DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING, UCLA
ECE 239AS: COMPUTATIONAL IMAGING

INSTRUCTOR: Prof. Achuta Kadambi
TA: Guangyuan Zhao

NAME:
UID:

HOMEWORK 2: SYNTHETIC APERTURE IMAGING

PROBLEM	TOPIC	MAX. POINTS	GRADED POINTS	REMARKS
0.1	Set up a Static Scene	0.5		
0.2	Capture a 4D Light Field	0.5		
0.3	Acquiring the Data	0.5		
0.4.1	Template and Window	0.5		
0.4.2	Normalized Cross Correlation	0.5		
0.4.3	Retrieving the Pixel Shifts	0.5		
0.5	Synthesizing an Image with Synthetic Aperture	1.0		
0.6	Repeating the Experiment for Different Templates	1.0		
1.1	Deriving the Blur Kernel Width	3.0		
1.2	Blur Kernel Shape	1.0		
1.3	Blur and Scene Depth	0.5		
1.4	Blur and Focal Length	0.5		
	Total	10		

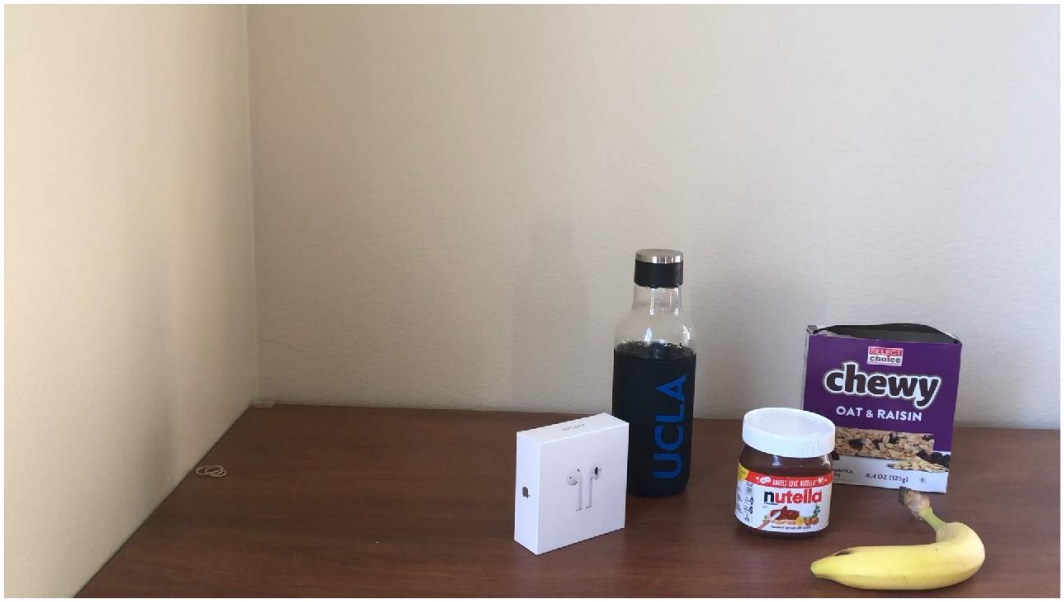
0.1 Set up a Static Scene (0.5 points)

Set up a static scene similar to the one shown in Figure **??**a. Try to have objects at different depths. For credit, place your image in the box below (replace our helmet scene with your own scene).

Figure 1: Insert an ordinary photograph of the scene (replace our example).

0.2 Capture a 4D Light Field (0.5 points)

Take a video by waving your camera in front of the scene by following a specific planar motion. The more you cover the plane, the better will be your results. Ensure that all objects are in focus in your video. For credit, place three frames of the video in the box below (replace our example). These frames differ in their *parallax*, i.e., an effect where object positions change in response to view.


UCLA
chewy
OAT & RAISIN
nutella


UCLA
chewy
OAT & RAISIN
nutella


UCLA
chewy
OAT & RAISIN
nutella

0.3 Acquiring the Data (0.5 points)

Write a function to read your video file and convert the video into a sequence of frames. Since this was captured from a cell phone, each frame image is in RGB color. Write a script to convert each frame to gray-scale. For credit, place the gray scale image of the first frame of your video in the box below (replace our example).

Figure 3: Insert the gray-scale image of the first frame (replace our example).

