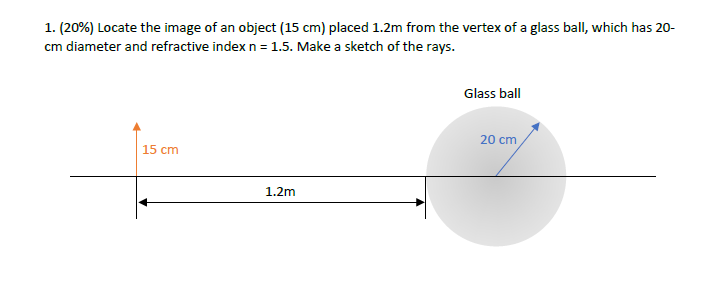
Problem 1.



Ans:

After first sphere surface, we have:

%parameters

n1 = 1.0003; % refraction index of air

n2 = 1.5; % refreactionn index of glass

so = 1200; % distance of object from sphere in cm

h = 15; % height of object in cm

r = 10; % radius of sphere

% Check that angle is small to use paraxial approximation

% if the angle made by the object path lo and so is small

% the the angle phi made by center of sphere and horizontal

% from projected line h is small;

angle\_obj = atan(h/so);

% convert to degrees

angle\_obj\_degrees = angle\_obj\*360/(2\*pi);

if angle\_obj\_degrees <= 1

disp('Angle small use paraxial approx')

else

msg = 'Angle too large do not use paraxial approx error';

error(msg)

end

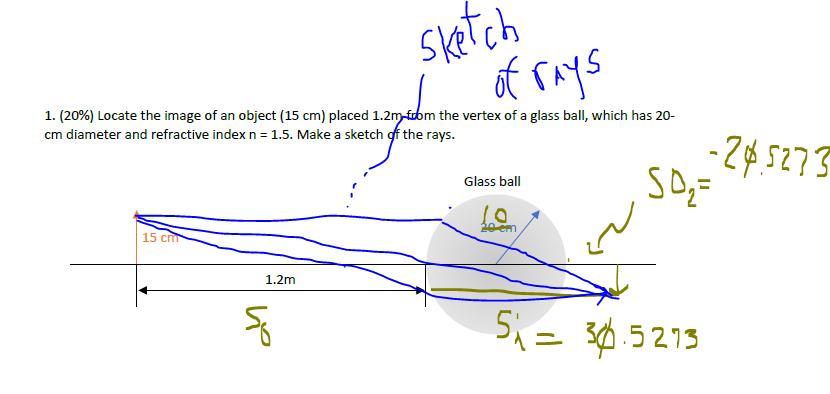
% Use paraxial approx; Solve for si

si = n2/( ((n2-n1)/r) - n1/so);

Problem 1 continued:

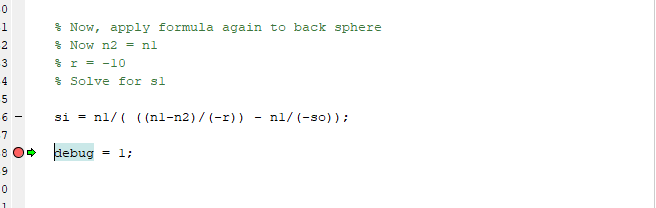
“si” = 30.5273

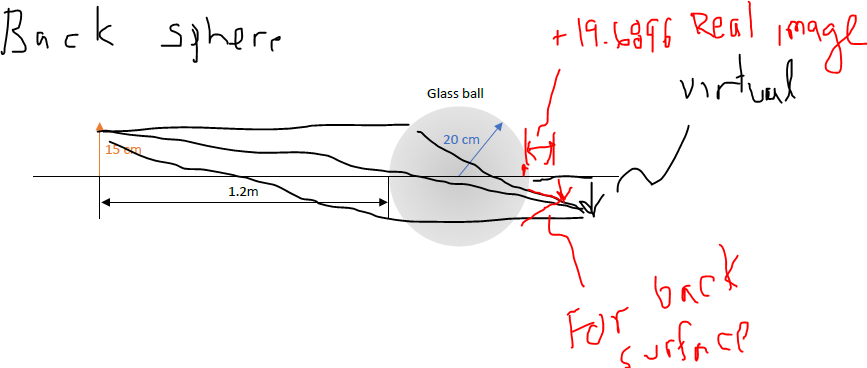
This is location of image from front of sphere.



