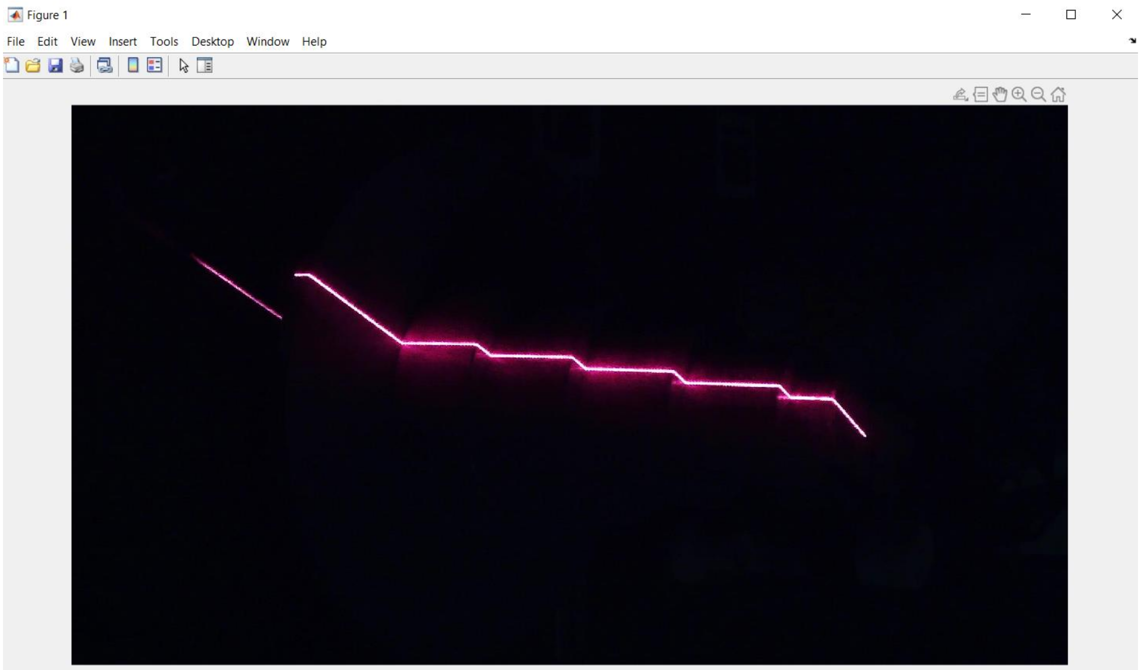
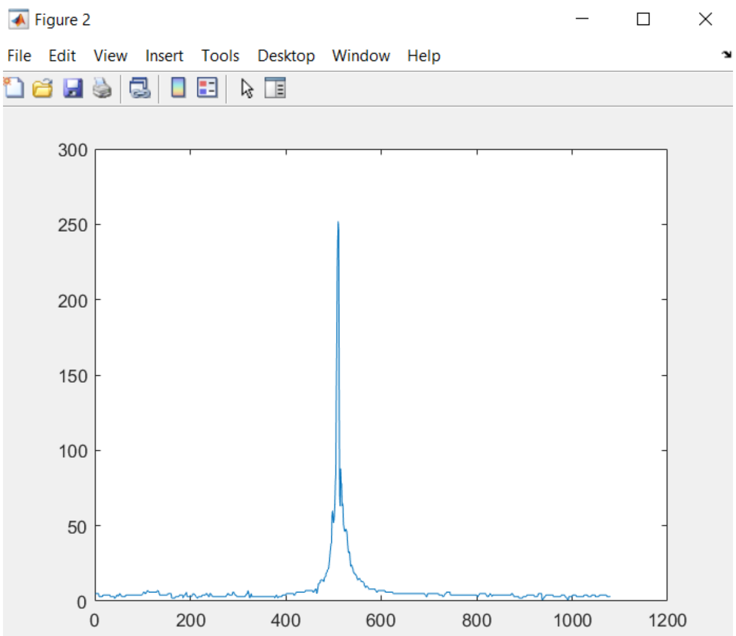
Calculation of the laser-pic

There are several algorithms to calculate the laser pic with sub-pixel precision. We review few of them. In order to understand it, we load first the image of a real laser projected in the calibration pattern, seen by the camera, and one horizontal line of it.

