02 Detector and noise

[Version 21.2.2020]

Nanophotonics and Metrology Laboratory

Institute of Microengineering,

School of Engineering,

École Polytechnique Fédéderale de Lausanne,

EPFL-STI-IMT-NAM

ELG 237

Station 11

CH-1015 Lausanne

Web : http://nam.epfl.ch

Contact person: Dr. Toralf SCHARF,
Phone: +41 21 6954286
E-mail : toralf.scharf@epfl.ch

Contents

1 Objective and overview

The practical work should introduce the following subjects to you:

- Evaluate the properties of the noise of the camera and noise reduction by averaging
- Increase the dynamic range of the camera through multiple exposures (HDR –high dynamic range imaging)

To get this done you need to read the reference document provided.

2 Safety Issues

In this experiments laser sources and low power electrical equipment are used. The laser is of class II. Class II: low-power visible lasers that emit a radiant power below 1 mW. The concept is that the human aversion reaction to bright light will protect a person. Only limited controls are specified. (http://www.osha.gov, Laser Hazards)
The laser is safe because the blink reflex of the eye will limit the exposure to less than 0.25 seconds. It only applies to visible-light lasers (400–700 nm). Intentional suppression of the blink reflex could lead to eye injury. In our experiment the laser sources are collimated and should be handled with care. A strongly divergent beam will not be focussed on the eyes retina and represents often no danger. Collimated beams will lead to small focus spots onto the retina special care is needed. ***Do not stare into a collimated beam!***
The electrical equipment used in the experiments is based on USB power (5V, 0.5 A, 2.5 W) and not subjected to any particular security issues. Nevertheless you should **not produce short circuits** on the printed circuit board (PCB) or to the computers USB connection to avoid damage to the material. Make proper use of screwdrivers. Do not force any mechanical parts.

3 Background

Camera

In our experiment we use a pixelated camera. The pixels are arranged in a regular array that is produced at very high precision. The camera chip has 1600 x 1200 pixels and is 4.536mm x 3.416mm wide and large respectively. Pixel pitch is 2.835 micron in both directions (square). We can safely assume that there is no deviation of pixel position.

Noise sources

Each camera has a limit of sensitivity, the light level needed to create a measurable signal. This minimum light level is influenced by several parameters. Let us consider one single pixel. Photons that hit the pixel will be converted into electrons at a certain rate (or probability). In conventional system this rate (or probability) is a number smaller than 1. The conversion rate defines the efficiency of the detector and is given by quantum effects.
The detector operates at a given temperature, which causes the creation of electrons by thermal fluctuations. This effect is called **thermal noise or dark noise**. As higher the temperature is as larger the dark noise becomes. These thermal electrons are equivalent to a

current which is called **dark current** (or noise) because it exists even when no light reaches the pixel. To measure a signal, the number of electrons created by the photon flux should be larger than the number of electrons created by thermal noise. We have found a first parameter that limits our measurement – the thermal noise. You might think of a certain threshold that is needed to get useful signal. In many situations it is desirable to start signal count at zero which can be reached by subtracting a certain signal level.

NOTE: The camera applies an automatic offset correction to subtract electronically the dark current. This correction does not eliminate the dark current but just makes a cosmetic correction to the measurement and leads to undesired effects. The measurement is influenced by this and sometimes even not possible.

Considering our signal conversion scheme, the next step is that electrons are collected, moved to a register and amplified to deliver a measurable quantity. During these operations, some electrons are lost and the signal is worsened. The magnitude of these effects is difficult to quantify.
The **amplification (gain in of the camera setting)** multiplies the electrons with the help of an electronic circuit and is one of the most important sources of noise in the system. This is because we would like to have high sensitivity, which means high amplification. But high amplification will also introduce high levels of electronic noise. In most measurements, it is the electronic noise of the amplifier that limits the detection sensitivity. We will be limited by the electronic noise.

REMEMBER: A high GAIN (electronic amplification) results in a high electronic noise level.

Figure 1 The same image taken at different amplification values: Left: low gain and right high gain. The inset shows the effect of electronic noise on the image quality.

Finally the electronic signal is digitized. In our case we have an 8 bit camera which delivers 256 grey levels (or counts) per pixel. **Digitalization** introduces a noise of about 1 count because if an error in counting is done it will cause a value false by a minimum of one count. Our best measurement without further treatment would have a signal to noise ratio of 256.

In todays cameras all information is saved in compressed format for instance as JPEG. This represents a loss of quality and information and is also a source of noise, which we call here **compression noise**! It often leads to artifacts and is especially annoying when the image is taken as a measurement. It is therefore important to save the date in its highest quality format to avoid such problems.

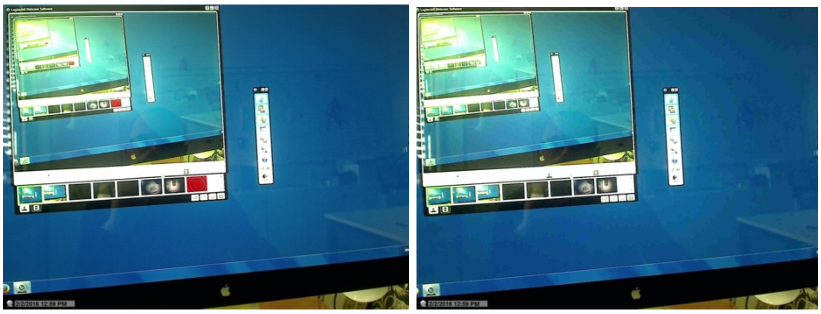
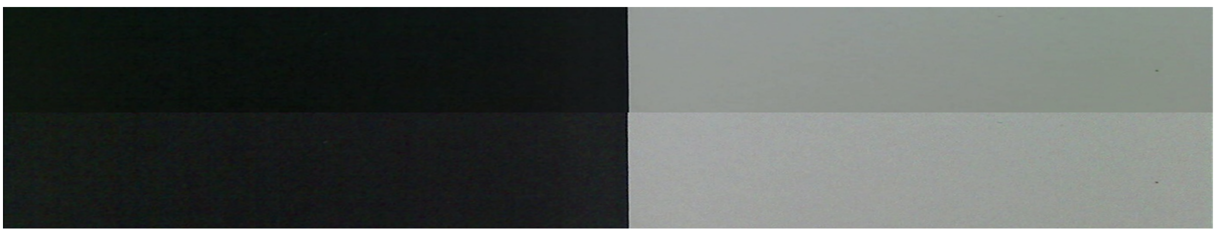


