İTÜ

BLG634E 3D Vision

Guideline for submitting your homework: In submitting your homework:

i. Comment your codes clearly.
ii. All your code should be written in Python language.
iii. Do not send your HWs via e-mail. No exception!

In this assignment, you will calibrate a camera and perform augmented reality (AR) by projecting a 3D model into an image.

Task 1: Camera Calibration

First, we'll need some images to perform the calibration. Use the images from **Checkerboard** folder. These images are obtained through stereo vision. The files **cameraParamsLeft.mat** and **cameraParamsRight.mat** contain camera parameters for each camera.

Use camera calibration functions from OpenCV. Display a scatter plot of the reprojection errors. In practice, reprojection error of less than 1 mm is typically considered "good", particularly for a webcam. Show the distorted images and lens undistorted images. Note the change in appearance, particularly further away from the centre of the image.

Once you are finished with the final nonlinear optimization (bundle adjustment/refinement), you have the camera calibration parameters, say cameraParams.

First, check the camera intrinsics, in terms of the focal lengths, principal point (centre of image), skew, lens distortion, and intrinsic matrix. There should be also N sets of extrinsic parameters, one for each image. Now, form you full camera projection matrix M.

Task 2: Augmented Reality (AR)

With the camera intrinsics and extrinsics known, we can project from 3D space into the images. To do this, we must:

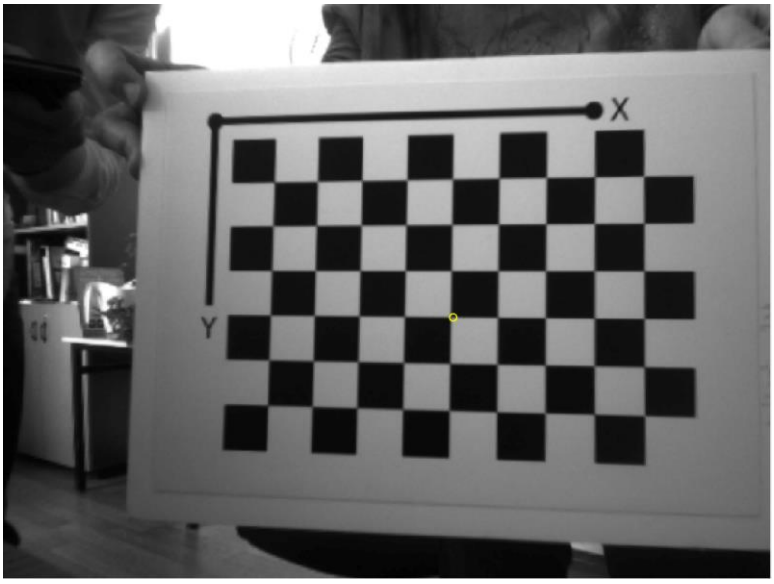
1. Undistort the image
2. Compute the projection matrix
3. Project 3D points to 2D points

Now: Create a new script. In this script, read and display one of the calibration images you saved however show the image in 3D space as you've done before. Also load the previously saved cameraParams. In your script, write code to undistort the image using the camera parameters. Then, create the projection matrix using the intrinsics, rotation matrix and rotation vector for the image. You can consult the lecture notes.

Then, in your script, you can project and display 3D points into the image. You can benefit from the following pseudocode.

```
P = [25 0 0 1]'
p = M*P
p = p / p(2)
scatter(p(0), p(1))
```

Here, M is the overall camera calibration matrix. In this example, the P is a homogenised 3D point $[25, 0, 0, 1]^T$, and is being projected to a 2D point p in the image. The 2D point is dehomogenised by dividing by the w component, and rendered as a yellow circle. Try other 3D points – for example, can you reproduce this image?



Now, project our famous cow model into the image translated to position (A, B, Z=0) and scaled by a factor of S by using the camera matrix M. Play with different scale and translation or even rotation parameters of the model and see if you can place it onto the image as follows. The cow should be on the center of the checkerboard and you should be sure about this by rotation the plane.