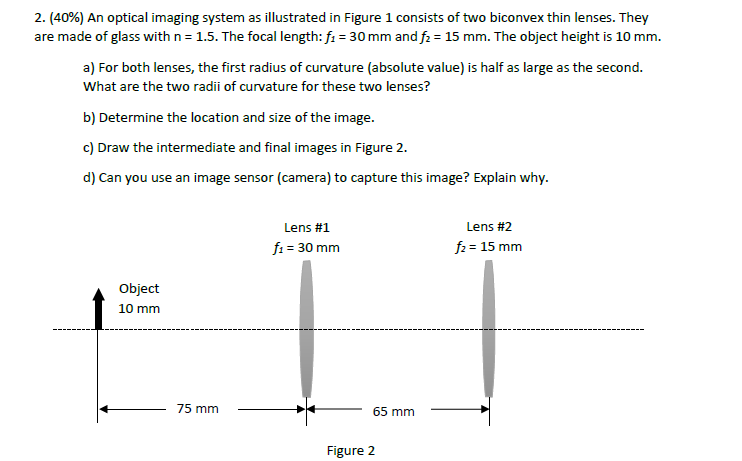
Now apply to back surface of sphere, where r < 0; “s0” < 0 (virtual) and swap n1 and n2 ;

That gives us a positive value for “s1” for vertex 2 with “s1” = 19.6896





Problem 2.

A

Ans:

Part a:

n1 = 1.0003;

n2 = 1.5; % refractive index

f1 = 30; % mm

so = 75; % mm

f2 = 15; % mm

% Part a.

% use equation: 1/f = (nl-1)(1/R1 - 1/R2)

% First radius of curvature R1 = 1/2\*R2 or R2 = 2R1

% Rewriting above thins lens formula:

% (1/R1 - 1/R2) = (n1-1)/f

% Let R2 = 2R1

% 1/(2R1) = (n1-1)/f

% 2R1 = f/(n1-1)

% R1 =f/(2\*(n1-1))

% Radius of curvature first lens

R1\_first\_lens = f1/(2\*(1.0003-1));

R2\_first\_lens = 2\*R1\_first\_lens;

R1\_second\_lens = f2/(2\*(1.0003-1));

R2\_second\_lens = 2\*R1\_second\_lens;

**So,for first lens**

**R1 = 5e4 mm**

**R2 = 1e5 mm**

**Second Lens:**

**R1 = 2.5e4 mm**

**R2 = 5e4 mm**

Part b and c

% Part b

% Determine intermediate image

% Use Gaussina lens formula

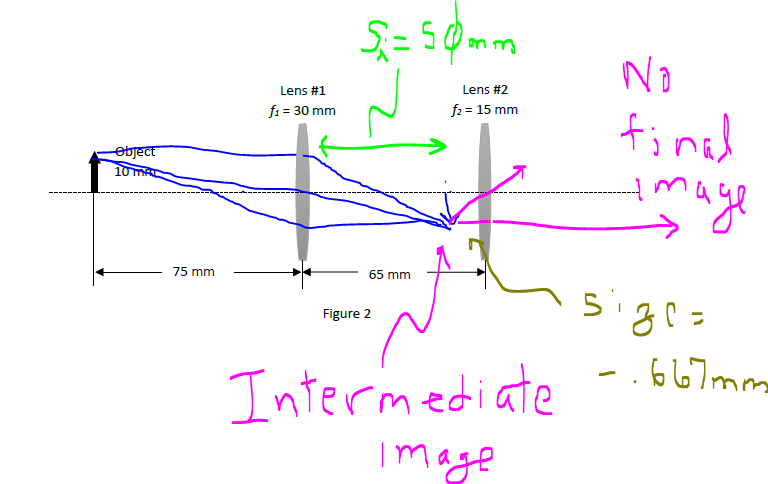
% 1/so + 1/si = 1/f

si = 1/( 1/f1 - 1/so ); % location of intermediate image

% size of intermediate image

M = -si/so;

