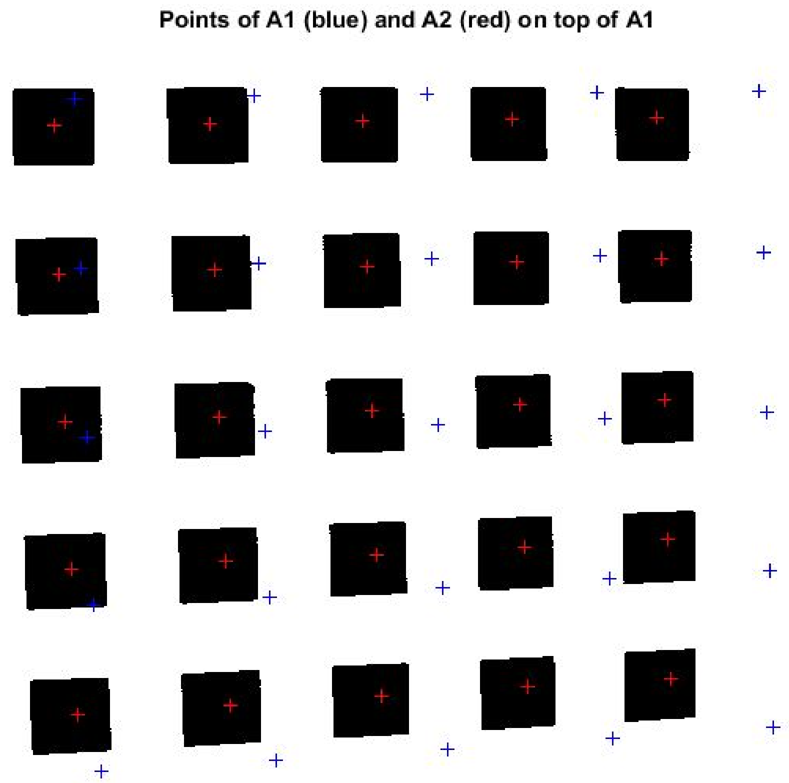
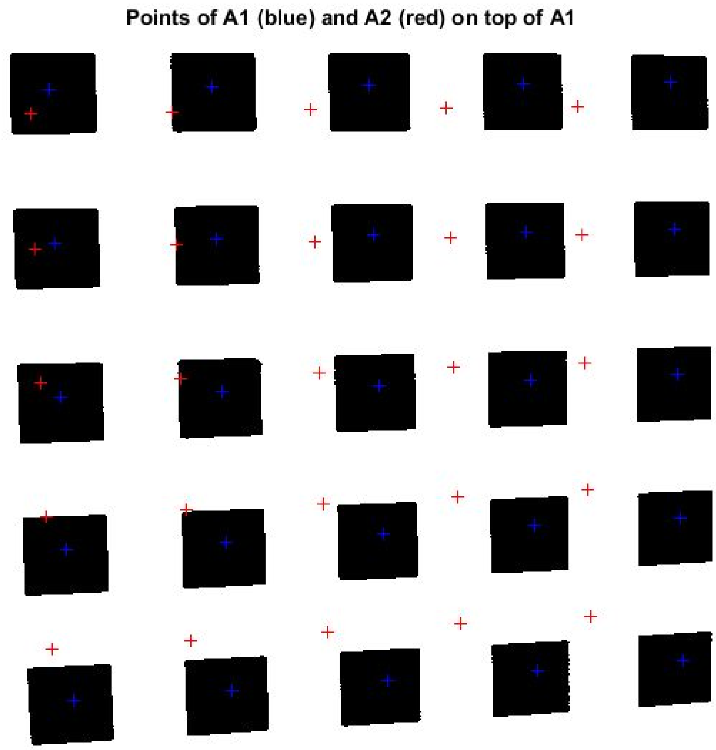
TP4 : Computer Vision
Aurélien Bernier Levalois & Yoann Fleytoux

