Question 2. For each pixel, you captured a sequence of measurements. Plot a histogram of these pixel values for a given pixel (you can use MATLAB's built in function hist). This plot serves as an estimate of the Probability Distribution Function (PDF) for the noise. Create a few plots for different pixels. What do you observe about the PDF? What is its approximate shape?

Answer: 4 points are randomly chosen and pixel values of each points are plotted in the Figure 1 and Figure 2. The horizontal axis is pixel value and the vertical axis is the quantity of every value. From the histograms we can see that the quantities of values for each pixels could be represented by Gaussian distributions versus the values.

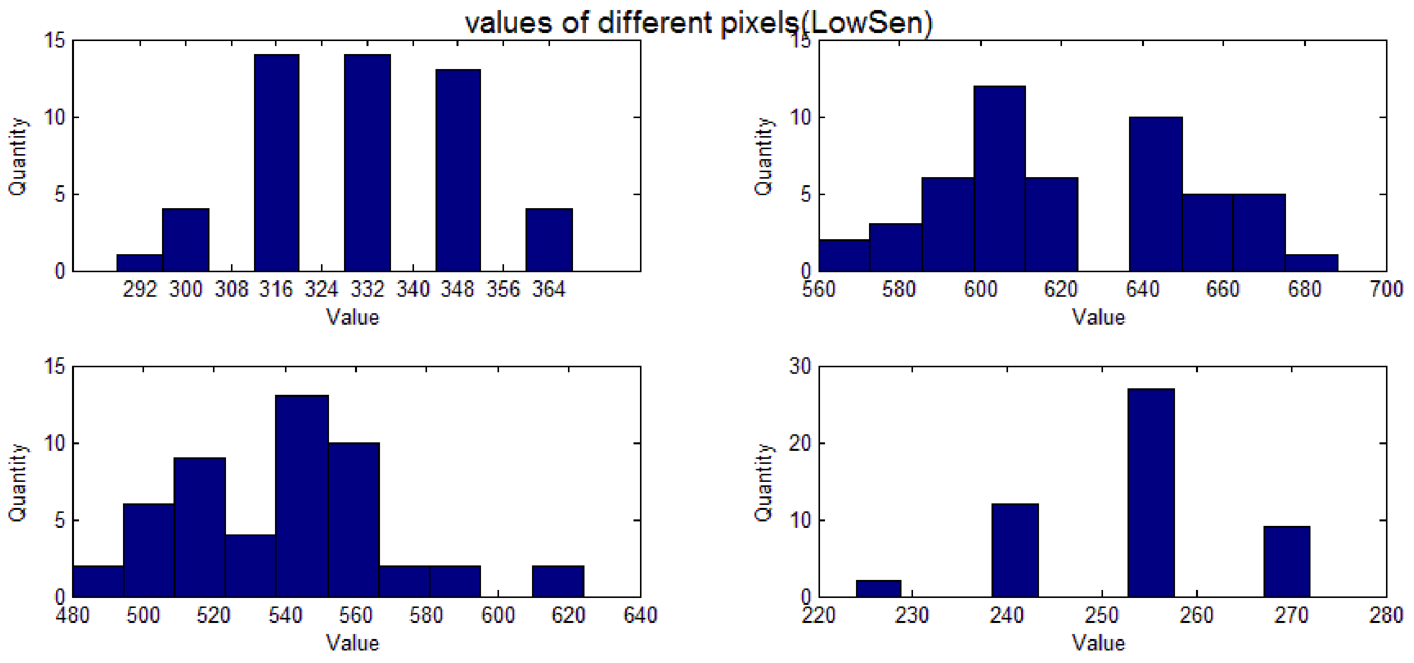

Figure 1. Low Sensitivity Setting

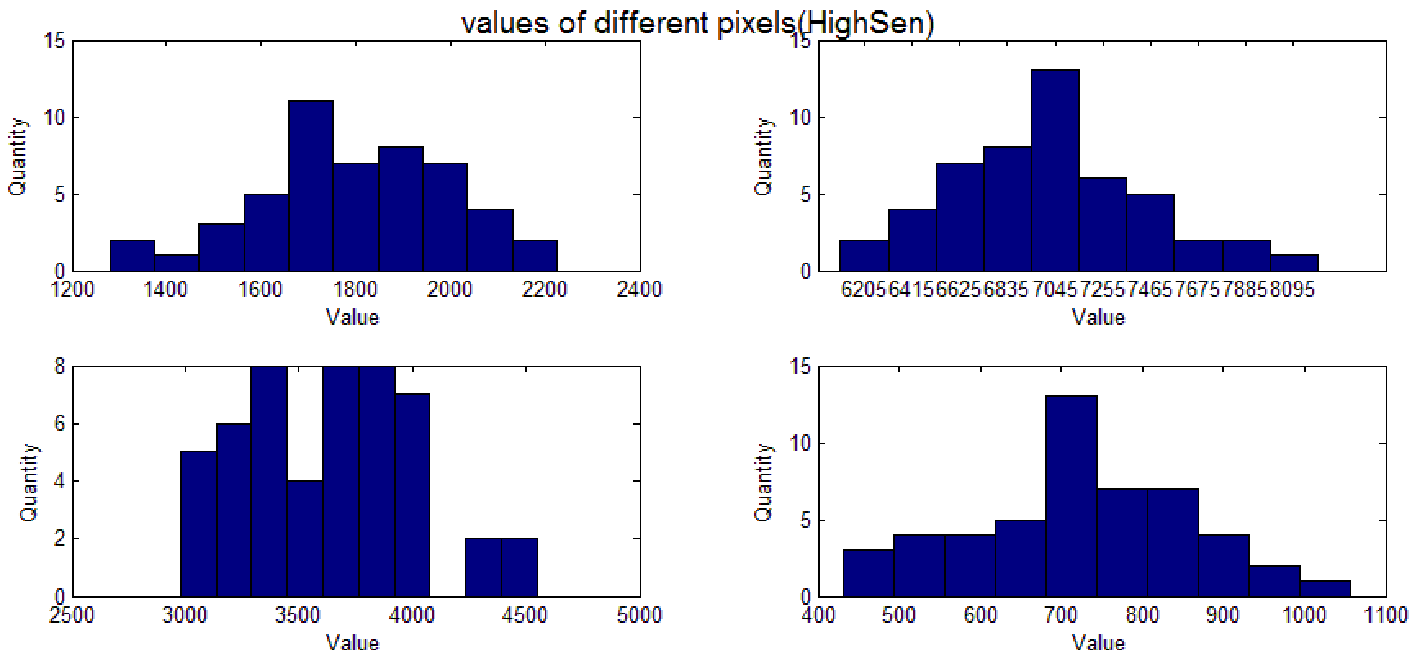

Figure 2. High Sensitivity Setting

Question 3. Use equations (2) and (3) to calculate the mean and variance for each pixel in the image. Do this separately for each experiment (i.e. for the different gain settings). Include a figure that shows both the mean and variance images, as well as a few of the original images you captured.

Answer: Figure 3. and Figure 4. are images of Low Sensitivity Setting experiment and High Sensitivity Setting experiment respectively.

