Problem 1.

1. (20%) There are two basic approaches for Time-of-flight camera: continuous wave modulation approach and pulsed-based approach. Briefly describe their principles and what are their major differences?

Ans(a): A continuous wave time of flight camera uses modulation to measure the phase and determine the properties of the return signal. This in turn can be used to calculate range.

The above is accomplished by first modulating a wave and detecting the return signal

*s(t)* = *a*1 + *a*2 cos*(*2*π f t),*

*r (t)* = *A* cos*(*2*π f t* − 2*π f τ)* + *B*

The cross-correlation gives us after some mathematical footwork:

Text, whiteboard

Description automatically generated

Now, note here that tau is our time delay, and that x is a variable in our original cross-correlation. Using equally spaced samples in one modulation period: 0, pi/2, pi. And 3\*pi/2 we can solve C and solve for

Phi, A and B.

Now, on the other hand a pulsed based time of flight camera sends out pulses of light by a laser or diode that once detected by the object are then computationally processed with time to digital converters (TDC) and time to amplitude circuits to reconstruct the signal

Ans(b)

The major difference is that CW cameras are measuring phase differences between a continuous signal and a pulse-based camera is directly measuring the round-trip. Related to each of these systems based on their measurement technique they each have limitations. For the CW this is dealing with the phase ambiguity and for pulsed it is the accuracy of the SPAD sensor.

Problem 2.

2. (20%) How does single photon avalanche diode work? Why it can be used to detect single photon?

Ans part a:

See the figure below. If you reverse bias a photodiode and dope the semiconductor material in a certain way you are essentially making it unstable. Now normally, materials are impure and there will be some electrons in p-type material and holes in n-type. So, if the material is properly doped then a SINGLE photon can start an avalanche breakdown. (a sort of runaway current as more electrons creates more holes).

Diagram, timeline

Description automatically generated

Ans part b:

The reason that these single photons can be detected as because instead of using an ADC(analog to digital converter) a time to digital converter is used.

Given the engineering limitations of ADCs, that include resolution and sampling times, using them to compute a single photon event does not seem practical. On the other hand if you model photon flux as:

A picture containing text

Description automatically generated

where g is the initial pulse, tau is a discrete Dirac function and a[t] is the ambiguity photon flux,

Now the SPAD outputs:

A picture containing chart

Description automatically generated

where eta if probability and d represent false events.

Next, if the above experiment can be repeated N times, then the probability of detecting a certain number of events is a Poisson distribution, given by:

Text

Description automatically generated

Now we can use the histogram to compute statistically single photons.

Problem 3.

3. (10%) Briefly describe how structured light 3D surface imaging works.

Ans:

Here in the diagram below two known patterns of grayscale column images are similar except one is phase shifted in relation to the other. Now, both are projected sequentially to the object. The deformation of the patter helps us to topologically map distances of the object correctly, helping us reconstruct the shape.

In addition there are various coding, binary, gray and random patterns(fringe and sinusoidal) that can be employed. More advanced techniques add modulation to the patterns and dithering which is a widely employed trick in image processing to give the appearance of more fidelity.

A picture containing diagram

Description automatically generated

Problem 4.

4. (20%) Name two techniques for ultrafast transient imaging and describe their basic principles.

One method is with femto-photography using streak camera. Here the system is comprised of light source, diffuser, beam splitter, synch circuits and the streak camera. (As seen below in the figure a). Now, the reason for all the complexity owes to the nature of light and its speed. Here, after the light has been converted into spherical waves after interacting with the object it enters the streak camera (figure b.) Operating in a way analogous to old-cathode tube early television, the photons are converted into electrons, where the speeds of the circuit become manageable. (Note, this is the reason for the synch circuits which must control the sweeping of electrons) The streak camera records the horizontal position in one axis and time on the other. (Figure c) Finally, with the use of a scanning mirror, the y axis of the object can be built up (figure d) and then a finally a motion video can be reconstructed.

Chart

Description automatically generated

Another method is with the use of single-photon avalanche diodes (SPADs).

The basic principle with techniques that use SPADs is that the measurement problem, speed of photons is too great, is solved by framing the problem as a probability measurement. As seen in the diagram below. The photon flux is input into the SPAD, with the output being fed into a time to digital converter, and transient flight time computed from a histogram. This makes the computation of the flight time again manageable as there are not practical ways of converting voltages using an analog to digital converter.

Diagram

Description automatically generated

Problem 5.

5. (30%) Transient image reconstruction using MATLAB. Download the dataset and code from Transient imaging with SPADS link, choose one dataset, and use MATLAB (demo code is provided and demonstrated during the class) to reconstruct the transient images. Try different PSF options for reconstruction and see which is better.

