones(N,M);
B(290:310, 490:510) = con;
B(410:430, 490:510) = con;
B(540:560, 505:525) = con;
B(660:680, 515:535) = con;
B(780:800, 525:545) = con;
B(595:615, 640:660) = con;
B(715:735, 650:670) = con;
B(840:860, 655:675) = con;
B(350:370, 625:645) = con;
B(225:245, 615:635) = con;
B(105:125, 605:625) = con;
```

```
B(410:430, 760:780) = con;
B(530:550, 765:785) = con;
B(650:670, 775:795) = con;
B(775:795, 780:800) = con;
B(285:305, 750:770) = con;
B(165:185, 740:760) = con;
B(170:190, 480:500) = con;
B(220:240, 875:895) = con;
B(340:360, 885:905) = con;
B(465:485, 895:915) = con;
B(590:610, 900:920) = con;
B(710:730, 905:925) = con;
B(235:255, 355:375) = con;
B(355:375, 360:380) = con;
B(475:495, 370:390) = con;
B(600:620, 380:400) = con;
B(720:740, 385:405) = con;
B(420:440, 235:255) = con;
B(540:560, 245:265) = con;
B(665:685, 255:275) = con;
B(295:315, 230:250) = con;
B(280:300, 1010:1030) = con;
B(400:420, 1015:1035) = con;
B(525:545, 1025:1045) = con;
B(645:665, 1030:1050) = con;
B(845:865, 395:415) = con;
B(830:850, 920:940) =con;
imshow(B); title('MASK')
```



Displaying the filtered image.

```
filtered('painting.jpg', B);
```

ORIGINAL

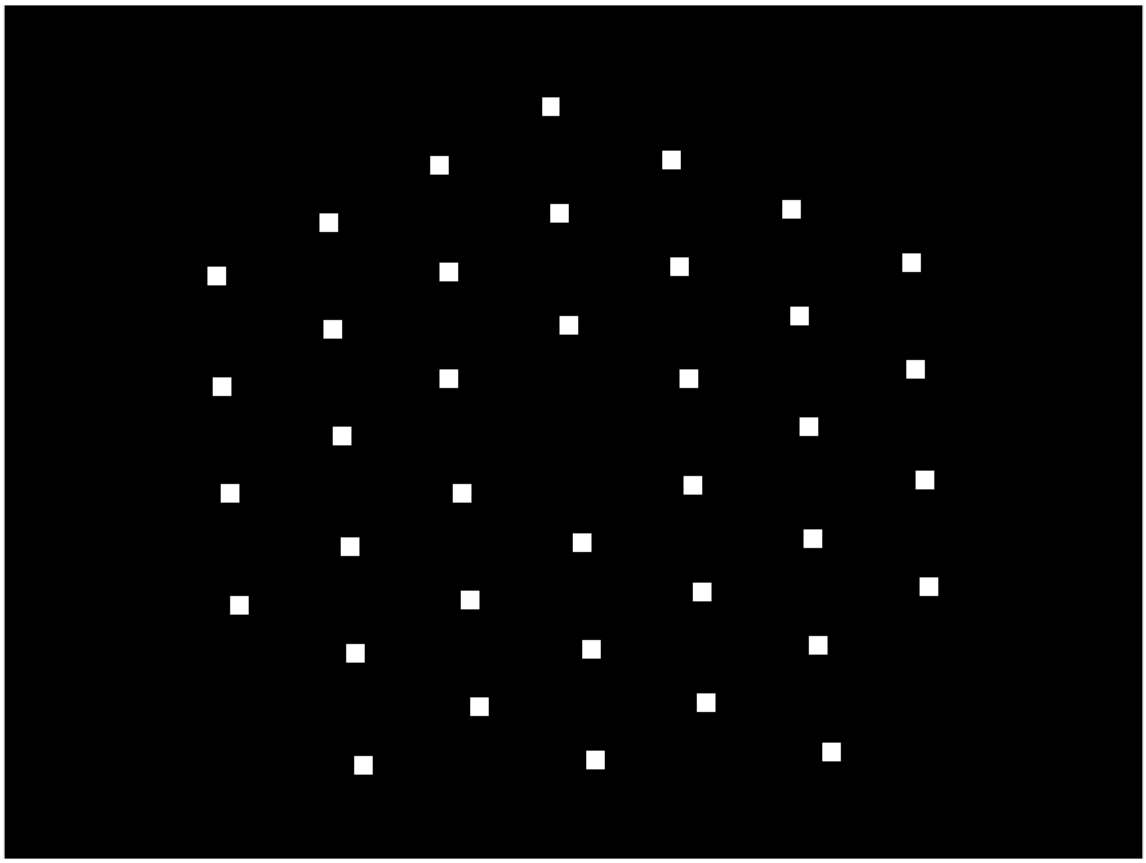
FILTERED

Trying to invert the image to get only the weaving patterns.

```
close; clear;
%Mask
N = 960; M = 1280;
con = 1;
B = zeros(N,M);
B(290:310, 490:510) = con;
B(410:430, 490:510) = con;
B(540:560, 505:525) = con;
B(660:680, 515:535) = con;
B(780:800, 525:545) = con;
B(595:615, 640:660) = con;
B(715:735, 650:670) = con;
B(840:860, 655:675) = con;
B(350:370, 625:645) = con;
B(225:245, 615:635) = con;
B(105:125, 605:625) = con;
B(410:430, 760:780) = con;
```

```
B(530:550, 765:785) = con;
B(650:670, 775:795) = con;
B(775:795, 780:800) = con;
B(285:305, 750:770) = con;
B(165:185, 740:760) = con;
B(170:190, 480:500) = con;
B(220:240, 875:895) = con;
B(340:360, 885:905) = con;
B(465:485, 895:915) = con;
B(590:610, 900:920) = con;
B(710:730, 905:925) = con;
B(235:255, 355:375) = con;
B(355:375, 360:380) = con;
B(475:495, 370:390) = con;
B(600:620, 380:400) = con;
B(720:740, 385:405) = con;
B(420:440, 235:255) = con;
B(540:560, 245:265) = con;
B(665:685, 255:275) = con;
B(295:315, 230:250) = con;
B(280:300, 1010:1030) = con;
B(400:420, 1015:1035) = con;
B(525:545, 1025:1045) = con;
B(645:665, 1030:1050) = con;
B(845:865, 395:415) = con;
B(830:850, 920:940) =con;
imshow(B); title('MASK 2')
```

MASK 2



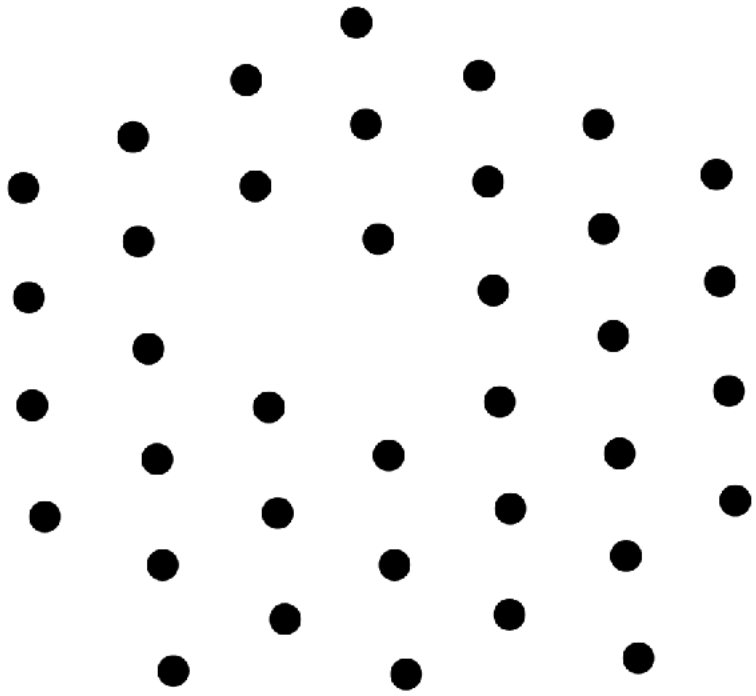
```
filtered('painting.jpg', B);
```

ORIGINAL

FILTERED

Trying a different mask. Same position of peaks but now with cicle patches instead of squares.

```
clear;close;
I=im2double(im2gray(imread("maskwhite.png")));
imshow(I);
```

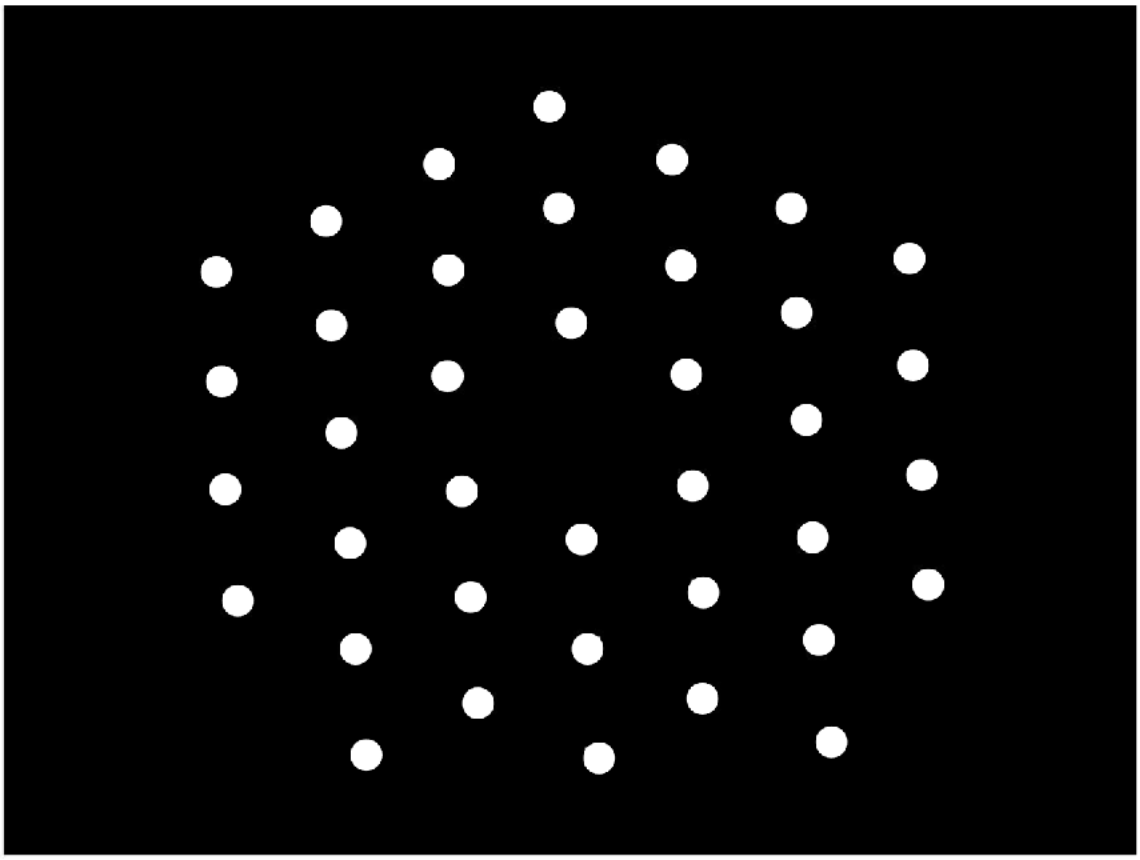


```
filtered('painting.jpg', I);
```

ORIGINAL

FILTERED

```
clear;close;
mask=im2double(rgb2gray(imread("maskpaint1.png")));
imshow(mask);
```



```
filtered('painting.jpg', mask);
```

ORIGINAL

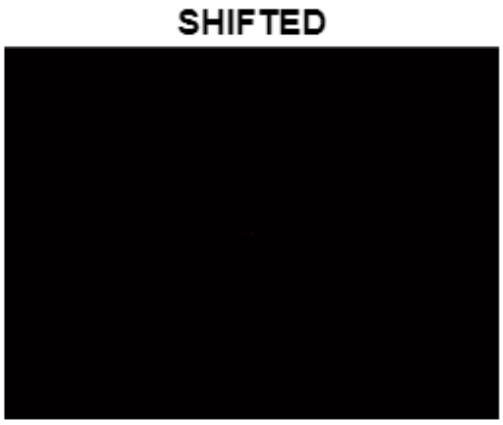
FILTERED

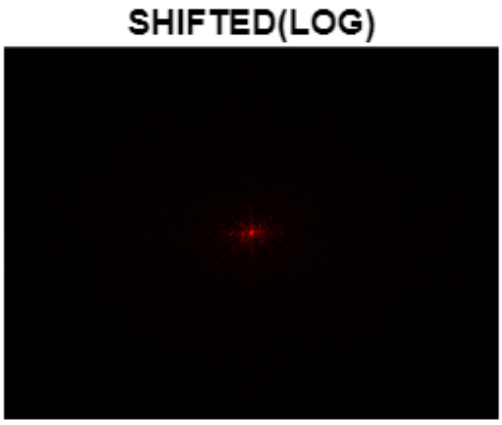
```
```

EXTRA CHALLENGE 1

Using the USANA eco bag logo picture, I obtained the mean-subtracted image and its FT.