0.4 Registering the Frames (1.5 points)

0.4.1 Template and Window (0.5 points)

From the first frame of your video, select an object as a template. We will be registering all other frames of the video with respect to this template. Once a template has been selected in the first frame, we search for it in the subsequent frames. The location of the template in a target frame image will give us the shift(in pixels) of the camera. Since we don't have to search for the template in the entire target frame image, we select a window to perform this operation. Note, however, that selecting a window is optional. This is done just to reduce the computation time. For credit, place the image of the first frame of your video in the box below with the template and the window markings (replace our example).

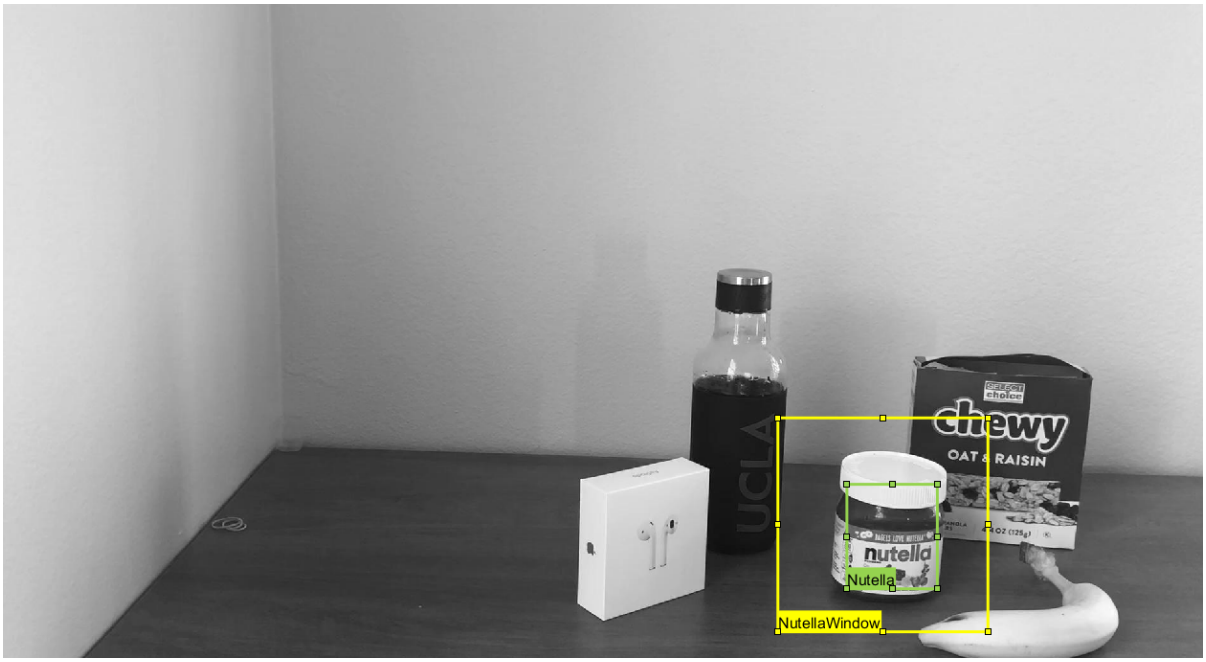


Figure 4: Insert your image with template object and search window marked (replace our example).

0.4.2 Normalized Cross Correlation (0.5 points)

Perform a normalized cross correlation of the template with the extracted search window.

Let $A[i,j]$ be the normalized cross-correlation coefficient. If $t[n,m]$ is our template image and $w[n,m]$ is our window, then from [**?**] we have:

$$A[i,j] = \frac{\sum_{n,m=1}^{T}[w(n,m)-\overline{w}_{i,j}][t(n-i,m-j)-\overline{t}]}{\{\sum_{n,m=1}^{T}[w(n,m)-\overline{w}_{i,j}]^2[t(n-i,m-j)-\overline{t}]^2\}^{0.5}}, \tag{1}$$

where, $\overline{t}$ is the mean of the template and $\overline{w}_{i,j}$ is the mean of the window $w[n,m]$ in the region under the template. Plot the cross correlation coefficient matrix $A[i,j]$ for one of the frames. For credit, place the plot in the box below. (replace our example)

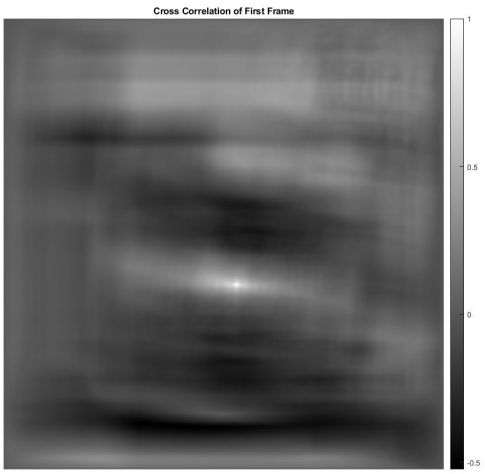


Figure 5: Insert the plot of the correlation coefficient Matrix (replace our example).

