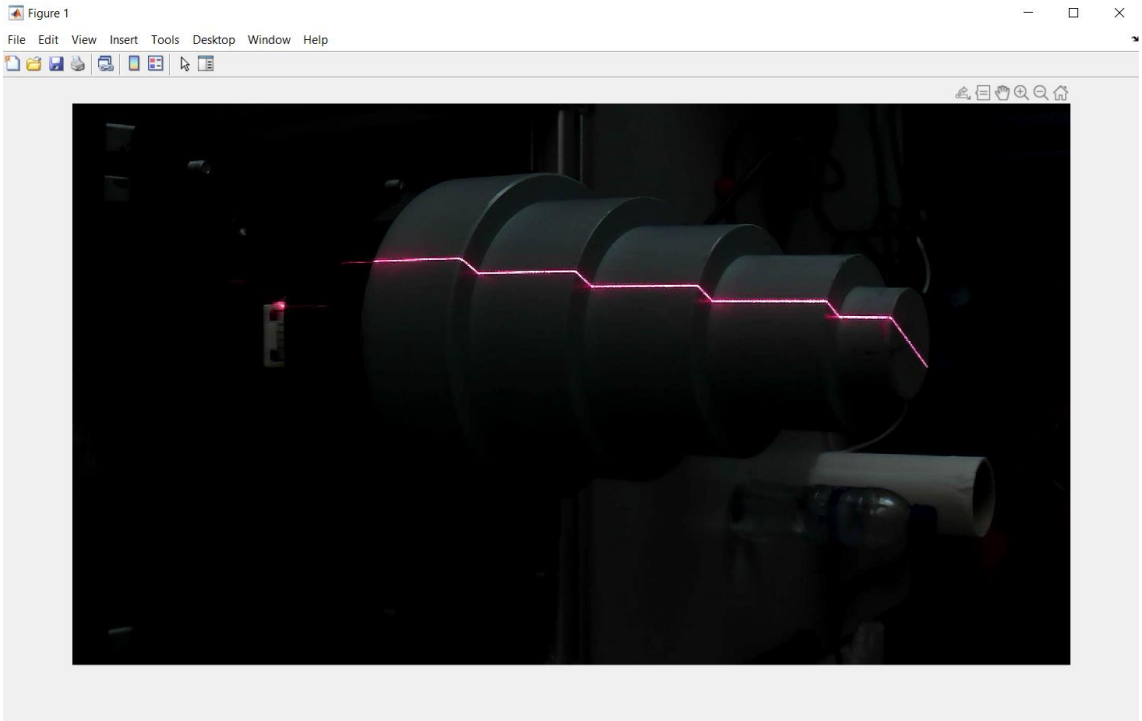
Calculate the Homography Laser-Image

First, load an image of the laser on the calibration cone. It is important that the camera is with low intensity from the outside, in order to detect correctly the laser line.