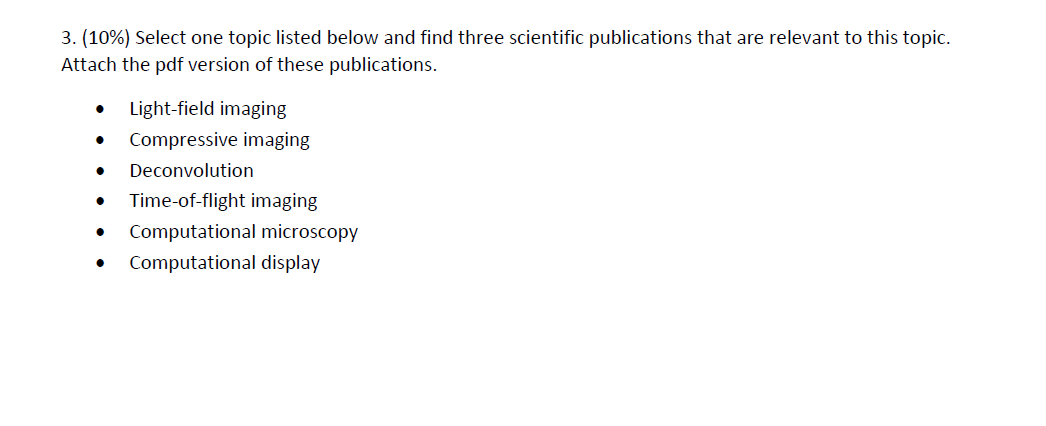
So, since intermediate image is at focal



**Intermediate image is at so = to 50 mm and size is -.667mm**

Part d **No, you cannot use an image sensor to capture image, because there is no final image..**

Problem 3.



The topic that I am selecting is : compressive imaging.

The three papers plus one tutorial that I have attached are:

“ **An Introduction to Compressive Sampling**” in the IEEE publication of Spectrum by Candes and Watklin. This is the original paper/tutorial that I read about 15 years ago that started me on my higher education journey.

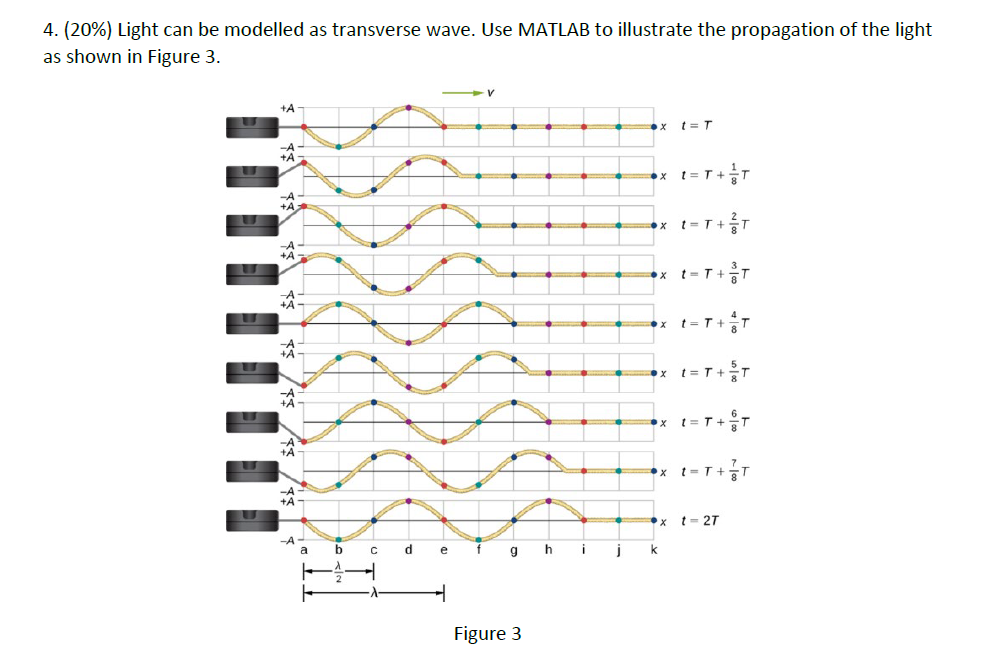
Here are the three papers:

“**Compressed Sensing for reduced hardware footprint in medical ultrasound” by Mitrovis, et. Al**

