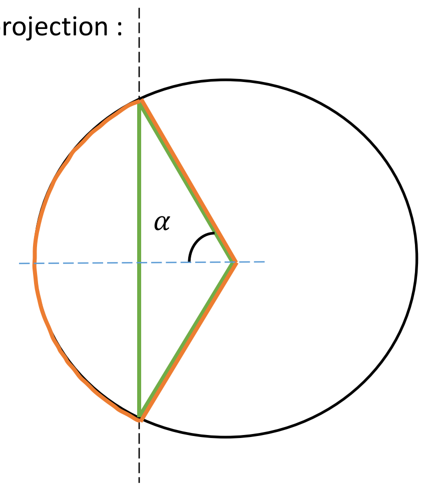
Pierre Oucif ME6406 : Machine Vision – HW#1

1. <u>Pin-hole optics</u>

a) We begin to find the expression of the area of the overlap region for one projection :



$\mathrm{\delta A_{bright}} = \pi R^2 - \mathrm{\delta A_{dark}}$

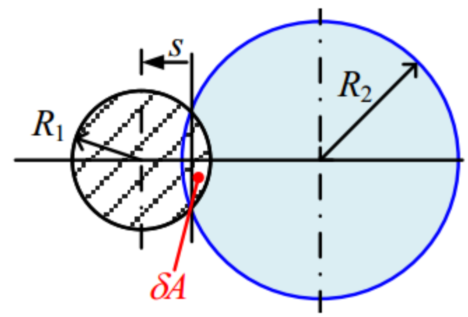
$\frac{\mathrm{\delta A_{bright}}}{\mathrm{\delta O}} = 1 - \frac{\mathrm{\delta A_{dark}}}{\mathrm{\delta O}} = 1 - (\mathrm{\delta A_{red}} - \mathrm{\delta A_{green}})$

$\mathrm{\delta A_{dark}} = \pi R^2\left(\frac{2\alpha}{2\pi}\right) - sh = R^2\alpha - s\sqrt{R^2 - s^2} = sR\sqrt{1 - \left(\frac{s}{R}\right)^2}$

$S = \frac{s}{R}$ and $\alpha = \cos^{-1} S$

$\frac{\mathrm{\delta A_{bright}}}{\mathrm{\delta O}} = 1 - \frac{1}{\pi}(\cos^{-1} S - S\sqrt{1 - S^2})$

Then, we can find a relationship between the two projections by using the distance between the two intersection points of this two projections. We get:
$R_1^2 - s_1^2 = R_2^2 - s_2^2$



So, $s_2^2 = R_2^2 - R_1^2 + s_1^2$

Then, $s_2 = \pm R_1\sqrt{\lambda^2 + S\_^2 - 1}$ where $S = s_1/R_1$

We take $s_2 = -R_1\sqrt{\lambda^2 + S\_^2 - 1}$ because when s_1 and s_2 are defined from the same axis formed by the two intersections of the circles, but they stop on the center of each circle. So $\mathrm{sign}(s_2) = -\mathrm{sign}(s_1)$

Finally, $S_2 = \frac{s_2}{R_2} = -\frac{R_1\sqrt{\lambda^2+S^2-1}}{R_2} = -\frac{\sqrt{\lambda^2+S^2-1}}{\lambda}$

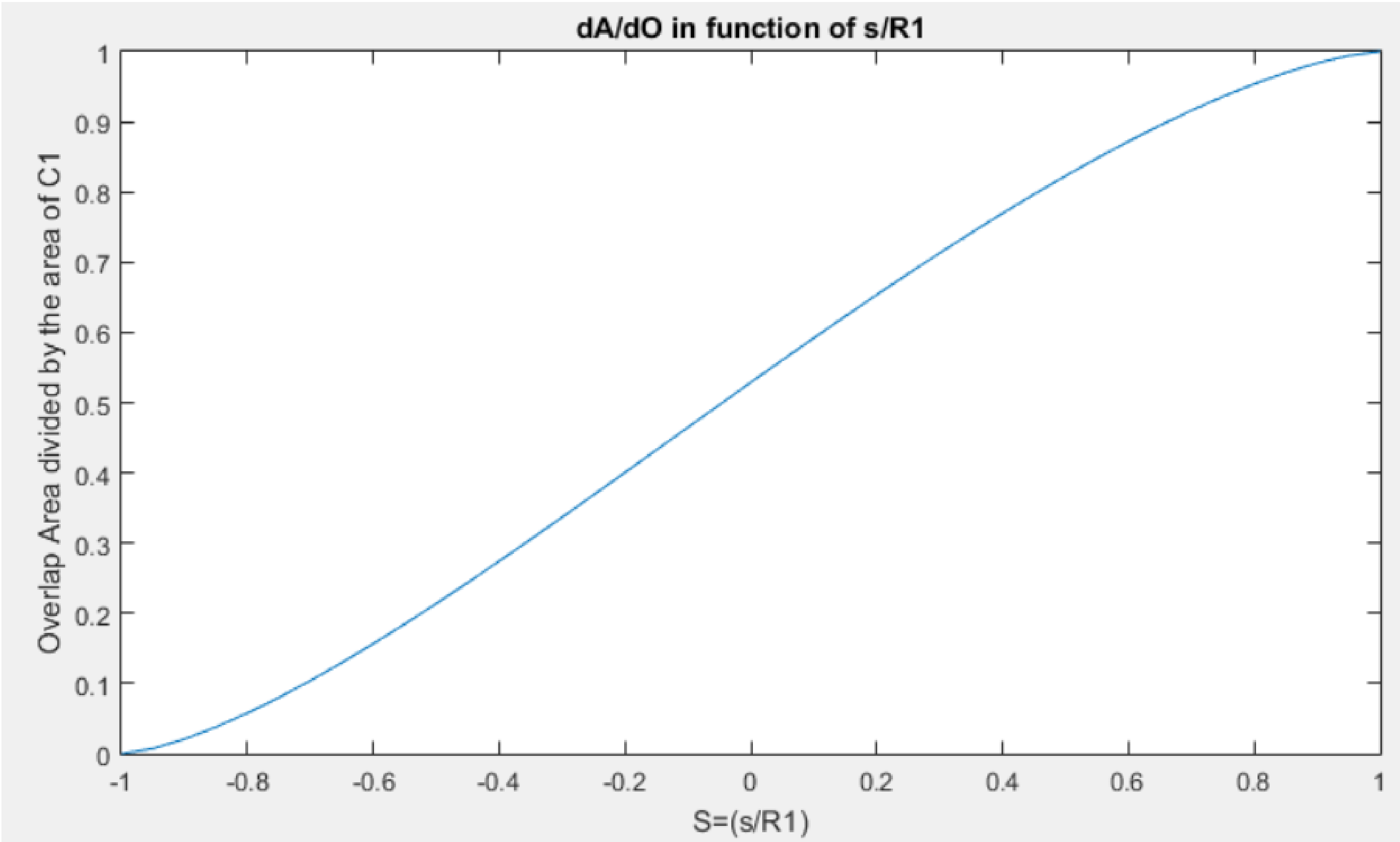
Now, to find an expression of the area of the overlap region, we have to sum the overlap area of each region, such that:

$$\frac{\delta A}{\delta O} = \left(\frac{\delta A}{\delta O}\right)_1 + \left(\frac{\delta A}{\delta O}\right)_2$$