[Hint: Use the MATLAB function *normx*2*corr* or *xcorr*2 to perform the 2D cross correlation function.]

0.4.3 Retrieving the Pixel Shifts (0.5 points)

The location that yields the maximum value of the coefficient$A[i, j]$ is used to compute the shift [**?**]. The shift in pixels for each frame can be found by:

$$[s_x, s_y] = max_{i,j}\{A[i, j]\}. \tag{2}$$

For credit, please place the plot of s_x v/s s_y in the box below (replace our example).

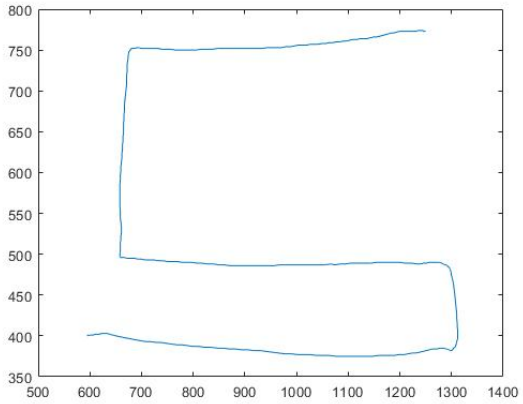


Figure 6: Insert the plot of X pixel shift v/s Y pixel shift (replace our example).

0.5 Synthesizing an Image with Synthetic Aperture (1.0 points)

Once you have the pixel shifts for each frame, you can synthesize refocused image by shifting each frame in the opposite direction and then summing up all the frames. (Note: in Section 1, you will need to explain why this operation works. Start thinking about this now!)

Suppose the pixel shift vector for Frame Image $I_i[n,m]$ is $[s_{x_i}, s_{y_i}]$. Then, the image output, $P[n,m]$ with synthetic aperture is obtained as:

$$P[n,m] = \sum_i I_i[n - s_{x_i}, m - s_{y_i}]. \tag{3}$$

For credit, place your synthetically "defocused" image in the box below (replace our example).

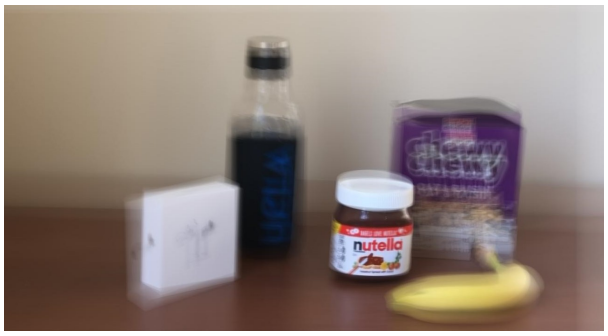


Figure 7: Insert an image with an object in synthetic focus (replace our example).

[Hint: Use the MATLAB function *imtranslate* to perform the shifting operation.]

0.6 Repeating the Experiment for Different Templates (1.0 points)

Now, we will exploit the fact that we can synthetically focus on different depths. To do this, select a new object as your template and repeat all the steps to generate an image that is focused on this new object. Here, we have selected the cup as our new object. For credit, place a de-focused image with a different template object in focus in the box below (replace our example).

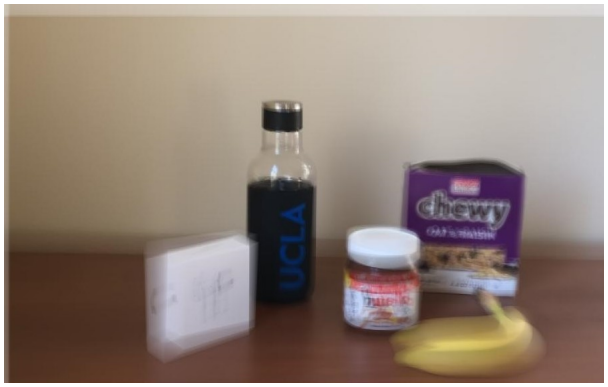


Figure 8: Insert an image with an object in synthetic focus. This object should be different from the previous box (replace our example).

