Vibhav Jha

BE 492 Section B5a

March 3, 2019

Lab 3: Microscopy I

**Part B: Image Blurring**

B1: In addition to the pixel size, the specific magnification of the objective as well as the 0.5x demagnification is needed to calculate FOV.



**Figure B1**: Grayscale Image 4x



**Figure B2:** 10x Grayscale



**Figure B3:** 40x Grayscale

B2: The resolution at 4x is 2.5 microns, at 10x it is 1 micron and at 40x it is 0.385 microns.

B3: The maximum light ray tilt angle at 4x magnification (or 2.5 microns) is 5.739 degrees. For 10x, it is 14.478 degrees. For 40x or 0.385 microns the maximum light ray tilt is 40.452 degrees.



**Figure B4:** 2D Fourier Transform, zoomed in and slightly centered of figure B1.

B4: The sum of pixels is: 147771717. The value of the center pixel is the same.



**Figure B5:** Filtered 4x, with cutoff frequency 45 Hz using circular disk filter.



**Figure B6:** Normalized Line profile of figure B5.



**Figure B7:** Filtered 4x, cutoff frequency 120 Hz using a circular disk filter.



**Figure B8:** Normalized line profile of B7.



**Figure B9:** Gaussian filtered B1 with cutoff frequency 45 Hz.



**Figure B10:** Normalized line profile of B9, zoomed in to differentiate lines easier.



**Figure B11:** Gaussian filtered B1 with cutoff frequency 120 Hz.



**Figure B12:** Normalized line profile of B12.

\***Note:** Figures B9-B12 do not look right. After applying the Gaussian Filter, in the plot profiles there should be a Gaussian visible. Rechecking the code below, I could not figure out why I was not seeing a Gaussian.

%BE492 Lab3 Part B

close all;

pb4xp = imread('01\_B1.bmp');

pb10xp = imread('02\_B2.bmp');

pb40xp = imread('03\_B2.bmp');

%Grayscale Conversion using NTSC standards

pb4xpgray = 0.2989\*pb4xp(:,:,1) + 0.5870\*pb4xp(:,:,2) + 0.1140\*pb4xp(:,:,3);

pb10xpgray = 0.2989\*pb10xp(:,:,1) + 0.5870\*pb10xp(:,:,2) + 0.1140\*pb10xp(:,:,3);

pb40xpgray = 0.2989\*pb40xp(:,:,1) + 0.5870\*pb40xp(:,:,2) + 0.1140\*pb40xp(:,:,3);

%Camera Size

pb4xp\_size = size(pb4xpgray);

pb10xp\_size = size(pb10xpgray);

pb40xp\_size = size(pb40xpgray);

%Pixel Size Scale

%num of pixels \* pixel size / magnification effective

pb4xpmicrons\_x = [0 ((1280 \* 3.6)/(0.5\*4))];

pb4xpmicrons\_y = [0 ((1024 \* 3.6)/(0.5\*4))];

pb10xpmicrons\_x = [0 ((1280 \* 3.6)/(0.5\*10))];

pb10xpmicrons\_y = [0 ((1024 \* 3.6)/(0.5\*10))];

pb40xpmicrons\_x = [0 ((1280 \* 3.6)/(0.5\*40))];

pb40xpmicrons\_y = [0 ((1024 \* 3.6)/(0.5\*40))];

xsize = 1280;

ysize = 1024;

FigureB1 = figure('Name', '4x Image NTSC Standard');

colormap(gray(256));

imagesc(pb4xpmicrons\_x,pb4xpmicrons\_y, pb4xpgray);

title('NTSC Standard Grayscaled 4x Magnification');

xlabel('Width of sample (microns)');

ylabel('Length of sample (microns)');

FigureB2 = figure('Name', '10x Image NTSC Standard');

colormap(gray(256));

imagesc(pb10xpmicrons\_x,pb10xpmicrons\_y, pb10xpgray);

title('NTSC Standard Grayscaled 10x Magnification');

xlabel('Width of sample (microns)');

ylabel('Length of sample (microns)');