Figure 2. Two images recorded with different JPEG compression ratios. Left: low compression Right: high compression. The compression leads to zones of equal intensities, which is an artifact.

There is an additional source of noise that comes from the statistical properties of light itself. If one considers the model where light is represented as a stream of photons we have the same problem as in digitalization. We might count one photon wrong, and noise related to this quantization of energy is called **"shot" noise**. If there are many photons, **"shot" noise** has a very small effect because the relation of 1 false to many photons is small. If the signal is weak, only few photons arrive on the detector and a false count has a relative large effect. In this case the shot noise might be the limiting factor. We will only consider the effects of electronic noise here.

Note: An ideal optical detection system is shot noise limited. In our case the detection system is limited by the electronic noise of the amplifier.

Averaging can reduce noise. Averaging can be done in time or space. Averaging over time is done by changing the integration time (**exposure**) of the camera. The longer the integration time, the less noise will be found in repeated measurements.
Average in space is only possible when the scene has certain symmetry. For instance, if we take an image of lines! In such a case one can average over a large number of pixels and reduce the noise.



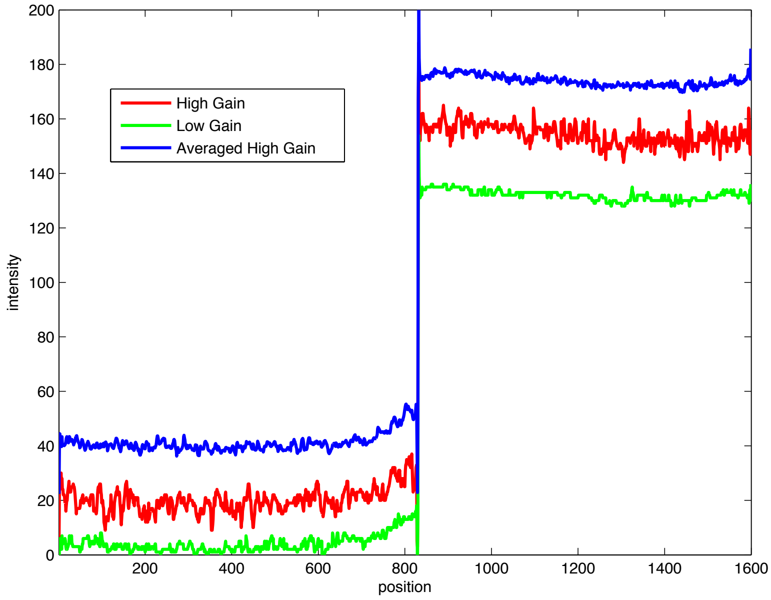


Figure 3 Looking at an image with an intensity step allows to quantify the amplification noise. On top we have crops of the original images with the low gain on top and the high gain below. Three example curves are shown in the lower part of the Figure: Low Gain (green), High Gain (red) and Average High Gain (blue). One clearly sees how averaging over 20 lines reduced noise (Blue line versa red line). Note: the "overshooting" at the edge which is caused by the camera edge enhancement algorithm to artificially sharpen the image. This is an artifact caused by the software signal treatments.

In standard digital photo cameras the amplification noise plays a role when the sensitivity of the camera is changed. The sensitivity is defined with the ISO value and represents the level of electronic amplification. Standard settings are ISO 200, which is optimized for having low amplification noise. Doubling the ISO value doubles the sensitivity of the detector and increases the noise because a higher amplification is used. There are only few cameras today that can take reasonable quality images at ISO 3200, which is 16 times more sensitive than the standard setting. The main problem is the size of the chip. Because the noise is approximately the same for all pixel sizes but a smaller area collects less photons for the same photon flux. Hence the signal to noise ratio is not as good.

In the course we will inspect the noise properties of the camera by taking images with intensity steps and making spatial averages.

Dynamic range

Our camera has only 256 grey levels, a value which limits its use very much. In todays technology combining hardware and software interaction the dynamic range can be artificially increased. It should be mentioned that if every parameters are held fixed (time, F# number, focal lengths etc., chip size, good optics, amplification…) the overall performance is comparable for all cameras. But one can use tricks to increase the performance. Imagine you take not only one image but several images at different exposure condition. Effectively that multiplies the exposure time by the number of images taken. But the single exposure time is often short and this is not disturbing the use of the camera. Two images are taken, one is overexposed, and another is underexposed. In the overexposed image, the visibility of dark areas will be good – in the underexposed image, bright areas will reveal more details.

Software can identify the area to consider and an image with a high dynamic range can be generated. The standard today is an enhancement value of 400% (4 times increase of dynamic range). In our case a 256 (8 bit) camera would operate as if it would have 1024 grey levels (10 bit). These techniques are called **HDR (High dynamic range)** photography and can be found today in nearly all nomadic devices (cell phones, digital photocameras). Obviously the generation of the image gets more complicated for color images because the color rendering has to be matched too.

Figure 4. Examples of the effect of HDR in a real scene (iPhone 4). The HDR technique allows to recover details in the shadow and in the overexposed areas of pictures

In the course we will show the basic operation of HDR by generating high dynamic range black and white images with Matlab.

4 Setup and equipment

4.1 Materials

- The camera (1600x1200 pixels, colour, pixel size 2.835 um) C600 from Logitech
- Three different light sources (Halogen, LED, Laserdiode) USB driven
- Sheet polarizers for intensity regulation
- Objective lens for the Logitech C600
- Mechanical setups

First we proceed to check the camera basic adjustments. To do so a simple setup has to be mounted. Start with the breadboard and mount the translation stage.

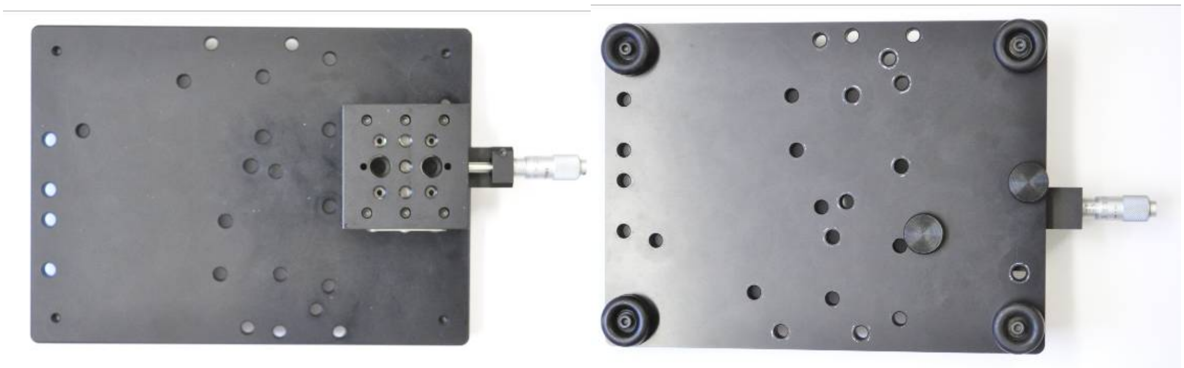


Figure 5: Breadboard with translation stage

The stage is fixed with screws arranged unsymmetric (see the right picture). Continue by mounting the adapter plate and the intermediate piece as below.

Figure 6: Adapter plate and intermediate fixing have to be screw together.

The cameras PCB (printed circuit board) special holder is mounted as shown below.

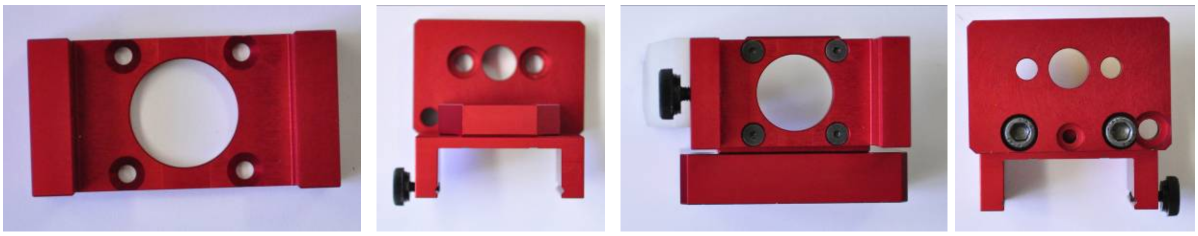


Figure 7. Camera holder (left) mounted on the adapter plate with the intermediate piece. Right: as seen from below.

The assembly has to be fixed on the translation stage before the camera PCB is put into place.

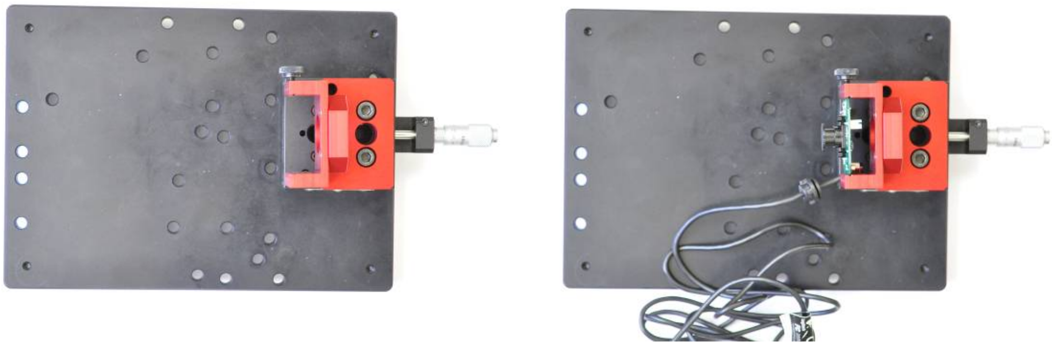


Figure 8: Camera mounts on the translation stage with and without camera in its holder.

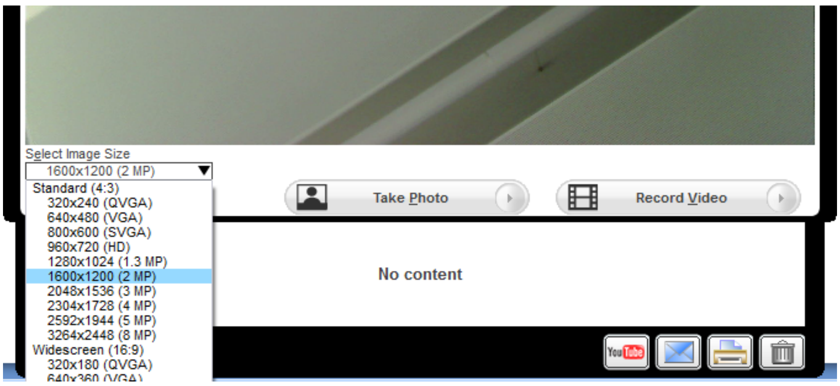
Figure 9: Camera PCB in the mount ready for shooting.

