Enhanced visualization: graded exercises

The following exercises on camera calibration will be graded.

A report in the form of one or two Jupyter notebooks should be delivered through moodle in the section named *First graded report*.

You can find below the rules that you should follow:

- You can work in groups with no more than 3 students and all the students in a group should belong to the same lab work group. In case the number of students in a labwork group is not a multiple of 3, one group of 4 students is allowed. You should ask your lab work instructor for a permission to form a group of 4 students.
- You should upload all your notebooks and files in a zip file. The zip file name should contain your family names, *e.g. dupont_denis_li.zip*.
- The deadline for uploading the report is 21/03/2021 23h59. You will not be able to upload your report through moodle passed this date, but you can still send your report to your lab work instructor. There will be a penalty of -4 points if you deliver your report after the deadline. Each extra day of delay will lead to an extra penalty of -4 points.

I. Calibration with planes

A. Calibration with OpenCV In this part we are going to follow the OpenCV tutorial on camera calibration. It is a tutorial that does calibration considering distortion[1].

[1] Source: `http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html`

Download the Python notebook *calibration.ipynb* and the folder *left.zip* from *moodle*. Extract the images from *left.zip* inside the same folder where the Python notebook is located.

1. Explain in details what is done in Cell 1 and 2.
2. Read the documentation of the command cv2.calibrateCamera and explain briefly which calibration method is used.
3. What are the output variables ret, mtx and dist?
4. What are the output matrices rvecs and tvecs?
5. What is the relationship between rvecs and the rotation matrix **R** in the extrinsic matrix? Which command from OpenCV can be used to retrieve **R** from rvecs ?
6. Compare the output images of Cell 3 and the original image. What is the purpose of the commands in Cell 3?
7. What quantity is evaluated in Cell 4? Is this quantity expected to have a high or a low value?

B. Teapot on the checkerboard In this part, you will use the calibration results from the previous part to project an animation of a moving teapot on the checkerboard images. The projections will be carried out only on the images for which the corners of the checkerboard have been found. You will consider a simplified projection model where the effects of distorsion are neglected. For the calibration part you can use OpenCV as in part A, however for the calculating the projections you will rely on your own code.

The 3D coordinates corresponding to the points on the teapot surface can be found in the file teapot.txt in *moodle*.

1. Load the file teapot.txt and build a matrix containing in its column vectors the 3D points forming the surface of the teapot in homogeneous coordinates.

2. For each of the images where ret=True, build the corresponding camera matrix **M**, from mtx, rvecs and tvecs.

3. Uniformly scale the teapot by a factor of $1/2$. Since the z axis in the reference frame points towards the back of the checkerboard, to be able to see the teapot correctly on the top of the teapot, you should also scale the z coordinate of each point by a -1 factor.

4. Using the teapot scaled points and the M matrices you have obtained, calculate the projections. Using pyplot's functions imshow and scatter (with the optional argument s=2) display the teapot projections. For the image corresponding to file left1.jpg you should obtain a figure similar to Fig. 1.

5. For 3 different images create animations where the teapot rotates around its center (rotation around its z axis) at the same time that it translates in circles on the top of the checkerboard. The center of the circular movements should correspond to the center of the checkerboard and the radius of the circles should be $r = 2.5$. See Fig. 2.

   To create the animation you can use similar functions that have been used in Assignment 2.

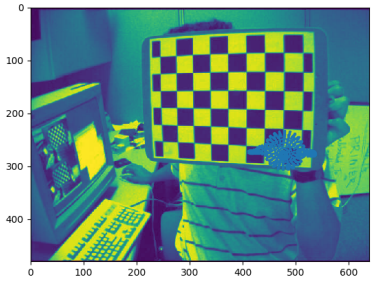

Figure 1: Projection of the teapot on the top of the checkerboard.

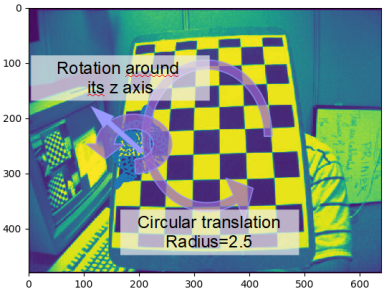

Figure 2: Circular movements and rotation of the teapot on the top of the checkerboard.

II. Goal

In this part you will use the calibration results previously obtained in the lab work for the image goal.jpg.

A. Ball's position

1. We would like to insert the image of a virtual ball in the goal image. The center of the ball should correspond to the origin of the

reference frame (not the position of the real ball) : $\mathbf{u}_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ m. Using the camera matrix $\mathbf{M}$ obtained during the lab work, evaluate the image position of the center of the ball.

2. Using pyplot's imshow and scatter, draw a small circle on the position that you have calculated in the previous question. Does the position of the ball look right?

3. Explain why we cannot obtain the 3D position of the real ball in the image based only on its image coordinates, even if the camera is calibrated.

   Could you propose which supplementary information you could use to retrieve the ball's position?

B. Players' positions In this part you will retrieve the positions in the world reference frame (same as the one from the labwork) of 11 players (positions in meters). The positions of the players will correspond to the coordinates of one of their feet. Since the selected coordinates will correspond approximately to points in the plane $x_w - z_w$ you will assume that $y_w = 0$m.

The 11 players whose positions should be retrieved are indicated by letters (A to K) in Fig. 3 below.

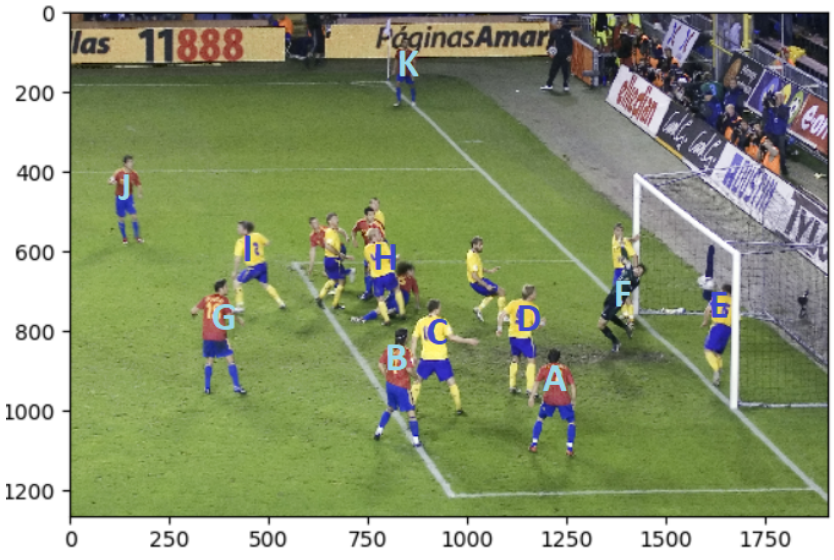

Figure 3: The positions in meters of the players indicated by letters (A - K) in this figure should be retrieved from the pixel position of one of their feet.

1. Obtain the matrix $\mathbf{M}'$ of dimensions 3×3 that allows you to retrieve the position of the foot of a player in the image $\mathbf{u}_{im}$ in homogeneous coordinates. *Hint:* $y_w = 0$.

2. How can you test if transformation from $\begin{bmatrix} x_w \\ z_w \\ 1 \end{bmatrix}$ to $\mathbf{u}_{im}$ is invertible? Is this transformation invertible?

3. Obtain the pixel positions of one of the feet of each of the 11 players. Pick only the pixel positions of feet which are touching the ground.

4. Retrieve the positions $\begin{bmatrix} x_w \\ z_w \end{bmatrix}$ in meters of the 11 players. Discuss whether your results seem adequate based on the known dimensions of the field (goal width, goal area width, etc).