```
clear; close;
FFT(meansubtract('usana.jpg'));
```

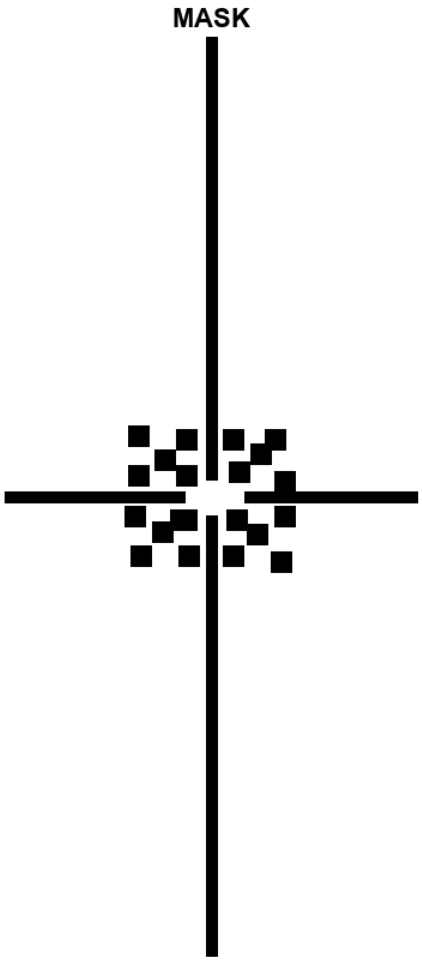




Creating the mask:

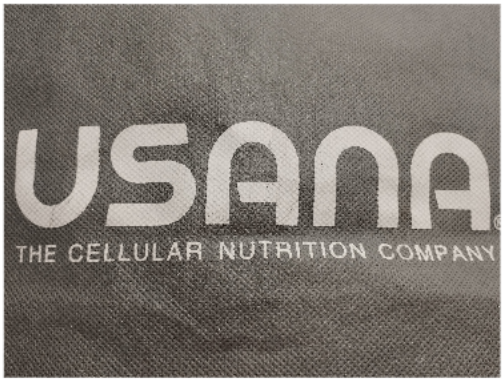
```
clear;close;
B = ones(3072, 4080);
B(1520:1560, 1350:1950) = 0;
B(1520:1560, 2150:2730) = 0;
B(1:1480, 2020:2060) = 0;
B(1600:3072, 2020:2060) = 0;
B(1580:1650, 1900:1990) =0;
B(1620:1690, 1840:1910) =0;
B(1430:1500, 1920:1990) = 0;
B(1380:1450, 1850:1920) = 0;
B(1420:1490, 2100:2170) = 0;
B(1360:1430, 2170:2240) = 0;
B(1580:1650, 2090:2160) = 0;
B(1630:1700, 2160:2230) = 0;
B(1700:1770, 1770:1840) = 0;
B(1300:1370, 1760:1830) = 0;
```

```
B(1310:1380, 2220:2290) = 0;
B(1720:1790, 2240:2310) = 0;
B(1310:1380, 1920:1990) = 0;
B(1430:1500, 1760:1830) = 0;
B(1570:1640, 1750:1820) = 0;
B(1700:1770, 1930:2000) = 0;
B(1700:1770, 2080:2150) = 0;
B(1570:1640, 2250:2320) = 0;
B(1450:1520, 2250:2320) = 0;
B(1310:1380, 2080:2150) = 0;
imshow(B); title('MASK');
```

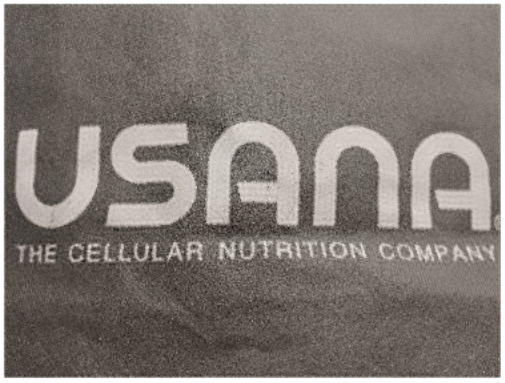



```
filtered('usana.jpg', B);
```

ORIGINAL



FILTERED

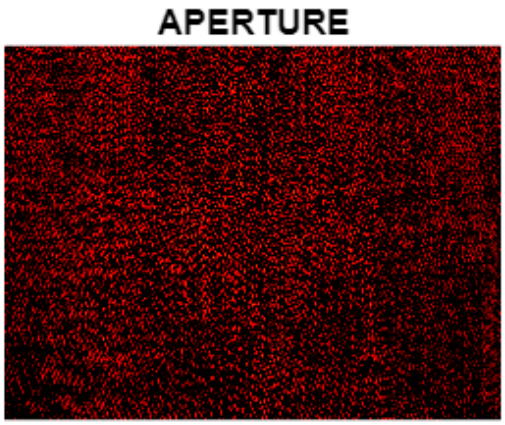


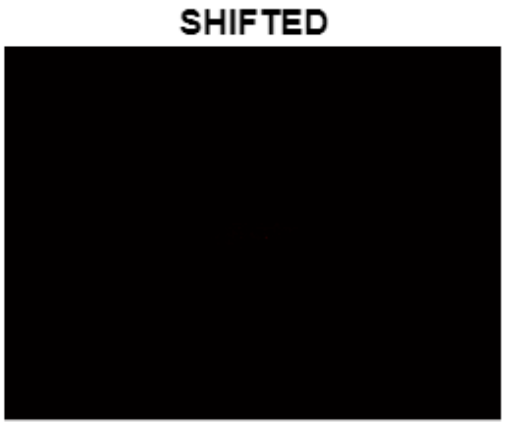
EXTRA CHALLENGE 2: KAKETSUGI

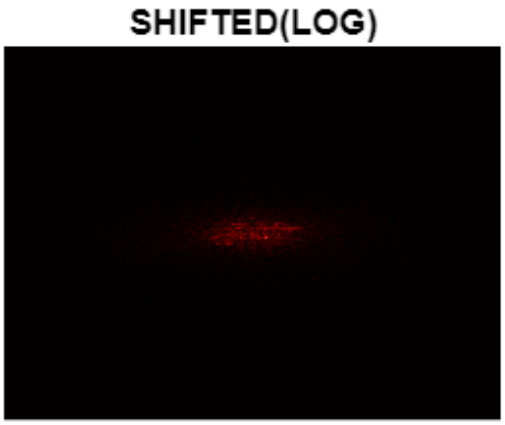
Here, I used two types of fabric: denim and katsa.

DENIM

```
clear; close;
FFT(meansubtract('denim.jpg'));
```

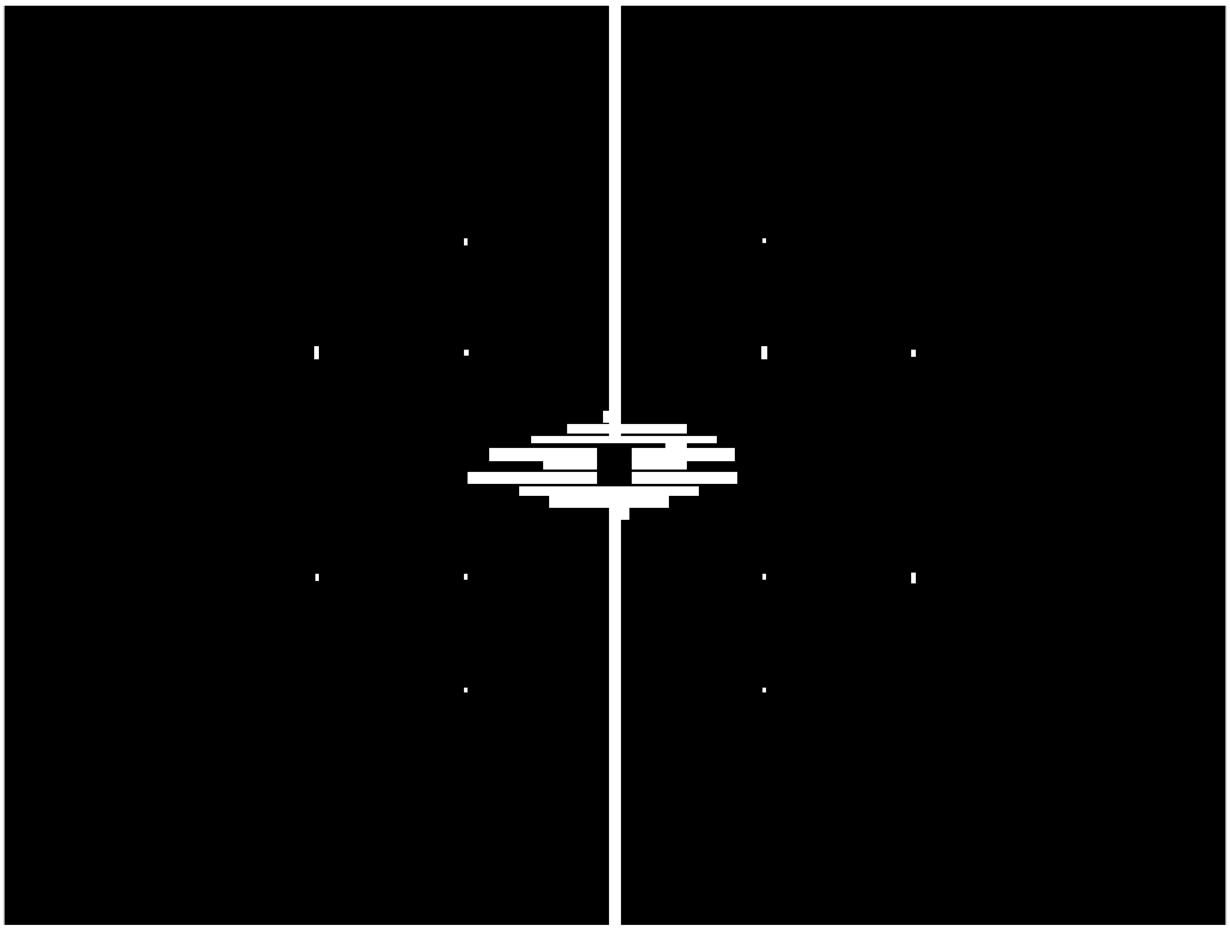







```
clear;close;
%Mask
C = zeros(3072, 4080);
C(1620:3072, 2020:2060) = 1; %straight bottom
C(1:1460, 2020:2060) = 1; %straight top
C(1520:1550, 1800:1980) =1;
C(1520:1550, 2100:2280) =1;
C(1355:1395, 2000:2050) = 1;
C(1400:1430, 1880:2280) = 1;
C(1440:1460,  1760:2380) = 1;
C(1440:1490, 2210:2280) =1;
C(1480:1520, 1620:1980) =1;
C(1480:1520, 2100:2440) =1;
C(1560:1600, 1550:1980) =1;
C(1560:1600, 2100:2450) =1;
C(1610:1640, 1720:2320) =1;
C(1640:1680, 1820:2220) =1;
C(1680:1720, 2030:2090) =1;
```

```
C(1895:1930, 3030:3045) = 1;
C(1900:1920, 2535:2545) =1;
C(2280:2295, 2535:2545) =1;
C(2280:2295, 1538:1546) =1;
C(1900:1920, 1538:1548) =1;
C(1900:1925, 1040:1050) =1;
C(1140:1180, 1035:1050) =1;
C(1150:1170, 1535:1550) =1;
C(780:800, 1538:1548) =1;
C(1140:1180, 2530:2550) =1;
C(780:795, 2535:2545) =1;
C(1150:1175, 3030:3045) =1;
imshow(C);
```



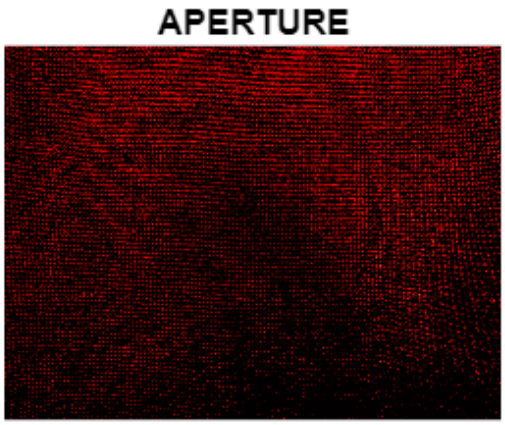
```
filtered('denim.jpg', C);
```

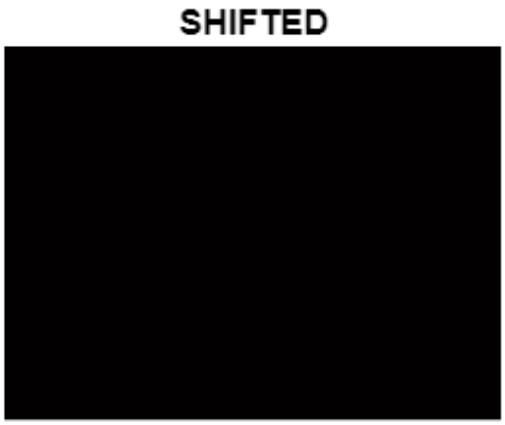
ORIGINAL

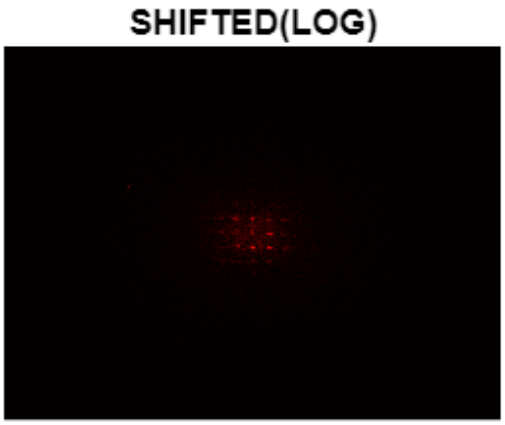
FILTERED

KATSA