1 Assessment

1.1 Deriving the Blur Kernel Width (3.0 points)

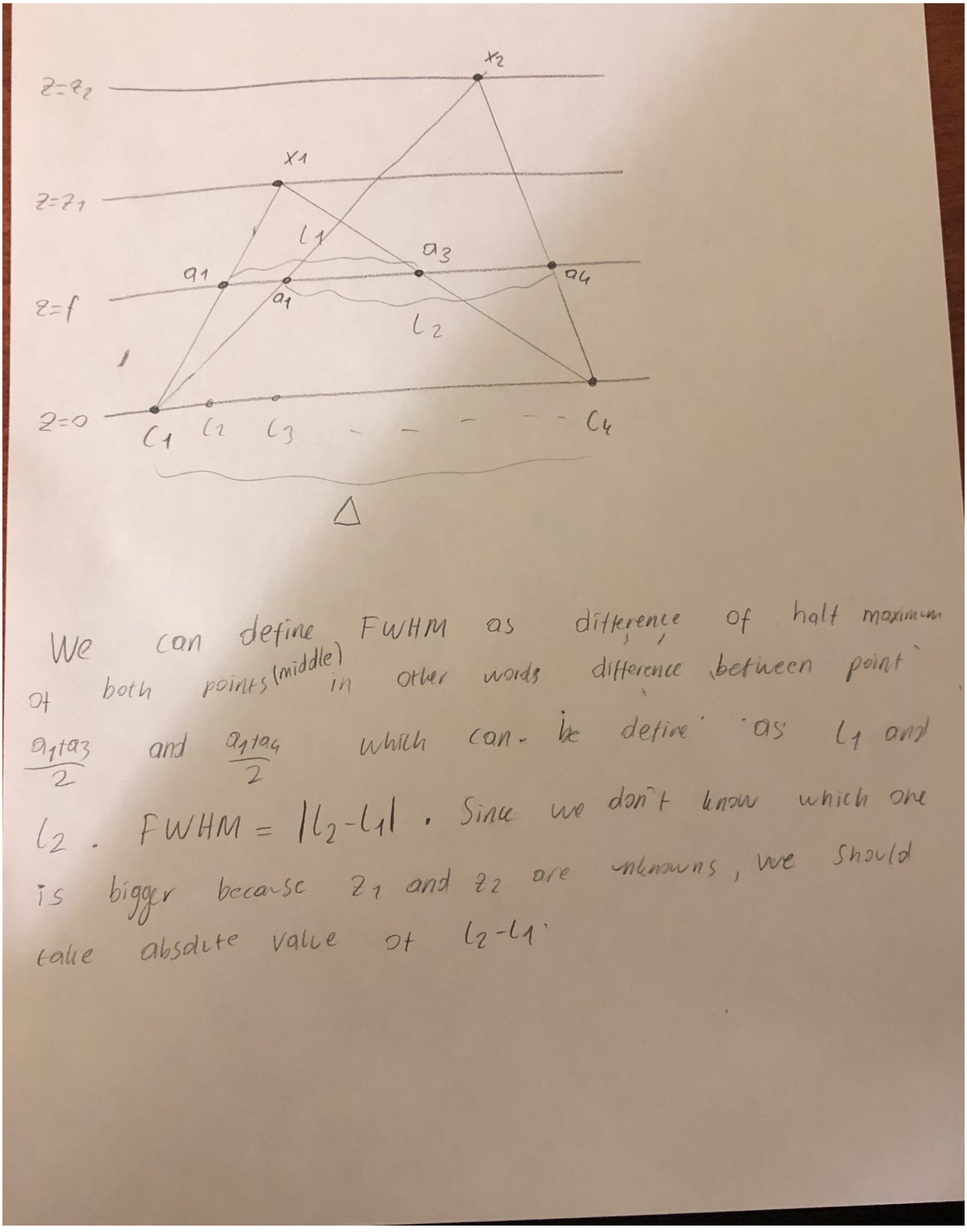
The goal is to understand how much blur is synthetically added by using a model of pinhole cameras. Consider the coordinate diagram shown in Figure 9. Here, $[X_1,Z_1]$ is a scene point of an object in the template, $[X_2,Z_2]$ is a scene point of an object in the background and $C^{(i)}$ for $i = 1,\ldots,k$ are positions of the apertures of cameras at which the scene is captured. The maximum camera translation is Δ and f is the focal length of the cameras (all are assumed to be the same).