**“Compressive Imaging for Video Representation and Coding” by Wakin , et. Al**

**“Snapshot Compressive Imaging: Principle, Implementation, Theory, Algorithms and Applications”, by Yuan, et. al**

Problem 4



Ans:

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

% file name : hmwk\_1\_prob\_4.m

% Student: Ray Duran

% Date: 9/12/21

% Class : EECS 590 Professor Liang, Fall Semester

% University of North Dakota

% Descr: Model traverse wave nature of light

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

A = 1;

t = [-3\*pi:pi/8:2\*pi]

y\_leading\_zeros = zeros(1,83);

y = sin(t - pi );

% pad zeros in front of wave

y\_padded = [ y y\_leading\_zeros];

%plot(t,y)

debug = 1;

t\_sample = 1 : 100;

% initial position

figure(1)

subplot(9,1,1)

plot(t\_sample,y\_padded(t\_sample+24))

axis([0 42 -1.1 1.1]);

title('Tranverse Wave Propagated Initial t = T')

% Transverse wave at 1/8T

y\_at\_18 = circshift(y\_padded,2);% propagate for light ; In model just do a circular shift

subplot(9,1,2)

plot(t\_sample,y\_at\_18(t\_sample+24))

axis([0 42 -1.1 1.1]);

title(' Tranverse Wave Propagated t = T + 1/8T')

% Transverse wave at 2/8T

y\_at\_28 = circshift(y\_padded,4);% propagate for light ; In model just do a circular shift

subplot(9,1,3)

plot(t\_sample,y\_at\_28(t\_sample+24))

axis([0 42 -1.1 1.1]);

title(' Tranverse Wave Propagated t = T + 2/8T')

% Transverse wave at 3/8T

y\_at\_38 = circshift(y\_padded,6);% propagate for light ; In model just do a circular shift

subplot(9,1,4)

plot(t\_sample,y\_at\_38(t\_sample+24))

axis([0 42 -1.1 1.1]);

title(' Tranverse Wave Propagated t = T + 3/8T')

% Transverse wave at 4/8T

y\_at\_48 = circshift(y\_padded,8);% propagate for light ; In model just do a circular shift

subplot(9,1,5)

plot(t\_sample,y\_at\_48(t\_sample+24))

axis([0 42 -1.1 1.1]);

title(' Tranverse Wave Propagated t = T + 4/8T')

% Transverse wave at 5/8T

y\_at\_58 = circshift(y\_padded,10);% propagate for light ; In model just do a circular shift

subplot(9,1,6)

plot(t\_sample,y\_at\_58(t\_sample+24))

axis([0 42 -1.1 1.1]);

title(' Tranverse Wave Propagated t = T + 5/8T')

% Transverse wave at 6/8T

y\_at\_68 = circshift(y\_padded,12);% propagate for light ; In model just do a circular shift

subplot(9,1,7)

plot(t\_sample,y\_at\_68(t\_sample+24))

axis([0 42 -1.1 1.1]);

title(' Tranverse Wave Propagated t = T + 6/8T')

% Transverse wave at 7/8T

y\_at\_78 = circshift(y\_padded,14);% propagate for light ; In model just do a circular shift

subplot(9,1,8)

plot(t\_sample,y\_at\_78(t\_sample+24))

axis([0 42 -1.1 1.1]);

title(' Tranverse Wave Propagated t = T + 7/8T')

% Transverse wave at 8/8T

y\_at\_88 = circshift(y\_padded,16);% propagate for light ; In model just do a circular shift

subplot(9,1,9)