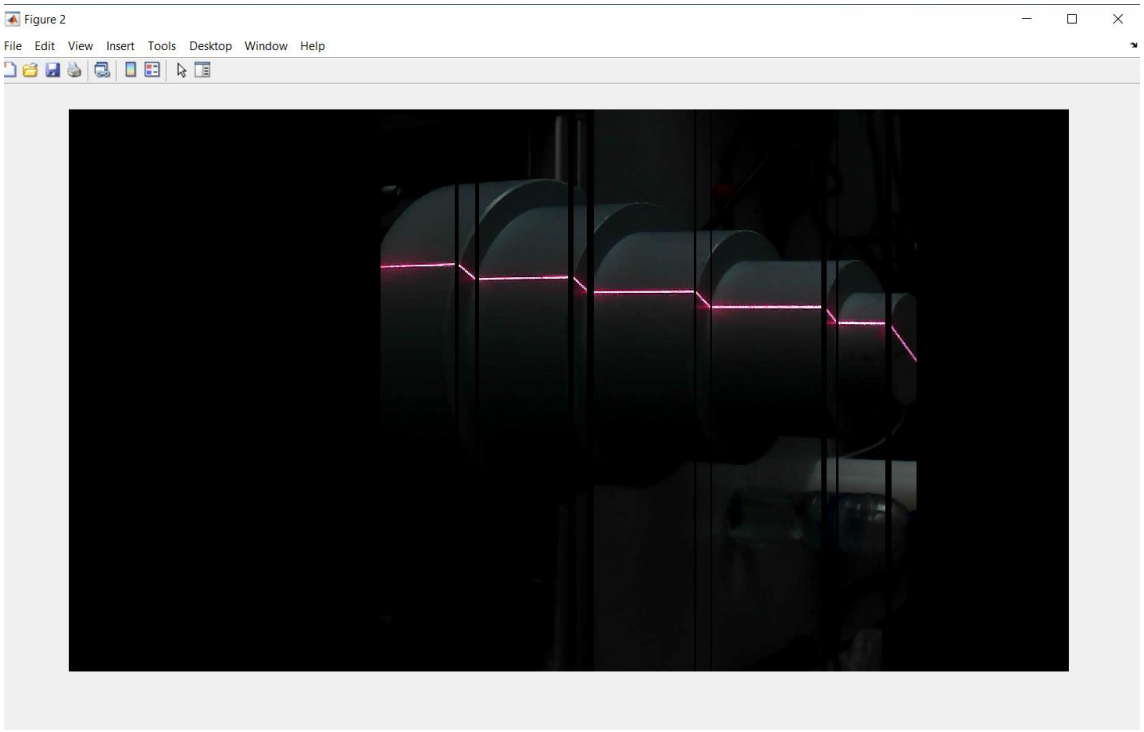
```
clear all; close all; clc;
% Calibrate the laser camera with the cone

% Take image of cone
cam = webcam(1);
img = snapshot(cam);

% For debugging
%imwrite(img, 'img.jpg');
img = imread('img.jpg');
imshow(img);
```



We next define the regions for each line, i.e. the different interval of the line.

```
% Black out the non interesting region
imgInteres = img * 0;

reg{1} = [600, 741];
reg{2} = [750, 780];
reg{3} = [788, 959];
reg{4} = [970, 995];
reg{5} = [1009, 1200];
reg{6} = [1205, 1232];
reg{7} = [1235, 1445];
reg{8} = [1456, 1473];
reg{9} = [1478, 1568];
reg{10} = [1581, 1627];

imgGray = rgb2gray(img);
```

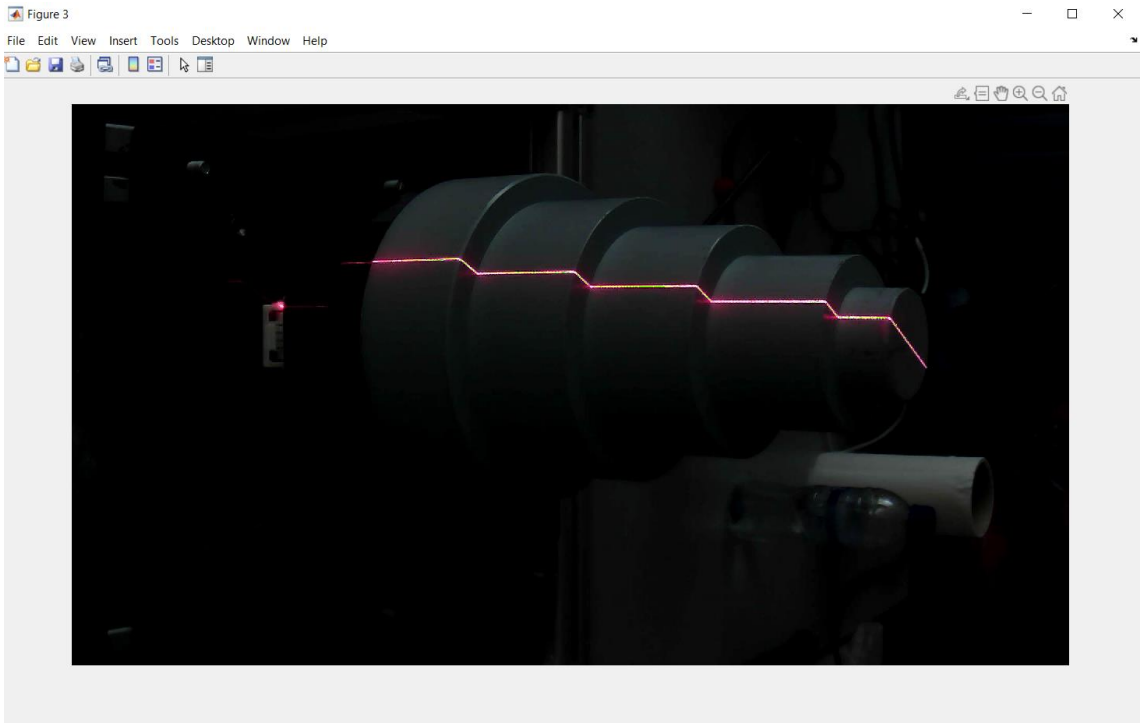
```
for i = 1:length(reg)
  imgInteres(:, reg{i}(1):reg{i}(2), :) = ...
     img(:, reg{i}(1):reg{i}(2), :);
end

figure;
imshow(imgInteres);
```