Figure 9: Example coordinate system and notation. In this figure, the dashed plane is the virtual film plane, placed one focal length *above* the apertures located at $C^{(1)},\ldots,C^{(k)}$. This is a common shorthand convention so we do not have to flip the camera images. In reality, the actual film plane would be one focal length below the aperture location. This coordinate system is used as a guide - you are welcome to modify as needed.

We will use the shift-and-add method for light field imaging such that X_1 is the point in focus (i.e. as the "template" that we shift and add"). Derive a mathematical expression for the full-width half maximum (FWHM) of the blur kernel (W) applied to X_2. Credit will be assessed both for technical correctness and the presentation of the derivation. You should not need figures, but are welcome to include them. Insert your derivation in the box below.

[Hint: Our solution to derive W was about a half page.]

[Hint: To check your solution, if $Z_1 = Z_2$ the width of the blur kernel should be zero.]



If we draw the above figure we can easily derive FWHM by the help of geometry. FWHM can be defined as follows:

$$FWHM = |l2 - l1|$$

We need to use triangle similarity equations which are as follows:

$$\frac{C_k - C_1}{l2} = \frac{Z_2}{Z_2 - f}$$

$$\frac{C_k - C_1}{l1} = \frac{Z_1}{Z_1 - f}$$

$$l1 = \frac{(Z_1 - f) * (C_k - C_1)}{Z_1}$$

$$l2 = \frac{(Z_2 - f) * (C_k - C_1)}{Z_2}$$

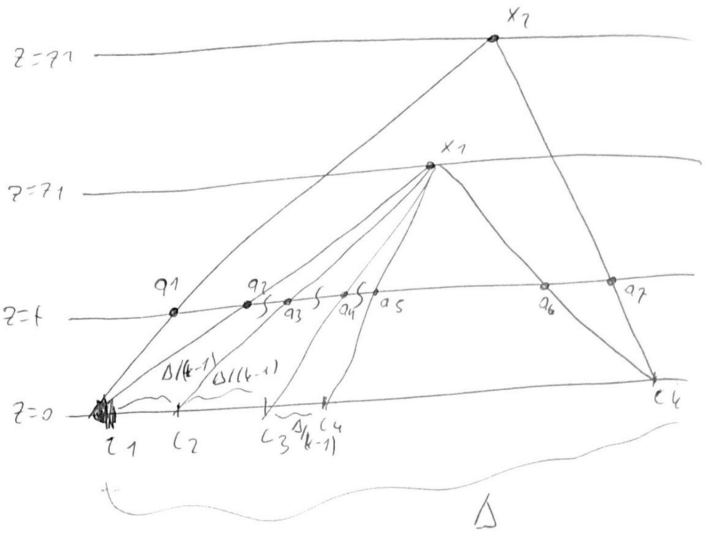
$$C_k - C_1 = \Delta$$

$$FWHM = |(\Delta) * f * (\frac{1}{Z_1} - \frac{1}{Z_2})|$$

1.2 Blur Kernel Shape (1.0 points)

Now that you have derived the FWHM of the blur kernel, please write the functional expression for the blur kernel. For example, is it a Gaussian blur?

Documents-page-001.jpg



$$\frac{a_7 - a_1}{\Delta} = \frac{z_2 - f}{z_2}$$

$$\frac{a_7 - a_2}{\Delta} = \frac{z_1 - f}{z_1} \qquad a_7 - a_2 = \frac{(z_1 - f)\Delta}{z_1}$$

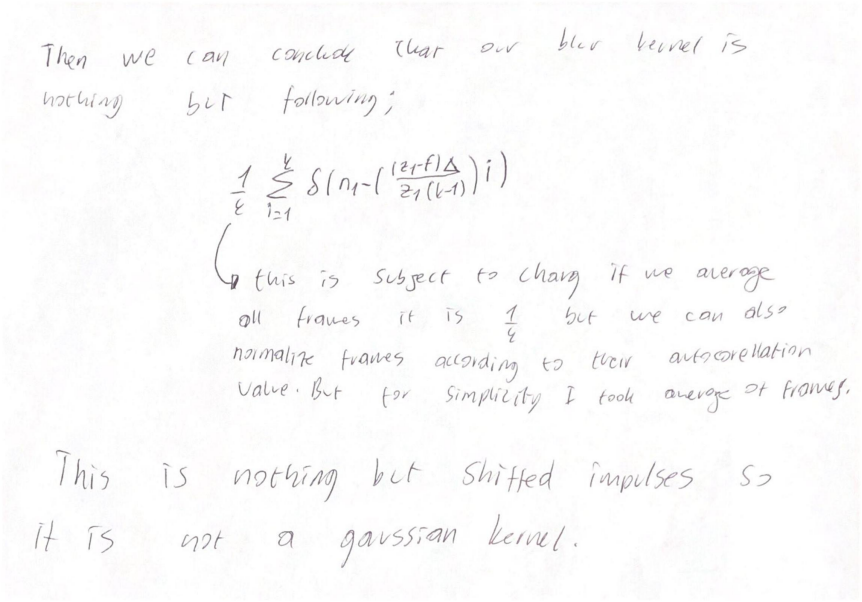
$$a_3 - a_2 = a_4 - a_3 = a_5 - a_4 = \frac{a_7 - a_2}{k-1} = \frac{(z_1 - f)\Delta}{z_1(k-1)}$$

The object x_1 at output will be calculated with same ideology in experiment. In other words I will shift frames taken by camera 2, camera 3 - - - , camera k to point a_2 which means combining every frame by shifting. In other words shifting point $a_3, \dots a_4$ to a_2 to and add them up. Then output image of object 1 will be as follows;

$$y_1(n_1) = x_1(n_1) * \sum_{i=1}^{k-1} \frac{1}{k} \left(\delta\left(n_1 - \left(\frac{(z_1 - f)\Delta}{z_1(k-1)} \right) i \right) \right)$$

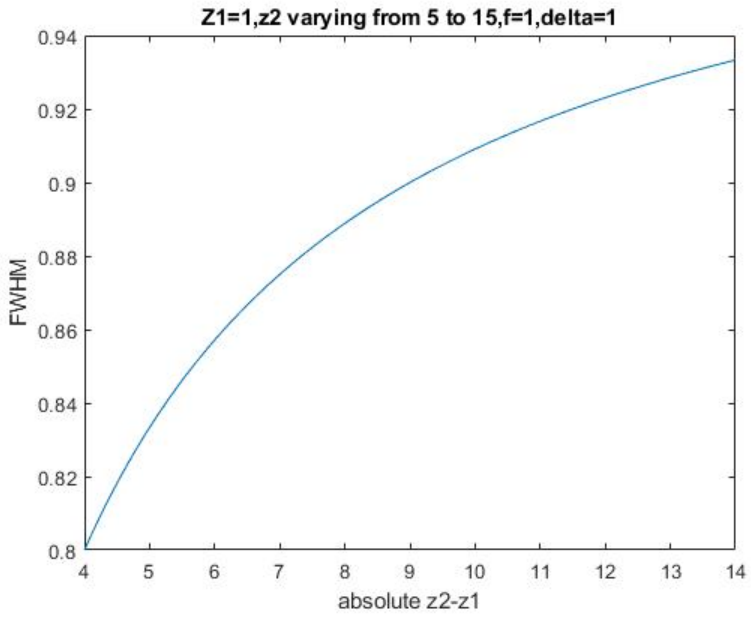
$$y_2(n_1) = x_2(n_1) * \sum_{i=1}^{k-1} \frac{1}{k} \left(\delta\left(n_1 - \left(\frac{(z_1 - f)\Delta}{z_1(k-1)} \right) i \right) \right)$$