So, $$\frac{\delta A}{\delta O} = 2 - \frac{1}{\pi}(\cos^{-1}\frac{s_1}{R_1} - \frac{s_1}{R_1}\sqrt{1 - \left(\frac{s_1}{R_1}\right)^2} + \cos^{-1}\frac{s_2}{R_2} - \frac{s_2}{R_2}\sqrt{1 - \left(\frac{s_2}{R_2}\right)^2}$$

Finally, we get: $$\frac{\delta A}{\delta O} = 2 - \frac{1}{\pi}(\cos^{-1} S - S\sqrt{1 - (S)^2} + \cos^{-1}(-\frac{\sqrt{\lambda^2+S^2-1}}{\lambda}) + \frac{\sqrt{\lambda^2+S^2-1}}{\lambda}\sqrt{1 - \left(-\frac{\sqrt{\lambda^2+S^2-1}}{\lambda}\right)^2}$$

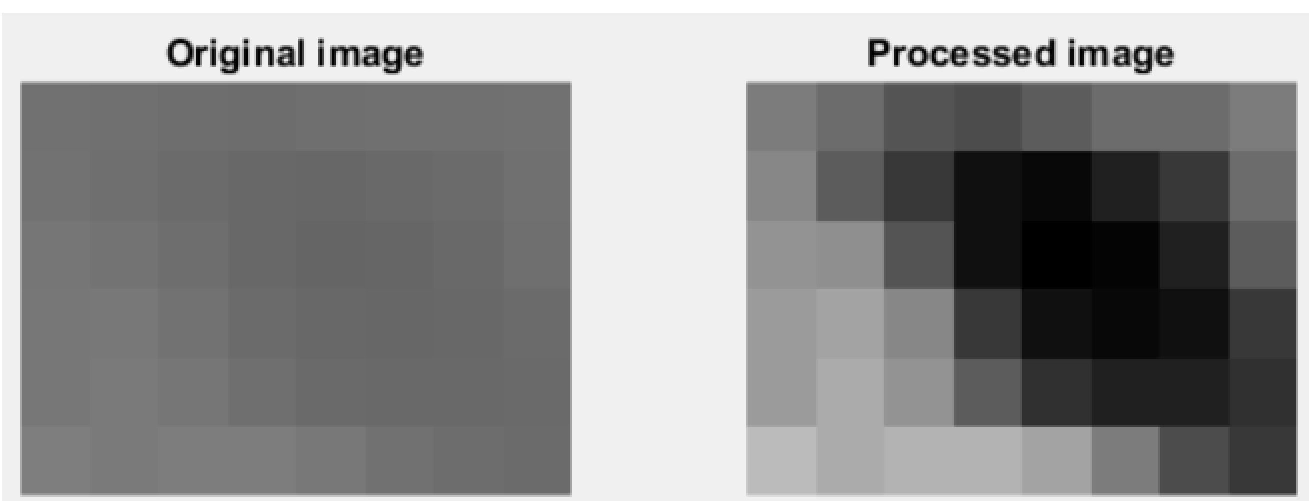
b) Plot of $\frac{\delta A}{\delta O}$ for $-1 \leq S \leq 1$ and $\lambda = 2$



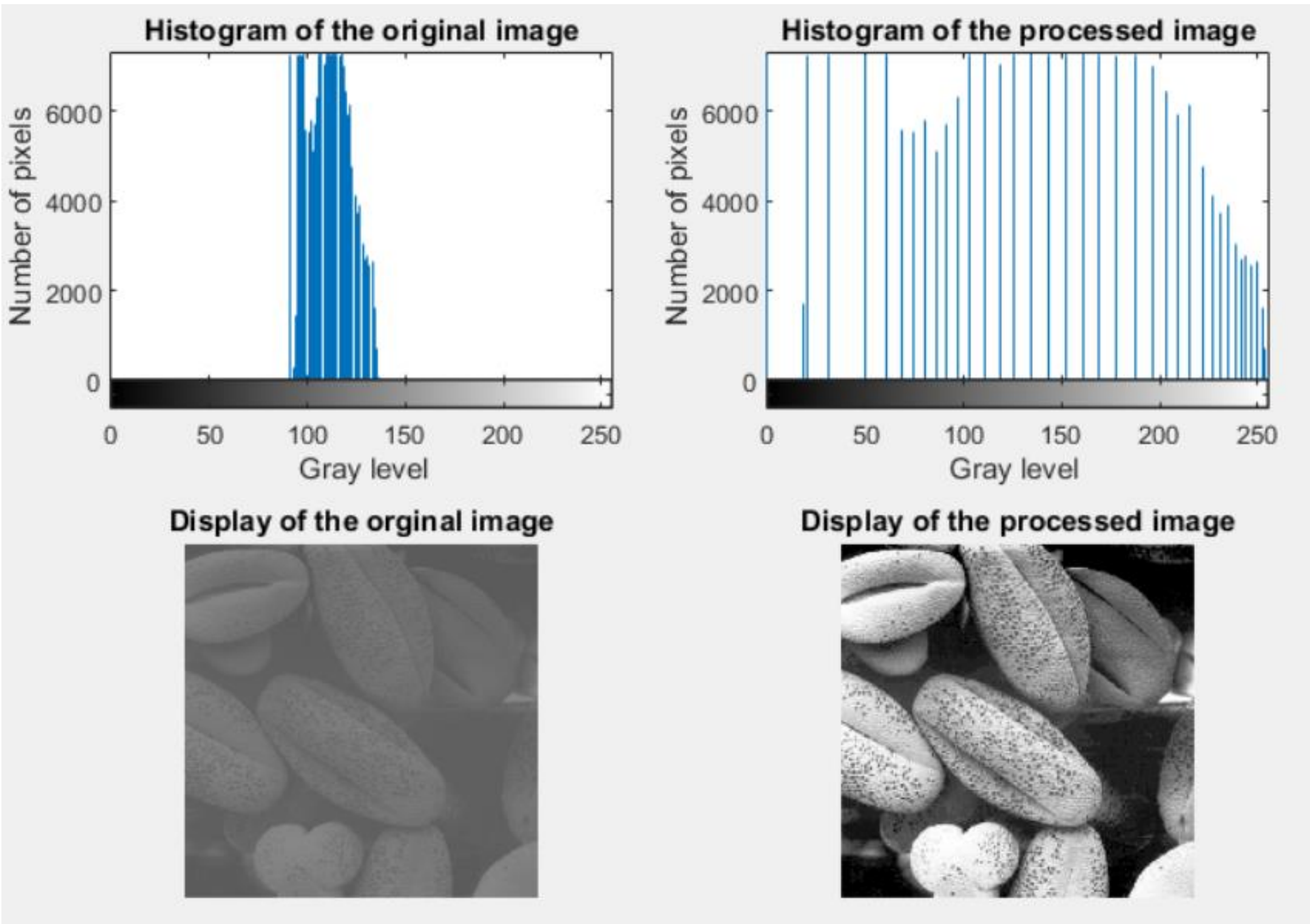
This plot confirm that when $s = -1$, which means that $s = R_1$, the small circle C1 is tangent to the big circle C2 and outside of C2, so the area of the overlap is equal to 0. Whereas, when $s = 1$, which means that $s = -R_1$, C1 is also tangent to C2 but this time totally into C2.