Report : Part 1
%See TP4_1.m for details of answers
It is intended to carry out the calibration of a chamber by doing the calculation of the matrix Of projection. The calibration process integrates the following steps:
1. Acquire the coordinates of the calibration points belonging to different planes of Calibration.
Note: The 3D coordinates can be obtained by the user manually. The 2D coordinates are to be obtained in the image by calculating the center of mass of the rectangles / squares placed in a calibration plane. The calibration pattern may consist of an equally spaced rectangle / squared array. The distance between the centers of the rectangles / squares must be known precisely. Consider the top left-hand point coincident with the origin of the OBJECT coordinate system, and the X and Y axes of the frame aligned with the rows and columns of the rectangle / squares matrix. Get the 3D coordinates of the Rectangles /



squares



2. Fill in the matrix A once the coordinates [X w, Y w Z w] and [x f, y f] of each of the points considered for calibration are known. Must use points that belong to more than one plane.
Note: The coordinates [x f, y f] are obtained by calculating the centers of mass in the image.

3. Calculate the pseudo-inverse matrix of A.

4. Calculate the elements of matrix M, considering M 12 = 1

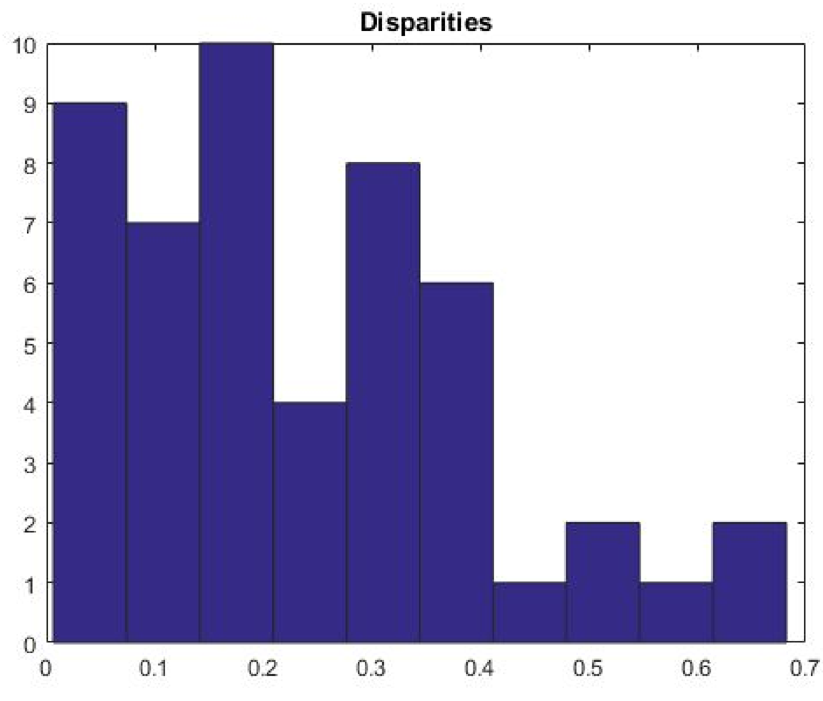
5. Once the perspective matrix M is known, project the 3D calibration points into Image (p = M. P w = [x 'f, y' f]), and compute the disparity in the image. The disparity is calculated for each point through

$$D = \sqrt{\left(x_f - x'_f\right)^2 + \left(y_f - y'_f\right)^2}$$

6. Calculate the maximum, minimum, median, mean and standard deviations.
This parameter is a good indicator of the quality of the calibration performed and "also" of the quality of the calculated model parameters. A perfect calibration gives zero mean disparity, that is, all points projected on the image coincide with those calculated through the perspective matrix.

```
minD =

    0.0064


meanD =

    0.2348


medD =

    0.1976


stdD =

    0.1680


maxD =

    0.6822
```