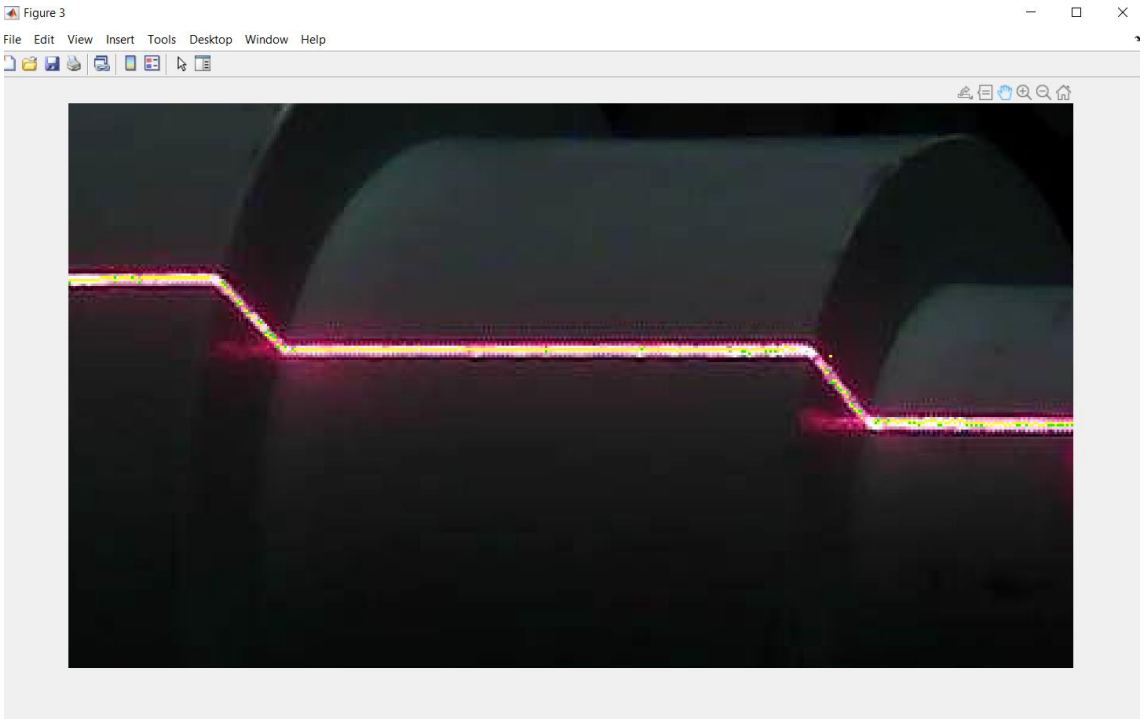


For each line interval, we calculate the maximum, first without subpixel precision, and afterwards with subpixel precision, drawing in green and yellow both of them. We use the fitting of the exponential with +- 7 (delta value) pixel from the maximum.

```
% Calculate the subpixel peaks
for i = 1:length(reg)
  %Calculate the max peaks
  [~, indexes{i}] = ...
    max(imgGray(:, reg{i}(1):reg{i}(2), 1));
  % Calcular los subpeaks
   delta = 7;
   indexSub{i} = maxSubPeak(imgGray, reg{i}(1):reg{i}(2), indexes{i},
delta);
end

imgMax = img;

for i = 1:10
  for j = 1:length(indexes{i})
    imgMax(indexes{i}(j), reg{i}(1) + j - 1, 1) = 0;
    imgMax(indexes{i}(j), reg{i}(1) + j - 1, 2) = 255;
    imgMax(indexes{i}(j), reg{i}(1) + j - 1, 3) = 0;

    indexesSub = round(indexSub{i}(j));
    imgMax(indexesSub, reg{i}(1) + j - 1, 1) = 255;
    imgMax(indexesSub, reg{i}(1) + j - 1, 2) = 255;
    imgMax(indexesSub, reg{i}(1) + j - 1, 3) = 0;
```

```
  end
end

figure;
imshow(imgMax);
```