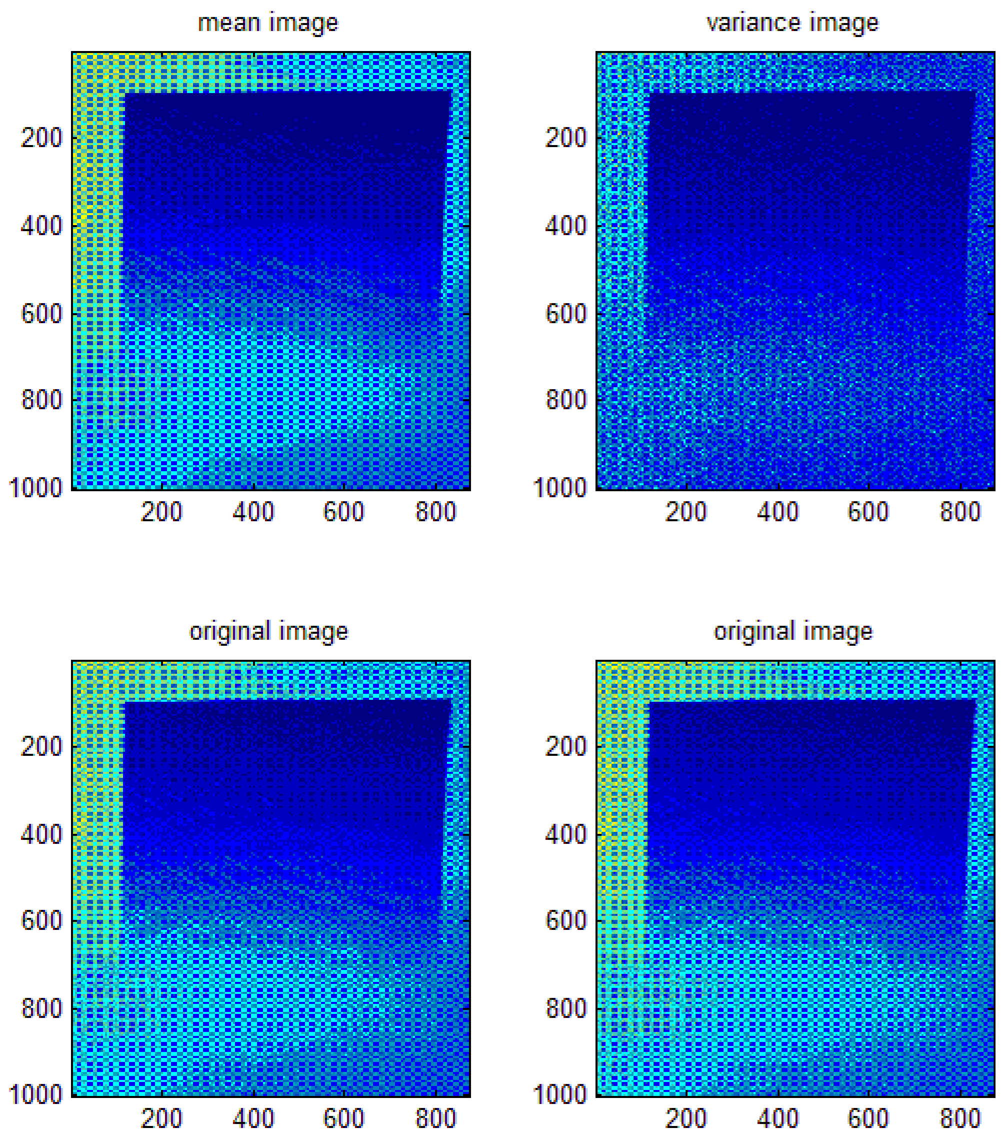

Figure 3. mean image, variance image and original images of Low Sensitivity

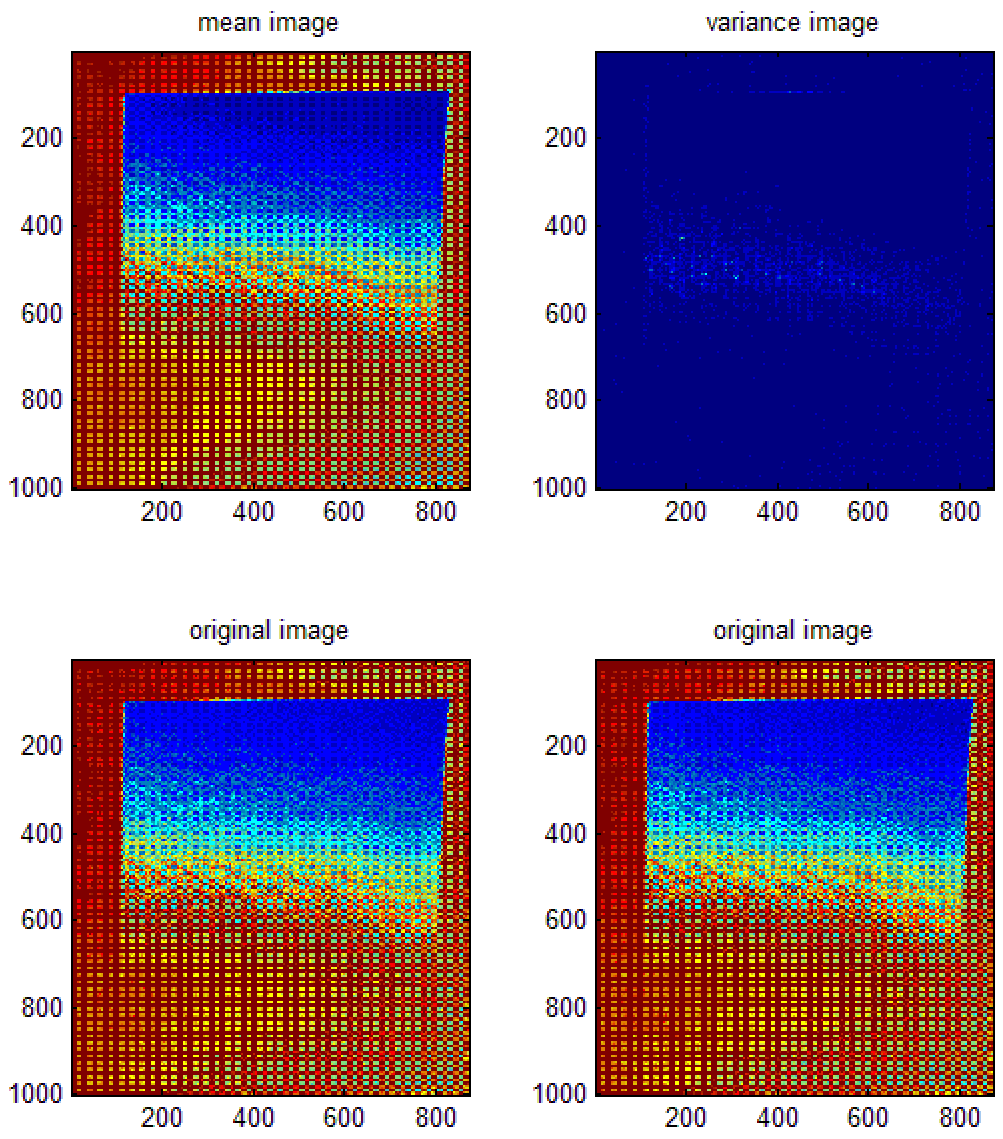

Figure 4. mean image, variance image and original images of high Sensitivity

Question 4. Plot the variance as a function of the mean for each experiment.

Answer: From results of the original plots, we can see that the minimum of mean is not 0. That is because the Tegra camera cannot store the pixel of the object, whose value in fact is 0, as 0 value. So, the error makes the fitting line doesn`t intersect y axis on its positive axis, which leads to the calculation of noise can be negative results. To avoid this kind of error, the values of mean have been shifted. Every value minus the minimum value of u so as to let the fitting cure has a positive y intersection and keeps the original shape. What`s more, when reading the photos, I transformed the unit16 to double to make calculation quicker. And when it comes to round the values of mean, because the values are all decimals, I rounded them by keeping a certain number of significant digits. Finally, I used 4 significant digits to do the next calculation work. Figure 5. and Figure 6. are plots of Low Sensitivity Setting experiment and High Sensitivity Setting experiment respectively after shifted without averaging the σ^2 of for same μ. Figure 7. and Figure 8. are plots of Low Sensitivity Setting experiment and High Sensitivity Setting experiment respectively after averaging the σ^2 of for same μ with 3 significant digits. Figure 9. and Figure 10. are plots of Low Sensitivity Setting experiment and High Sensitivity Setting experiment respectively after averaging the σ^2 of for same μ with 4 significant digits.