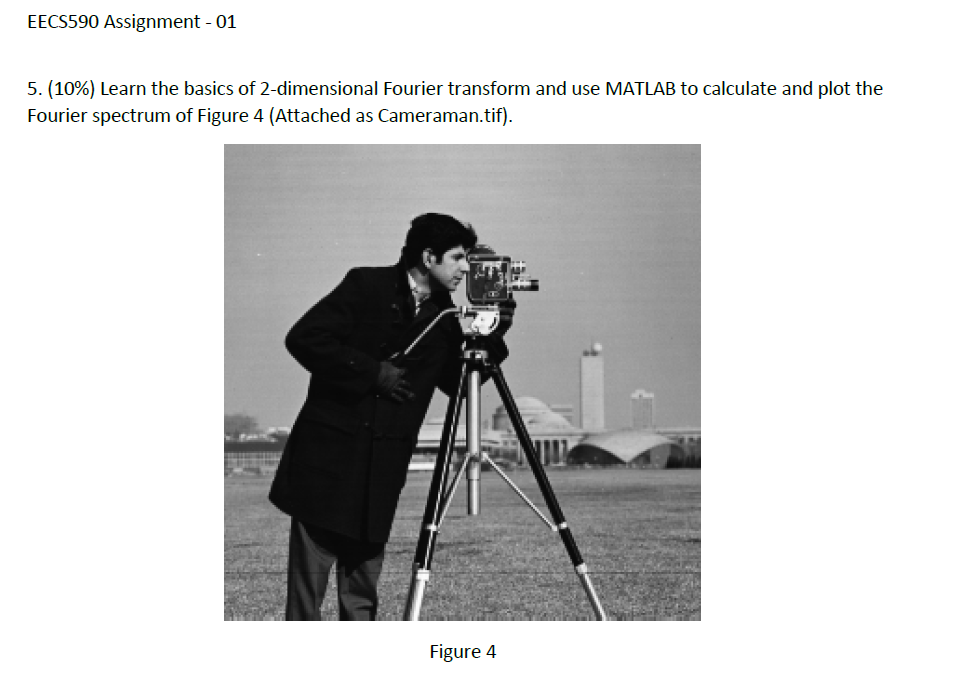
plot(t\_sample,y\_at\_88(t\_sample+24))

axis([0 42 -1.1 1.1]);

title(' Tranverse Wave Propagated t = 2T')



Problem 5



Ans :

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

% file name : hmwk\_1\_prob\_5.m

% Student: Ray Duran

% Date: 9/9/21

% Class : EECS 590 Professor Liang, Fall Semester

% University of North Dakota

% Descr: 2D FFT of Camerman

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

X = imread('cameraman.tif');

debug = 1;

sizeX = size(X,1);

sizeY = size(X,2);

% Setup Figure

lims = [ -256 256];

clim = [ 0 2000];

figure(1), clf

plot(1)

imageh = imagesc(zeros(sizeX));

axis square, axis off, axis xy

set(gca,'xlim',[lims(2)-30 lims(2)+30],'ylim',[lims(2)-30 lims(2)+30],'clim',[clim(1) clim(2)])

title('Amplitude Spectrum')

% 2-D FFT

img = abs(fftshift(fft2(X)));

set(imageh,'CData',img);

debug = 1;

Adjust picture and then create figure with this code:

function createfigure(cdata1)

%CREATEFIGURE(cdata1)

% CDATA1: image cdata

% Auto-generated by MATLAB on 11-Sep-2021 18:42:32

% Create figure

figure1 = figure;

% Create axes

axes1 = axes('Parent',figure1);

axis off

hold(axes1,'on');

% Create image

image(cdata1,'Parent',axes1,'CDataMapping','scaled');

% Create title

title('Amplitude Spectrum');

% Uncomment the following line to preserve the X-limits of the axes

% xlim(axes1,[0.5 256.5]);

% Uncomment the following line to preserve the Y-limits of the axes

% ylim(axes1,[0.5 256.5]);

box(axes1,'on');

axis(axes1,'square');

% Set the remaining axes properties

set(axes1,'CLim',[0 2000],'Layer','top');

% Create colorbar

colorbar(axes1);

Results:



Figure 1. Amplitude Spectrum of Camera Man with 2-D FFT

Analysis:

This is a 2-D FFT which means **before** a shift that centers everything that a 1-D FFT is applied to the columns resulting in a low frequency at the top of figure up to 2\*Pi high frequency at the bottom. Then a 1-D FFT is applied along the rows with the left side of the figure at dc and the right side again at the sampling frequency. A FFTshift moves the low frequency to the center. So, this is a natural image which means that most of the energy is at the low frequencies as is shown by figure 1.

As a sanity, we can run a 2-D wavelet, which instead of using a basis of cosine and sines uses a mother wavelet expressing the image in time and frequency. What the addition of the wavelet shows us(figure 2) in addition to the 2-D FFT is how much energy is at the lowest frequency , but that there is still energy at the highest frequencies(edges) that you can see in the wavelet level 1 and in the corners of the 2-D FFT plot.