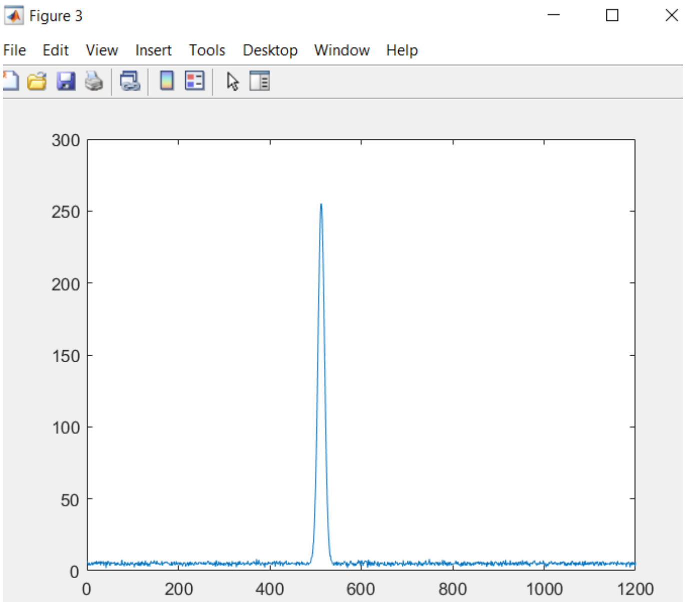
```
clear all
close all
clc

im = imread('6.jpg');
imgr = rgb2gray(im);
imshow(im)
figure
plot(imgr(:, 1000));
```





This line is very similar to a Gaussian function: $y = K * e^{-\left(\frac{x-x_0}{\sigma}\right)^2}$.

```
% Gaussian curve y = A  * e(-(x - x0) / sigma)^2)
x = 1:1200;
x0 = 512.7;
sigma = 10;
y = 250 * exp(-((x - x0) / sigma).^2) + 5 + 1 * randn(1, 1200);
figure
plot (x, y);
```



The first method consists on the calculation of the maximum of the laser line, with the problem that it is not sub-pixel.

```
% First look for the maximum of the peak
[~, xMax] = max(y)
```

xMax = 513

The second method consist of making the fitting of the gaussian. The easiest way is to calculate the logarithm of the gaussian, i.e. $\log(y) = \log(K) - \left(\frac{x-x_0}{\sigma}\right)^2$. This results in the linear system:

$$[\log(yi)] = [-xi^2, 2 * xi, 1] * \left[\frac{1}{\sigma^2} \quad \frac{1}{\sigma} \quad (\log(K) - x_0^2/\sigma^2)\right]^T$$

In practice, we use log(abs(yi)) in order to prevent logarithms of values <= 0, and choose an interval of fitting around the maximum.

```
% We take the log of the equation log(y) = log(250) - (x - x0) /
sigma)^2
interval = [xMax - 10: xMax + 10];
logy = log(abs(y(interval)) + 1)'; % to prevent log(0)
xi = x(interval)';

B = logy;
A = [-xi.^2, 2 * xi, - ones(size(xi))];
```

```
Res = pinv(A) * B;
sigma2Res = 1 / Res(1);
x0Exp = Res(2) * sigma2Res
```

x0Exp = 512.6835

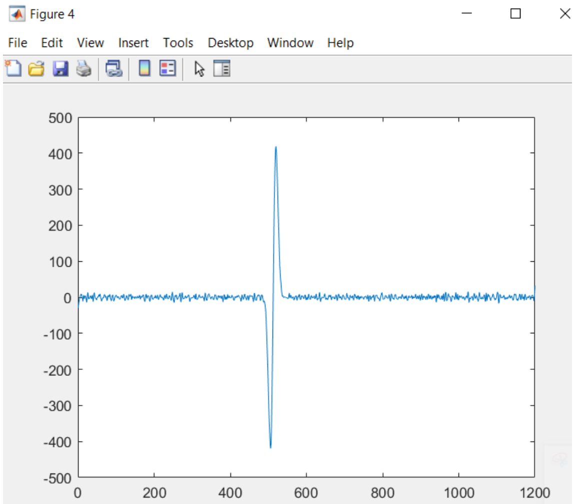
The next method consists of calculating the center of gravity of the curve around the maximum.

```
% We take the center of gravity of the line
intervalCoG = [xMax - 20: xMax + 20];
x0CoG = sum(y(intervalCoG).*x(intervalCoG)) / sum(y(intervalCoG))
```