4.2 Noise evaluation

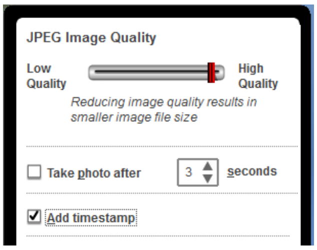
For the noise evaluation a uniform illumination of the sensor is used and measurements are done at two different conditions: without light and with a light level sufficient to **nearly** saturate the camera signal.

Image quality setting.

Check the image quality settings in the lower part of the measurement window.



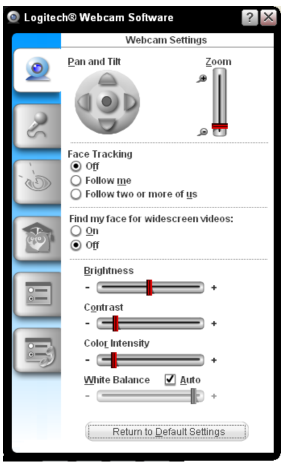
- Set the image size to 1600x1200 (2MP)
- In **"Take Photo"** choose the highest quality JPEG setting.
- In **"Take Photo"** click the option **"Timestamp"** to have the date to appear on all your photos.



Please switch off first the camera control LED. Go to further options under advanced settings  and switch of the LED under LED control.

Dark Noise measurement without light

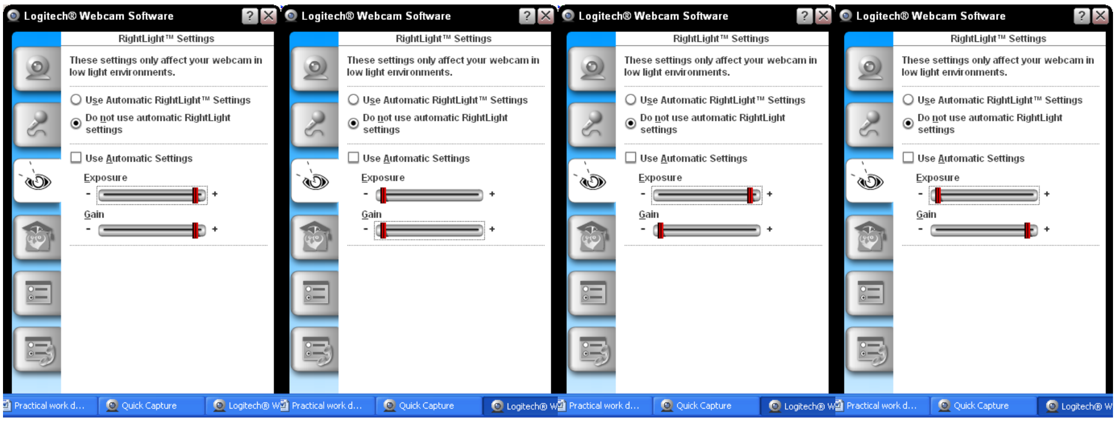
- **Block the light** (leave the objective, cover the lens, use the black tissue)
- Set the Brightness-Contrast-Colour conditions on the default settings as shown below by pressing **"Return to Default Settings"**



- Go to the Right light settings menu
- Switch off "Use automatic settings"

The **Exposure** sets the duration of the exposure. It is the shutter time known from photography. **Gain** sets the electronic amplification. A high gain corresponds to a large amplification and a high sensitivity at a given shutter speed (exposure). To see the effect of the gain and exposure on the noise level take pictures under different settings.

- Take a picture for the four different conditions as shown below



Now you have recorded all images without incident light. These images should be black (minimum level). All other values are due to the dark noise for different exposure time (exposure) and different amplifications (gain).

Measurements for nearly saturated camera

To have a uniform illumination on the detector, unscrew the objective and place the camera in front a uniformly illuminated surface. This might be a sheet of paper, the black cloth or something else. You can change the light level by changing the distance between the detector/camera and the test surface.

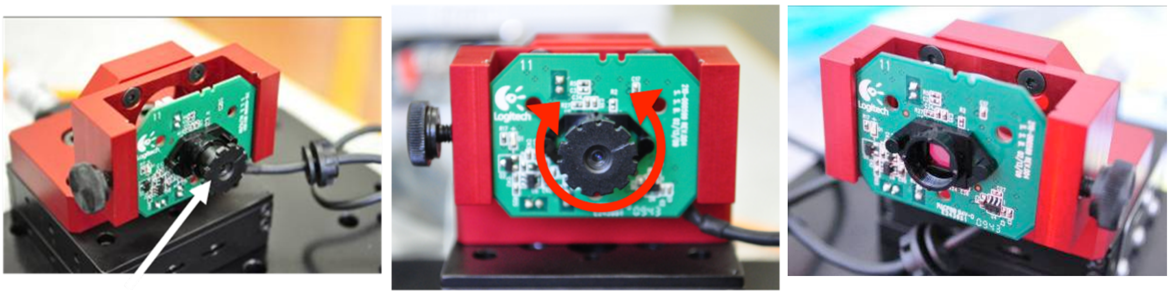


Figure 10. Camera PCB with objective. You can unscrew the objective to get access to the detector. On the right is the arrangement that should be used.

To effectively illuminate the camera chip use it without objective. Illuminate as uniform as possible by holding it in front of a uniformly illuminated surface (white grey or black). Please take care to manipulate the camera with greatest care.