7. Display a histogram of disparity values.



2a parte - 2nd part

See TP4_2.m
Results with A1_2

1 - In calculating the matrix M instead of imposing the constraint M 12 = 1 use the decomposition in Singular values as described in the theoretical class;

M =

0.0380	-0.0002	-0.0045	-0.6763
0.0015	0.0403	0.0015	-0.7345
0.0000	0.0000	-0.0000	0.0068

2-Calculate the intrinsic and extrinsic parameters (from matrix M) as described in the theoretical class;

K =

0.0383	0	0.0000
0	0.0404	0.0000
0	0	1.0000

R_T =

0.9932	-0.0064	-0.1165	-17.6788
0.0378	0.9986	0.0374	-18.1990
0.0000	0.0000	-0.0000	0.0068

3-Calculate the maximum, minimum, median, mean and standard deviation differences as in part 1.
minD =

0.0056

meanD =

0.2342

medD =

0.1912

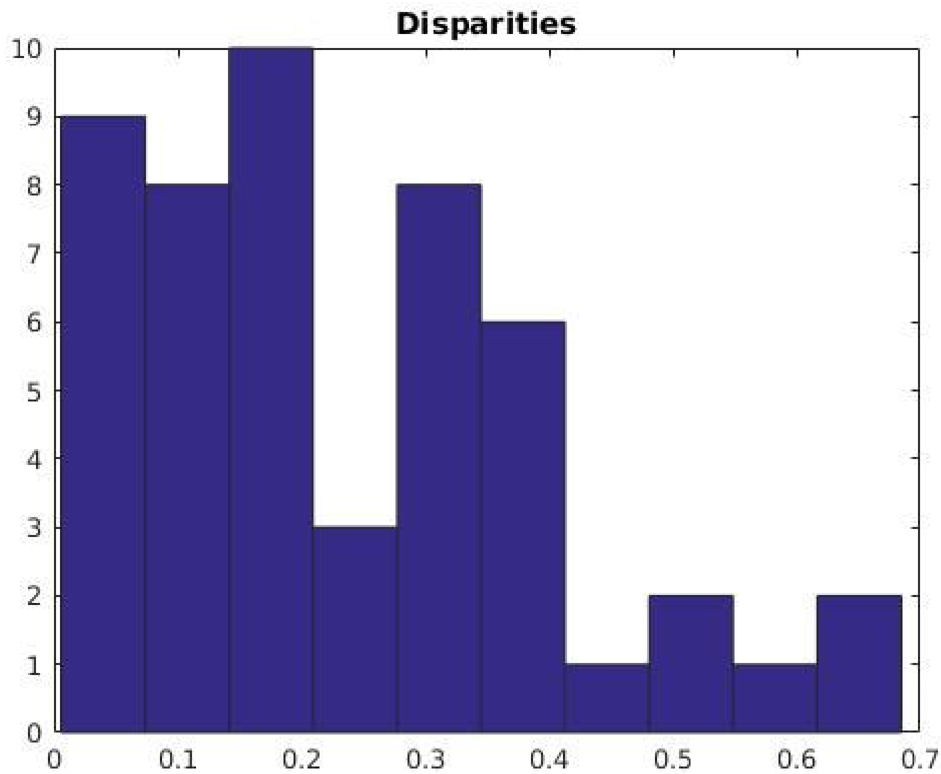
stdD =

0.1687

maxD =

0.6846

4-Present a histogram of the disparity values.



5-Calculate the intrinsic and extrinsic parameters using the QR decomposition. Compare the results with those obtained in the previous paragraphs and make a critical analysis.

Q =

-0.9992	0.0401	-0.0000
-0.0401	-0.9992	-0.0001
-0.0000	-0.0001	1.0000

R =

-0.0380	-0.0014	0.0044	0.7053
0	-0.0403	-0.0017	0.7067
0	0	-0.0000	0.0069

We can see that both M=K*R_T=Q*R and that R and K are quite similar (if we consider the absolute value).