x0CoG = 512.7404

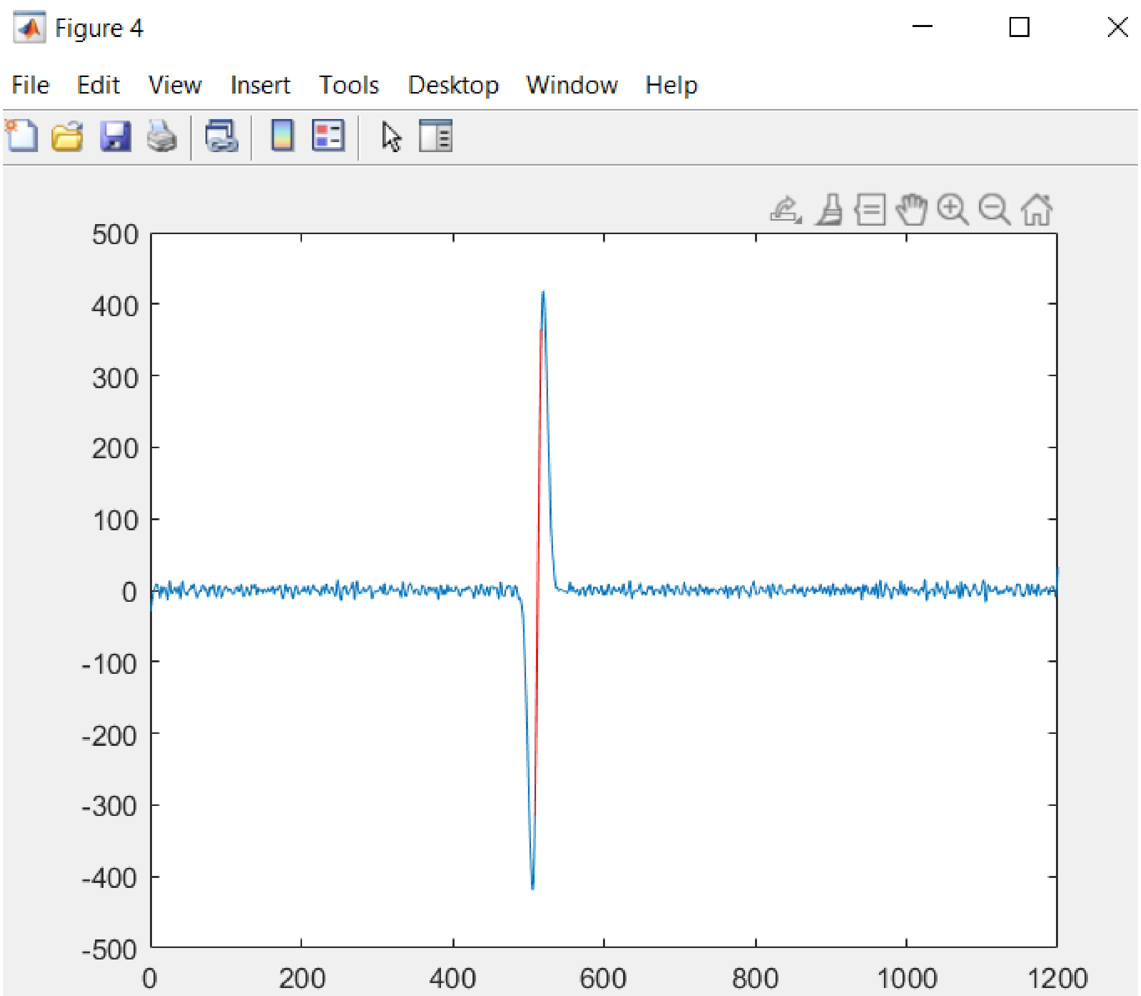
The las method consist of making a derivative convolution with a mask (equivalent to a filter FIR). Thereafter, a zero crossing is calculated.

```
% We take the convolution (FIR) of the line with a mask
mask = [-1 -2 -3 0 3 2 1];
yConv = conv(y, mask, 'same');
figure
plot(x, yConv);
```




```
% We look at the zero crossing of this curve.
% First fitting a line a * x + b * y + c = 0
intervalZCross = [xMax - 4: xMax + 4];
A = [x(intervalZCross)', yConv(intervalZCross)',
ones(size(x(intervalZCross)'))];
[~, ~, V] = svd(A);
Recta = V(:, end);

hold on
plot(x(intervalZCross), (- Recta(1) * x(intervalZCross) - Recta(3)) /
Recta(2), 'r');
x0ZCross = - Recta(3) / Recta(1)
```

x0ZCross = 512.7045



Next Practice

The following practice is to calculate the point cloud of an object, given the calibrated homography between the lase-plane world tha passes through a rotating line and the camera. The image takes are made at a discretization of 1º. The student should calculate the points in 3D in function of the angle of rotation.