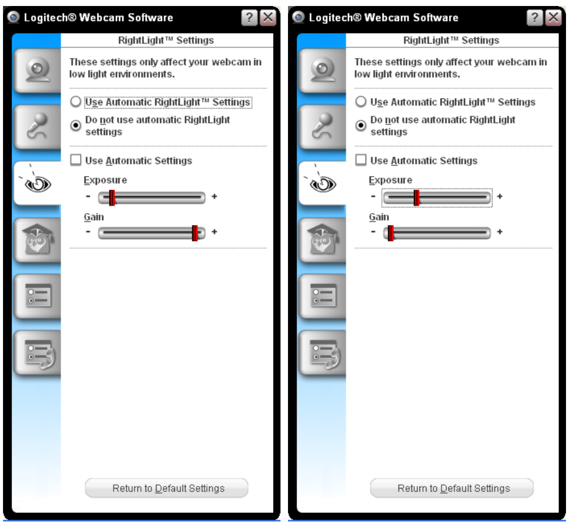
Make the following two measurements

- Select high gain and choose exposure conditions to just NOT saturate the detector
- Select low gain and choose exposure conditions to just NOT saturate the detector

To show the quality of the intensity adjustment you can plot a line plot with the Matlab script that is given below. (filename: center_line_plot, based on 1200 x 1600 pixels)

```
% Read image 1
I_1 = imread('Picture_8.jpg');
% select a channel
Red_1 = I_1(:,:,1);
Green_1 = I_1(:,:,2);
Blue_1 = I_1(:,:,3);
% format conversion
Red_1 = double(Red_1);
Green_1 = double(Green_1);
Blue_1 = double(Blue_1);
% select a line of interest ROI (1 x 1600), here center line on position
600
CenterLine_Red = Red_1(600:600,1:1599);
CenterLine_Green = Green_1(600:600,1:1599);
CenterLine_Blue = Blue_1(600:600,1:1599);
%plot a single line
figure
plot(CenterLine_Red,'LineWidth',2,'Color','r');
xlabel('position [pxl]')
ylabel('signal [counts]')
figure
plot(CenterLine_Green,'LineWidth',2,'Color','g');
xlabel('position [pxl]')
ylabel('signal [counts]')
figure
plot(CenterLine_Blue,'LineWidth',2,'Color','b');
xlabel('position [pxl]')
ylabel('signal [counts]')
```

You might find position as below but this depends on setups.



NOTE: It is important to have uniform illumination conditions! If you would have a regular intensity pattern on your detector the measurement would be useless. AVOID saturation.

The images below give you an idea how the image should look like. The intensity variation over the detector should be as small as possible.

Figure 11. Two cases of intensity variation over the detector surface. The left is badly adjusted while the right show a high uniformity.

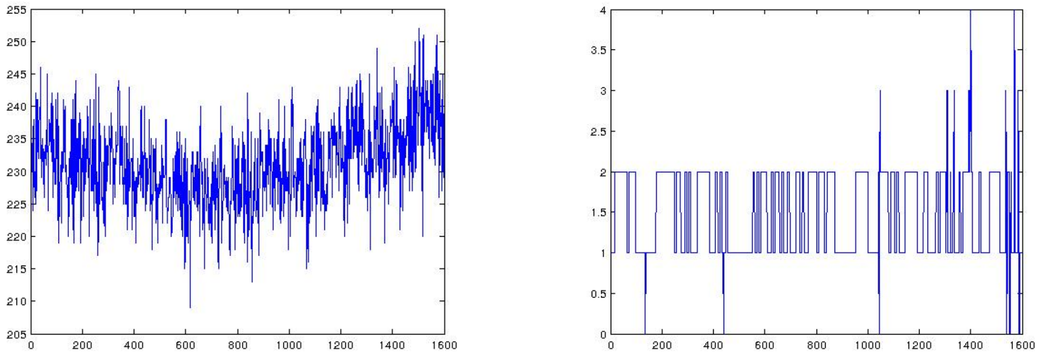


Figure 12. Typical plots of the centreline for a very well adjusted intensity profile at high intensity (Left) and for low intensity (Right) at high gain.

Evaluation

The noise level in the image can be qualified by using statistics. A good measure of the noise properties is the mean value and the standard deviation. This can be calculated by considering each pixel as a single measurement and by computing the distribution of these values. For each image one should find the **mean of counts in the image and the standard deviation** using Matlab. Please refer to the attached examples.
The particularity here is that the pictures have three channels, one for each color and need to be converted to different formats to perform the sum operations. Please find the mean and standard deviation for all channels (colours) separately.

(filename: Noise_mean_std.m)

```
% Read image 1
I_1 = imread('Picture 8 high gain.jpg');
% select the red channel
Red_1 = I_1(:,:,1);

% format conversion
Red_1 = double(Red_1);
% mean intensity and standard deviation is calculated
```

```
MEAN_red = mean(mean(Red_1))
STD_red = mean(std(Red_1))

% select the green channel
Green_1 = I_1(:,:,2);
% format conversion
Green_1 = double(Green_1);
% mean intensity
MEAN_green =mean(mean(Green_1))
STD_green =mean(std(Green_1))

% select the blue channel
Blue_1 = I_1(:,:,3);
% format conversion
Blue_1 = double(Blue_1);
% mean intensity
MEAN_blue =mean(mean(Blue_1))
STD_blue =mean(std(Blue_1))
```

To be done for the report: Please provide a table with the different means and standard deviations including parameters for each image and each color, for blue, green and red. Give example line plots with a width of one line taken in the middle of the image for the measurement. (select **one channel** – red, green or blue, six plots (4 dark noise, 2 nearly saturated).