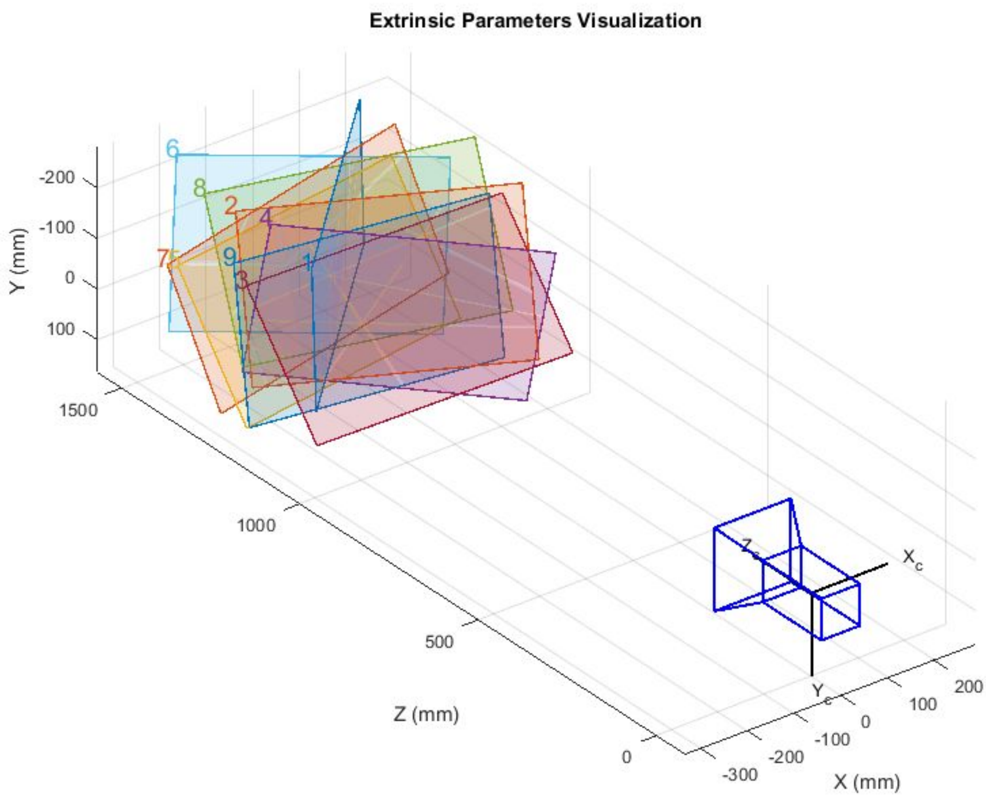
3a parte - 3rd part

%See TP4_3.m for details of answers

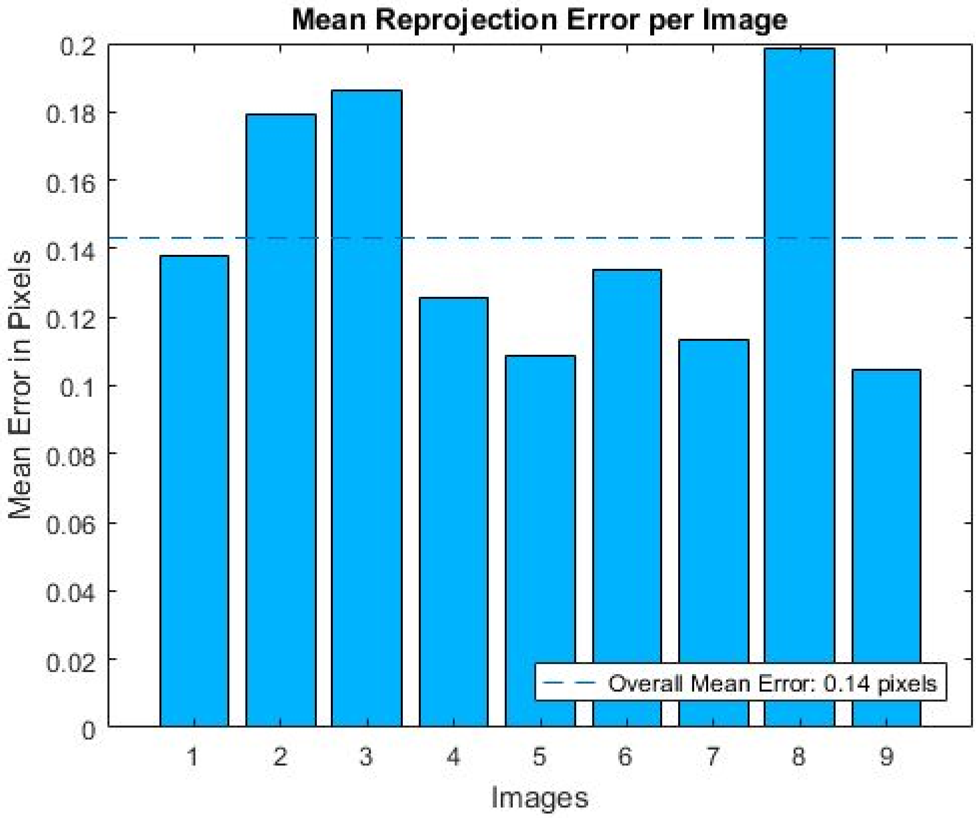
Now consider the images "proj_image_i.jpg" (i = 5 ... 13). The side of each square measures 38 mm.

A) Calibrate the camera that acquired these images using the following Matlab functions:
--" detectCheckerboardPoints";
--"generateCheckerboardPoints";
--"estimateCameraParameters";

B) Use the "showExtrinsics" function to represent the location of the pattern in the Camera coordinate system and to represent the camera in the Coordinates of the two patterns;



C) Use the "showReprojectionErrors" function to determine the histogram of reprojection


Mean Reprojection Error per Image
Mean Error in Pixels
Images
Overall Mean Error: 0.14 pixels
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
2
3
4
5
6
7
8
9

D) Use the code you developed in Part 2 and compare the calibration results.

```
cameraParams.RotationMatrices
ans(:,:,1) =

    0.7281   -0.1069    0.6770
    0.0833    0.9942    0.0675
   -0.6804    0.0072    0.7328