```
clear; close;
FFT(meansubtract('tote.jpg'));
```








```
clear;close;
%Mask
D = zeros(3072, 4080);
D(1:1440, 2010:2080) = 1;
D(1640:3072,2010:2080) =1;

D(1160:1230, 2140:2210) =1;
D(1270:1340, 2140:2210) =1;
D(1370:1440, 2140:2210) =1;
D(1510:1580, 2140:2210) =1;
D(1620:1690, 2140:2210) =1;
D(1740:1810, 2140:2210) =1;
D(1850:1920, 2140:2210) =1;

D(1160:1230, 2260:2370) =1;
D(1270:1340, 2260:2370) =1;
D(1370:1440, 2260:2370) =1;
D(1510:1580, 2260:2370) =1;
```

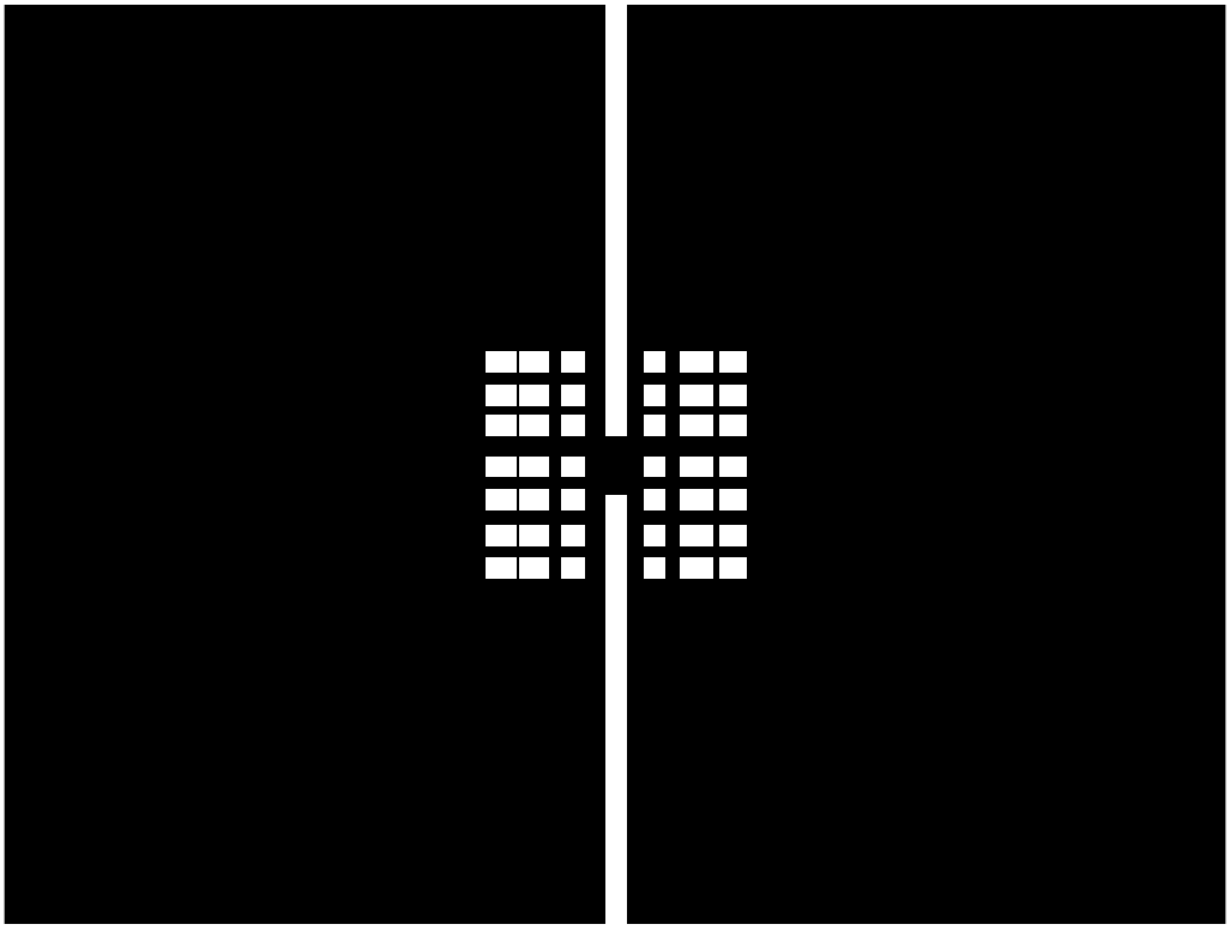
```
D(1620:1690, 2260:2370) =1;
D(1740:1810, 2260:2370) =1;
D(1850:1920, 2260:2370) =1;

D(1160:1230, 2390:2480) =1;
D(1160:1230, 2390:2480) =1;
D(1270:1340, 2390:2480) =1;
D(1370:1440, 2390:2480) =1;
D(1510:1580, 2390:2480) =1;
D(1620:1690, 2390:2480) =1;
D(1740:1810, 2390:2480) =1;
D(1850:1920, 2390:2480) =1;

D(1160:1230, 1860:1940) =1;
D(1160:1230, 1860:1940) =1;
D(1270:1340, 1860:1940) =1;
D(1370:1440, 1860:1940) =1;
D(1510:1580,1860:1940) =1;
D(1620:1690,1860:1940) =1;
D(1740:1810, 1860:1940) =1;
D(1850:1920, 1860:1940) =1;

D(1160:1230, 1720:1820) =1;
D(1160:1230, 1720:1820) =1;
D(1270:1340, 1720:1820) =1;
D(1370:1440, 1720:1820) =1;
D(1510:1580,1720:1820) =1;
D(1620:1690,1720:1820) =1;
D(1740:1810, 1720:1820) =1;
D(1850:1920, 1720:1820) =1;

D(1160:1230, 1610:1710) =1;
D(1160:1230, 1610:1710) =1;
D(1270:1340, 1610:1710) =1;
D(1370:1440, 1610:1710) =1;
D(1510:1580,1610:1710) =1;
D(1620:1690,1610:1710) =1;
D(1740:1810, 1610:1710) =1;
D(1850:1920, 1610:1710) =1;
imshow(D)
```

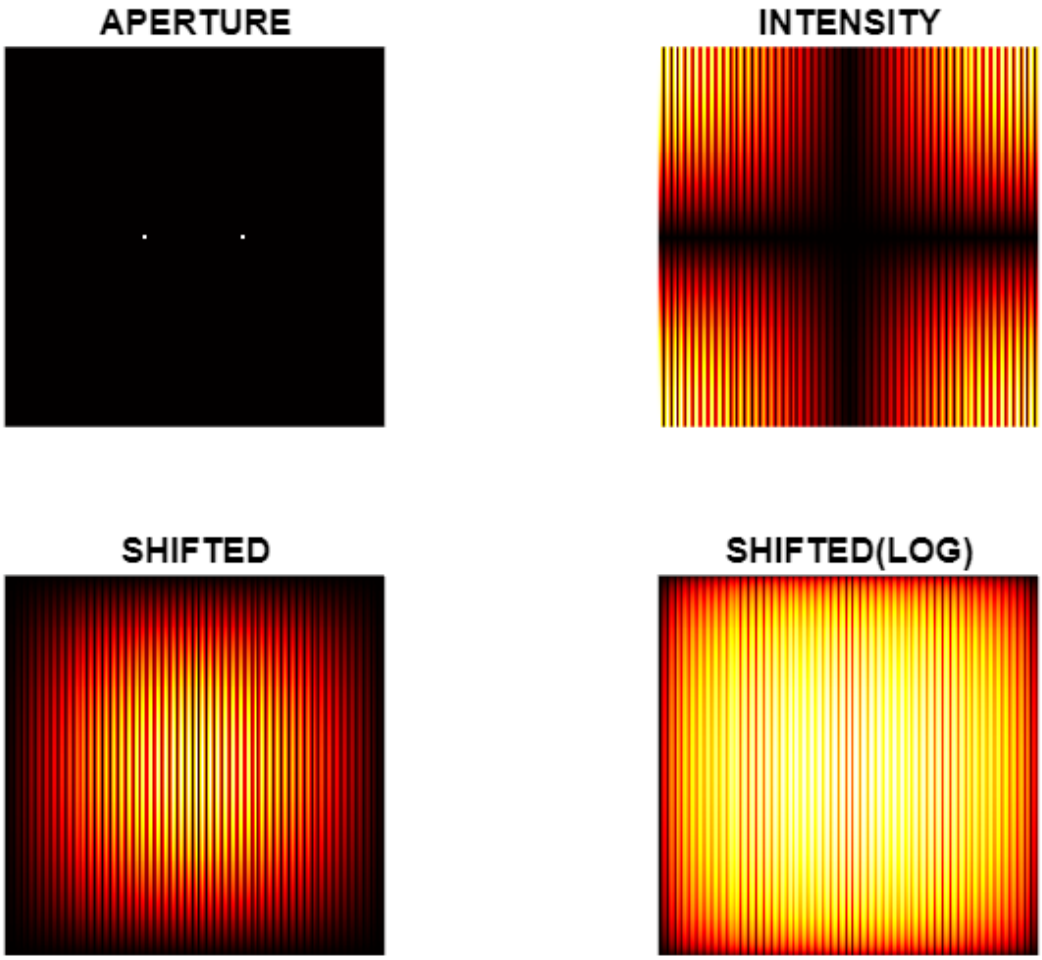


```
filtered('tote.jpg', D);
```

Activity 2.2.3. Convolution Theorem Redux

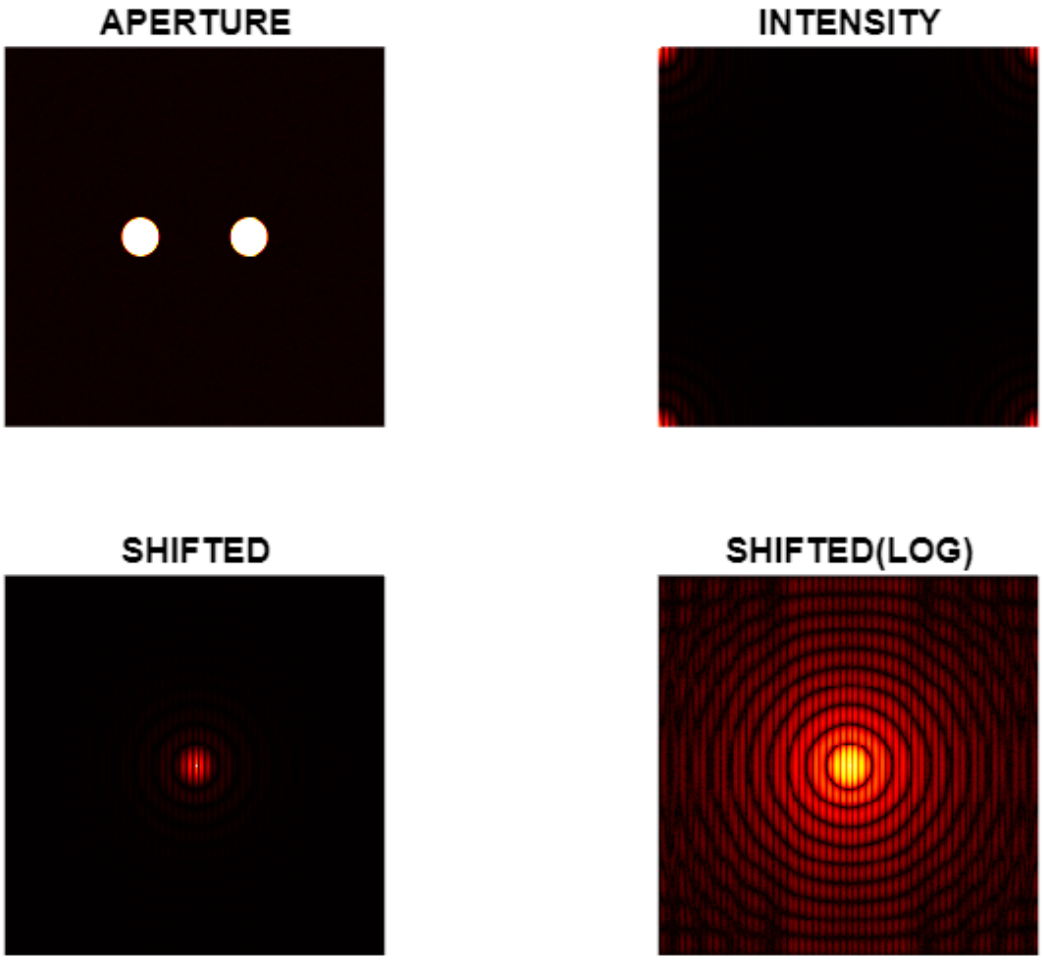
1. Create a binary mage of two dots (one pixel each) along the x-axis symmetric about center. Take the FT and display the modulus.