		RED		GREEN		BLUE	
		Mean	STD	Mean	STD	Mean	STD
Dark	Lowest Gain – Short exposure						
	Highest Gain – Short exposure						
	Lowest Gain – Long exposure						
	Highest Gain – Long exposure						
Bright	Lowest Gain – exposure adapted						
	Highest Gain – exposure adapted						

4.3 Noise reduction by averaging

Noise reduction is effectively done by averaging independent images or lines. This can be easily done in Matlab by computing the mean over an image area or by adding several images. We want to demonstrate the power of averaging by comparing the noise levels for different gain settings before and after averaging.
Mount the objective to be able to take images! Project an edge adjust focusing. **Defocussing helps to avoid aliasing effects!**
Take an image of the edge for a pattern as shown below.

Figure 13. Intensity step to show and evaluate the effect of averaging on the noise level.

- Align the camera with the edge to a vertical edge direction (because we will average it vertically)
- Take images for **highest and lowest gain**. Exposure conditions will be different. Adjust the exposure conditions to prevent saturation.
- Create line plots with averaging and for the situation where several lines are averaged. As a criteria to measure the noise we will use the standard deviation of a certain area in the image

To be more precise, images should not be saturated (constant values of 256 counts) at the high intensity area or underexposed (0 to 2 counts constant) in the black area. Play around to find a good adjustment.
For evaluation the dark and bright areas needs to be considered separately. Try to set the edge in the middle of the screen and use the script **noise_line_ROI_red_step** to evaluate the numbers for the dark and bright zone. In our example the dark area is left and the bright area is right.

Example Matlab script:
(filename: noise_line_ROI_red_step)

```
% Read image 1
I_1 = imread('Picture 9 low gain.jpg');
% select a channel (here red)
Red_1 = I_1(:,:,1);
% format conversion
Red_1 = double(Red_1);
% plot the a line plot from x=(1 x 1600) at the center line at y = 600
CenterLine = Red_1(600:600,1:1600);
figure
plot(CenterLine,'LineWidth',2);
xlabel('position [pxl]')
ylabel('signal [counts]')
% select a Region of Interst RIO with  values between 1 and 1600 in x
direction, and the center line in y direction (value 600), choose the left
area of the image: here from 10 to 400 in x direction
CenterLine_left = Red_1(600:600,10:400);
%average over several lines
% select a region of interest (ROI) with several number of lines N in y
direction (horizontal lines are averaged), in the example below x is on the
left from 10:400 and the lines that are averaged are 590 tp 610 hence 20
lines, all calculations will be done for this region of interest RIO
```

```
N = 1;
ROI_Red_1_left  = Red_1((600-N/2):(600+N/2),10:400);
%caculate the mean of the RIO over several rows
N_Avg_Red_1_left = mean (ROI_Red_1_left);
%plot the averaged line
figure
plot(10:400, N_Avg_Red_1_left,10:400, CenterLine_left,'LineWidth',2);
xlabel('position [pxl]')
ylabel('signal [counts]')
MEAN_red_left = mean(mean(ROI_Red_1_left))
STD_red_left = mean(std(ROI_Red_1_left))
% repeat for the right side
CenterLine_right = Red_1(600:600,1100:1580);
ROI_Red_1_right  = Red_1((600-N/2):(600+N/2),1100:1580);
%caculate the mean of the RIO over several rows
N_Avg_Red_1_right = mean (ROI_Red_1_right);
%plot the averaged line
figure
plot(1100:1580, N_Avg_Red_1_right,1100:1580, CenterLine_right,'LineWidth',2
);
xlabel('position [pxl]')
ylabel('signal [counts]')
MEAN_red_right = mean(N_Avg_Red_1_right)
STD_red_right = std(N_Avg_Red_1_right)
```

A typical series of result for one image is shown in the images below.

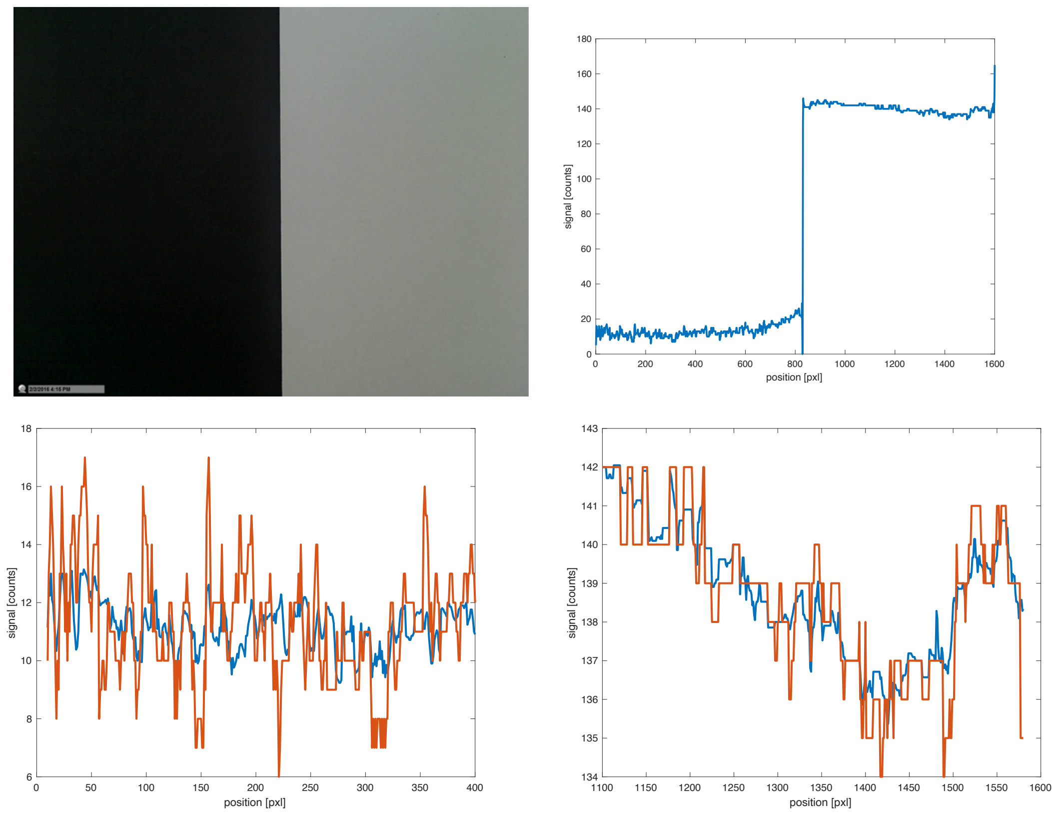


Figure 14 Sample image and line plots for one exposure condition. X axis is pixel position and y axis are counts. Middle Figure gives a line plot at y position 600. Bottom Left is the left side measurement raw (green) and averaged (blue). Bottom Right is the right side measurement raw (green) and averaged (blue). 20 averages are done

EPFL

The figure above shows clearly the improvement of signal quality by averaging. To compare the noise under different conditions one can choose a region of interest with **uniform** illumination in the image (i.e. a flat signal level) and compare the noise for lowest and highest gain. **A typical value would be below 20 averages**. The script is providing this automatically be delivering values

MEAN_red_left = 11.1907
STD_red_left = 1.3338
MEAN_red_right = 138.8250
STD_red_right = 0.7229

in the Matlab command window.
We do not have a lot of possibilities with our camera but we can compare the noise increase induced by high gain to the noise level at low gain that should be smaller. The idea is to average several lines the high gain signal, calculate the standard deviation and compare it with the standard deviation of a single signal at low gain. The number of averages needed to equal the standard deviations can be used as a measure for the amplification noise.

To do so make the following

- Check if the region of interest ROI falls in a zone of flat intensity. If not adapt it by changing the x value (actually set to 10-400 and 1100-1580)
- Set the averaging value N to 1 in the script **noise_line_ROI_red_step**
- Treat the image with **lowest gain** first, and write down the values for a single line!
- Treat the image with **highest gain**; and write down the values for a single line!
- Treat the image with highest gain for different N values to **find the N value that gives the same STD** as the lowest gain single line measurement!
- **How many averages in the highest gain image are necessary to get the noise level of the lowest gain image?**

NOTE: Measurement with the step image simulate a real situation where one has high and low intensity regions in the image. You should choose a projection image with a contrast that allows you to have a high contrast, low intensity values on one side but NO SATURATION. Take care that you avoid aliasing effects caused be the pixelation the screen and camera!

To be done for the report: Show one set of example images example images like in Figure 14 for the high gain! (one channel – red). Make a table with the value for single line for low an high gain. Find out how many lines from the high gain image have to be averaged to compensate the noise added by the gain!

	Low gain N=1	High gain N=1	High Gain for N=??
MEAN_red_left			
STD_red_left			
MEAN_red_right			
STD_red_right			

4.4 High dynamic range imaging

The dynamic range of cameras is limited. An 8 bit camera has 2^8=256 grey levels. For measurements this is very few. Note that camera producers count slightly different. They say that they have 8 bit times 3 colour channels and claim a depth of 24 bit. For black and white measurements that will not work of course. A way to increase the dynamics is controlled over- and under exposure and assembling of images with software. This method is called HDR (High dynamic range) photography. For colour images, HDR photography is rather complicated because the colour rendering has to match the colour impression. We will only make black and white HDR images to proof the concept.

Evaluation of maximum dynamic range (gain)

To start with We want to determine the factor of sensitivity increase of our camera which can be obtained between the highest and the lowest gain. For the Logitech C600 camera the gain can be set in 100 steps and the exposure time has 15 different values.
To do the evaluation, we need to **match a known value of intensity for different conditions and extrapolate**. One takes first an image of a scene with balanced exposure conditions in which one find **large uniform areas**. It is important to have very good uniformity, which can be evaluated by considering the standard deviation of all pixel counts in a given area. We work with a fixed exposure time and change the gain value to the two extremes. The picture is taken at the **highest gain** after adjustment of the exposure time to have a high signal level, but **avoid saturation** of the uniform area. Then take a second image of the same scene with the same exposure time but the **lowest** gain. Check that there is some signal in the region of interest, because only then we can match numbers.

NOTE: Pixel values in the dark area should not be too low to avoid artifact due to the automated base line correction of the camera software. Adjust the exposure and gain to have above 20 counts in the dark area.

Figure 15: Example for the gain factor evaluation. On the left, the image was taken with the highest gain. On the right, the image was taken with the lowest gain. The region of interest is indicated by a small rectangle. Their level is as uniform as possible with saturation and residual signal at low gain.

In the example above the region of interest was set at x=240 and Δx=+40, y = 200 and Δy=+60.

To be more precise you can use the courser tool to read values in a Figure! 
Click the courser tool in the toolbar of the figure and than move the mouse into the figure to find values!

The Matlab script below gives values for an averaged intensity over this area and calculates the signals standard deviation.

(filename: readimage_gain)

```
% Read an image
I = imread('Picture 52.jpg');
% display an image
figure
imagesc(I);
% convert to B&W, we take the red channel here
I = mean(I,3);
figure
imagesc(I,[0 255]); colormap(gray);
% select a region of interest ROI
%starting point x
x = 240;
%starting point y
y = 200;
%width w in x direction
w = 40;
%height h in y direction
h = 60;
%region select
I = I(y:y+h,x:x+w);
%plot region of interest
figure
imagesc(I,[0 255]); colormap(gray);
% mean intensity
MEAN_ROI = mean(mean(I))
STD_ROI = mean(std(I))
```

In the measurement the standard deviation of the ROI has to be smaller than 4 counts to assure validity of the basic assumptions. In our example we find

Image highest gain:	MEAN_ROI = 185.4900	STD_ROI = 1.3874
Image lowest gain:	MEAN_ROI = 39.2401	STD_ROI = 0.9232

The dynamic range factor with gain is therefore G = 185.49/39.24 = 4.83.