ans(:,:,2) =

    0.8650   -0.0377   -0.5003
   -0.2054    0.8832   -0.4216
    0.4578    0.4675    0.7563


ans(:,:,3) =

    0.9783   -0.1079   -0.1768
   -0.0539    0.6915   -0.7204
    0.2000    0.7143    0.6707


ans(:,:,4) =

    0.8274    0.1583   -0.5388
   -0.1011    0.9858    0.1343
    0.5524   -0.0566    0.8317


ans(:,:,5) =

    0.9950   -0.0526    0.0846
    0.0891    0.8495   -0.5200
   -0.0445    0.5250    0.8500


ans(:,:,6) =

    0.9179    0.1691   -0.3589
   -0.2861    0.9089   -0.3035
    0.2749    0.3813    0.8826


ans(:,:,7) =
```

```
    0.9721   -0.2332   -0.0261
    0.2231    0.9531   -0.2042
    0.0725    0.1927    0.9786


ans(:,:,8) =

    0.9576   -0.0221   -0.2873
   -0.1417    0.8320   -0.5364
    0.2509    0.5544    0.7936


ans(:,:,9) =

    0.9866   -0.0089   -0.1632
   -0.0189    0.9856   -0.1679
    0.1623    0.1688    0.9722


cameraParams.FocalLength =

   1.0e+03 *

    2.0732    2.0706


cameraParams.IntrinsicMatrix =

   1.0e+03 *

    2.0732         0         0
         0    2.0706         0
    0.6900    0.5690    0.0010
```