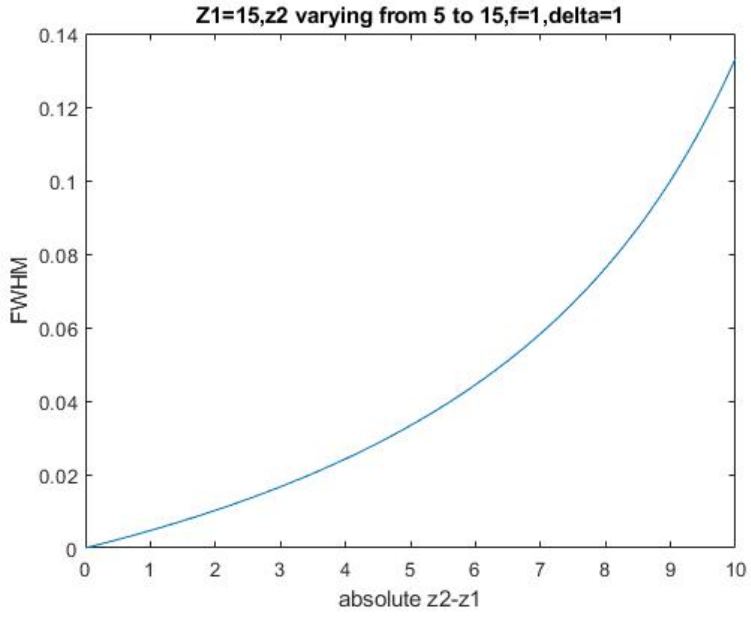
Documents-page-002.jpg



1.3 Blur and Scene Depth (0.5 points)

Plot the width of the blur kernel, W, as a function of the difference in depth planes, $|Z_2 - Z_1|$. Insert your plot in the box below. Comment on the relationship between these variables.

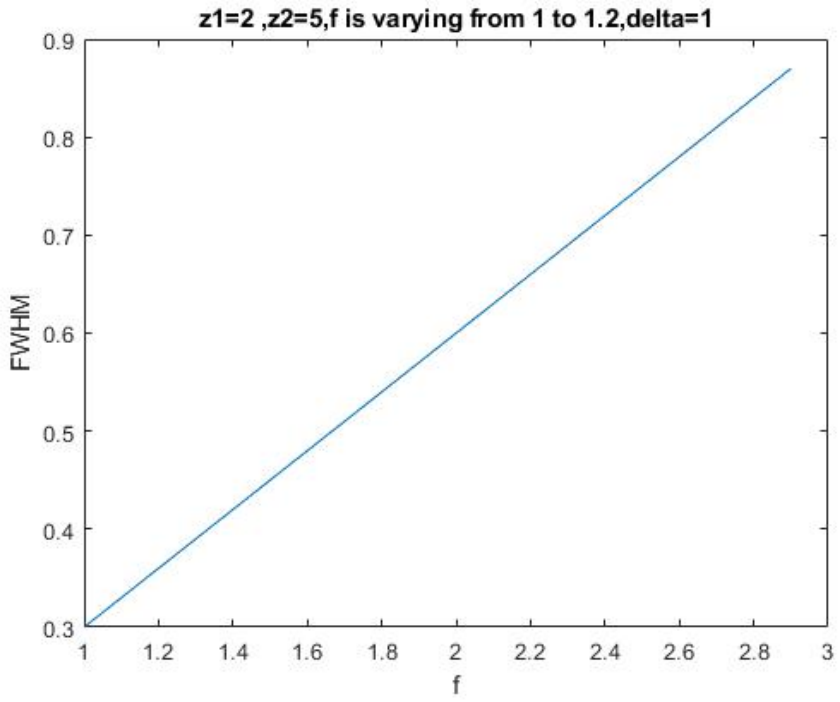




As the distance between the object in focus and other object increases blur gets worse.We already saw this result in our experiment above. So this is easy to understand from the geometry that I drew while finding FWHM. It is easy to understand from that figure that as the object which is not in focus goes further away from the the object in focus the distance between a1 and a4 becomes larger. This means that we are mapping a point to a much larger area this means more bluer according to object which is put much closer to object which is in focus. So to conclude as the distance between focused object and other object ,in the direction z only, increases mapping area for single point to image space becomes larger which causes more bluer.When Z1 is smaller than z2 at certain amount of relative distance it becomes imposiible to differentiate the FWHM.At very large distance it becomes like it.If the z1 and z2 are equal it is nothing but putting them in same focus so there is no blur kernel which means zero FWHM.So it can be said that if the object is behind the focus its bluer increase decreases as it goes further.If the object is in front of the focus object its bluer increase increases as it come closer to the camera.This is nothing but a linear relation

1.4 Blur and Focal Length (0.5 points)

Plot the width of the blur kernel, W, as a function of the focal length of the camera, f. Insert your plot in the box below. Comment on the relationship between these variables.