FigureB3 = figure('Name', '40x Image NTSC Standard');

colormap(gray(256));

imagesc(pb40xpmicrons\_x,pb40xpmicrons\_y, pb40xpgray);

title('NTSC Standard Grayscaled 40x Magnification');

xlabel('Width of sample (microns)');

ylabel('Length of sample (microns)');

%Calculating Resolutions using lambda/2NA

pb4xp\_res = (0.5E-6)/(2 \* 0.1);

pb10xp\_res = (0.5E-6)/(2 \* 0.25);

pb40xp\_res = (0.5E-6)/(2 \* 0.65);

%Calculating Max light ray angle using NA=nsin(angle) n=1

pb4xp\_angle = asind(0.1);

pb10xp\_angle = asind(0.25);

pb40xp\_angle = asind(0.65);

FigureB4 = figure('Name' , 'ftpic abs');

ftpic = fftshift(fft2(pb4xpgray));

pb4xp\_fftabs = abs(ftpic);

colormap(gray);

imagesc(pb4xp\_fftabs);

zoom(FigureB4, 150);

title('150x Zoom of Absolute Value of FFT of 4x Image');

xlabel('K\_x');

ylabel('K\_y')

%Central Pixel

ftpic\_size = size(ftpic);

csize = ftpic\_size(2)/2+1;

rsize = ftpic\_size(1)/2+1;

pb4xp\_sum = sum(sum(pb4xpgray));

central\_pixel = ftpic(rsize,csize);

isequal(pb4xp\_sum, central\_pixel);

%45 Hz frequency filter

cutoff\_freq4x = 45;

[x, y] = meshgrid(1:xsize, 1:ysize);

rr\_pb4x = (x-xsize/2-1).^2+(y-ysize/2-1).^2 <= cutoff\_freq4x^2;

filtered = ftpic.\*rr\_pb4x;

new\_pb4xp = ifft2(ifftshift(filtered));

new\_pb4xp\_real = real(new\_pb4xp);

FigureB5 = figure('Name', 'Circular Disk Filtered 45Hz');

colormap(gray(256));

imagesc(pb4xpmicrons\_x, pb4xpmicrons\_y, new\_pb4xp\_real);

title('Circular Disk Filtered Picture with Cutoff Frequency 45Hz');

xlabel('Microns');

ylabel('Microns');

%LineProfile

ftpic\_ln = ftpic(ysize/2+1,:);

ftpic\_ln\_prof = abs(ftpic\_ln)/max(abs(ftpic\_ln));

rr\_prof = rr\_pb4x(ysize/2+1,:);

filt\_cent = abs(filtered(ysize/2+1,:));

filt\_cent\_prof = abs(filt\_cent)/max(abs(filt\_cent));

FigureB6 = figure('Name', 'Line Profile B5');

plot(ftpic\_ln\_prof, 'LineWidth', 5);

hold on

plot(rr\_prof, 'LineWidth', 3);

hold on

plot(filt\_cent\_prof, 'LineWidth', 4);

title('Normalized Line Profile with Cutoff Frequency 45Hz');

legend({'Filtered','Transform','Filtered Transform'});

xlabel('Pixel Number');

ylabel('Pixel Number');

%Frequency Filter 120Hz

cutoff\_freq120x = 120;

[x, y] = meshgrid(1:xsize, 1:ysize);

rr\_pb120x = (x-xsize/2-1).^2+(y-ysize/2-1).^2 <= cutoff\_freq120x^2;

filtered120 = ftpic.\*rr\_pb120x;

new\_pb120xp = ifft2(ifftshift(filtered120));

new\_pb120xp\_real = real(new\_pb120xp);

FigureB7 = figure('Name', 'Filtered Picture with Cutoff Frequency 120Hz');

colormap(gray(256));

imagesc(pb4xpmicrons\_x, pb4xpmicrons\_y, new\_pb120xp\_real);

title('Circular Disk Filtered Picture with Cutoff Frequency 120Hz');

xlabel('Microns');

ylabel('Microns');

%Line Profile

ftpic\_ln120 = ftpic(ysize/2+1,:);

ftpic\_ln\_prof120 = abs(ftpic\_ln120)/max(abs(ftpic\_ln120));

rr\_prof120 = rr\_pb120x(ysize/2+1,:);

filt\_cent120 = abs(filtered120(ysize/2+1,:));

filt\_cent\_prof120 = abs(filt\_cent120)/max(abs(filt\_cent120));