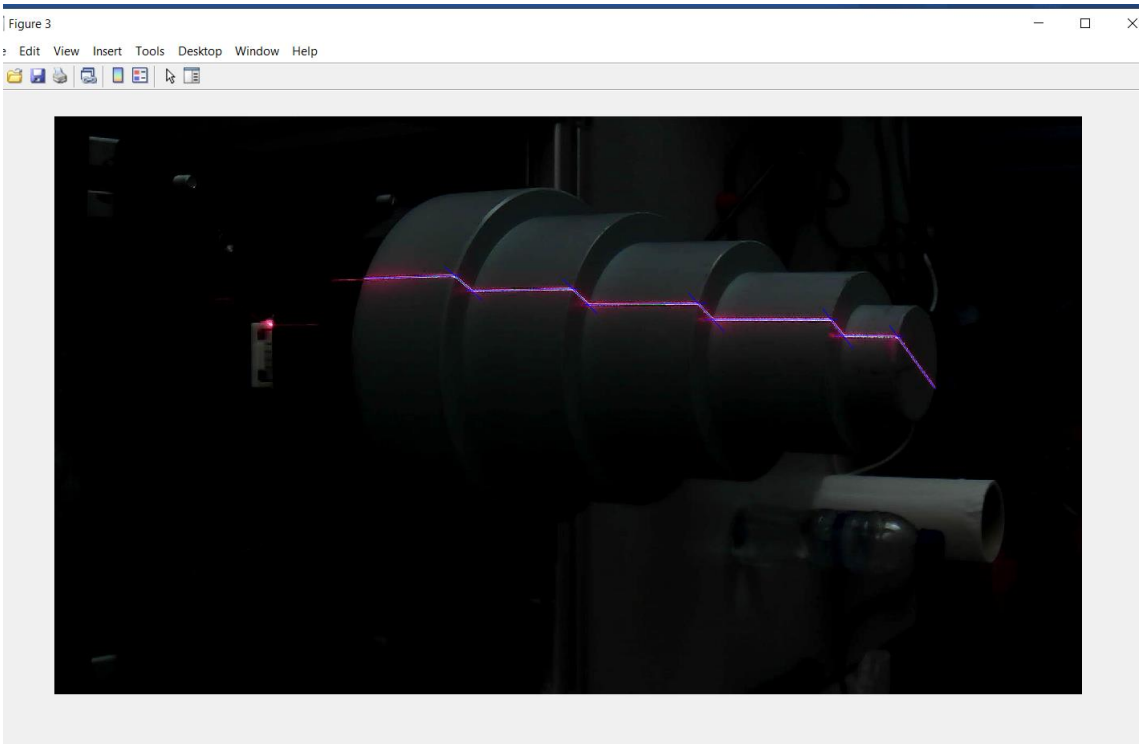




Next, we calculate the fitting of the straight line, taking out the outlayer.