2. Histogram equalization

   a) To complete the table 1 to carry out the histogram equalization, we use the formula from the class note. You can see directly in the Matlab code the table 1 completed. Next, you can see the original image displayed with the processed image that we got by making the histogram equalization:

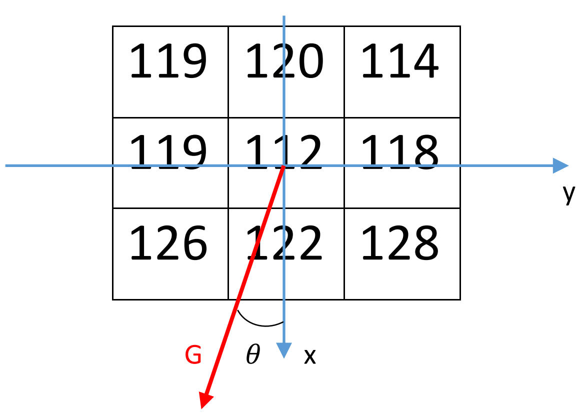


b) You can see on the commented script Matlab the different step to carry out the histogram equalization. The following plots are the results that we obtained with this script.

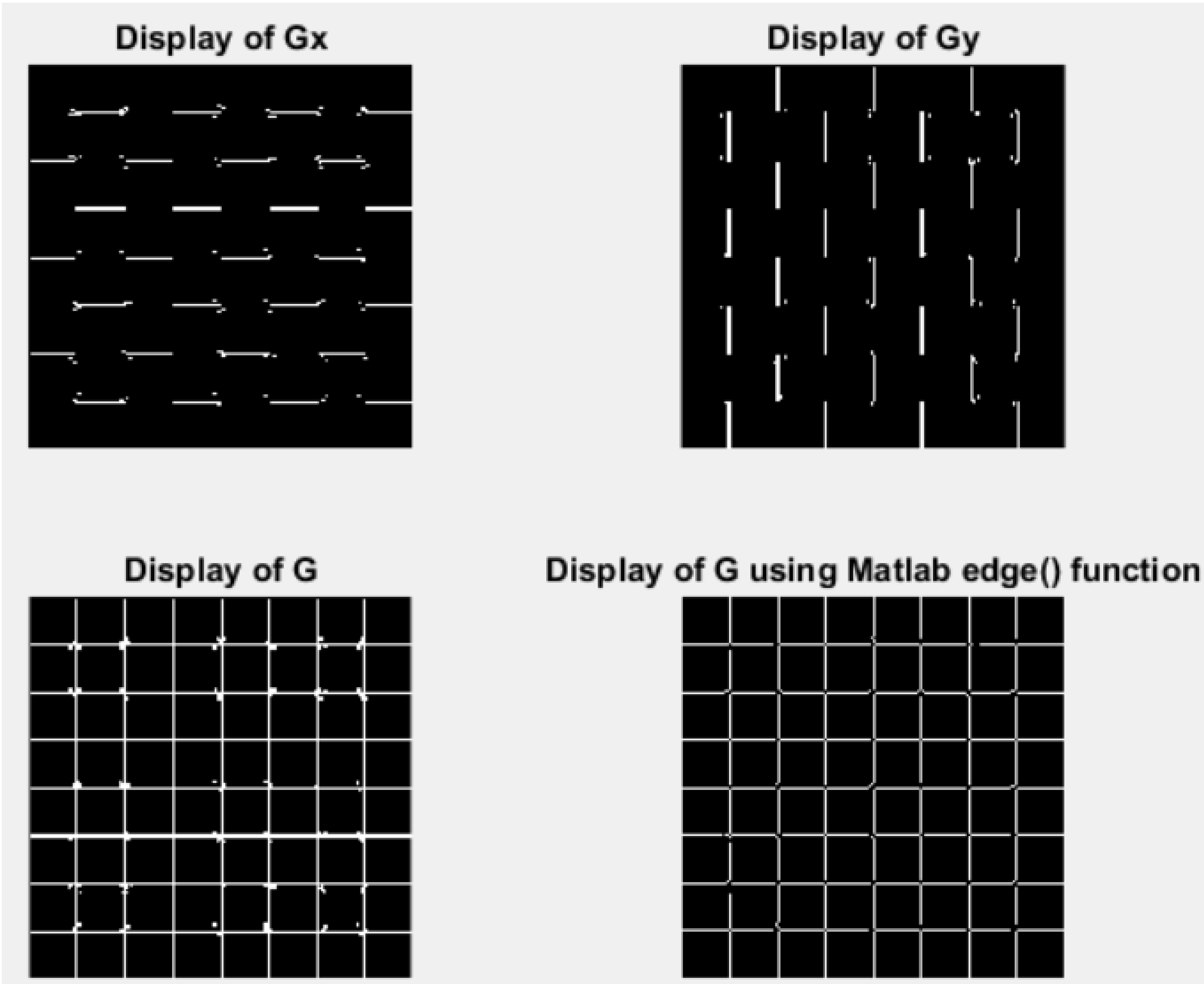


3. Filtering masks

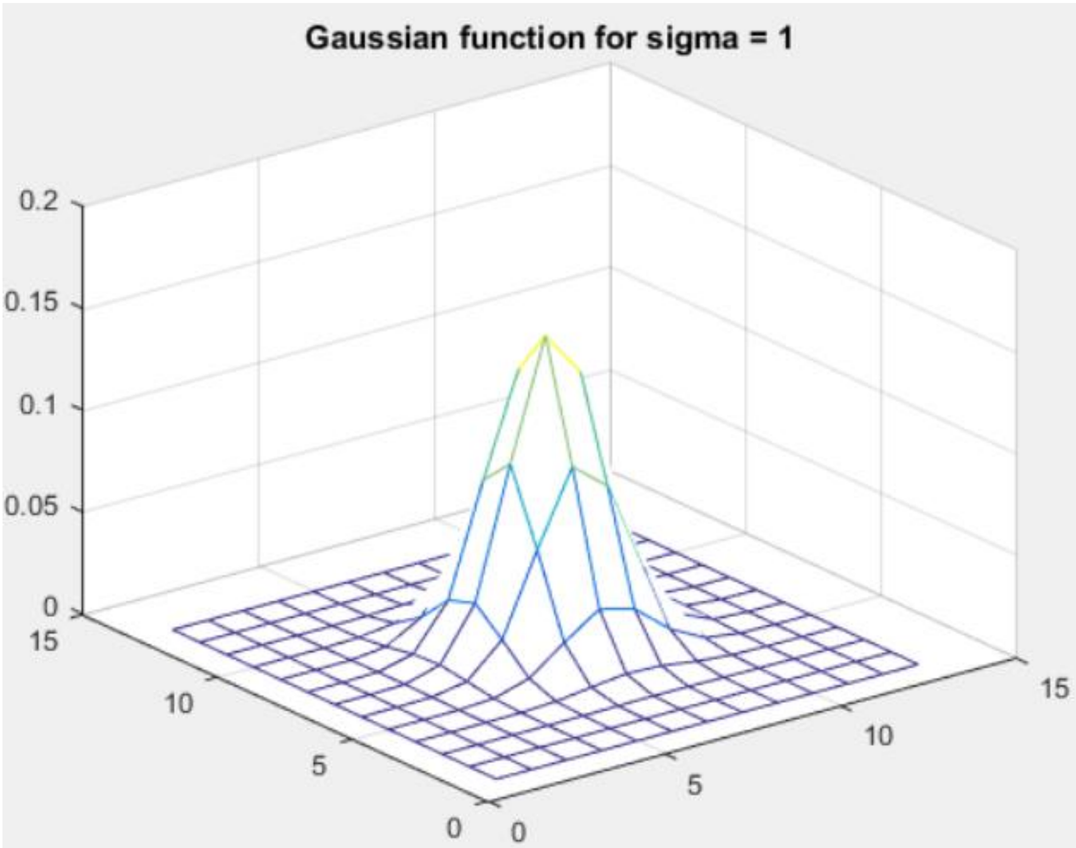