When we increase the focal length, f, the mapping area of the object x2 on imaging plane increases which causes more bluer.The only difference is the increased area in mapping. So as we increase the focal length mapping area increases linearly. This result can be easily seen by the triangular similarities.We have also seen this mapping area increase in our classes.So when we increase the focal length image gets bluer proportional to that bluer. This is reasonable since as the focal length increases shift distance for frames to match with first frame increases which will cause much bigger difference to required shift of x2 and how much we shifted.This can be easily seen from the triangular similarities.This causes linear increase in bluer since it is exactly related to the triangular similarity which is linear.It can obviously seen from the figure that it passes through origin and the slope changes according to other fixed variables.If we choose f as zero we will not see any bluer and if we choose it very large it will make so much bluer.So to sum up we can say that the mistake we do while we are shifting our object which is not in focus gets much bigger when we increase the focal length.In other words it goes further away where it supposed to be when we add frames together this cause much bluer.In other words center of the object in other frames go very far apart where it supposed to be and this causes bluer.

MATLAB CODE

```
clear all; load('labels2.mat') load('v.mat') load('vvv.mat')

vcolor=framereadercolor('30fpsvideo.MOV'); xofobject=gTruth.LabelData.Nutella1(1); yofobject=gTruth.Label
widthofobject=gTruth.LabelData.Nutella1(3); heightofobject=gTruth.LabelData.Nutella1(4); xofwin-
dow=gTruth.LabelData.NutellaWindow1(1); yofwindow=gTruth.LabelData.NutellaWindow1(2); widthofwin-
dow=gTruth.LabelData.NutellaWindow1(3); heightofwindow=gTruth.LabelData.NutellaWindow1(4);

frame1=v(:,:,1);

object=frame1(yofobject:yofobject+heightofobject-1,xofobject:xofobject+widthofobject-1);

for i=1:350 frame1=v(:,:,i); xp=zeros(1); yp=zeros(1);

crosscor=normxcorr2(object,frame1); if i ==1 figure; foto=crosscor; imshow(crosscor,[]); figure;
surface(crosscor) figure; end [xp,yp]= find(crosscor==max(crosscor(:))); normvector(i)=crosscor(xp,yp);
yoffSet(i) = yp-size(object,1); xoffSet(i) = xp-size(object,2); end

framenumber=100; gg=normvector; normvector=normvector(1:framenumber)./sum(normvector(1:framenumber)
temp=normvector(1)*vcolor(:,:,:,1);

for i=2:framenumber aa=(imtranslate(vcolor(:,:,:,i),[yoffSet(1)-yoffSet(i),xoffSet(1)-xoffSet(i)]));
temp=imadd(double(temp),normvector(i)*double(aa)); end imshow(uint8(temp))

function f = framereader(n) vvv = VideoReader(n); video = read(vvv);

framelength=int64((vvv.Framerate)*(vvv.Duration)); for i=1:framelength videogray(:,:,i)=rgb2gray(video(:,:,:,i));
end f=videogray;

end function g= framereadercolor(n) vvv = VideoReader(n); video = read(vvv);

g=video;

end clear all z1=1; z2(1)=5; xx=1000 for i=2:xx z2(i)=z2(i-1)+0.01; end

for j=1:xx absolute(j)=(z2(j)-z1); y(j)=(abs(z2(j)-z1)/(z1*z2(j))); end

figure

plot(absolute,y) title('Z1=1,z2 varying from 5 to 15,f=1,delta=1') xlabel('absolute z2-z1') yla-
bel('FWHM') z1=15; for j=1:xx absolute(j)=abs(z2(j)-z1); y(j)=(abs(z2(j)-z1)/(z1*z2(j))); end
figure plot(absolute,y) title('Z1=15,z2 varying from 5 to 15,f=1,delta=1') xlabel('absolute z2-z1')
ylabel('FWHM') z1=5; for j=1:xx absolute(j)=abs(z2(j)-z1); y(j)=abs((z2(j)-z1)/(z1*z2(j))); end
figure plot(absolute,y) title('When z1 is both smaller and bigger z2')

a2=5; a1=2; f(1)=1; for j=2:20 f(j)=f(j-1)+0.1; end y=[] for j=1:20

y(j)=f(j)*((abs(a2-a1)/(a1*a2))); end figure plot(f,y) title('z1=2 ,z2=5,f is varying from 1 to 1.2,delta=1')
xlabel('f') ylabel('FWHM')
```