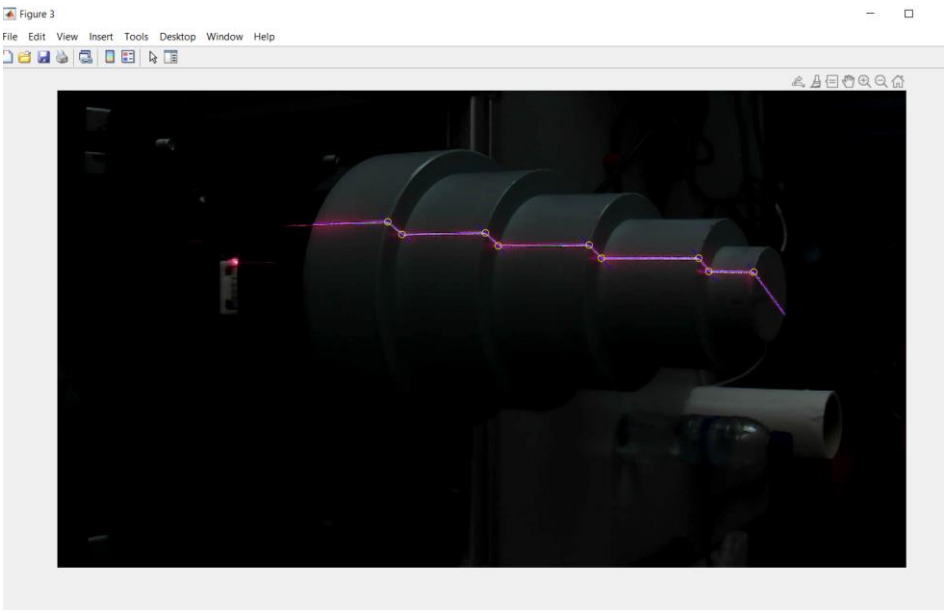
```
% Perform the fitting of the lines
for i = 1:length(reg)
  Recta{i} = fitLine(reg{i}(1, 1):reg{i}(1, 2), indexSub{i});
```

```
end

hold on
% Plot the lines on the image
for i = 1:length(reg)
  intervalX = reg{i}(1, 1) - 20:reg{i}(1, 2) + 20;
  y = -(Recta{i}(1) * intervalX + Recta{i}(3)) / Recta{i}(2);
  plot(intervalX, y, '-b');
end
```



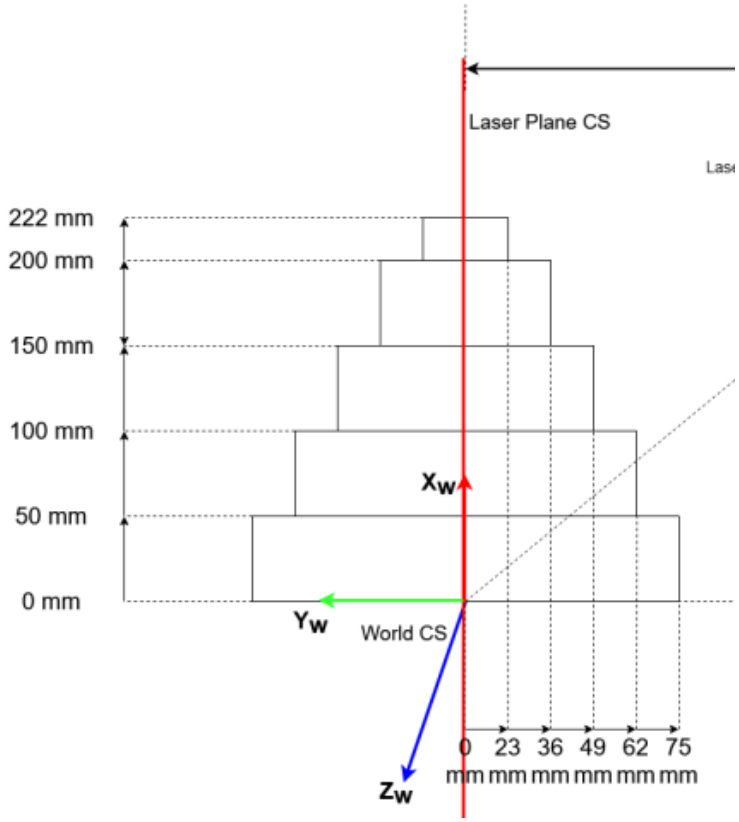
Then, we plot the intersection of the lines:

```
% Calculate the intersection of the lines
for i = 1:length(reg) - 1
  PointsIm(:, i) = -pinv([Recta{i}(1:2)'; Recta{i + 1}(1:2)']) * ...
    [Recta{i}(3); Recta{i + 1}(3)];
  hold on
  plot(PointsIm(1, i), PointsIm(2, i), 'oy');
end
```



Finally, we calculate the homography between the points of the cone and the image

```
% Calculate the homograhy
PtsCono = [0 23  23  36  36  49  49 62  62 75 75;...
       222 222 200 200 150 150 100 100 50 50 0];

H_matrix = get_homography(PointsIm, PtsCono);
```



Complete the functions of this code, understanding it.