a) We use a 3x3 Sobel operator to calculate the gradient on the pixel (5,2). You can see in the script the details to find the gradient. We obtain a gradient of magnitude G equals to 23.41 and an angle θ of -19.98°:

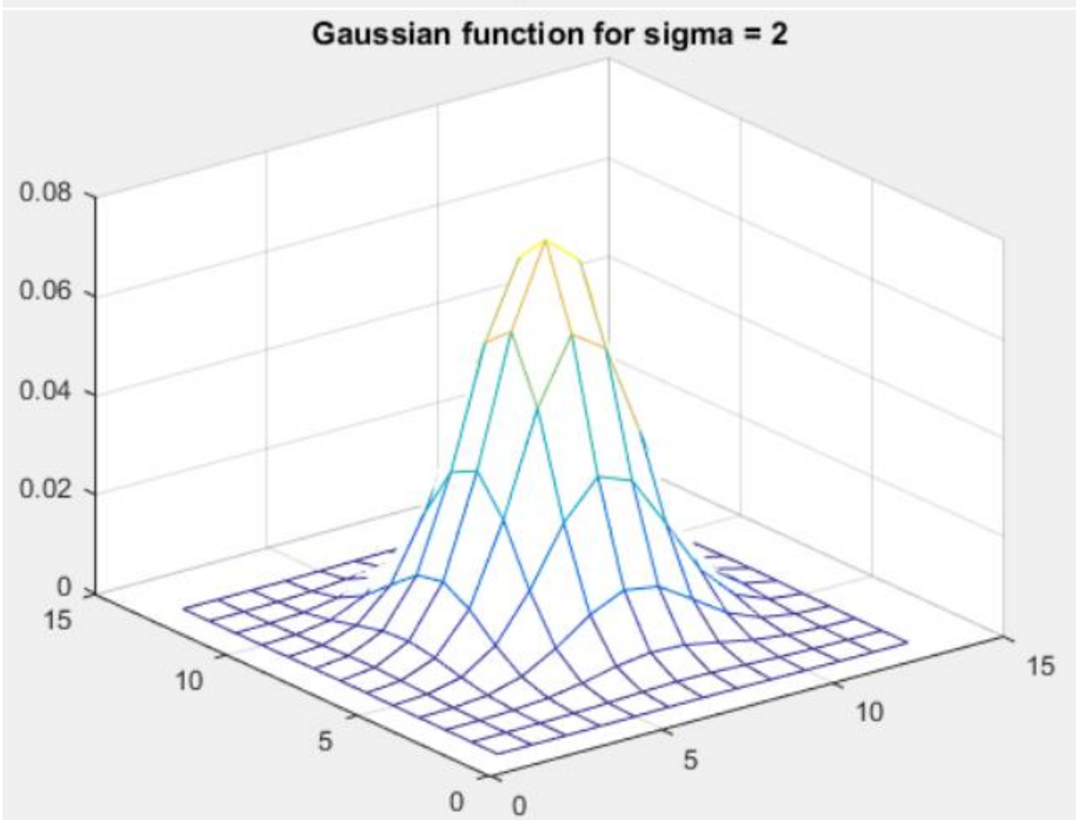


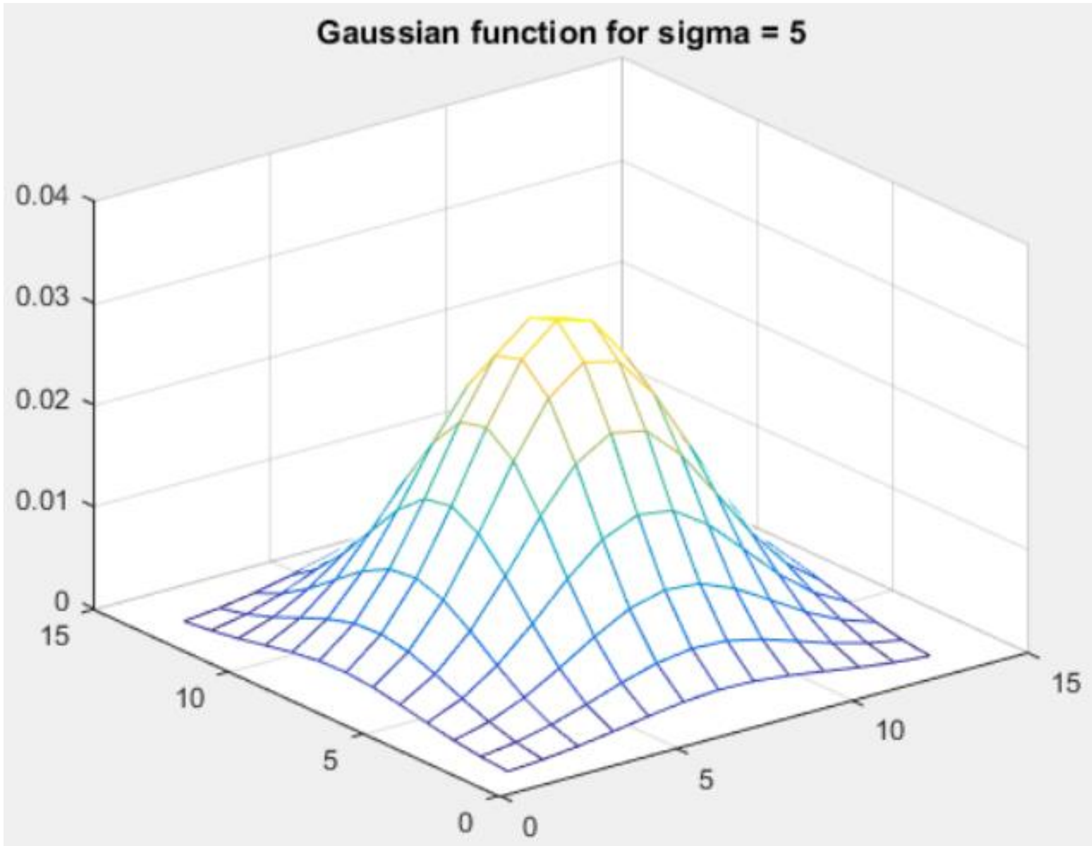
b) We can also calculate the gradient at each pixel of an image by using the Sobel operator. We just cannot find it at the edge of the image. You can see in the corresponding Matlab script how to calculate the x and y components of the gradient at each pixel of the image. We also plot the gradient gives by the Matlab function :