FigureB8 = figure('Name', 'Normalized Line Profile 120Hz');

plot(ftpic\_ln\_prof120, 'LineWidth', 5);

hold on

plot(rr\_prof120, 'LineWidth', 3);

hold on

plot(filt\_cent\_prof120, 'LineWidth', 4);

title('Normalized Line Profile with Cutoff Frequency 120Hz');

legend({'Filtered','Transform','Filtered Transform'});

xlabel('Pixel Number');

ylabel('Pixel Number');

%Gaussian Filter 45Hz

cutoff\_freq4x = 45;

[x, y] = meshgrid(1:xsize, 1:ysize);

gg45 = exp(-((x-xsize/2-1).^2 + (y-ysize/2-1).^2/cutoff\_freq4x^2));

filtered\_45x = ftpic.\* gg45;

new\_pb45x\_g = ifft2(ifftshift(filtered\_45x));

new\_pb45x\_gr = real(new\_pb45x\_g);

FigureB9 = figure('Name', 'Gaussian Filtered 45Hz');

colormap(gray(256));

imagesc(pb4xpmicrons\_x, pb4xpmicrons\_y, new\_pb45x\_gr);

title('Gaussian Filtered Picture with Cutoff Frequency 45Hz');

xlabel('Microns');

ylabel('Microns');

%Lineprofile

ftpic\_ln45g = ftpic(ysize/2+1,:);

ftpic\_ln\_prof45g = abs(ftpic\_ln45g)/max(abs(ftpic\_ln45g));

gg\_prof45 = gg45(ysize/2+1,:);

filt\_cent45g = abs(filtered\_45x(ysize/2+1,:));

filt\_cent\_prof45g = abs(filt\_cent45g)/max(abs(filt\_cent45g));

FigureB10 = figure('Name', 'Normalized Gaussain Plot Profile 45Hz');

plot(ftpic\_ln\_prof45g, 'LineWidth', 5);

hold on

plot(gg\_prof45, 'LineWidth', 3);

hold on

plot(filt\_cent\_prof45g, 'LineWidth', 4);

title('Normalized Gaussian Line Profile with Cutoff Frequency 45Hz');

legend({'Filtered','Transform','Filtered Transform'});

xlabel('Pixel Number');

ylabel('Pixel Number');

cutoff\_freq120x = 120;

[x, y] = meshgrid(1:xsize, 1:ysize);

gg120 = exp(-((x-xsize/2-1).^2 + (y-ysize/2-1).^2/cutoff\_freq120x^2));

filtered\_120x = ftpic.\* gg120;

new\_pb120x\_g = ifft2(ifftshift(filtered\_120x));

new\_pb120x\_gr = real(new\_pb120x\_g);

FigureB11 = figure('Name', 'Gaussian Filtered 120Hz');

colormap(gray(256));

imagesc(pb4xpmicrons\_x, pb4xpmicrons\_y, new\_pb120x\_gr);

title('Gaussian Filtered Picture with Cutoff Frequency 120Hz');

xlabel('Microns');

ylabel('Microns');

%LineProfile

ftpic\_ln120g = ftpic(ysize/2+1,:);

ftpic\_ln\_prof120g = abs(ftpic\_ln120g)/max(abs(ftpic\_ln120g));

gg\_prof120 = gg120(ysize/2+1,:);

filt\_cent120g = abs(filtered\_120x(ysize/2+1,:));

filt\_cent\_prof120g = abs(filt\_cent120g)/max(abs(filt\_cent120g));

FigureB12 = figure('Name', 'Normalized Gaussain Plot Profile 120Hz');

plot(ftpic\_ln\_prof120g);

hold on

plot(gg\_prof120);

hold on

plot(filt\_cent\_prof120g);

title('Normalized Gaussian Line Profile with Cutoff Frequency 120Hz');

legend({'Filtered','Transform','Filtered Transform'});

xlabel('Pixel Number');

ylabel('Pixel Number');

**Part C: Image Deblurring**



**Figure C1:** Focused 4x image of USAF 1951.



**Figure C2:** Out of focus target.

C1: Calculated in-focus resolution is 228 line-pair/mm in group 7 element 6. The NA required for this is 0.1140, which is similar to the actual NA of 0.1 provided by the 4x objective. The smallest set of lines for the unshifted out of focus image is group 4 element 6 with a resolution 28.5 line-pair/mm. The required NA to produce this is 0.014, which is vastly different to the actual NA of the objective.



