CS635: Assignment #1 – Camera Calibration

Out: January 24, 2025
Due: February 7, 2025

Objective

The objective of this assignment is for you to calibrate the parameters of a digital camera. The theory of camera calibration is simple but in practice it is much harder than it seems. Precision is very important: small errors can lead to big calibration errors.

You are to base your work off the Tsai camera model and to use an optimization package to do the calibration work. You may use reasonable initial guesses to start off the optimization but ultimately you have to let the optimizer do the work. The verification of proper calibration is by visual re-projection of the calibration points and by reporting the optimization error.

This assignment requires some mental planning work, some lab work, and some programming. The whole assignment is not that much work and mostly requires you to understand the process. I do recommend STARTING EARLY! (i.e. -- you will not be able to complete the assignment if you start the day before its due --).

Detailed Description

Step 0 – Internal Parameters Calibration (33%)

In this assignment, you will calibrate a digital CCD camera. If you have a digital camera of your own (e.g., a phone), you can use it. If you do not have one, or prefer not to use it, I can lend you a camera for some hours so that you can take the necessary pictures (which at the minimum are just two pictures).

To perform the calibration, you will need a calibration pad. You can have fun making your own – remember it needs to be *flat*. In the lab, there are calibration pads that you can optionally use. **PLEASE TREAT IT CAREFULLY**. The pad is flat and the checkerboard pattern is regularly spaced at intervals of 4 inches (or 101.6mm). When you take your pictures, make sure the images have some perspective foreshortening (i.e., the camera image plane is not perpendicular to the calibration pad plane).

You will need a minimum of one picture to internally calibrate the camera. In the picture, you will see checkerboard patterns. You can easily compute the 3D position of the corners. Use the mouse to click on the corners; this gives you their 2D projections. Now you have many 3D to 2D correspondences (x_i, y_i) and you can calibrate the camera model as described in class.

The minimum set of internal camera parameters is:
a) Focal length (f)
b) Image plane center (p_x, p_y)
c) Radial lens distortion correction factor (k)

Note: Depending on your camera k might be insignificant (e.g., $k = 0$). It is ok if you assume zero, but then don't use a camera with obvious lens distortion.

The optimization will take as input (f, p_x, p_y, k) set to initial guesses and compute their optimized values as output. The observations (u_i, v_i) and their 3D equivalents (x_i, y_i, z_i) are used as data for the optimization. These 5-tuples are fed into a nonlinear least squares optimization. You may look at the last slides of the camera calibration lecture, in particular those of bundle adjustment. Choose either to directly solve for matrix elements or for the functional form. For extrinsic parameters, use a guess.

NOTE: To perform this optimization and all those needed for this assignment, I recommend you use Levenberg-Marquardt's nonlinear least squares method (lmdif). An implementation and sample interface code is available on the course website. The implementation is from the Numerical Recipes in C code. The file lmdif.c is "C" code but the interface sampleopt.C is a "C++" program. The actual lmdif.C code is not very human readable – you are only meant to see its function call syntax and usage example. However, you may use other implementations if you prefer.

Step 1 – External Parameters Calibration (33%)

Once your camera is internally calibrated (approximately), you must take two pictures of the calibration pad from significantly different viewing positions (i.e., for a 2-meter camera-to-pad distance, the two camera viewing points should differ by about 1 meter) and estimate the external camera pose of each picture using the same set of internal camera parameters. *You will need to prevent the camera from changing focal length and zoom settings. Moreover, ideally the aperture should not change either but it is not mandatory.* You may in fact choose the same images as for internal calibration. The objective though is to verify that using the *same internal camera parameters* yields reasonable external-parameter convergence for two (very different) pictures.

Similar to internal calibration, obtain a corresponded set of 2D-3D feature points. Use these features points to calibrate the external camera parameters, namely the 3D position of the camera's center of projection (t_x, t_y, t_z) and its 3D orientation (total of 3 positional scalars and 3 angles). Your optimization will take as input guesses for the pose and return the optimized pose.

Step 2: Linear Estimation of Camera Instrinsics and Extrinsics (17%)

Based on the slides and Zhang's paper, implement a linear estimation to camera extrinsics/intrinsics. You may use the slides and/or other sources but you must write this extension (i.e., you will not receive credit if you simply cut and paste code from elsewhere). This option should be able to "used" or not with something like a command line parameter or a GUI button. After you perform this initialization of camera extrinsics/instrinsics, you can/should perform the same optimization as in Step 1 and/or 2.

"A Flexible New Technique for Camera Calibration", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

Step 3 – Rendering and Verification (17%)

In theory, you can perform the entire assignment using just two images from two viewpoints. Assuming this, below are the minimum presentation requirements (if you have more images, no problem, just provide the equivalent functionality).

For this assignment, please demo a program that can show both of your captured images (one a time, or both simultaneously) and via a user interface can draw on top of the image the following:

- All user clicked 2D feature points
- All subpixel accurate refined 2D feature points (if implemented)
- A reprojection of the points used for external pose estimation (after calibration)

- A reprojection of the location of the camera viewpoint and its orientation (e.g., by drawing the camera's coordinate frame) relative to the points used for calibration; please also draw the calibration points. This would be a bird's eye view of the camera and calibration pad.
- The numeric values of the per-point re-projection error (in pixels, and in world-units) and the average total re-projection error (this is essentially the error of the numerical optimization process).

Grading/Demonstration

Your demonstration will consist of you showing me your program in a short demo session to be arranged (in my office). On or before the due date, please upload to Brightspace a zip file with a single directory called **"<your-name>-asgn1**" containing:

- Windows PC Executable
- Data files (i.e., images)
- Other necessary files, DLLs, etc…

During your demo session, I will use the provided zip-file to grade your program. Your grade will be influenced by how well your particular camera is calibrated, by the presentation and usability of your program, and by how well you complete the assignment requirements. To give you some reference, a calibration of near one-pixel error is "good". More than a few pixels is "large error".

In this assignment, you may collaborate *only* to help with the mechanics of the assignment (e.g., using the pad, your camera, etc). *Everybody must take their own pictures*! Practically speaking, this means that nobody should have the same pictures or calibration results.

If you have questions, please come see me ASAP – do not wait until the last moment.

Have fun and good luck!