This will be different for each camera!!

To be done for the report: Show the images with the ROI. Evaluate the dynamic range increase obtained by the changing of the gain from low to high and **c**alculate the dynamic range factor G with its error! (use the standard deviations as errors for the input values).

High dynamic range imaging example (exposure)

Our approach here is simple. We take one image of a **contrasted** scene with saturated areas and another where the same areas are not saturated. This can be done by reducing the exposure time. Then we combine the two images

- Take two images at different exposure conditions of a contrasted scene
- One image should have overexposed areas – adapt the exposure manually by setting the exposure time to different values
- The other image should have information in the former overexposed area and **no saturation**
- Combine the two images using Matlab by replacing the pixel that where overexposed in the first image with the pixel values of the second image.

An example is shown below

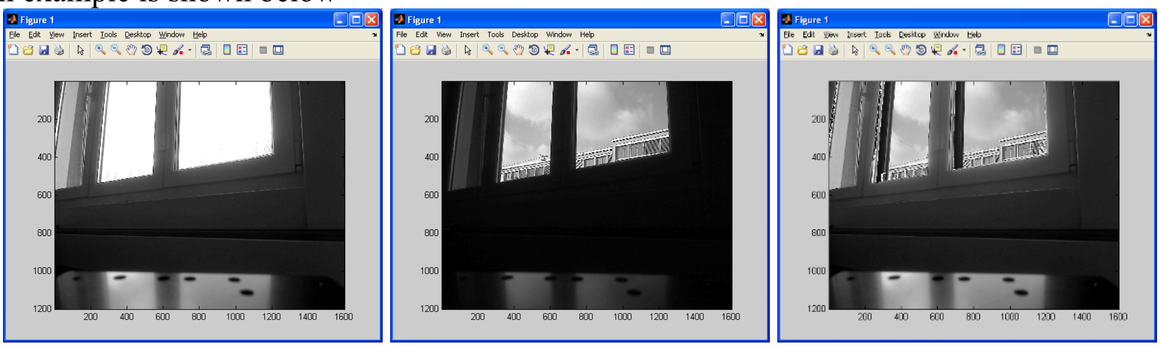


Image one Image two Combined image

Figure 16. High dynamic range imaging example where the overexposed pixels in image one are replaced by the pixels taken from an underexposed image (image two) and replaced to form a combined image that has much higher information content.

The Matlab routine below will help you to do this: (filename: readimage_HDR)

```
% Read two images I1 and I2 with different exposure levels
I1 = imread('Picture 43.jpg');
I2 = imread('Picture 44.jpg');
% display the color images
%imagesc(I1);
%imagesc(I2);
% convert to B&W
% The green channel is taken
I1 = I_1(:,:,2);
I2 = I_2(:,:,2);

figure
imagesc(I1,[0 255]); colormap(gray);
figure
imagesc(I2,[0 255]); colormap(gray);
% replace saturated pixels from I1 using the corresponding pixels from I2
% require
%    I1    B&W image with saturated region
%    I2    B&W image without saturation
R = I1;
% find the saturation level of R - adjustment by 5 counts, (max(R) - 5)
```

```
m = max(max(R))-5;
% find the indices of the saturated pixels;
i = find(R > m);
% fusion of images by replacing the saturated pixel i in I1 by that from
image I2
R(i) = I2(i);
% display the combined image
figure
imagesc(R,[0 255]); colormap(gray);
```

To be done for the report: Show an example of high dynamic range imaging and provide the three images as shown in the example above.

REMARK:

The MATLAB toolbox for imaging has a more professional HDR at hand which is available as command in Matlab. If you are interest you might compare your result with our approach. If no exif metadata are available one needs to define the exposure times in a vector called `expTimes`.

A possible script reads:

```
%create a list of images to be used in HDR
files = {'picture 43.jpg', 'picture 44.jpg'};
% define the list of relative exposure time
expTimes = [5, 1];
%compose the image with optional 'relative exposure'
%exif data would be used if available attached to the file for composition
of the image
hdr = makehdr(files, 'RelativeExposure', expTimes ./ expTimes(1));
%Render high dynamic range image for viewing
rgb = tonemap(hdr);
figure; imshow(rgb)
```

5. Summary of tasks of the experimental work

5.2 Noise evaluation (45 min)

Please provide a table with the different means and standard deviations including parameters for each image and each color, for blue, green and red. (4 dark noise, 2 nearly saturated). Give example line plots with a width of one line taken in the middle of the image for the measurement. (select one channel – red, green or blue, **six plots** (4 dark noise, 2 nearly saturated, select one channel – red, green or blue).

5.3 Noise reduction by averaging (45 min)

Show one set of example images example images like in Figure 14 for the high gain! (one channel – red, **four plots**). Make a table with the value for single line for low and high gain. Find out how many lines from the high gain image have to be averaged to compensate the noise added by the gain!

5.4 High dynamic range imaging (45 min)

Show the images with the ROI (**two plots**). Evaluate the dynamic range increase obtained by the changing of the gain from low to high and **c**alculate the dynamic range factor G with its error! (use the standard deviations as errors for the input values)

Show an example of high dynamic range imaging and provide the **three images** as shown in the example. Indicate in which area one can gather supplementary information.

5.5. Web example for HDR

Find a HDR example image on the web and cite correctly!

The time indicates the time to be spending on experimental task without evaluation!