**Figure C3:** Filter function determined from the Fourier transforms of in-focus and out-of-focus images.



**Figure C4:** Shifted out of focus image.



**Figure C5:** Refocused image with a fudge factor of 0.1.

%Part C

close all;

USAF1951\_focused = imread('04\_C5.bmp');

USAF1951\_defocused = imread('05\_C6.bmp');

USAF1951\_shifted = imread('06\_C7.bmp');

%Grayscale Conversion using NTSC standards

Focusedgray = 0.2989\*USAF1951\_focused(:,:,1) + 0.5870\*USAF1951\_focused(:,:,2) + 0.1140\*USAF1951\_focused(:,:,3);

Defocusedgray = 0.2989\*USAF1951\_defocused(:,:,1) + 0.5870\*USAF1951\_defocused(:,:,2) + 0.1140\*USAF1951\_defocused(:,:,3);

Shiftedgray = 0.2989\*USAF1951\_shifted(:,:,1) + 0.5870\*USAF1951\_shifted(:,:,2) + 0.1140\*USAF1951\_shifted(:,:,3);

FG\_size = size(Focusedgray);

DFG\_size = size(Defocusedgray);

SG\_size = size(Shiftedgray);

microns\_x = [0 ((1280 \* 3.6)/(0.5\*4))];

microns\_y = [0 ((1024 \* 3.6)/(0.5\*4))];

FigureC1 = figure('name', 'Focused');

colormap(gray(256));

imagesc(microns\_x,microns\_y, Focusedgray);

title('NTSC Standard Grayscaled 4x Magnification in Focus');

xlabel('Microns');

ylabel('Microns');

FigureC2 = figure('name', 'Defocused');

colormap(gray(256));

imagesc(microns\_x,microns\_y, Defocusedgray);

title('NTSC Standard Grayscaled 4x Magnification Defocused');

xlabel('Microns');

ylabel('Microns');

%in-focus: Group 7 element 6

%out: Group 4 Element 6

infocus\_real = 2^(7+((6-1)/6));

outfocus\_real = 2^(4+((6-1)/6));

%NA

infoc\_m = 1000/(2\*infocus\_real);

outfoc\_m = 1000/(2\*outfocus\_real);

NA\_infoc = 0.5/(2\*infoc\_m);

NA\_outfoc = 0.5/(2\*outfoc\_m);

%filter function

ftpic\_in = fftshift(fft2(Focusedgray));

ftpic\_out = fftshift(fft2(Defocusedgray));

filterfxn = ftpic\_out./ftpic\_in;

filterfxn\_abs = abs(filterfxn);

filterfxn\_abs(filterfxn\_abs>1) = 1;

FigureC3 = figure('Name', 'Filter Function');

colormap(gray(256));

imagesc(microns\_x,microns\_y, filterfxn\_abs);

title('Filter Function');

xlabel('Microns');

ylabel('Microns');

%Deblur

ftpic\_shift = fftshift(fft2(Shiftedgray));

fudgefactor = 0.1;

f = filterfxn\_abs./((abs(filterfxn).^2) + fudgefactor);

ftpic\_fr = ftpic\_shift.\*f;

iftpic = ifft2(ifftshift(ftpic\_fr));

iftpic\_real = real(iftpic);

iftpic\_rabs = abs(iftpic\_real);

FigureC4 = figure('Name', 'Defocused Shifted');

colormap(gray(256));

imagesc(microns\_x,microns\_y, Shiftedgray);

title('Defocused Shifted Image');

xlabel('Microns');

ylabel('Microns');

FigureC5 = figure('Name', 'Deblurred Defocused Shifted');

colormap(gray(256));

imagesc(microns\_x,microns\_y, iftpic\_rabs);

title('Deblurred Defocused Shifted Image');

xlabel('Microns');

ylabel('Microns');