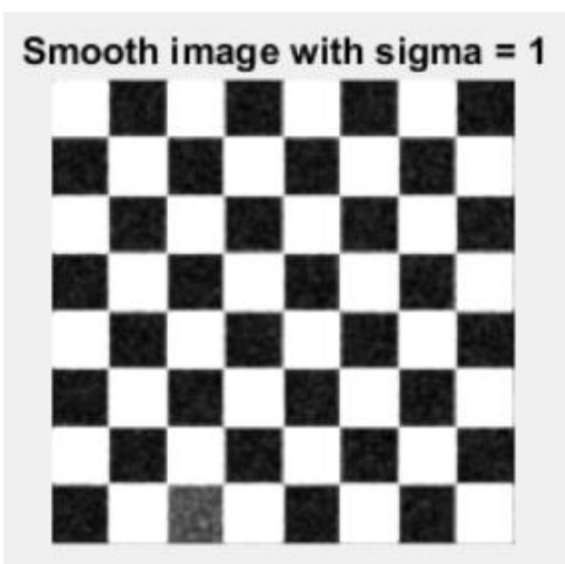
c) Plots of the Gaussian function :

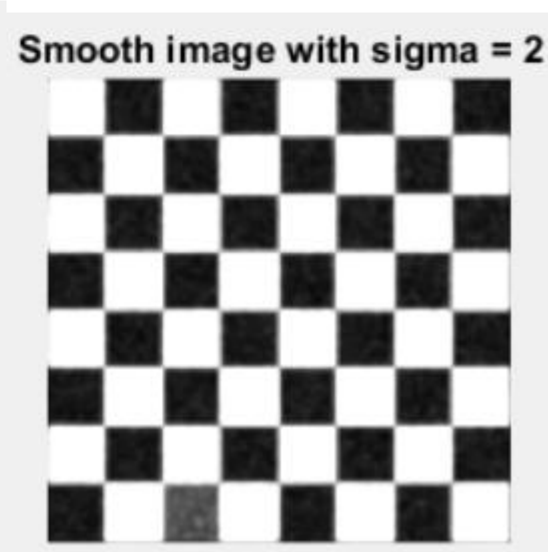


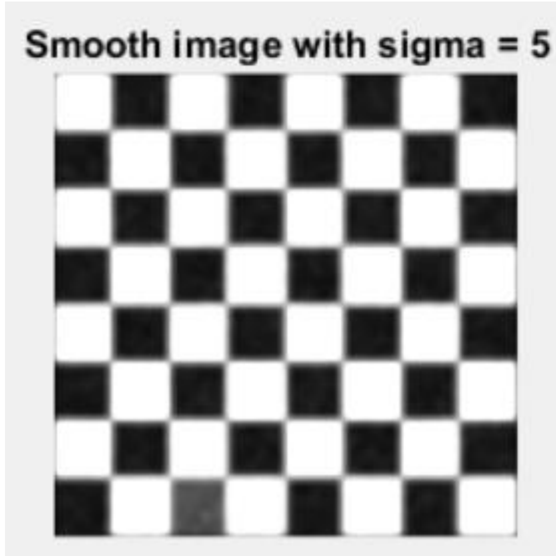




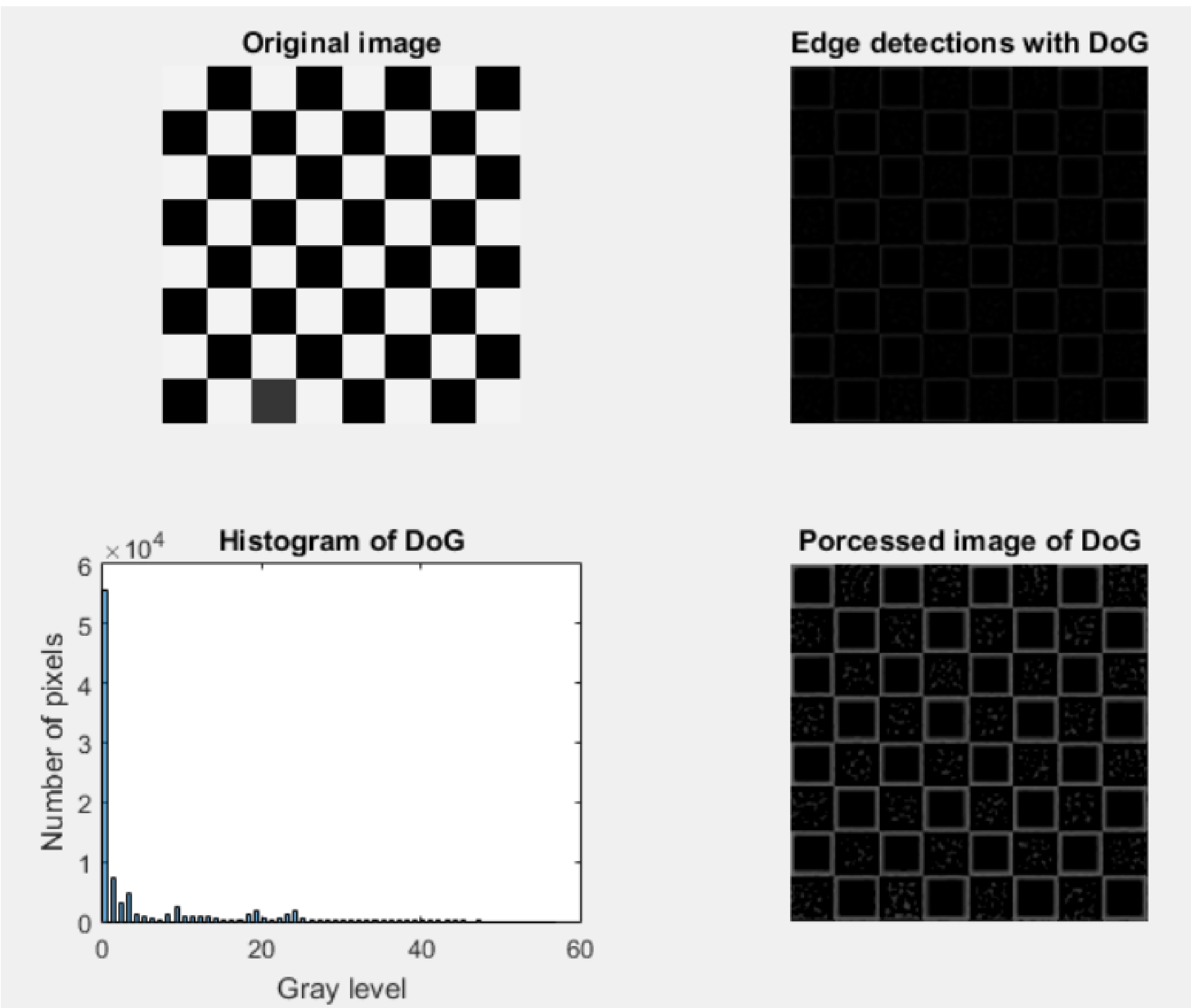
d) We use the previous Gaussian functions to make convolutions with the image to smooth. With the use the Matlab function “imfilter” to get the following images :