```
clear; close;
FFT(Slit(100,101,74,75,100,101,125,126));
```

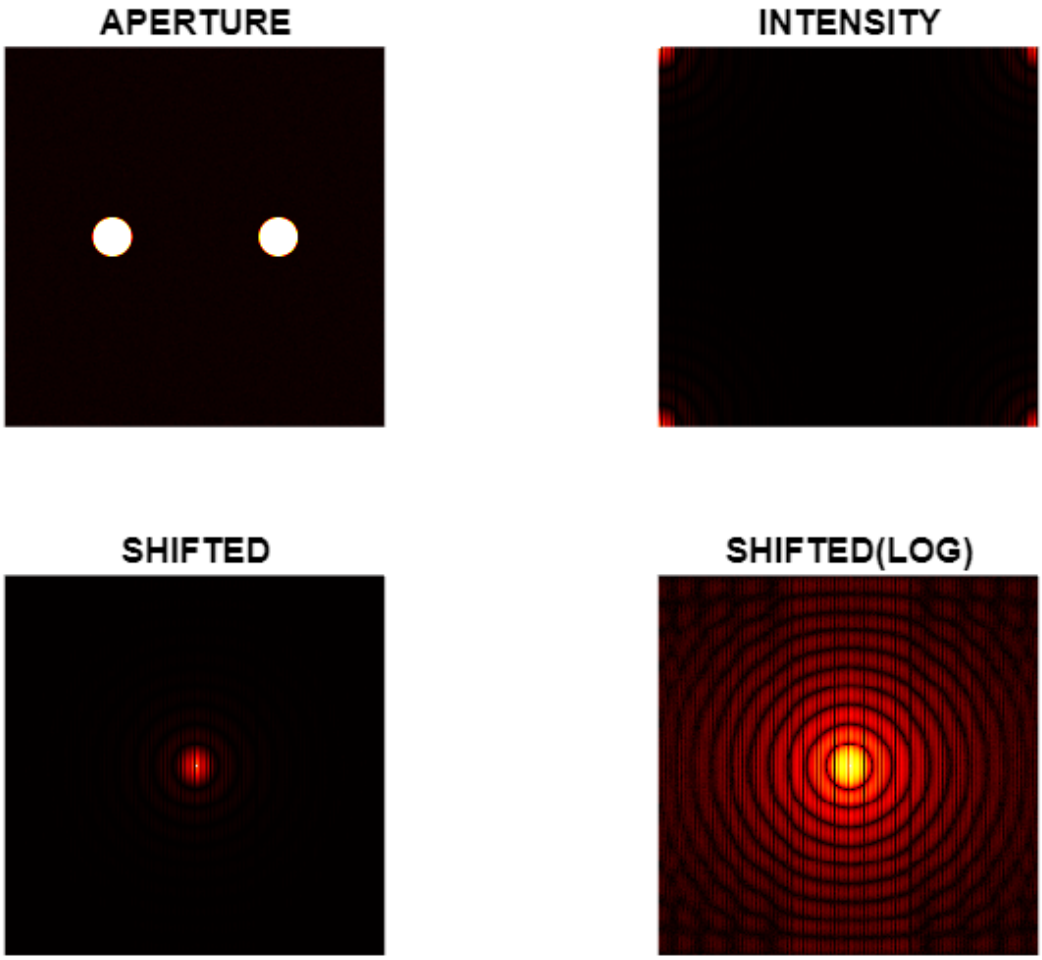


2. Replace the dots with circles of some radius. Discuss what you observe in the FT modulus as you vary the radius.

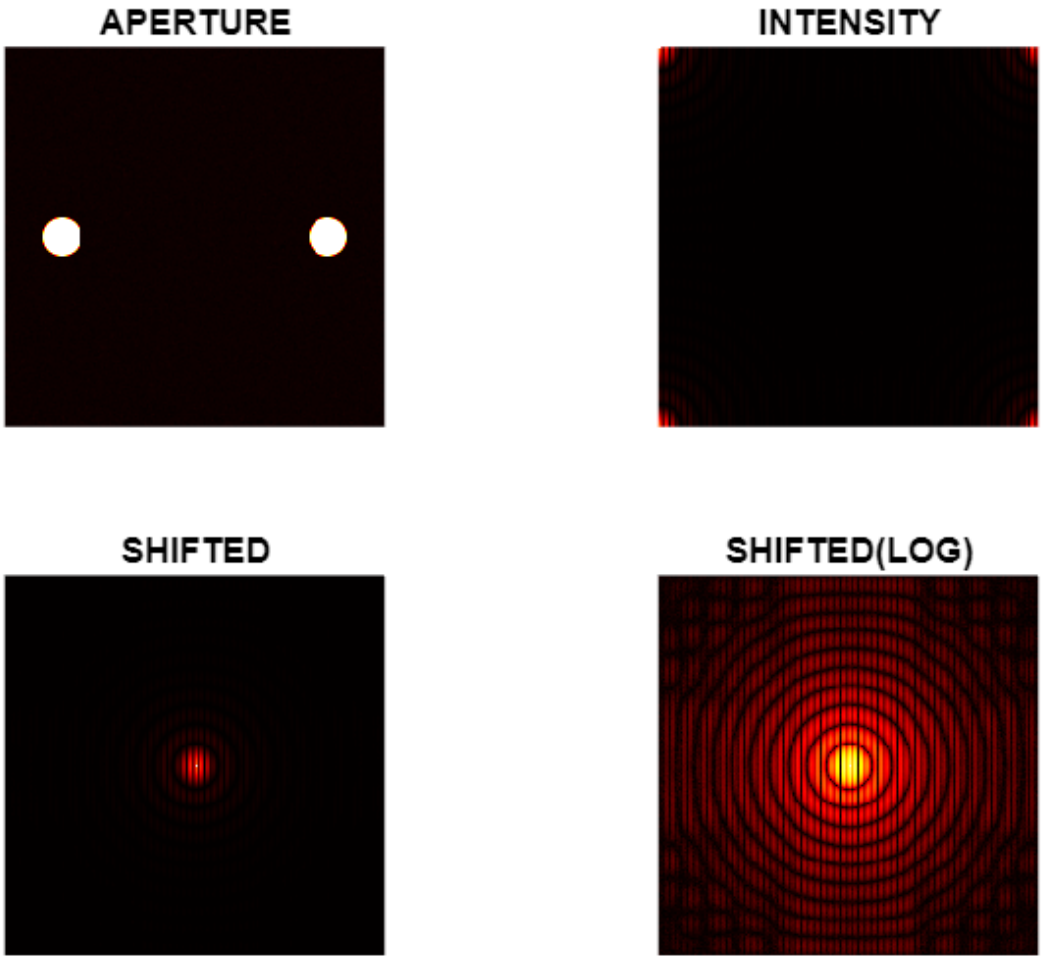
```
clear;close;
I = im2gray(imread('small1.png'));
FFT(I);
```



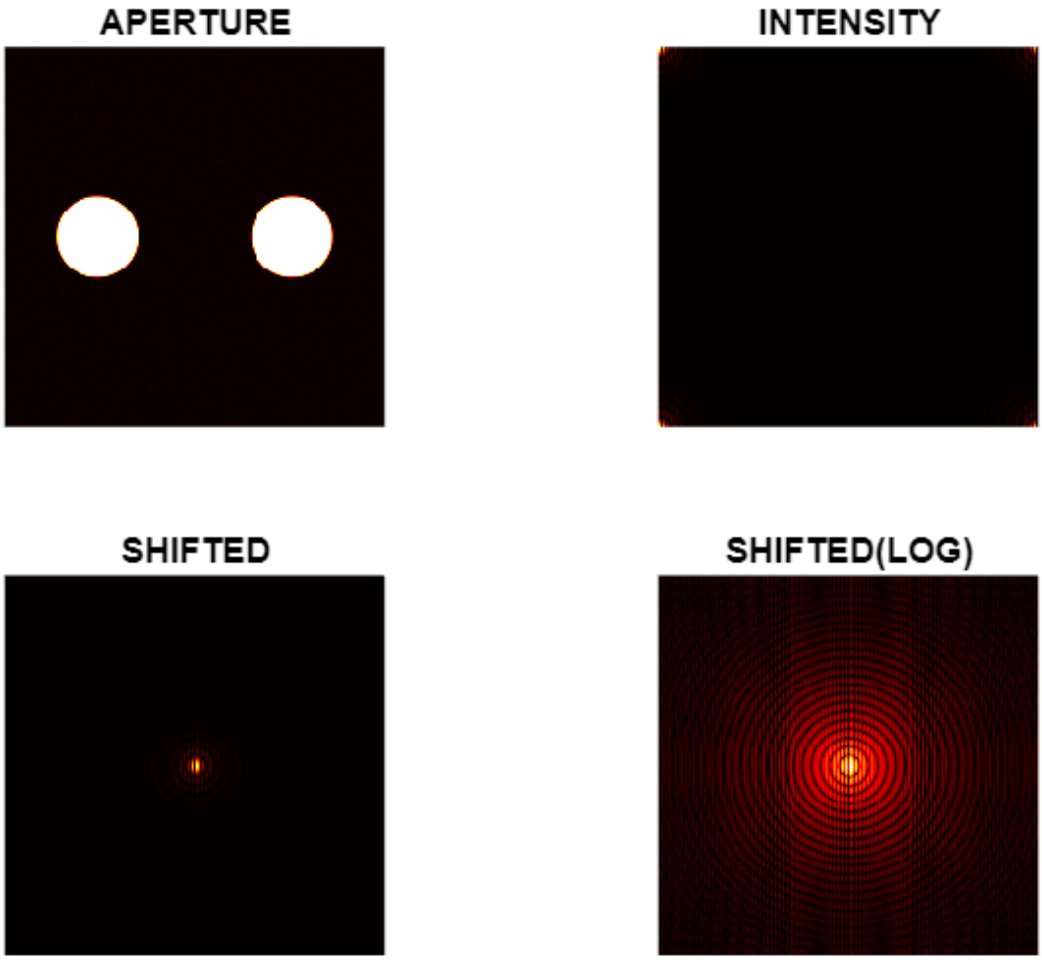