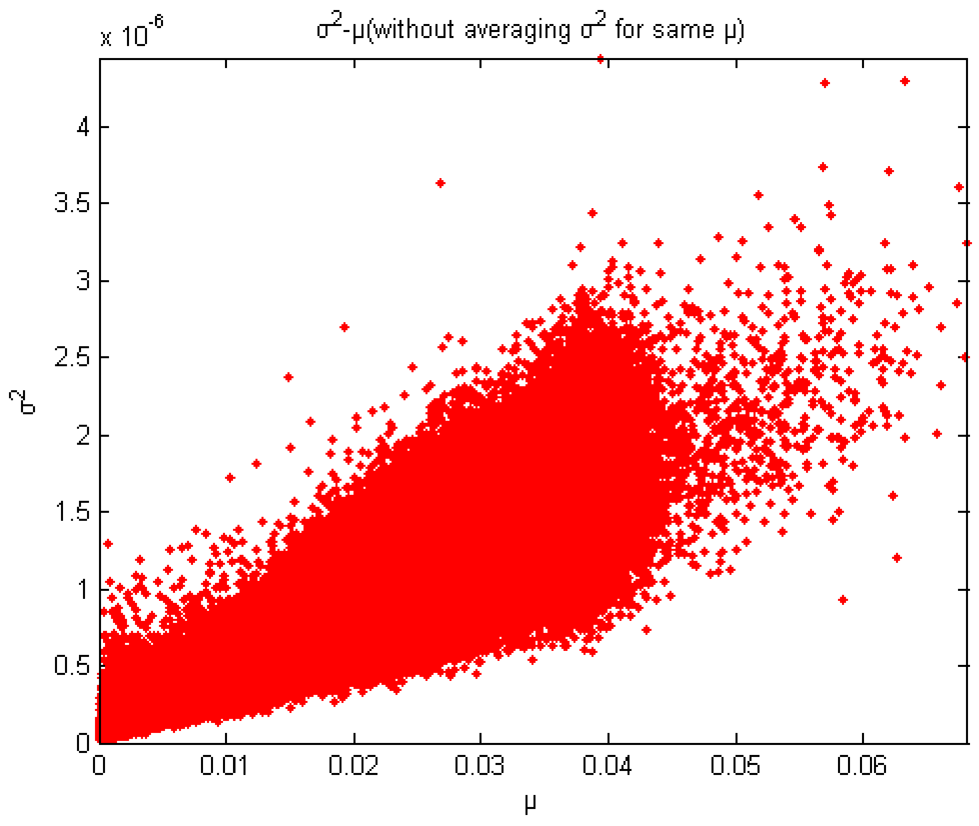

Figure 5. Low Sensitivity Setting

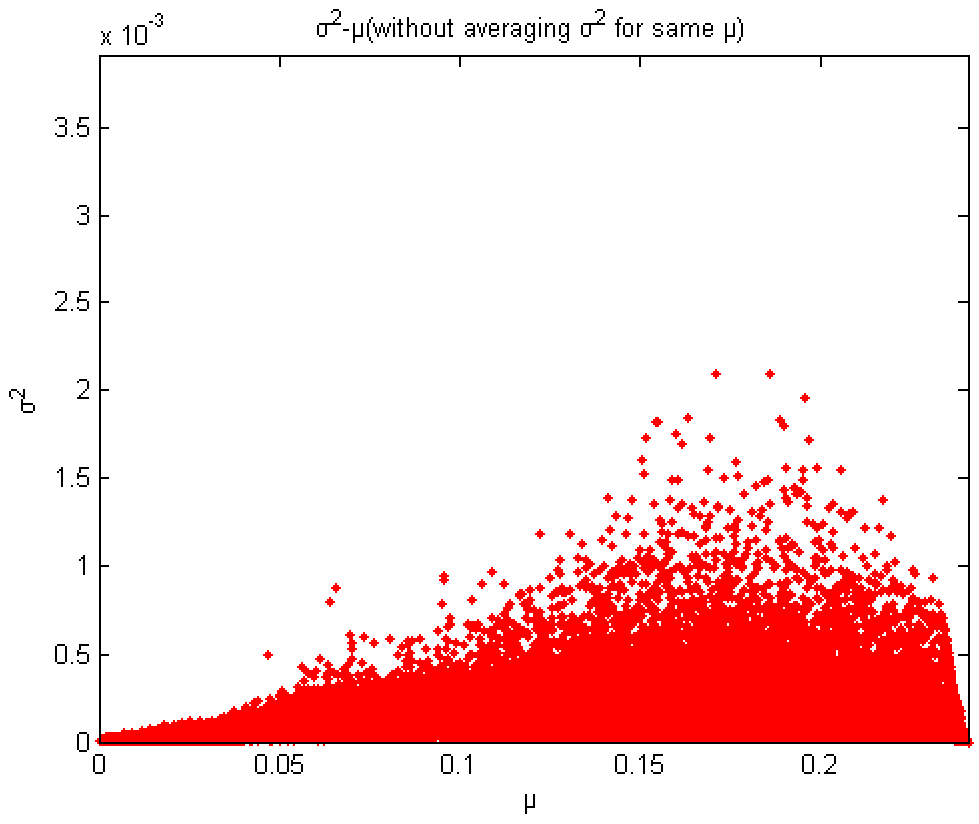

Figure 6. High Sensitivity Setting

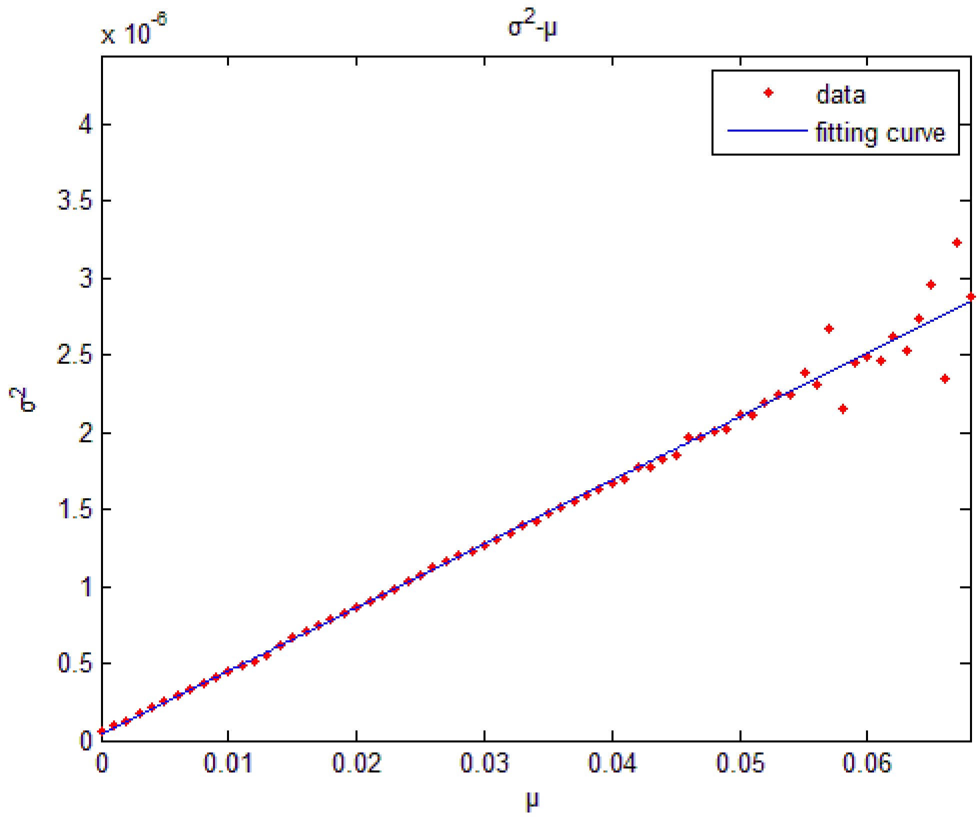

Figure 7. Fitting curve of Low Sensitivity Setting, 3 significant digits

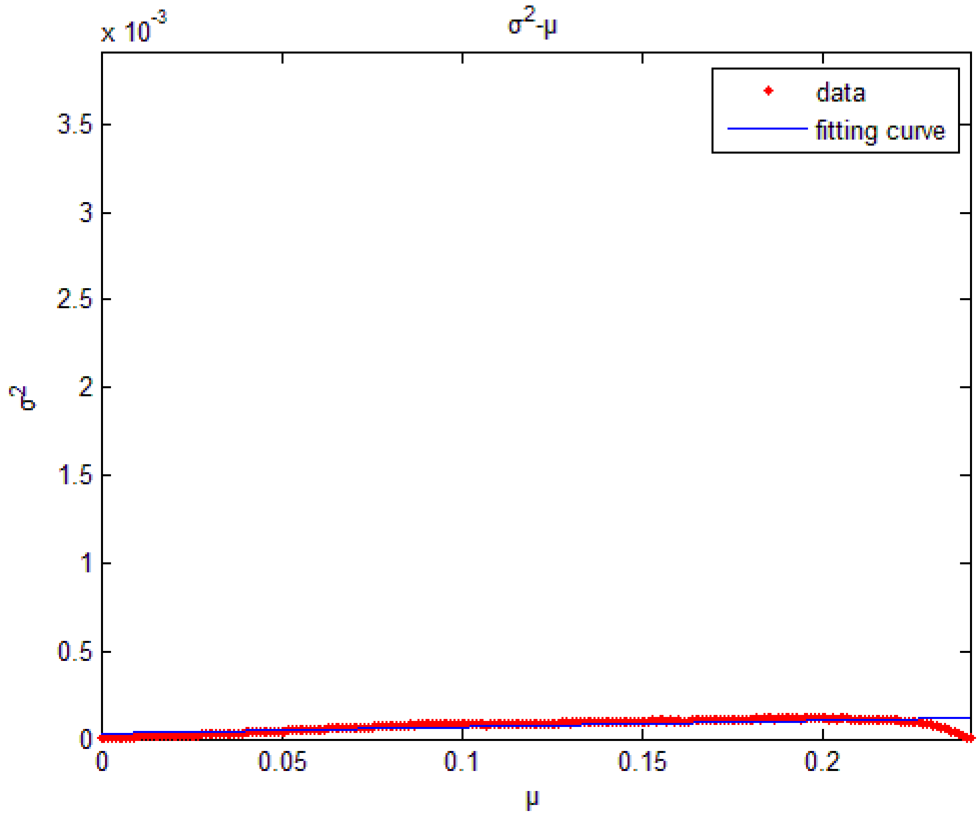

Figure 8. Fitting curve of High Sensitivity Setting, 3 significant digits

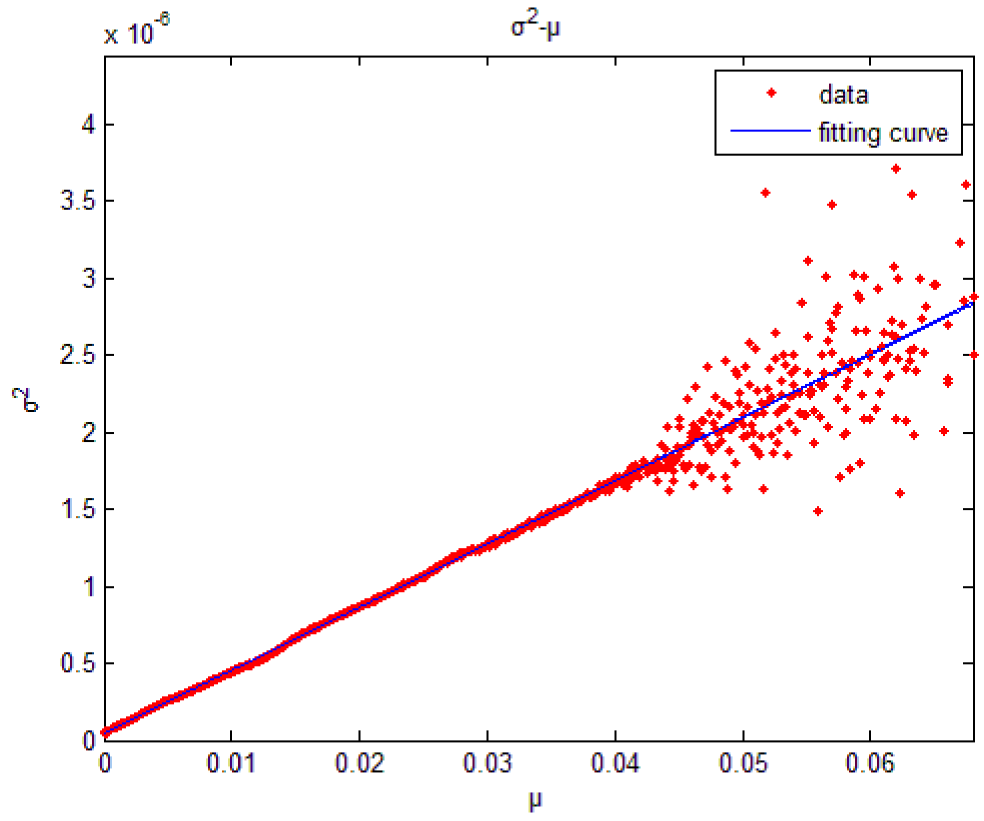

Figure 9. Fitting curve of Low Sensitivity Setting, 4 significant digits

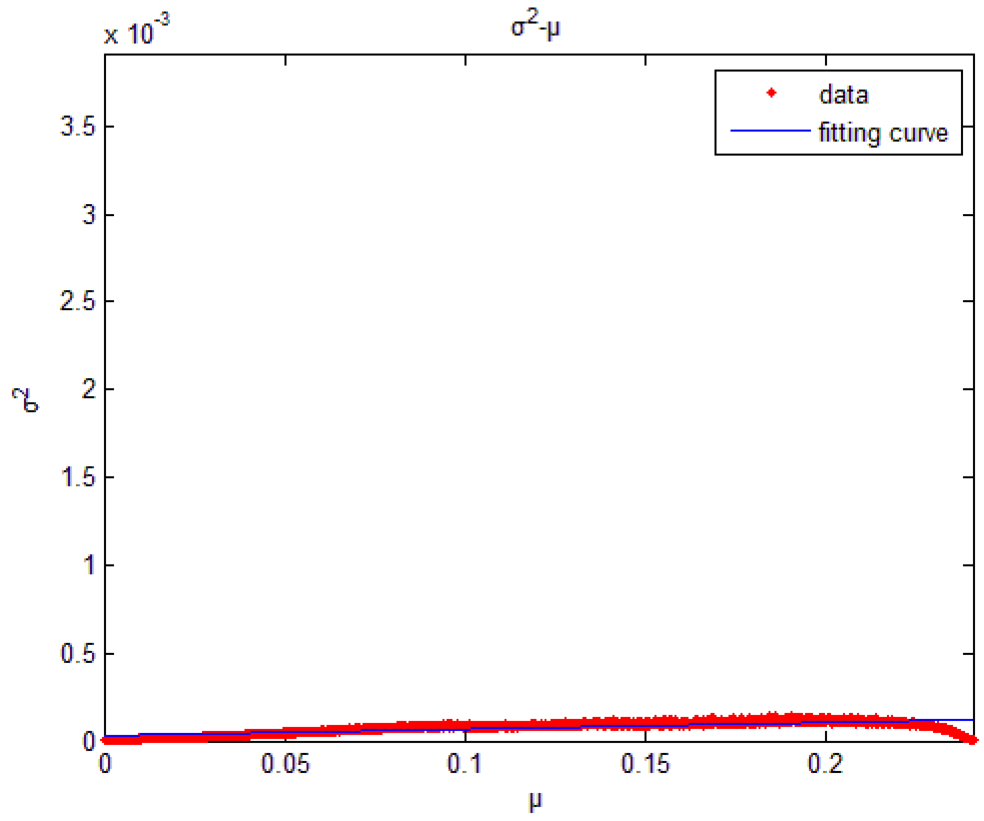

Figure 10. Fitting curve of High Sensitivity Setting, 4 significant digits

Question 5 and Question 6. For each experiment, fit a line to the plotted data. You can do this using the built-in MATLAB function polyfit. For example, if your mean values are stored in a vector x and your variance values are stored in a vector y, your function call would be p = polyfit(x,y,1). Use your fitted line data to calculate the camera gain g, the read noise variance (measured in photo-electrons), and the ADC noise variance (measured in gray levels).

Answer: As shown in Figure 5. and Figure 6., fitting curves are plotted using function polyfit. By the function we can obtain the slop of each fitting curves, which are the gain of different sensitivity settings.

$$\text{Low Sensitivity Gain} = 4.0963e-05$$

$$\text{High Sensitivity Gain} = 3.7525e-04$$

We can choose two points respectively from Low Sensitivity Setting and High Sensitivity Setting and then get two sets of parameters, $\sigma_i$, $\mu_i$ and $g$, of which we know the exact values. According to the equation

$$\sigma_i^2 = \mu_i \cdot g + \sigma_{read}^2 \cdot g^2 + \sigma_{ADC}^2 \tag{1}$$

, we set up a system of linear equations with two unknowns, $\sigma_{read}$ and $\sigma_{ADC}$ and then solve it.

$$\sigma_L^2 = \mu_L \cdot g_L + \sigma_{read}^2 \cdot g_L^2 + \sigma_{ADC}^2 \tag{2}$$

$$\sigma_H^2 = \mu_H \cdot g_H + \sigma_{read}^2 \cdot g_H^2 + \sigma_{ADC}^2 \tag{3}$$

Now I chose two points from the each curves,

$$\left(\sigma_L^2, g_L, \mu_L\right) = \left(1.2742\text{e-}06, 4.0963e-05, 0.03\right)$$

$$\left(\sigma_H^2, g_G, \mu_H\right) = \left(8.6699\text{e-}05, 3.7525e-04, 0.15\right)$$

According to equation (2) and (3), we can get

$$\sigma_{read} = 14.7733$$

$$\sigma_{ADC} = 5.6648e\text{-}4i$$

Question 7. Plot the SNR as a function of mean pixel value. What is your interpretation of this plot? How does it relate to the three types of noise from equations (4) and (5)? What is the maximum SNR that can be achieved by the Tegra camera?

Answer: Figure 9. and Figure 10. are SNR plots of Low Sensitivity Setting experiment and High Sensitivity Setting experiment respectively. From the two figures we can find out SNR increases as u increase. Low level values pixels are more likely to be influenced than high level values pixels, because their low value make themselves compatible to noise. As u increases, the impact that noise has on u becomes less and less. In result of that, the fitting curve goes upward. The Tegra camera can achieve 33.82dB and 75.52dB in the low sensitivity setting and in the high sensitivity setting respectively.

