Problem 1.

1. (20%) Use MATLAB to perform convolution of x(t) and h(t), where x(t) is the triangular waveform and h(t) is the impulse response of a system. Write your **own convolution function** instead of using MATLAB function *conv*.

𝑥(𝑡)={0 𝑤ℎ𝑒𝑛 𝑡≤24𝑡−8 𝑤ℎ𝑒𝑛 2<𝑡≤5−4𝑡+42 𝑤ℎ𝑒𝑛 5<𝑡≤70 𝑤ℎ𝑒𝑛 𝑡>7

ℎ(𝑡)=2(𝑒−𝑡−𝑒−5𝑡)

Answer:

Here is code. We broke the problem into two functions and convoluted each by distributing with the exponential function. This is legal since convolution is distributive.

%---------------------------------------------------------------------

% file name : hmwk\_3\_prob\_1\_conv.m

% Student: Ray Duran

% Date: 10/5/21

% Class : EECS 590 Professor Liang, Fall Semester

% University of North Dakota

% Descr:

% Convolution example by time-reversal and slide and summation

% direct- method, not using a function

%---------------------------------------------------------------------

clf

clear

%-----------------------------

%% Convolve first function

%-----------------------------

% First Function

p = 2.1 : .1 : 5;

f1 = 4\*p -8;

figure(1)

subplot(2,2,1)

plot(p,f1)

title(' Original f first function')

% Time reverse our two defined slope function and generate a vector

t1 = 0; % Before start of convolution

p\_tr = -5: .1 : -2.1;

f1\_tr = 4\*(t1 -p\_tr) -8;

%figure(2)

subplot(2,2,2)

plot(p\_tr,f1\_tr) % is our time-reversed vector that we will sweep across

% our exponential function

title(' Time -reversed f first function')

% Generate second vector to convolve with first function

t2 = -5:.1:10;

%

h = 2\*exp(-t2);

g = -2\*exp(-5\*t2);

s = h+g;

% Now Let's convolve the functions s and f\_tr

time\_steps = size(s,2);

length\_of\_function = size(f1\_tr,2);

conv1 = zeros(1,time\_steps);

prod1 = zeros(1,length\_of\_function);

N = time\_steps - length\_of\_function;

for i = 1 : N

i1 = i-1;

for k = 1 : length\_of\_function

prod1(k) = f1\_tr(k)\*s(i1+k);

%debug

if k == length\_of\_function-1

debug = 1;

end

end

conv1(i) = sum(prod1);

%debug

if i == N-5

debug = 1;

end

prod1 = zeros(1,length\_of\_function);

end

%figure(3)

subplot(2,2,3)

plot(t2,conv1)

title('Approx conv of x with h : first function portion')

conv1\_true = conv(f1,s);

length\_conv1\_true = size(conv1\_true,2);

length\_t2 = size(t2,2);

conv1\_true\_plot = zeros(1,length\_t2);

%for m = 1 : length\_conv1\_true

for m = 1 : length\_t2

conv1\_true\_plot(m) = conv1\_true(m);

end

%figure(4)

subplot(2,2,4)

plot(t2,conv1\_true\_plot)

title('True conv of x with h : first function portion')

debug = 1;

%-----------------------------

%% Convolve second function

%-----------------------------

% Second Function

p = 5.1 : .1 : 7;

f2 = -6\*p + 42;

figure(2)

subplot(2,2,1)

plot(p,f2)

title(' Original f second function')

% Time reverse our two defined slope function and generate a vector

t1 = 0; % Before start of convolution

p\_tr = -7: .1 : -5.1;

f2\_tr = -6\*(t1 -p\_tr) + 42;

%figure(6)

subplot(2,2,2)

plot(p\_tr,f2\_tr) % is our time-reversed vector that we will sweep across

% our exponential function

title(' Time -reversed f second function')

% Generate second vector to convolve with first function

t2 = -7:.1:10;

%

h = 2\*exp(-t2);

g = -2\*exp(-5\*t2);

s = h+g;

% Now Let's convolve the functions s and f\_tr

time\_steps = size(s,2);

length\_of\_function = size(f2\_tr,2);

conv2 = zeros(1,time\_steps);

prod2 = zeros(1,length\_of\_function);

N = time\_steps - length\_of\_function;

for i = 1 : N

i1 = i-1;

for k = 1 : length\_of\_function

prod2(k) = f2\_tr(k)\*s(i1+k);

%debug

if k == length\_of\_function-1

debug = 1;

end

end

conv2(i) = sum(prod2);

%debug

if i == N-5

debug = 1;

end

prod2 = zeros(1,length\_of\_function);

end

%figure(7)

subplot(2,2,3)

plot(t2,conv2)

title('Approx conv of x with h : second function portion')

conv2\_true = conv(f2,s);

length\_conv1\_true = size(conv2\_true,2);

length\_t2 = size(t2,2);

conv2\_true\_plot = zeros(1,length\_t2);

%for m = 1 : length\_conv1\_true

for m = 1 : length\_t2

conv2\_true\_plot(m) = conv2\_true(m);

end

%figure(8)

subplot(2,2,4)

plot(t2,conv2\_true\_plot)

title('True conv of x with h : second function portion')

%-----------------------------

%% Combine funcitons

%-----------------------------

% convolution property is distributive

% So, h\*(f\_g) = h\*f + h\*g

% We need to pad the first function as it goes from -5 to 10

% while the second function goes from -7 to 10 by .1

pad\_zeros = zeros(1,20);

conv1\_padded = [ pad\_zeros conv1];

conv\_final = conv1\_padded + conv2;

figure(3)

plot(t2,conv\_final)

title('Final convoloution sum')







Problem 2.

2. (20%) Explain PSF, OTF, PTF, and MTF and describe their relationship (assume incoherence illumination).

Answer:

The PSF is the point spread function and is called the impulse response of the system. The impulse response, h(t) of a linear time-invariant (LTI) system in one dimension is the response y(t) when an impulse is injected. Here the PSF is the impulse response in 2-D. For an incoherent system the squared PSF is the impulse response of the system.

The OTF or optical transfer function (OTF) is the Fourier transform of the impulse response H (or the PSF squared) of an incoherent imaging system. In general this is a complex function and is in the frequency domain.

The PTF is the phase of the OTF, since the OTF can be complex.

The MTF is the magnitude of the complex OTF.

Problem 3.

3. (20%) An optical imaging system (Figure1, no aberrations; 1:1 magnification) has a circular exit pupil (12.5-mm diameter) and the distance of exit pupil to image plane zi is 125 mm. Assume the system is under incoherent illumination (light wavelength 520 nm). An object (USAF-1951.png, size: 0.5mmx0.5288mm) is place in the object plane. Use MATLAB to (1) calculate the optical transfer function and modulation transfer function of the optical system; and (2) simulate and plot the resulting image of the target.

Answer:

%---------------------------------------------------------------------

% file name : hmwk\_3\_prob\_3\_incoh\_diff.m

% Student: Ray Duran

% Date: 10/11/21

% Class : EECS 590 Professor Liang, Fall Semester

% University of North Dakota

% Descr:

% Calculate the Optical and modulation transfer functions, OTF and

% MTF of a black box aperture

%

%

% All code Borrowed from Prof Bo Liang in his Fourier Optics MATLAB demo,

% from EECS 590

%

%---------------------------------------------------------------------

A = rgb2gray(imread('USAF-1951.png'));

[M,N] = size(A);

Ig = single(A);

Ig = Ig/max(Ig(:));

Ug = sqrt(Ig); % Amplitude

L1 = 0.5e-3;

L2 = 0.5288e-3;

du = L1/N;

dv = L2/M;

u = -L1/2:du:L1/2-du;

v = -L2/2:dv:L2/2-dv;

[uu,vv] = meshgrid(u,v);

lambda = 0.5\*10^-6;

wxp = 6.25e-3;

zxp = 125e-3;

f0 = wxp/(lambda\*zxp);

fu = -1/(2\*du):1/L1:1/(2\*du)-(1/L1);

fv = -1/(2\*dv):1/L2:1/(2\*dv)-(1/L2);

[Fu,Fv] = meshgrid(fu,fv);

H = circ(sqrt(Fu.^2+Fv.^2)/f0); % This is logical

H\_doub = double(H);

H\_area = nnz(H\_doub); % Calculate area of aperture

PSF\_incoh = abs(fftshift(fft2(H)))/H\_area; % This is also known as the

% Optical transfer function, where

% we have to square the H because the

% impulse responce of such a system is

% the sqaure magnitude of the amplitude

% impulse response.

% Note: This is essentially the foundation

% of diffraction modeling

% Fraunhofer Diffraction fomrula where

% if we use a far-field diffreaction

% as an approx then the transfer

% fucntiuo is a 2D transform!