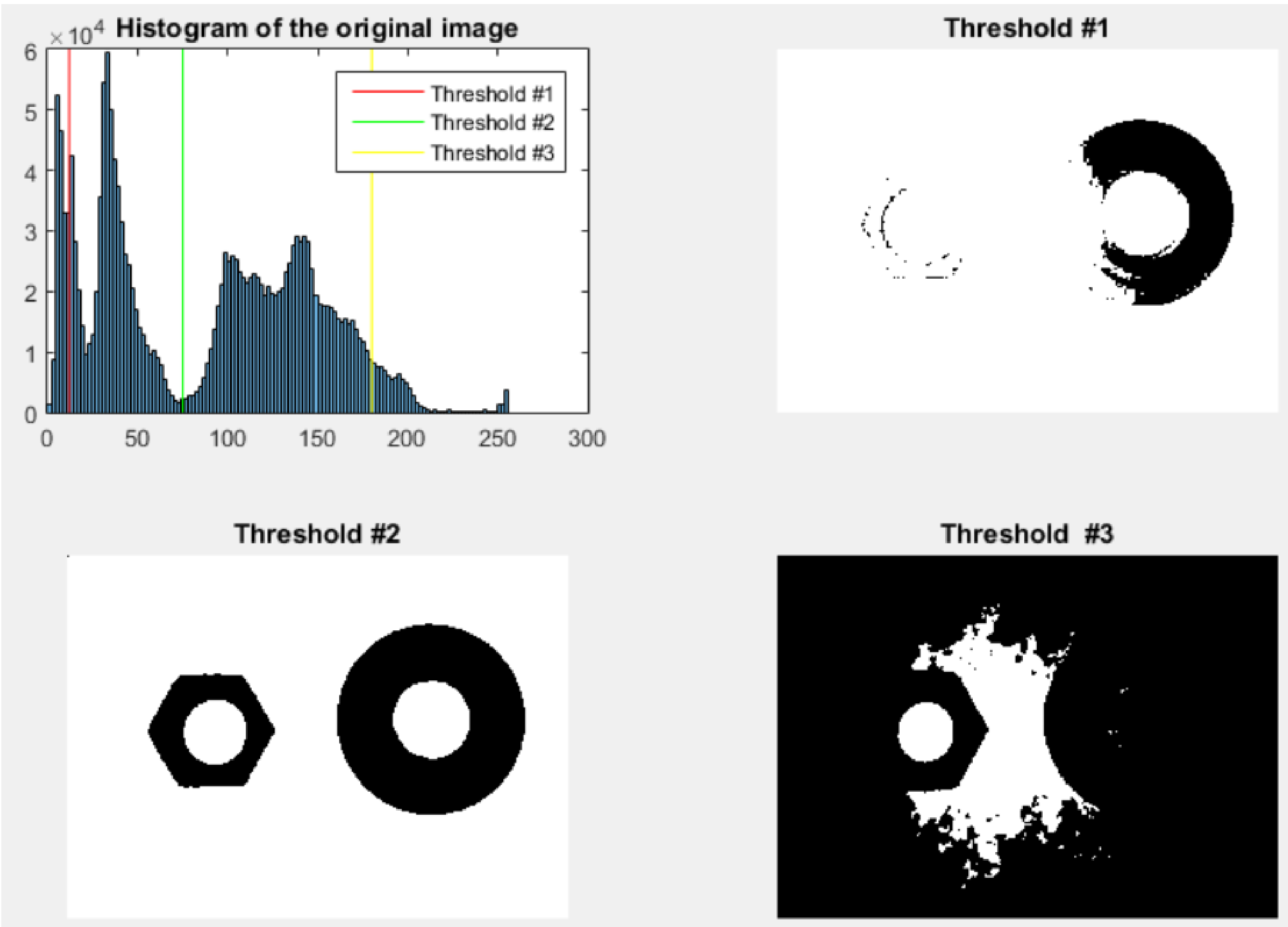
e) Then, if we make the difference between two Gaussian with different σ's but with the same mask size, we can perform an edge detection. So, we calculated the difference of the two Gaussian named DoG and we make the convolution with the image by using the previous Matlab function. We get:



We also made a histogram equalization to be able to see the result. We use the script from the problem 2 to realize the histogram equalization.

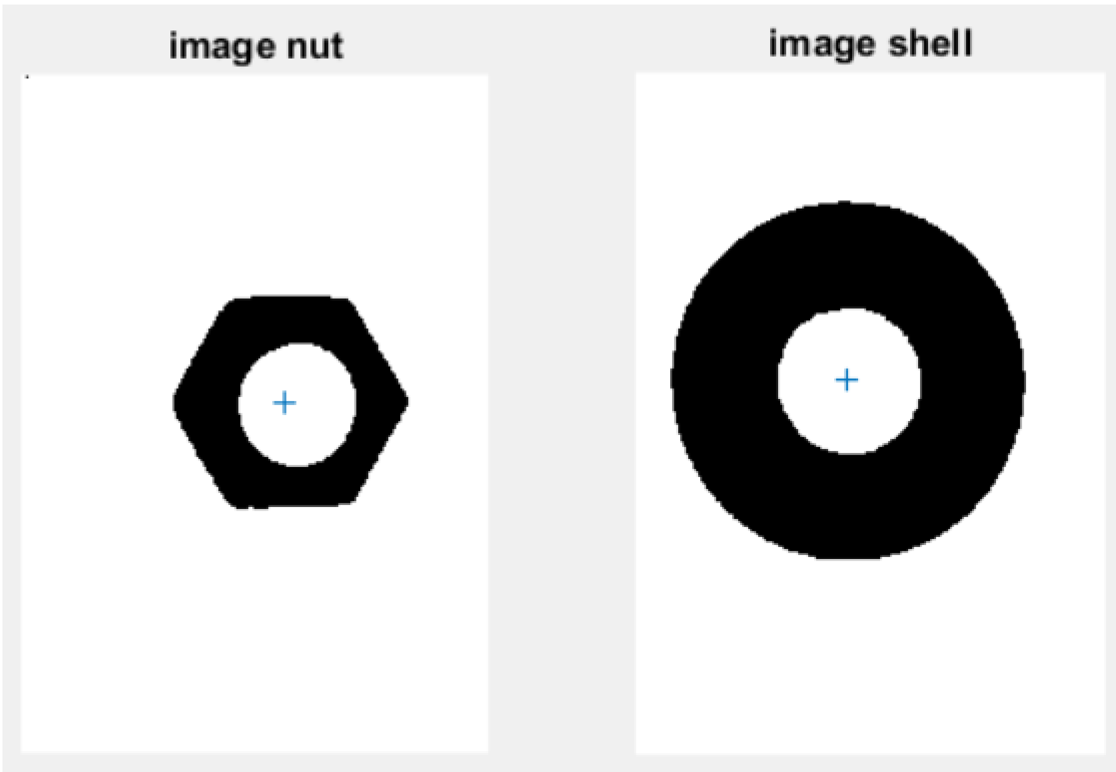
4. <u>Low-level information processing</u>

   a) First, we use a histogram of the gray scale of the image in order to determine the 3 different thresholds. The first one is an under-estimate threshold (red line on the plot) equals to 12/255, the second one is an appropriate threshold (green line on the plot) to binarize correctly the image. We fund it by using the histogram of the image, such that the threshold is between the two main pixel areas of the histogram. This threshold is equal to 75/255. To finish, the third threshold is an over-estimate threshold (yellow line on the plot) equals to 180/255.

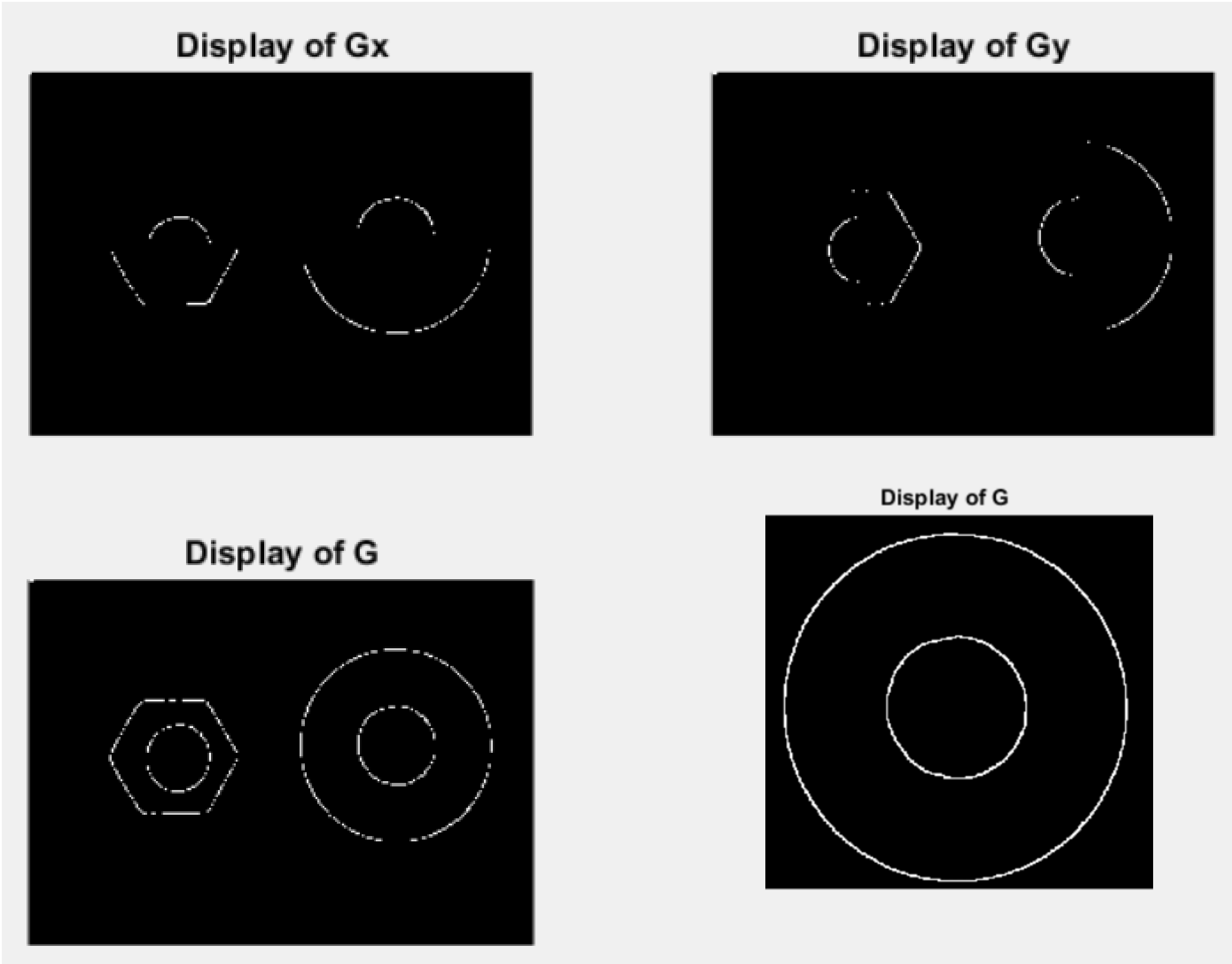


The two objects of this image are darkest than the rest of the image. So, they should be represented on the histogram by the high values of gray (on the right of the green line). So, if we take a threshold near the small values of the gray scale, we miss some pixels of the two objects as we can see on the right top image. Whereas, if we take a threshold to high on the gray scale, we take too much pixels and we get more than the two objects as we can see on the right bottom image. Finally, we a correct threshold, we take almost all the pixels of just the two objects as we can see on the left bottom image.

b) To find the area of the two objects, we cut into two pieces the original object in order to get one image per object. Then, we just have to count the number of black pixels on each image to get the area. We find $Area_{nut} = 25765\ pixels$ and $Area_{shell} = 79066\ pixels$.
Moreover, to find the centroid of each object, we calculate the barycenter of each object. This means that we calculate the average x-coordinate and y-coordinate of the black pixels. We also have to be careful about the coordinate axis in which we make our calculations and our plots (x and y are inversed). Finally, $centroid_{nut} = (260\ ,231)$ and $centroid_{shell} = (667\ ,301\ )$ (with respect to the plot coordinate system on the original image).



c) To find the outer boundaries of the nut and shell, we can calculate the gradient at each pixels to find the shell and nut boundaries. The following curves which represent the x and y coordinates of the gradient and the gradient are continuous, but because of the scale, they are displayed not continuous, but you can see on the zoom that this is just a scale effect.



Then, we just have to scan the gradient and get the outer boundaries. We did not use this technique. We used the two separate black and white image to find the outer boundaries. We scanned the two images on the x-direction symmetrically with respect to the median of the x axe and then we scanned on the y-direction symmetrically with respect to the median. So, at each first black pixels, we got the outer boundaries.