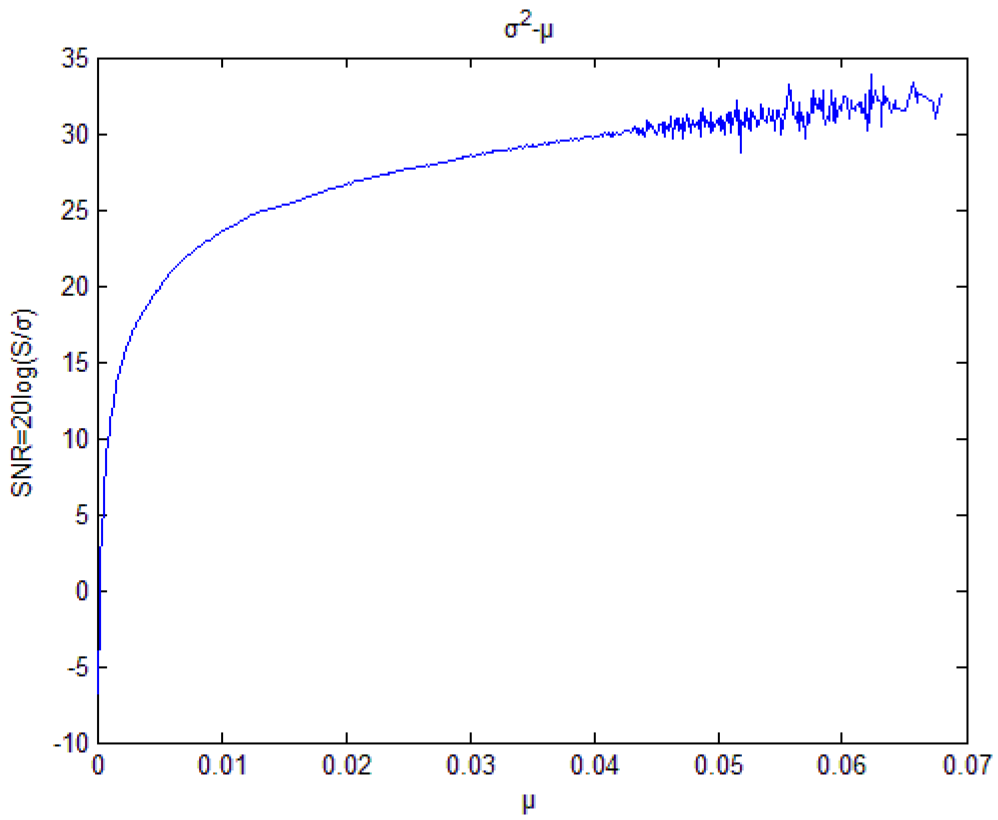

Figure 11. SNR curve of Low Sensitivity Setting

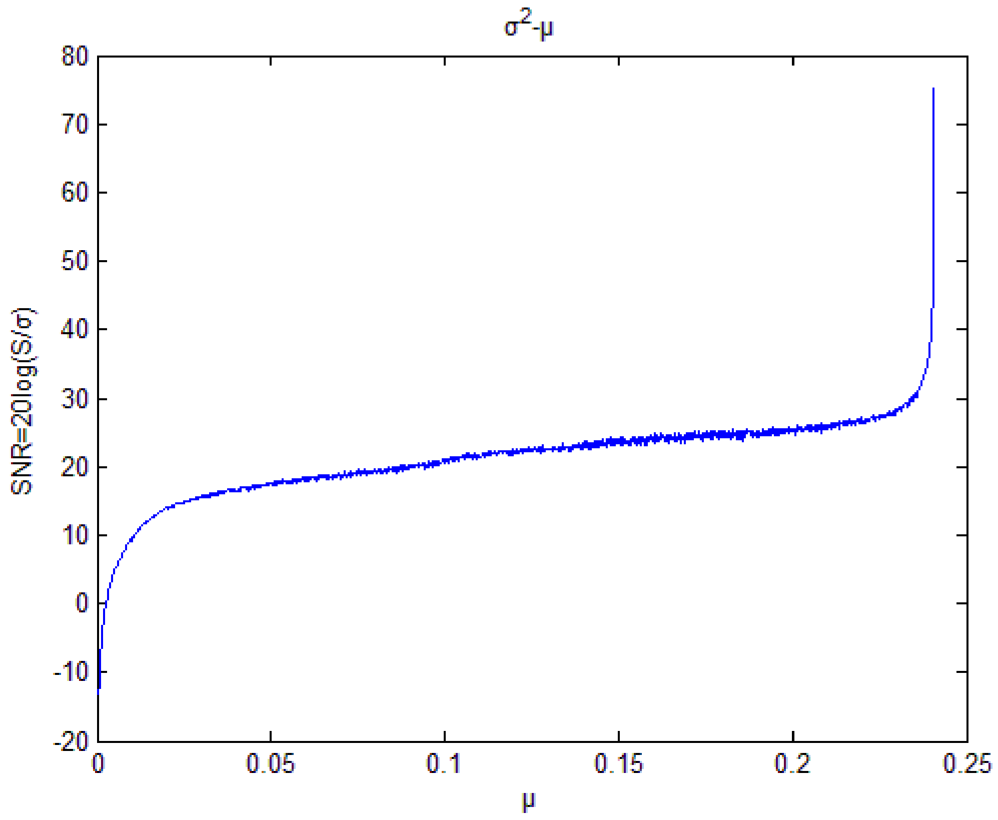

Figure 12. SNR curve of High Sensitivity Setting

Android Studio Codes

```
public void captureRAW() {
            //TODO:hw2
            //Set different capture numbers, but no larger than the buffer size (defaults to 51)
            for (int i = 0; i < 53; i++) {//These is a bug of the number setting. We have to set 53 to capture 50
pictures.
                  //sleep the system clock for 200 ms
                  SystemClock.sleep(300);
                  if (mCaptureSession != null) {
                        Log.e(TAG, "setting up raw capture call backs");
                        try {
                              CaptureRequest.Builder requester =

mCameraDevice.createCaptureRequest(mCameraDevice.TEMPLATE_STILL_CAPTURE);
                              requester.addTarget(rawCaptureBuffer.getSurface());
                              requester.setTag(mRequestCounter.getAndIncrement());

                              //TODO:hw2
                              //Set different gains
                              //Gains = k* Sensitivity (true);
                              //TODO: getting range of sensitivities supported using
SENSOR_INFO_SENSITIVITY_RANGE:
                              Range<Integer> sstt =
characteristics.get(CameraCharacteristics.SENSOR_INFO_SENSITIVITY_RANGE);

                              //TODO:getting lower and higher sensitivity
                              Integer lower_sstt = sstt.getLower();
                              Integer higher_sstt = sstt.getUpper();
                              Log.e(TAG, "lowersstt " + lower_sstt);
                              Log.e(TAG, "highersstt " + higher_sstt);
                              //TODO:Change the capture request's sensitivity
                              requester.set(CaptureRequest.CONTROL_AE_MODE,
CaptureRequest.CONTROL_AE_MODE_OFF);
                              requester.set(CaptureRequest.SENSOR_SENSITIVITY,lower_sstt);
                              try {
                                    // This handler can be null because we aren't actually attaching any callback
                                    rawChars.add(getCharacteristics());
                                    mCaptureSession.capture(requester.build(), mCaptureCallback,
mBackgroundHandler);
                              } catch (CameraAccessException ex) {
                                    Log.e(TAG, "Failed to file actual capture request", ex);
                              }
                        } catch (CameraAccessException ex) {
                              Log.e(TAG, "Failed to build actual capture request", ex);
```

```
                }
            } else {
                Log.e(TAG, "User attempted to perform a capture outside the session");
            }
        }
    }
```

Matlab Codes

```
%/////////Codes of Question 2 go as follows.//////////%
clc;
clear;
I=cell(1,50);
A=[ 500,600;
        800,1200;
        1200,1800
        1500,2400];%the 4 positions of points to be shown and image size is 1944*2592
B=zeros(50,4);%store values and 4 columns for 4 points
for i=1:50%load 50 images at a time
        %Image_lowSen or Image_highSen
        imgname=strcat('D:\Courses Files\Introduction to Computational
Photography\HW2\pattern_High2\',num2str(i),'.dng');
        I{i}=imread(imgname);
end

for i=1:4%store the values of each points
        for j=1:50
                B(j,i)=I{j}(A(i,1),A(i,2));
        end
end
figure(1);
suptitle('values of different pixels(HighSen)');
% suptitle('values of different pixels(LowSen)');
subplot(2,2,1);
hist(B(:,1));
xlabel('Value');
ylabel('Quantity');

subplot(2,2,2);
hist(B(:,2));
xlabel('Value');
ylabel('Quantity');

subplot(2,2,3);
hist(B(:,3));
xlabel('Value');
ylabel('Quantity');

subplot(2,2,4);
hist(B(:,4));
xlabel('Value');
ylabel('Quantity');
```

```
%/////////Codes of Question 3 go as follows.//////////%
clc;
clear;
num=49;
I=cell(1,num);

for i=1:num%load 50 images at a time
        %Image_lowSen or Image_highSen
        imgname=strcat('D:\Courses Files\Introduction to Computational
Photography\HW2\pattern_High2\',num2str(i),'.dng');
        I{i}=im2double(imcrop(imread(imgname),[1350,830,870,1000]));% find the interesting area--the
pattern area
end

[H,W]=size(I{1});
A=zeros(H,W);%mean matrix
B=zeros(H,W);%varience matrix

for i=1:H
        for j=1:W
                C=0;
                for k=1:num
                        C=C+I{k}(i,j);
                end
                A(i,j)=C/num;%mean
        end
end

for i=1:H
        for j=1:W
                C=0;
                for k=1:num
                        C=C+(I{k}(i,j)-A(i,j))^2;
                end
                B(i,j)=C/num;%mean
        end
end
figure(1);
subplot(2,2,1);
imagesc(A);
title('mean image');
C=reshape(A,1,H*W);%change dimension of mean metrix
subplot(2,2,2);
imagesc(B);
```

```
title('variance image');
subplot(2,2,3);
imagesc(I{1});
title('original image');
subplot(2,2,4);
imagesc(I{2});
title('original image');

%/////////Codes of Question 4 and 5 go as follows.//////////%
D=reshape(B,1,H*W);%change the dimension
figure(2);
c=find(C==min(C));%shift min u to zero point
C=C-C(c(1));
plot(C,D,'.','color','red');
axis([0,max(max(C)),0, max(max(D))]);
xlabel('μ');
ylabel('σ^2');
title('σ^2-μ(without averaging σ^2 for same μ)');

[E,ind]=sort(C);
Eround=roundn(E,-4);
j=1;
k=1;
a=0;
n=0;
i=1;
while i<(H*W)%averaging varience of same u
       n=length(find(Eround==Eround(i)));
       for j=i:(n+i-1)
              a=a+D(ind(j));
        end
       F(k)=Eround(i);
       G(k)=a/n;
       k=k+1;
       i=i+n;
        a=0;
end

figure(3);
plot(F,G,'.','color','red');
p = polyfit(F,G,1);
hold on
x=0:(max(max(F)))/1000:max(max(F));
y=p(1)*x+p(2);
plot(x,y);
```

```
axis([0,max(max(F)),0,max(max(D))]);
title('σ^2-μ')
legend('data','fitting curve','location','northeast')
xlabel('μ');
ylabel('σ^2');

%/////////Codes of Question 7 go as follows.//////////%
figure(4);
b=20*log10(F./(G).^0.5);
plot(F,b);
xlabel('μ');
ylabel('SNR=20log(S/σ)');
title('σ^2-μ');
```