Ans: See code below, that is from Prof. Bo Liang’s class developed by Lindell et. Al. We just used options to generate figures. As we can see from the figures below, we get our best results by using spatial and temporal blur.

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

% file name : hmwk\_5\_prob\_5\_transient\_imaging.m

% Student: Ray Duran

% Date: 11/21

% Class : EECS 590 Professor Liang, Fall Semester

% University of North Dakota

% Descr:

% See Below; All code from Lindell et. al below. In this exercise we just

% experimented with the different options

% Note: Need to include folders and sub-folders of scripts3D, data and

% results for proper operation.

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

% TOWARDS TRANSIENT IMAGING AT INTERACTIVE RATES WITH SINGLE-PHOTON DETECTORS

% Supplementary Datasets

% David B. Lindell, Matthew O'Toole, Gordon Wetzstein

%

% Datasets

% The raw captured results are stored in the 'data' directory. These .mat files

% contain both the raw data captured from the sensor, and the calibrated

% measurements formatted as transient images with dimensions arranged as

% [transient\_image\_idx, transient\_frame\_idx, row, column]. Processed results are

% stored in the results folder. Processing code can be downloaded separately

% from the project webpage. Before running the code, be sure to place the 'data'

% and 'results' folders into the same directory as the README file included with

% the code.

%% Prepare the dataset

% filenames of all scenes as well as temporal slices to use

scenes = { 'ghost\_highres', 710, 919; ...

'mirror\_highres', 730, 939; ...

'fiber\_highres', 800, 1400; ...

'david\_highres', 760, 879; ...

'wall\_highres', 760, 879; ...

'diffuser\_highres', 710, 889;

};

data\_path = 'transient\_imaging\_code\data\';

sceneIDX = 3;

filename = scenes{sceneIDX,1};

load([data\_path, filename '.mat'], 'transient');

% remove dead SPADs

transient(:,:,181,:) = [];

transient(:,:,77,:) = [];

transient(:,:,7,:) = [];

% get number of frames

numFrames = size(transient,1);

disp(['scene: ' filename ' has ' num2str(numFrames) ' frames']);

frameIDX = 1;

raw = squeeze(transient(frameIDX,scenes{sceneIDX,2}:scenes{sceneIDX,3},:,:));

%% Reconstruction

% use dark count or not

bUseDarkcount = true;

% options for the spatio-temporal psf:

% 0 - Dirac (no blur at all)

% 1 - spatial Gaussian (only spatial blur, no temporal blur)

% 2 - only temporal psf

% 3 - both spatial and temporal blur

psfoption = 2;

% ADMM parameter - should increase with smaller signal (more noise)

rho = 1e-1; % this seems to work great

lambda = 0.085;

% spatial convolution kernel

d = 5;

c = fspecial('gaussian', [d d], 1);

if psfoption==0

psf = 1;

elseif psfoption==1

% only 2D psf

psf = reshape(c, [1 d d]);

elseif psfoption==2

% load temporal psf

load('transient\_imaging\_code\scripts3D\misc\timePSF.mat');

% 3D psf

psf = timePSF;

else

% load temporal psf

load('transient\_imaging\_code\scripts3D\misc\timePSF.mat');

% 3D psf

psf = repmat(reshape(c, [1 d d]), [numel(timePSF) 1 1]) .\* repmat(timePSF, [1 d d]);

end

% normalize psf

psf = psf ./ sum(psf(:));

if bUseDarkcount

load('transient\_imaging\_code\data\dark.mat');

% create 2D volume and scale by dividing by total number of time bins in raw data

dark = dark ./ 1536;

% remove dead SPADs

dark(:,181,:) = [];

dark(:,77,:) = [];

dark(:,7,:) = [];

end

% max number of iterations

maxIters = 25;

tic;

% 3D OTF

otf = psf2otf\_3D(psf,size(raw));

% define function handle for 3D convolution

Afun = @(x) ifftn( fftn(x).\*otf, 'symmetric');

% run deconvolution

if bUseDarkcount

x = deconvADMMTV(Afun, raw, otf, maxIters, rho, lambda, false, repmat(dark, [size(raw,1) 1 250]));

else

x = deconvADMMTV(Afun, raw, otf, maxIters, rho, lambda, false);

end

toc;

%% Visualization

figure

[K,M,N] = size(x);

for ii = 1:K

img = squeeze(x(ii,:,:));

subplot(1,2,1)

imagesc(squeeze(raw(ii,:,:)))

subplot(1,2,2)

imagesc(img)

pause(0.05);

end