```
clear;close;
I = im2gray(imread('small2.png'));
FFT(I);
```



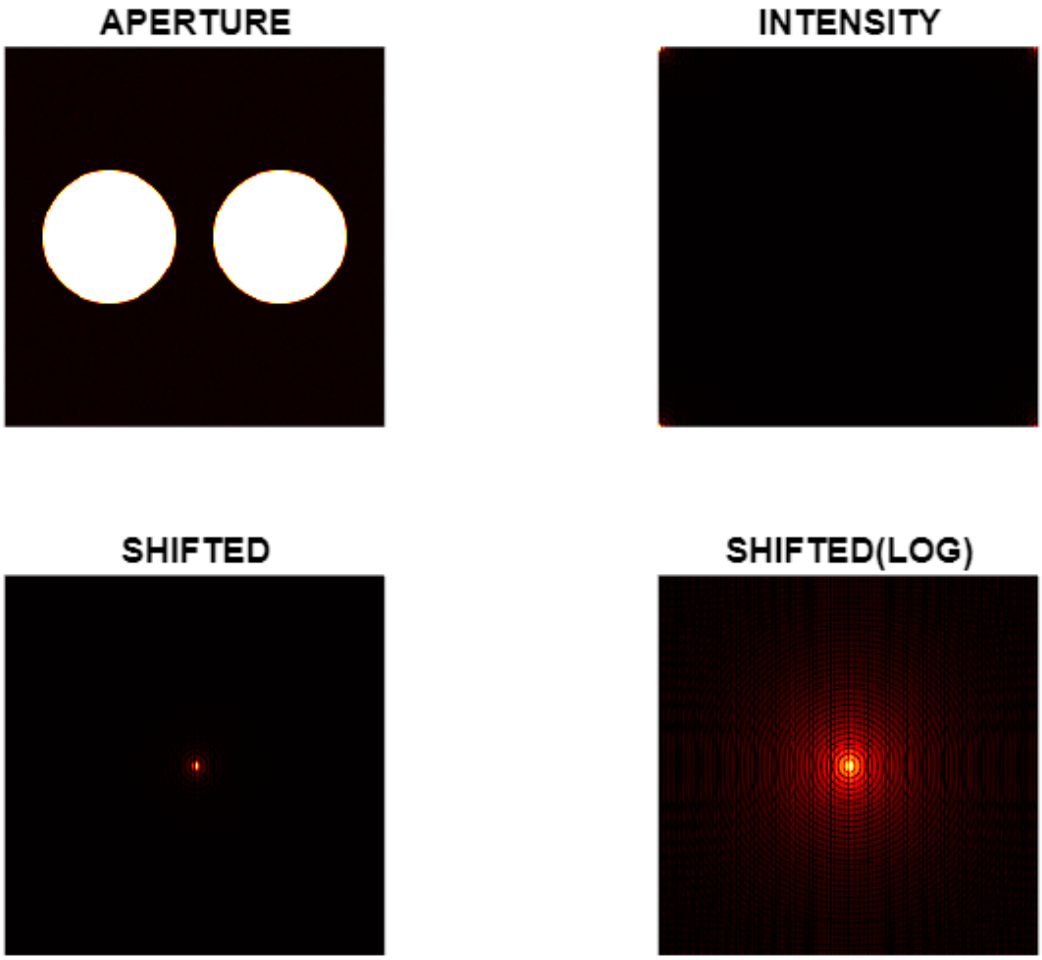

```
clear;close;
I = im2gray(imread('small3.png'));
FFT(I);
```




```
clear;close;
I = im2gray(imread('medium.png'));
FFT(I);
```

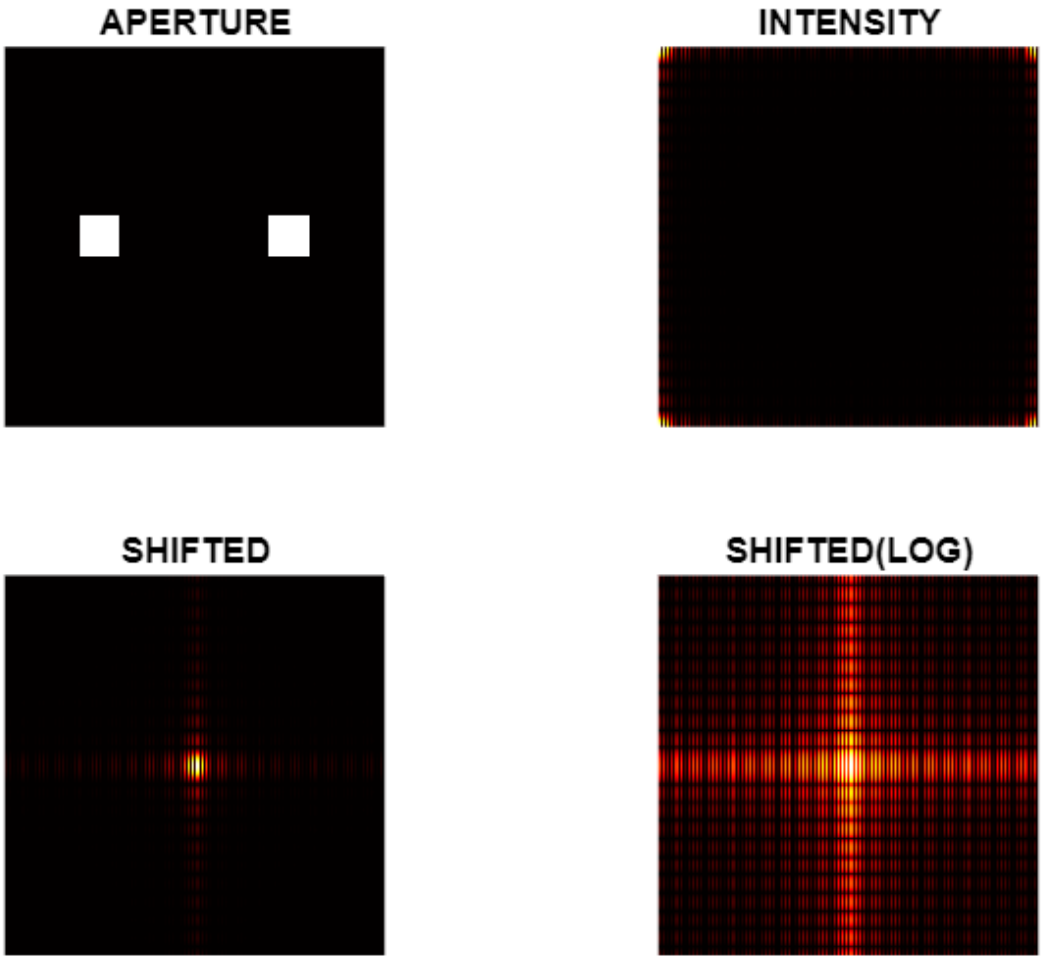



```
clear;close;
I = im2gray(imread('large.png'));
FFT(I);
```

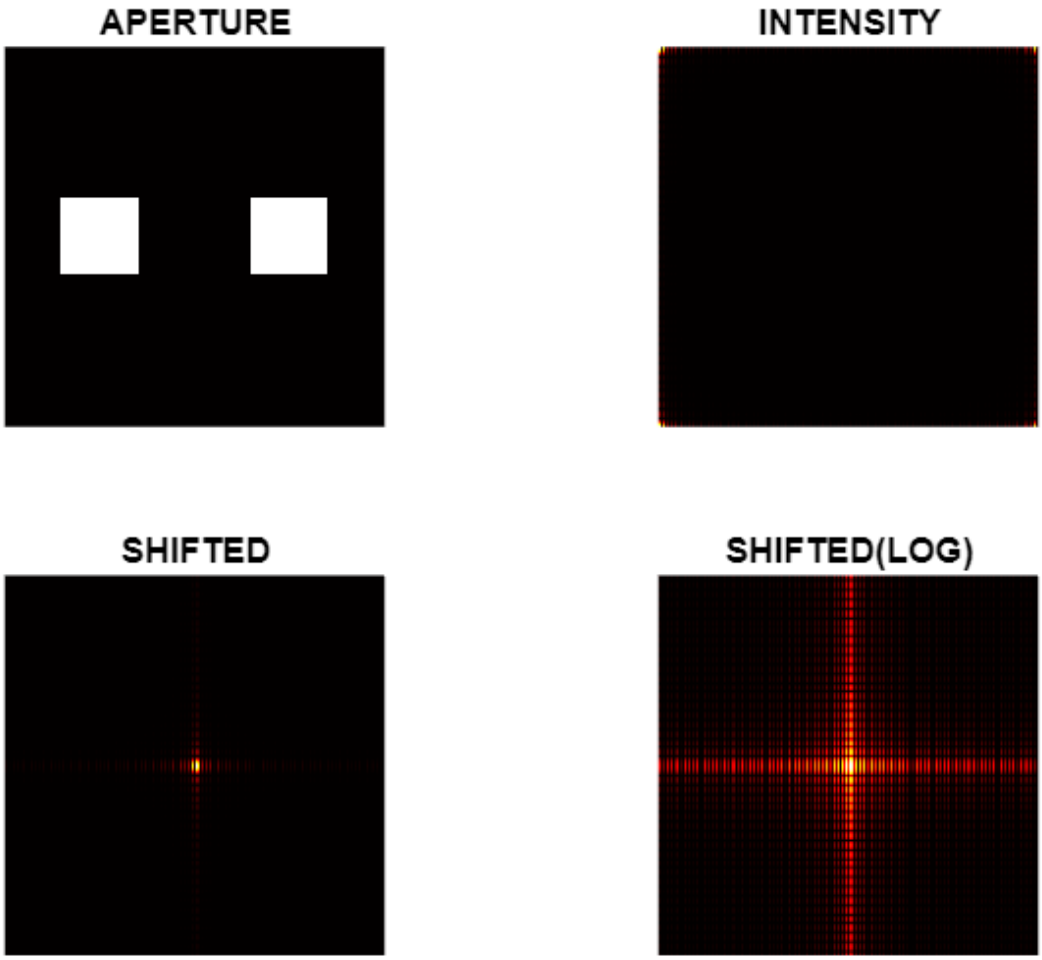


3. Replace the dots with squares of some width. Discuss what you observe in the FT modulus as you vary the width.

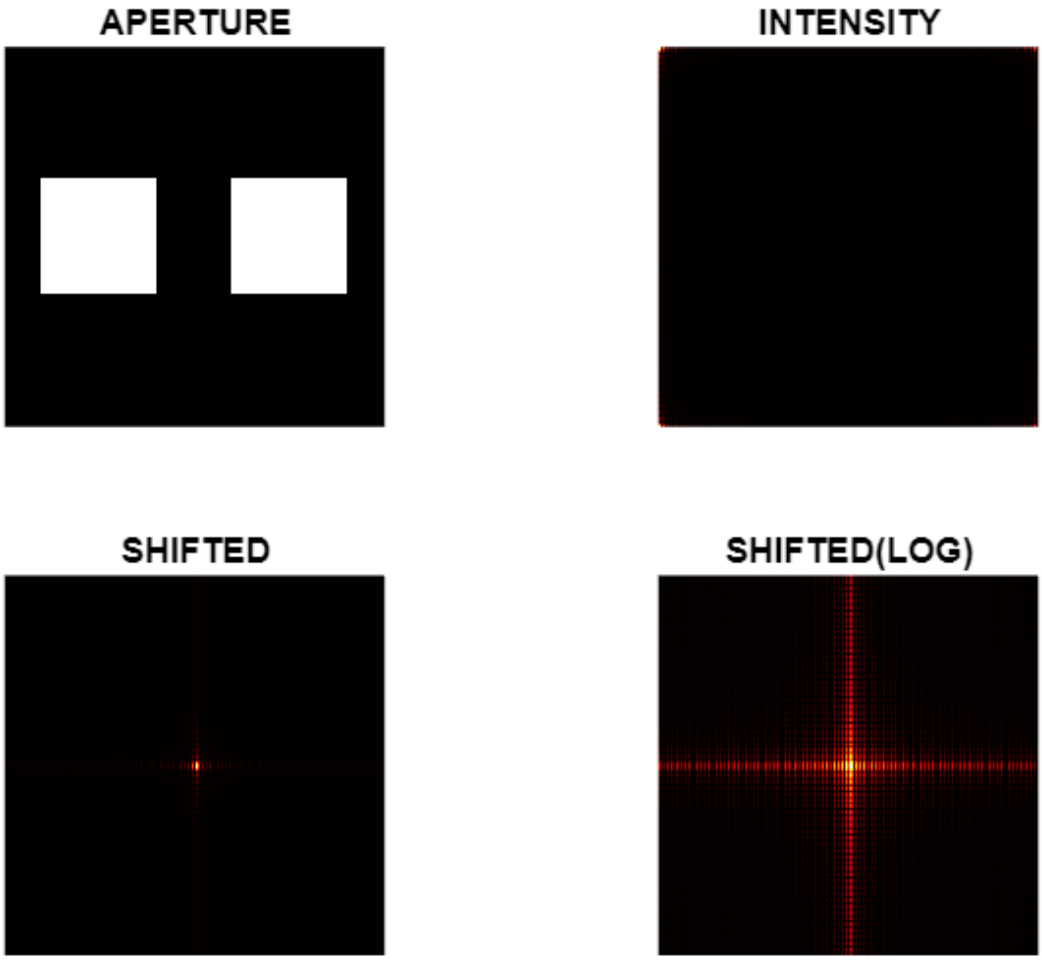
```
clear; close;
FFT(Slit(90,110,40,60,90,110,140,160));
```




```
clear; close;
FFT(Slit(80,120,30,70,80,120,130,170));
```

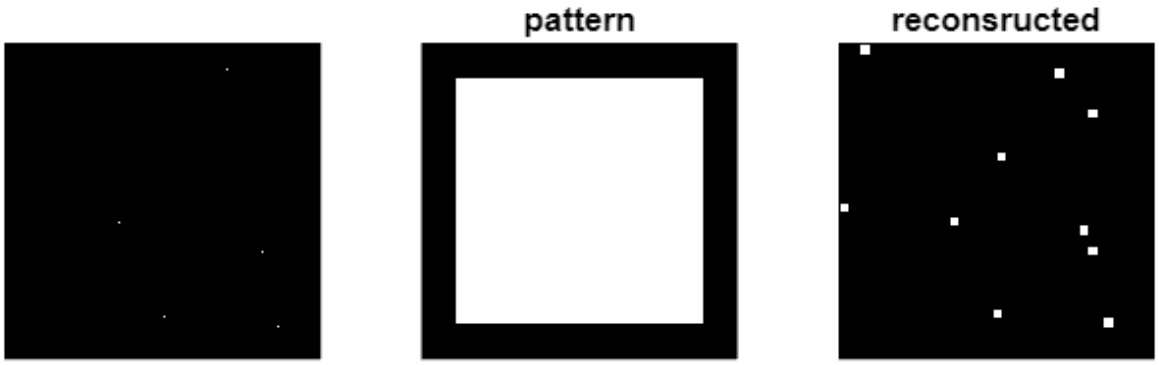



