Capture of images

To capture the images, we first switch on the laser, and we communicate with the Arduino through the serial line. We set the directory name in which we will capture the images, that will be created if it doesn´t exist. We also connect the camera and the serial line.

```
% Create a set of images and save them step by step
clear

directory = 'captures_Rabbit_22_02_2023';

if ~exist(directory, 'dir')
       mkdir(directory)
end

if ~exist('cam')
  cam = webcam(1);
end

pause(2);

% open the line to send steps to motor.

if (not(exist('s')))
  s = [];
  s = serialport('COM5', 9600);
end

pause(2);
write(s, "1", "string"); % send step
pause(2);
```

We write the images with the laser on, and minimum ambient light, in a loop of 200 images (each images is captured after sendin one step to the motor, via the serial line):

```
for i = 1:200
  write(s, "1", "string"); % send step
  pause(1.5);
  img = snapshot(cam);
  % save captured image
  strNum = sprintf('%03d', i);
  imwrite(img, strcat(directory, '\imgLaser_', strNum, '.jpg'));
  disp(strcat(directory, '\imgLaser_', strNum, '.jpg'));
end
```

Next, we turn off the light and take other 200 images with the ambient light on, in order to capture the real color of the laser points.

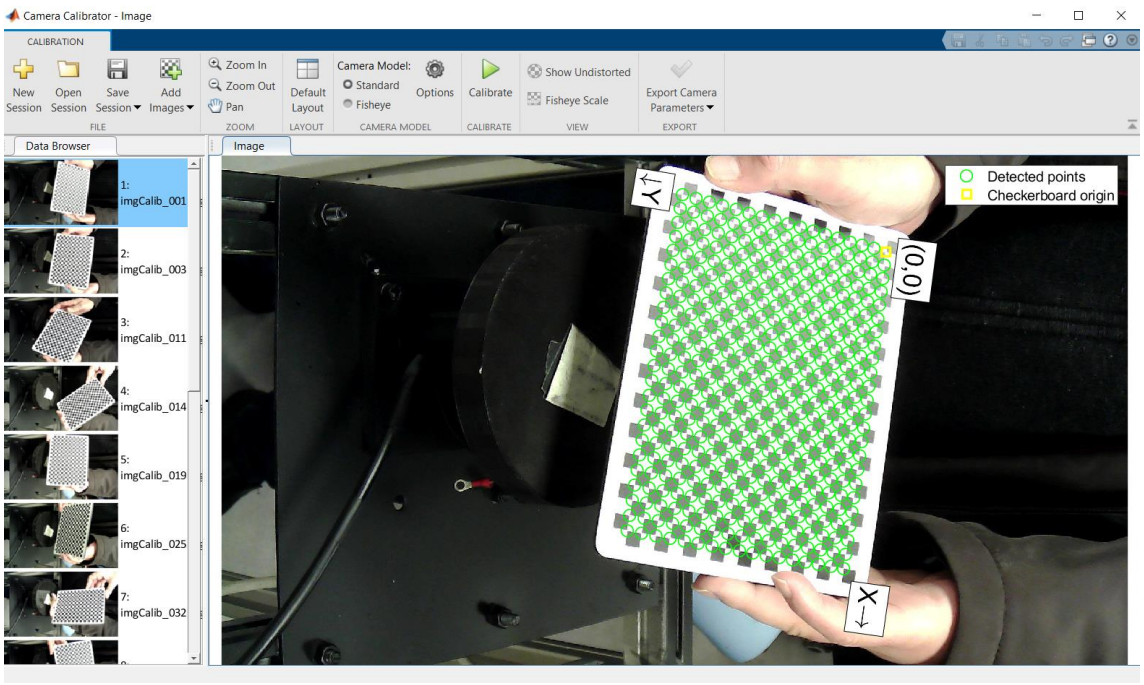
```
disp('15 seconds to take out laser and give lights')
pause(15);

for i = 1:200
  write(s, "1", "string"); % send step
  pause(1.5);
  img = snapshot(cam);
  % save captured image
  strNum = sprintf('%03d', i);
  imwrite(img, strcat(directory, '\imgNoLaser_', strNum, '.jpg'));
```

```
  disp(strcat(directory, '\imgNoLaser_', strNum, '.jpg'));
end
```

Reconstruction of the point cloud

In order to reconstruct the point cloud from the images, we have calibrated first the camera. To do it, we took a set of images and put in a directory “captures_Calib_22_02_2023”. The calibration assistant allows an easy calibration (square size = 7.5 mm).



Once calibrated, I save the cameraParams in ‘cameraParams.mat’. Be carefull, as depending on the Matlab version, the format of the cameraParams structure varies (best way, calibrate from the images in your Matlab version). The reconstruction consists on the definition of the directory with the captured images, and loading of the consecutive images:

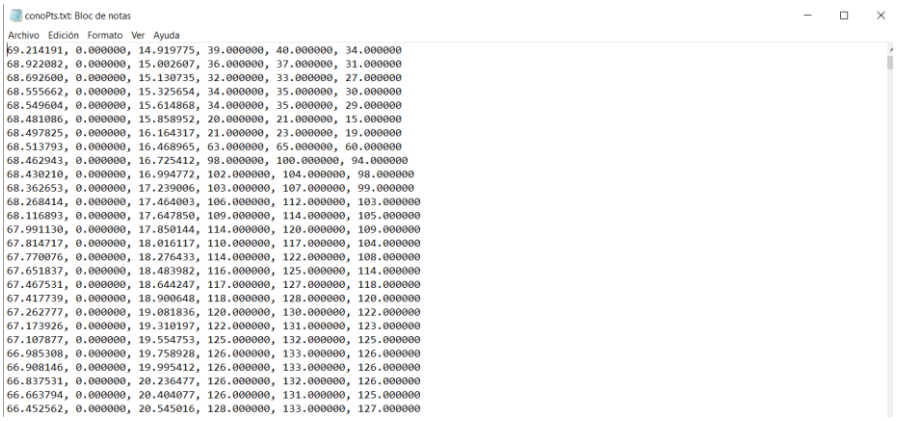
```
clear all; close all; clc;
% Calibrate the laser camera with the cone
load('cameraParams');
load('H_matrix.mat');

directory = 'captures_Rabbit_22_02_2023';
Files = dir(strcat(directory, '\imgLaser*.jpg'));
FilesColor = dir(strcat(directory, '\imgNoLaser*.jpg'));
file = fopen(strcat(directory, '\conoPts.txt'), 'w');

for i = 1:length(Files)
  disp(strcat(directory, '\', Files(i).name));
  img = imread(strcat(directory, '\', Files(i).name));
  img = undistortImage(img, cameraParams);
  imgColor = imread(strcat(directory, '\', FilesColor(i).name));
  imgColor = undistortImage(imgColor, cameraParams);
  imshow(img);

  imgGray = rgb2gray(img);
  PtsTourn = [];
  PtsColor = [];
```

The previous script shows how to capture the consecutive images (with and without laser). The images are undistorted. The next steps (not showed; it is your homework) consists of detecting the laser peaks with subpixel accuracy, calculate the in the X(laser)-ZLaser plane through the inverse of the homography, and put the corresponding value in an output file (together with the color of the pixel). The format of one output point cloud is (X, Y, Z, R, G, B):



As an example, we took pictures from a cup:

Resulting in the following point cloud: