Practical Exercise

PI 2.1 .- Calibrate a simulated camera
Related to lecture Camera Calibration

Description: Calibrate a simulated camera. Construct the transformation matrix from a set of parameters. Get 3D and 2D points. Calibrate by using the method of Hall. Check the accuracy against the increase of noise in the image points.

Workload: 16 hours

- 2 preparatory hours
- 4 hours in the laboratory (programming / solving doubts)
- 4 hours programming
- 6 hours for reporting the work

Programming platform: Matlab

Part 1

Step 1. Define the intrinsic parameters and extrinsic parameters with the following values:

```
au=557.0943; av=712.9824; u0=326.3819; v0=298.6679;
f=80 mm.;
Tx=100 mm.; Ty=0 mm.; Tz=1500 mm.;
Phix=0.8*pi/2; Phiy=-1.8*pi/2; Phix1=pi/5;      Euler XYX1
Image size:640x480
```

Step 2. Get the intrinsic and extrinsic transformation matrices

Step 3. Define a set of 3D points in the rang [-480:480;-480:480;-480:480]. Note the points should be non-linear and non-coplanar. At least you need to define a set of 6 points. It's not necessary to demonstrate mathematically the non-linearity/non-coplanarity, just define 6 points randomly in the 3D space.

Step 4. Compute the projection on the image plane by using the camera transformation matrix. Do not remove the subpixel precision of the obtained points.

Step 5. Open a window in matlab which will be used as the image plane and draw the 2D points. Are the points well spread in the image plane? Will the distribution of points in the image affect the accuracy in the computation?

Step 6. By using the points of Step 3 and their projection obtained in Step 5, compute the 3x4 transformation matrix by using the method of Hall.

Step 7. Compare the matrix obtained in Step 6 to the one defined in step 2.

Step 8. Add some Gaussian noise to all the 2D points producing discrepancies between the range [-1,+1] pixels for the 95% of points. Again repeat step 6 with the noisy 2D points and the ones defined in step 3. Compare the obtained matrix to the one you got in step 6 with the non-noisy points. Now compute the 2D points with the obtained matrix and compare them to those obtained in step 4 (you can check accuracy computing the discrepancy between points)?

Step 9. Increase the number of 3D points up to 10 points and then up to 50 points and repeat step 8. More the points we use more accurate is the matrix obtained.

Part 2.
Step 10. Define the vector X of the method of Faugeras. Compute X using the points of step 3 and 4, without noise. Extract the camera parameters from both computations. Compare the obtained parameters with the ones defined in Step 1.

Step 11. Add Gaussian noise to the 2D points (produce noise so that the 95% is in the range [-1,1], then in the range [-2,2] and finally in the range [-3,3]) and compute vector X (repeat step 10) for each rang. Which method is more accurate (Faugeras or Hall) with such noise, compute the accuracy from 2D point discrepancy?

Part 3.
Step 12. Open another window in matlab and draw the world coordinate system, the camera coordinate system, the focal point, the image plane, and both 3D points and their corresponding projections on the image plane by using the parameters of step 2 and points without noise. Check if camera position corresponds to (tx, ty, tz) defined in step 2. Check if all the optical rays cross at the focal point? Check if 2D points lie on the optical rays defined by the 3D points?

Deadline: Sunday, 15th March 2015
Note: Before the deadline send an email to mhabib82@hotmail.com including:

- your name and family name
- The matlab file(s) indicating the main file, if necessary.
- The write-up of the assignment (only PDF or WORD) (should not exceed 8 pages in 12 point size font) including:
    - Explain how you have solved every step
    - Include the .m code that solve each step
    - Include the obtained results per step
    - Discuss, if necessary, the results obtained.
    - In the end of the document include general comments & suggestions regarding the adequacy of the exercise, difficulty, workload, etc.