To plot the outer boundaries on the original image, we modified its colors at each pixels on the outer boundaries.

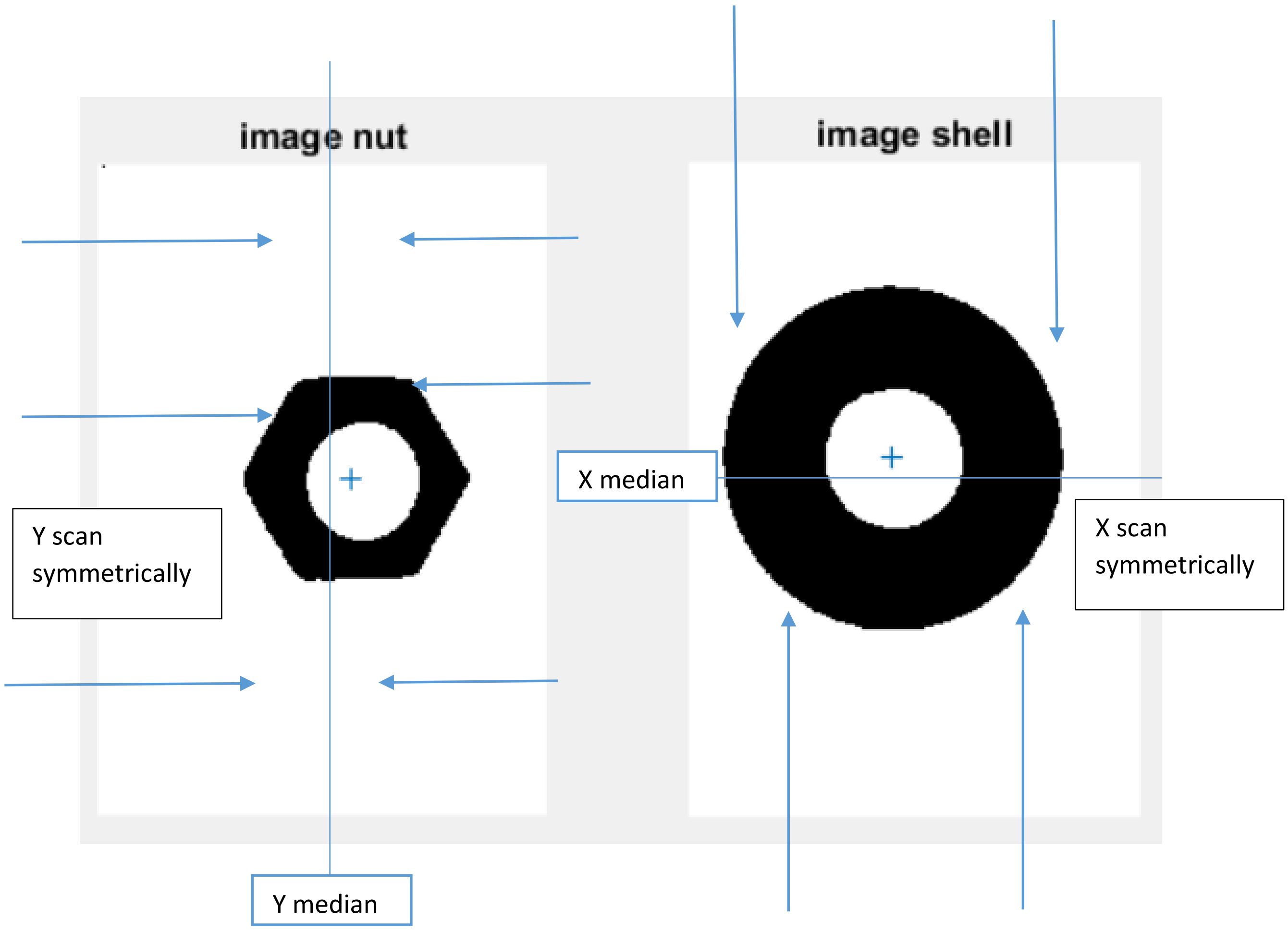
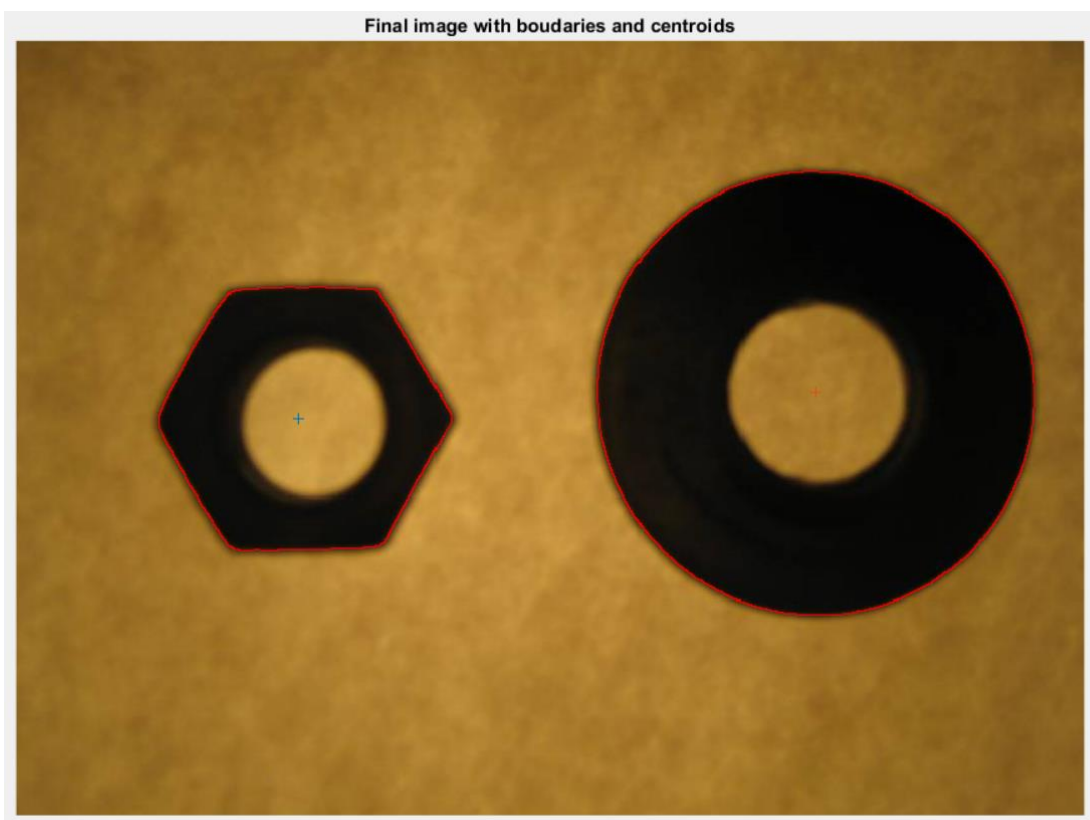


image nut
image shell
X median
Y scan
symmetrically
X scan
symmetrically
Y median


Final image with boudaries and centroids