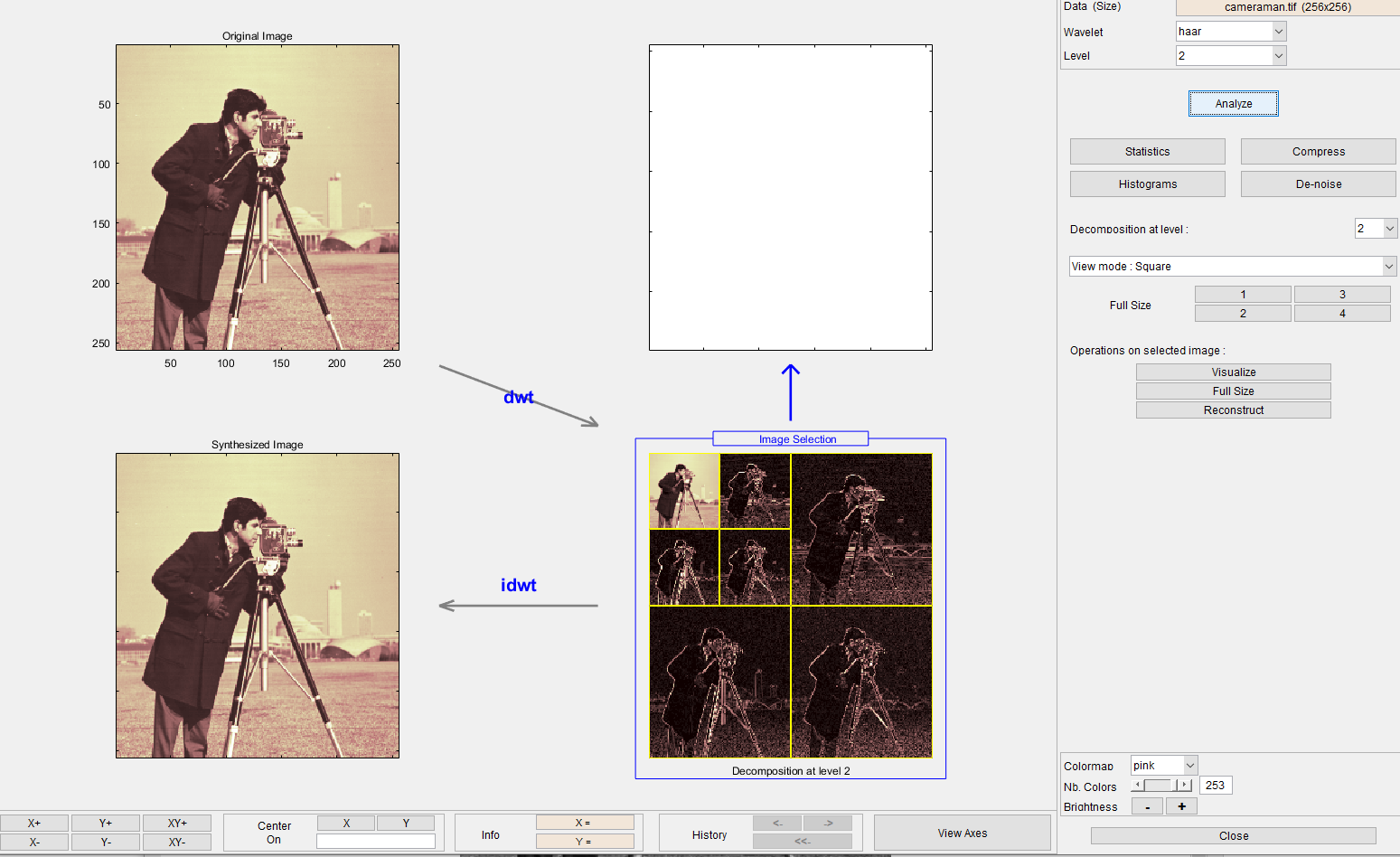


Figure 2. Wavelet Decomposition of Camera Man

All Matlab code at: <https://github.com/mathFPGAseek/und_eecs590_F21>