PSF = PSF\_incoh;

OTF = fft2(PSF);

OTF = OTF/OTF(1,1);

figure

subplot(2,2,1)

mesh(uu,vv,PSF)

subplot(2,2,2)

MTF = abs(fftshift(OTF));

mesh(Fv,Fu,MTF)

subplot(2,2,3)

imagesc(u,v,PSF\_incoh)

subplot(2,2,4)

MTF = abs(fftshift(OTF));

imagesc(fu,fv,MTF)

%

Gg = fft2(Ig); % Use Ig for incoherent system

Gi = Gg.\*OTF;

Ui = ifftshift(ifft2(Gi));

Ui = abs(Ui);

Ii = Ui.^2;

%

figure

subplot(1,2,1)

imshow(Ig)

subplot(1,2,2)

imshow(Ii)

Diagram

Description automatically generated



Problem 4.

4. (20%) What is light field imaging? Describe the advantages and limitations of light field imaging.

Answer:

A fundamental definition of a light field would be the parameterization of the flow of light thru an empty region of 3D space. Now, there are many ways to represent these parameters. In the most redundant of systems you can use a 7D system, which would include (x, y, z, theta(angle), alpha(angle), t, and lambda(wavelength). There are other systems, in the paper “Light Field and Computational Imaging”, Levoy references Moon and Hanrahan which annotate a 4D system as a light field, defined as the image plane (x, y) that passes thru an aperture plane (u, v), thus comprising the dimensions (x,y,u,v)

There are some limitations of light-field imaging. One disadvantage is that occlusion prevents information of concave objects. Also, if time is part of your parameters there are natural limitations of illumination of dynamic scenes. Although not a strict disadvantage, as there are ways to compensate, but geometrical complex objects present challenges to light field processing as multiple reflections, refractions and scattering affect the light ray. This last area is an active area of research. Perhaps the biggest disadvantages are economics and adoption of the technology. Because of the ascendancy of CMOS cameras in ubiquitous smart phones like the iPhone, the public does not see much extra value with plenoptic functions and light field systems. The widespread adoption of light fields would have to add salient extra benefits not found in today’s cameras. However, to take advantage of that would require a technology increase in bandwidth, sensors, memory, and computer speed many times greater than today to make light fields camera applications practical.

On the other hand, light field imaging provides new ways of capturing images and brings some benefits that traditional photography and imaging do not have. One advantage of light-field imaging is that with one shot you can provide “plenoptic” information of angles which would allow, with computational imaging, the reconstruction of focusing on different regions of the image. Closely related to this advantage is that you do not have to decide at the moment of image capture the focus as “post-processing” can extract different depths of field. Another advantage is that there are at least two different ways to capture light fields in an inexpensive way: sequential capture and spatial multiplexing. Spatial multiplexing uses a “robotic arm” to position lamps and cameras to capture the plenoptic information, while spatial multiplexing makes use of micro-lenses. Finally, because light field provides more information than just 2-D, views and angles that were not even captured or sampled with the given light field system, a light field rendering can generate new images, a form of interpolation that provides new custom views and presentations of the original image.

Problem 5.

5. (20%) Draw a diagram of a typical light-field camera, indicate the key components, and explain their functions.

Answer:



The light field can be described as a ray of light that passes thru two sets of points in two planes, here the (x, y) coordinate of the image plane and the (u, v) plane. The image plane is our object of interest.

The aperture plane is considered the main lens of the system as can also act as a relay system to the lens array module that follows. For example, two points in the image plane that appear lateral to each other will be captured at or near same point in the aperture but can be resolved further down the system with more information. At the aperture plane, with the use of a shutter, the function of digitally stopping down can be performed that sums the central portion of each micro lens. Another feature not found in conventional cameras systems is the ability to move the observer by moving the window at the aperture. This movement allows one to shift the focus of the image from the same single shot.

Whereas in a conventional camera the photosensors allow no extra information from the image plane, a lens array allows the spreading of the light field (x, y, u, v) and disperses the ray to sensors that provide angle information. This dispersion is essentially the key to light field cameras that use spatial multiplexing, since all information of a light field is 7D (x, y, u, v, theta, phi, lambda, and time(video)) has been reduced to 4D by our two-plane system.

A function provided from a light field system using a lens array is the ability to refocus from the same shot different depths of fields of the image plane. This is done by summing windows extracted from several micro lenses.

The sensor array can ultimately resolve and reconstruct with computational computing the intensity, wavelength (in some systems), angles and (x, y) locations of the image plane.