```
clear; close;
FFT(Slit(70,130,20,80,70,130,120,180));
```

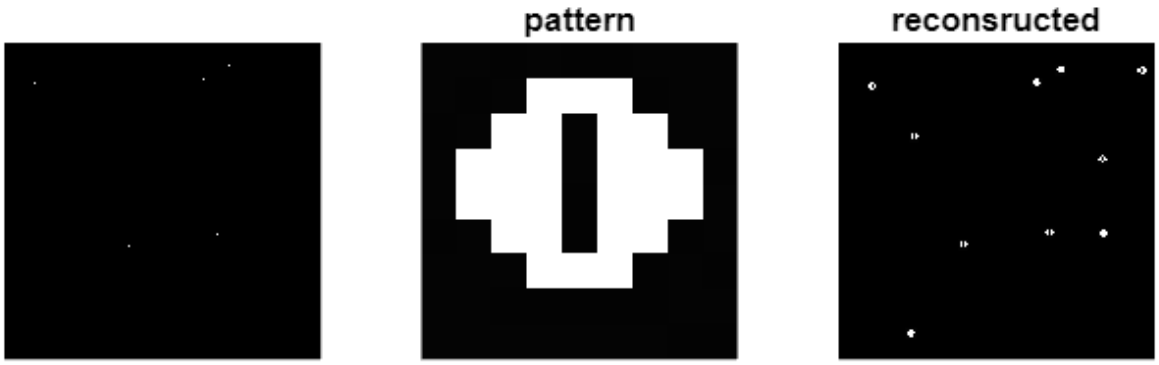


4. Create a 200x200 array of zeros. Put 10 1's in random locations in the array. These ones will approximate dirac deltas. Call this array A. Create an arbitrary 9x9 pattern, call it d. Convolve A and d.

```
clear; close;
%pattern d
M = 200;
d = zeros(9,9);
d(2:8, 2:8) = 1;
A = random(10);
subplot(1,3,1); imshow(A); axis image; axis off;
subplot(1,3,2); imshow(d); axis image; axis off; title('pattern');
subplot(1,3,3);imshow(conv2(d,A)); axis image; axis off; title('reconsructed');
```

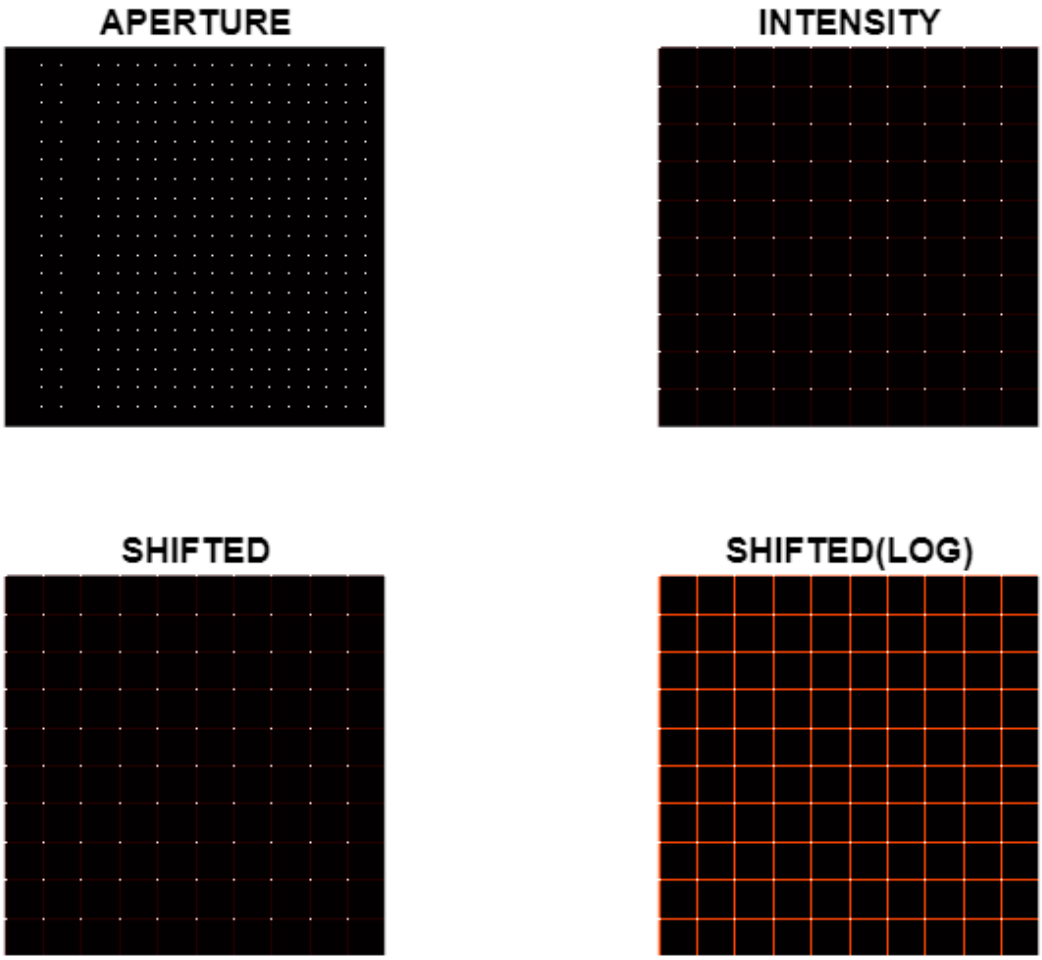



```
clear;close;
I = rgb2gray(im2double(imread('patternn.png')));
A=random(10);
subplot(1,3,1); imshow(A); axis image; axis off;
subplot(1,3,2); imshow(I); axis image; axis off; title('pattern');
subplot(1,3,3);imshow(conv2(A,I)); axis image; axis off; title('reconsructed');
```

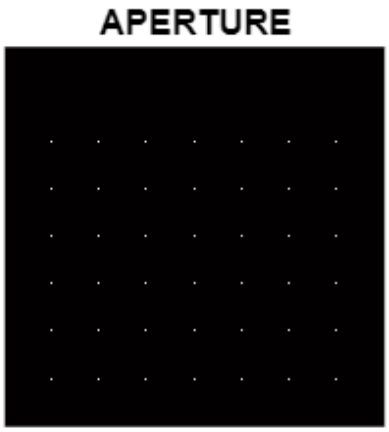


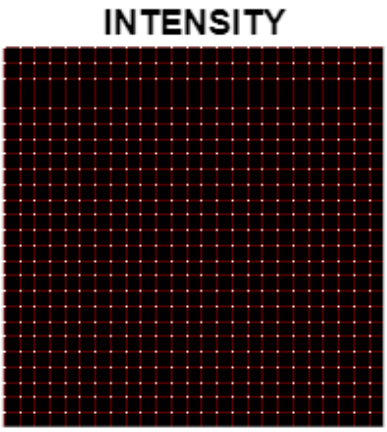
5. Create another 200x200 array of zeros but this time put equally spaced 1's along the x and y axis in the array. Get the FT and display the modulus. Change the spacing of the 1's and repeat.

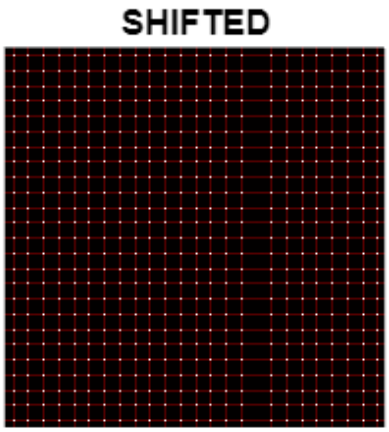
```
clear;close;
A = zeros(200,200);
for i = 10:10:190
    for j = 10:10:190
        A(i,j) = 1;
    end
end
FFT(A);
```

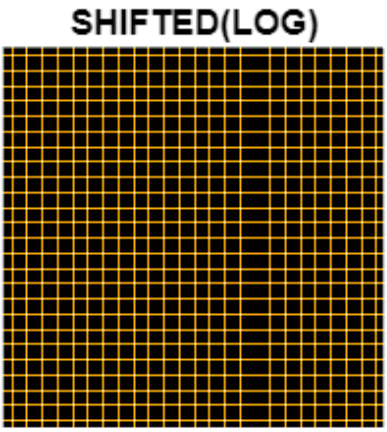



```
clear;close;
A = zeros(200,200);
for i = 25:25:175
    for j = 25:25:175
        A(i,j) = 1;
    end
end
FFT(A);
```

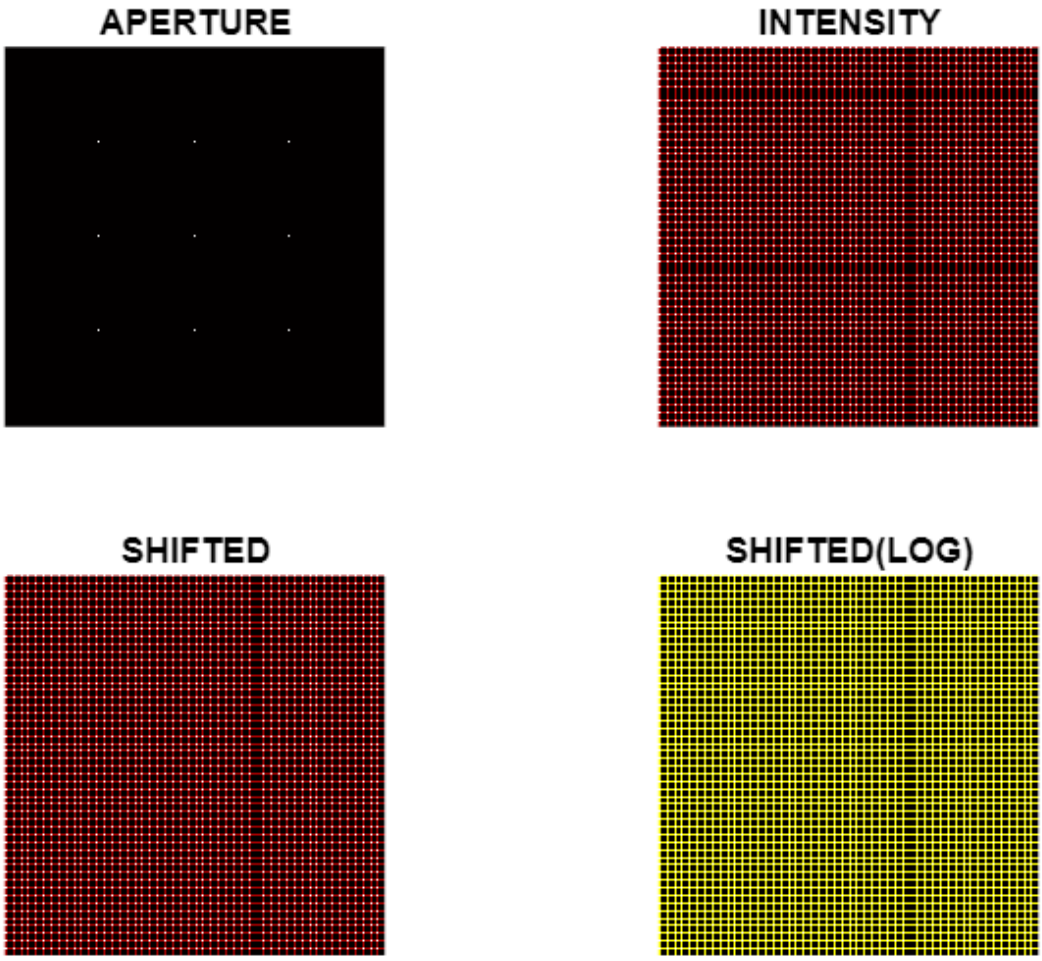









```
clear;close;
A = zeros(200,200);
for i = 50:50:150
    for j = 50:50:150
        A(i,j) = 1;
    end
end
FFT(A);
```



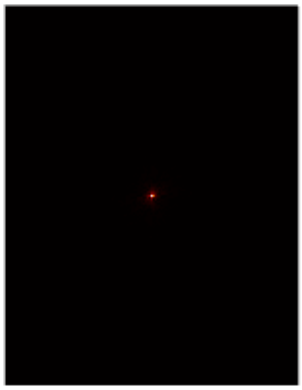
Acitivity 2.2.4. Fingerprints: Ridge Enhancement

```
clear;close;
FFT(meansubtract('fpjanen.jpg'));
```

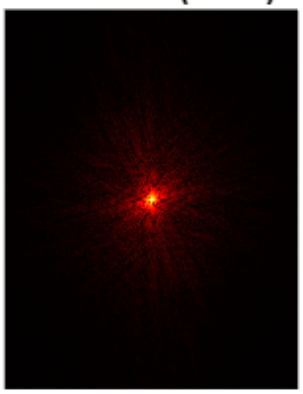
APERTURE

INTENSITY

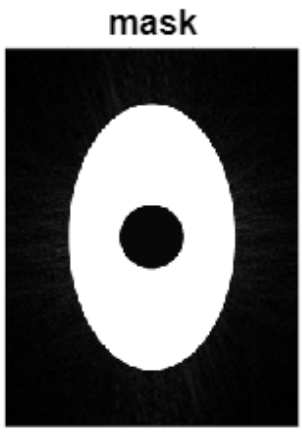
SHIFTED



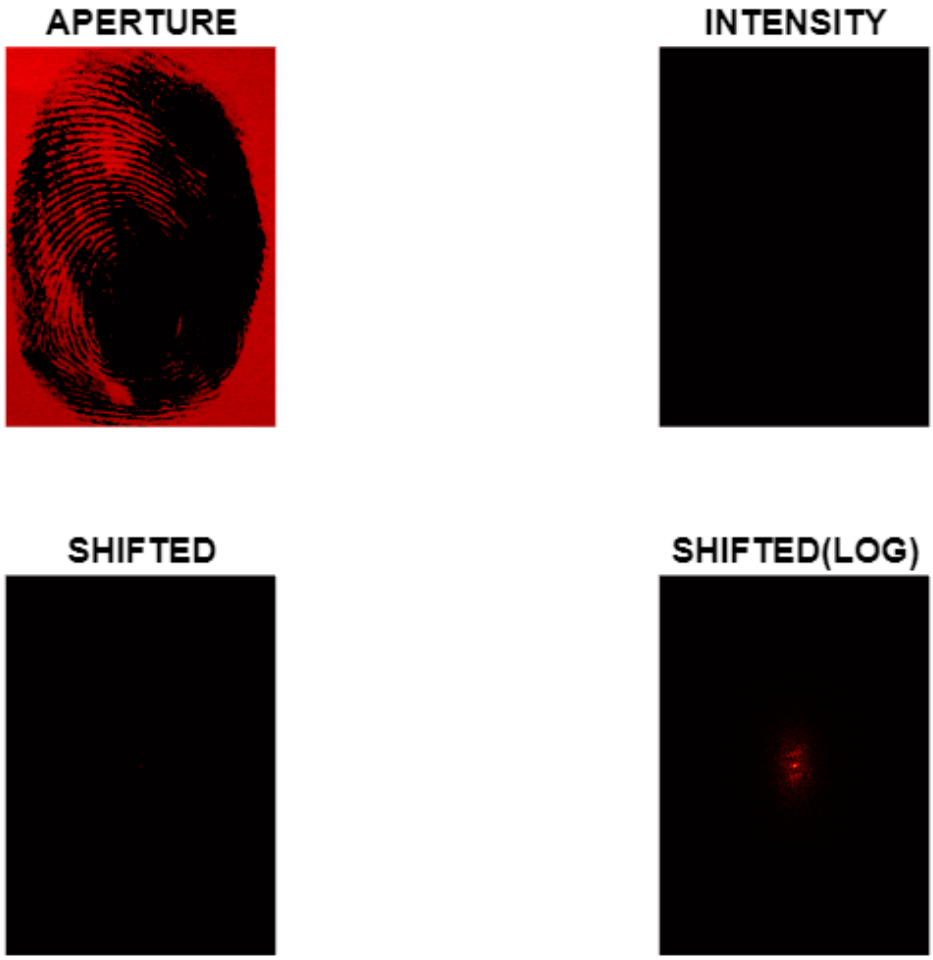
SHIFTED(LOG)



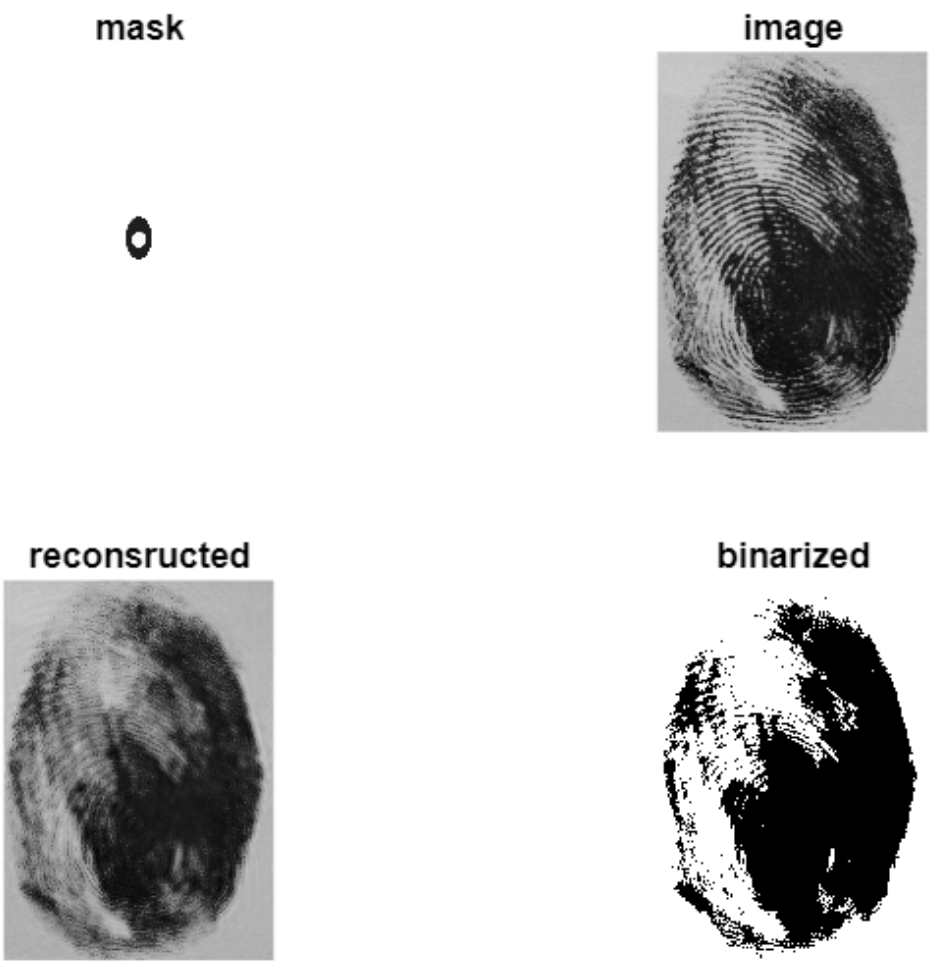
```
clear;close;
Convolution('fpjanenmask.png','fpjanen.jpg');
```




```
clear;close;
FFT(meansubtract('fingerprintmom.jpg'));
```




```
clear;close;
Convolution('maskmom3.png', 'fingerprintmom.jpg');
```




```
clear;close;
Convolution('maskmom4.png', 'fingerprintmom.jpg');
```

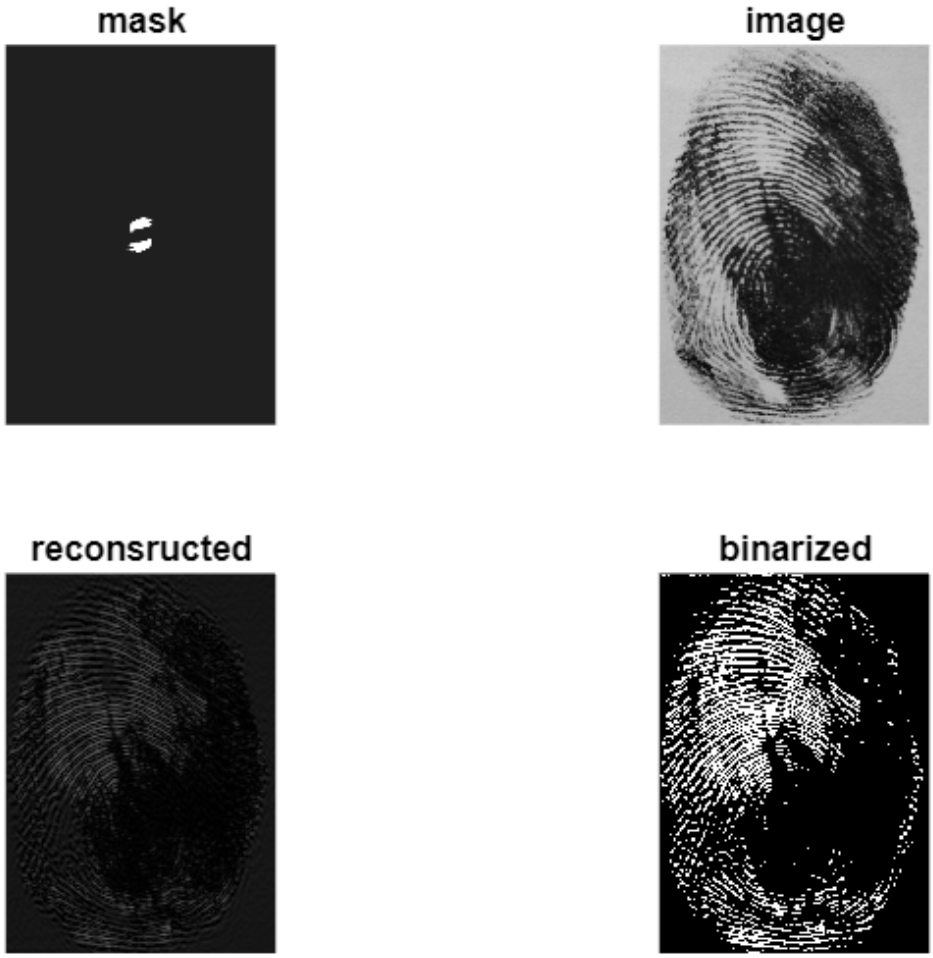
mask

image

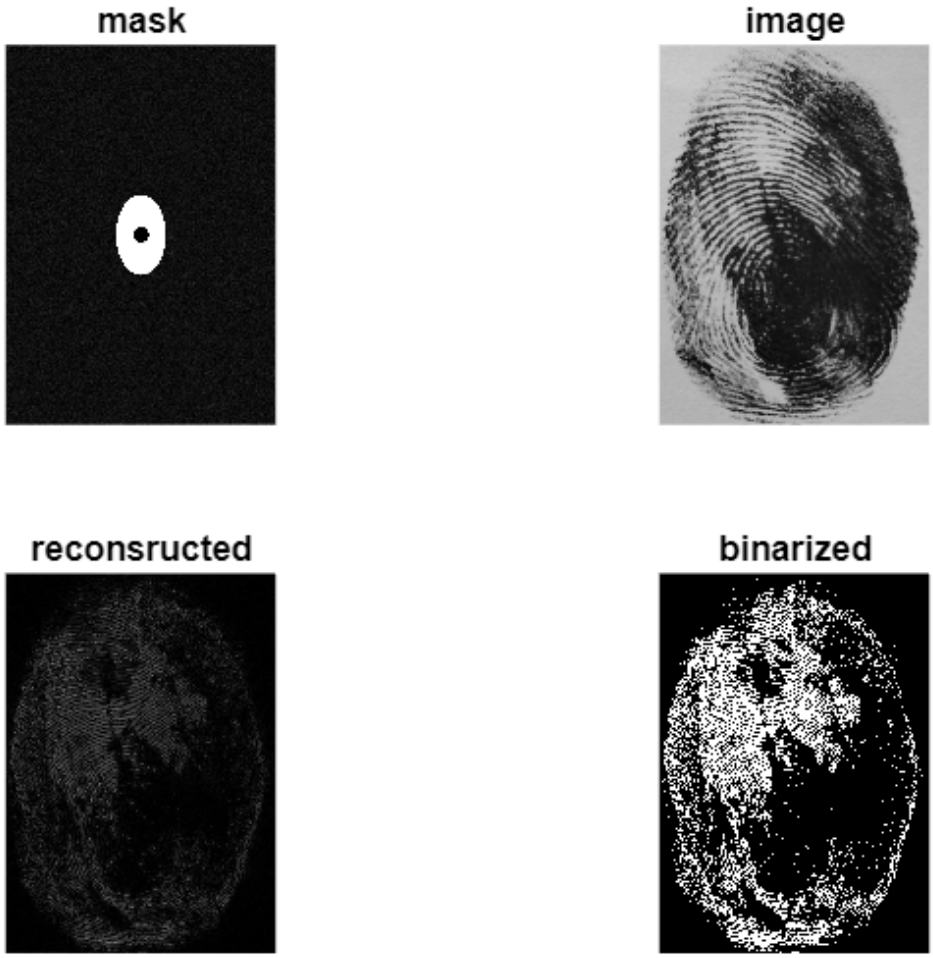
reconsructed

binarized

```
clear;close;
Convolution('maskmom.png', 'fingerprintmom.jpg');
```



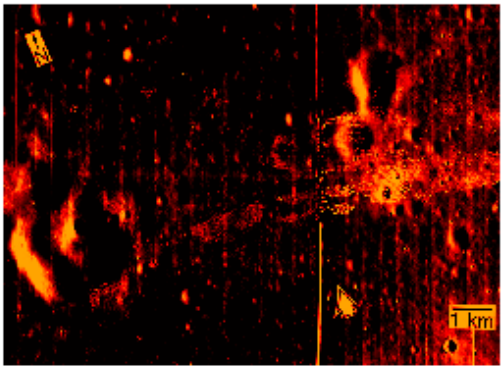

```
clear;close;
Convolution('maskmom2.png', 'fingerprintmom.jpg');
```



Activity 2.2.5. Lunar Landing Scanned Pictures: Line Removal

```
clear;close;
FFT(meansubtract('sapo_S05.jpg'));
```

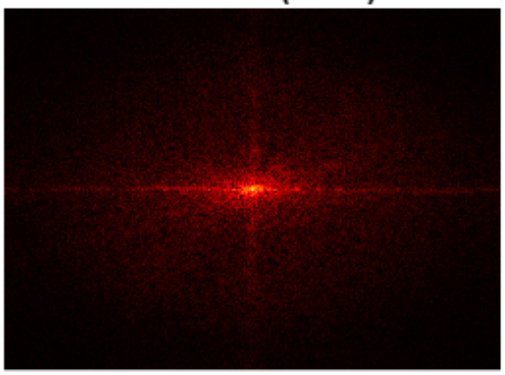
APERTURE



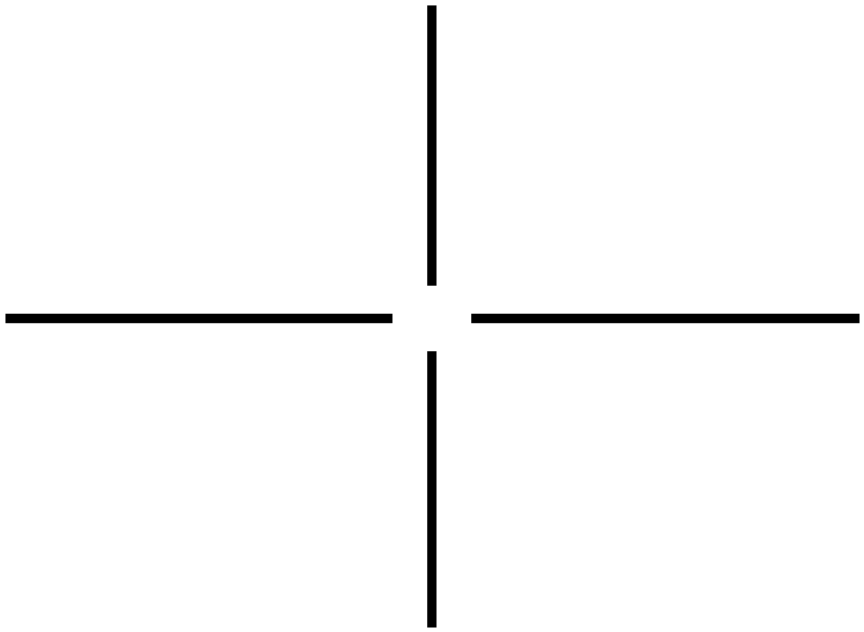
INTENSITY

SHIFTED

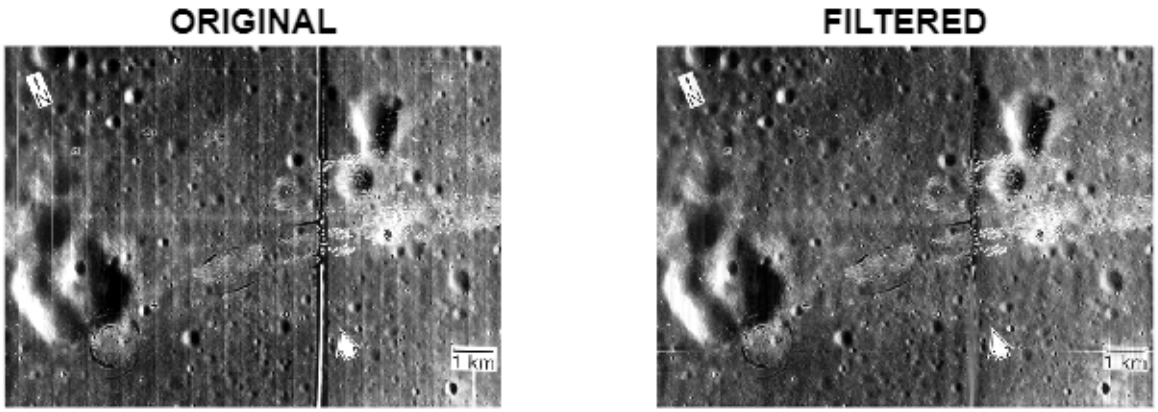
SHIFTED(LOG)



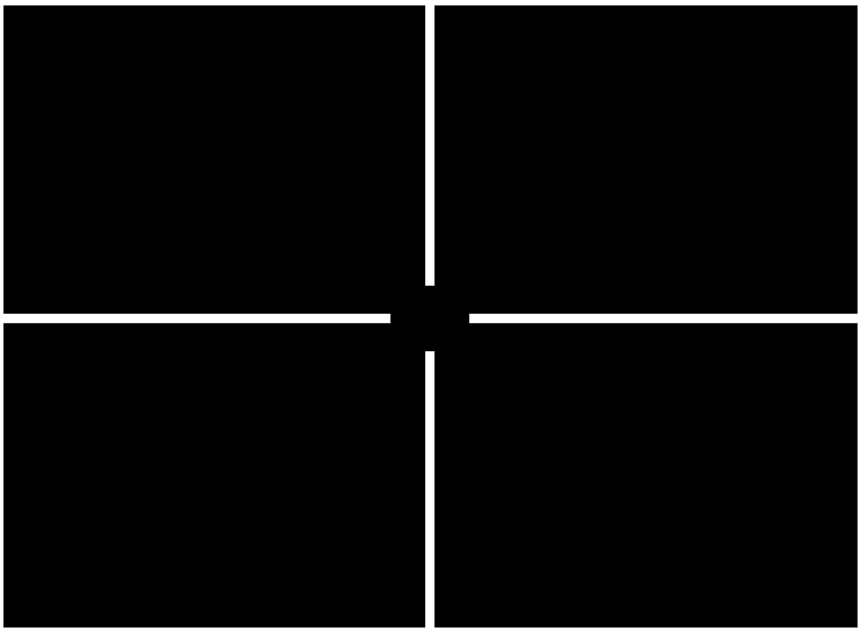
```
%Mask
clear;close;
E = ones(466,640);
E(232:238, 1:290) = 0;
E(232:238, 350:640) = 0;
E(1:210, 317:323) = 0;
E(260:466, 317:323) = 0;
imshow(E);
```



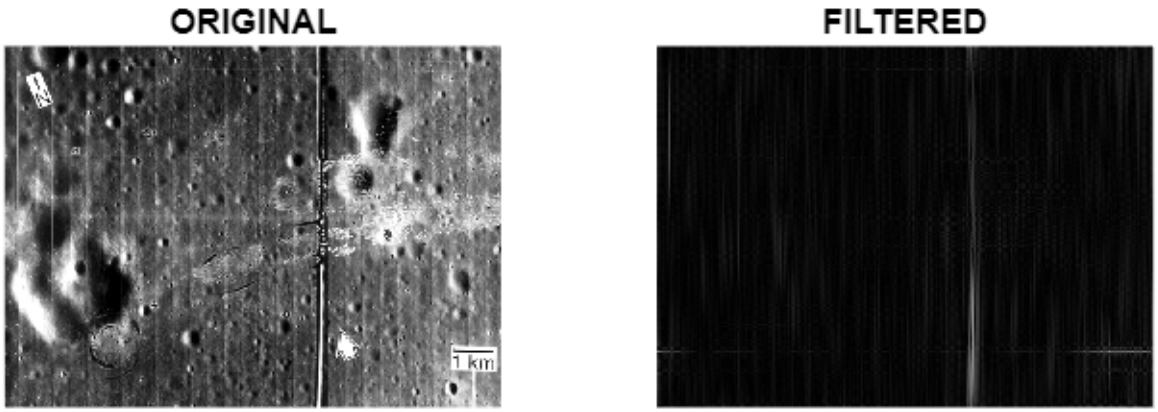
```
filtered('sapo_s05.jpg', E);
```




```
%Mask
clear; close;
E = zeros(466,640);
E(232:238, 1:290) = 1;
E(232:238, 350:640) = 1;
E(1:210, 317:323) = 1;
E(260:466, 317:323) = 1;
imshow(E);
```



```
filtered('sapo_s05.jpg', E);
```



Functions

```
function[a,b, c,d] = FFT(aperture)
    FFT = fft2(aperture);
    FFTmag = abs(FFT);
    FFTshifted = fftshift(FFT);
    FFTmagshift = rescale(abs(FFTshifted), 0, 256);
    FFTlog = log(FFTmagshift+1);
    a = subplot(2,2,1); imshow(aperture); title("APERTURE");
    b = subplot(2,2,2); imagesc(FFTmag); colormap hot; title("INTENSITY"); axis off; axis image
    c = subplot(2,2,3); imagesc(FFTmagshift); colormap hot; title("SHIFTED"); axis off; axis im
    d = subplot(2,2,4); imagesc(FFTlog); colormap hot; title("SHIFTED(LOG)"); axis off; axis im
end

function A = Sinusoid(f, d)
    N = 256;
    x = linspace(0,pi,N);
    y = x;
    [X,Y] = meshgrid(x,y);
    A = sin(2*f*(sin(d)*Y + cos(d)*X));
end
```

```
function I = meansubtract(image)
    I = im2gray(imread(image));
    meangray = mean2(I);
    I = I - meangray;
end
function [a,b] = filtered(unfiltered, filter)
    orig = im2double(imread(unfiltered));
    R = orig(:,:,1);
    G = orig(:,:,2);
    B = orig(:,:,3);
    %fftshift the filter A
    shiftfilter = fftshift(filter);
    %complex FT of RGB channels
    fft2R = fft2(R);
    fft2G = fft2(G);
    fft2B = fft2(B);
    %filtered FT of RGB channels
    filteredR = double(shiftfilter.*fft2R);
    filteredG = double(shiftfilter.*fft2G);
    filteredB = double(shiftfilter.*fft2B);
    %inverse
    invR = double(abs(ifft2(filteredR)));
    invG = double(abs(ifft2(filteredG)));
    invB = double(abs(ifft2(filteredB)));
    %Igray = rgb2gray(I);
    %Inew2 = abs(ifft2(shiftfilter.*fft2(I)));
    Inew(:,:,1) = invR;
    Inew(:,:,2) = invG;
    Inew(:,:,3) = invB;
    a = subplot(1,2,1); imshow(orig); axis image; axis off; title('ORIGINAL');
    b = subplot(1,2,2);imagesc(Inew); axis image; axis off; title('FILTERED');
end
function A = Slit(a,b,c,d,e,f,g,h)
    N = 200;
    A = zeros(N,N);
    A(a:b, c:d) = 255;
    A(e:f, g:h) = 225;
end
function A = equalone(p1,p2)
    N = 200;
    A = zeros(N,N);
    A(100:101,p1) = 1;
    A(100:101,p2) = 1;
    A(p1,100:101) = 1;
    A(p2,100:101) = 1;
end
function A = random(n)
    N = 256;
    num1 = n;
    A = zeros(N,N);  % set all to zero
```

```
    ix = randperm(numel(A)); % randomize the linear indices
    ix = ix(1:num1); % select the first
    A(ix) = 1; % set the corresponding positions to 1
end
function [a,b,c,d] = Convolution(mask, fingerprint)
    A = im2double(rgb2gray(imread(mask)));
    Ashift = fftshift(A);
    N = im2double(rgb2gray(imread(fingerprint)));
    Nfft = fft2(N);
    H = Ashift.*Nfft;
    h = ifft2(H);
    habs = abs(h);
    bw = imbinarize(habs);
    a = subplot(2,2,1); imshow(A); axis image; axis off; title('mask');
    b = subplot(2,2,2); imshow(N); axis image; axis off; title('image');
    c = subplot(2,2,3);imshow(habs); axis image; axis off; title('reconsructed');
    d = subplot(2,2,4); imshow(bw); axis image; axis off; title('binarized');
end
```