Homework 1: AutoCalib!

Rohan Chandra

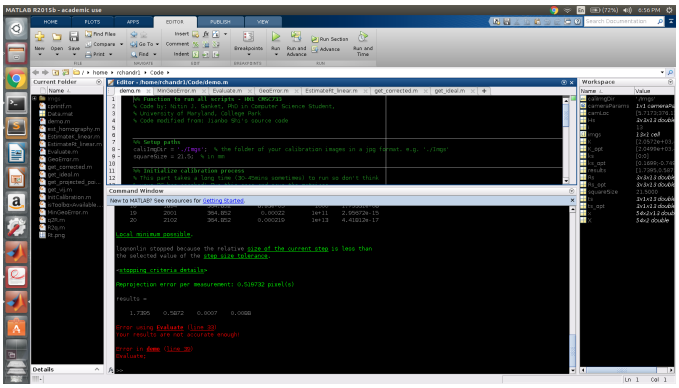

Fig. 1. Projection Error

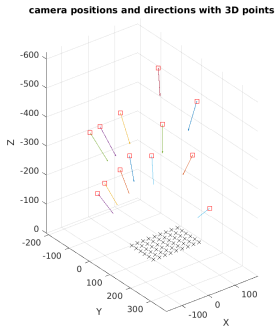

Fig. 2. Rt Image

***Abstract*—This homework aims to help us understand how a camera is calibrated and I also learned how even an error as small as 0.0072 can be disastrous enough to destroy your experiment.**

I. Introduction

The pipeline for this homework was as follows:

A. Intrinsic Matrix,(K)

Fortunately, we were already given the world coordinates and its corresponding image coordinates and we did not have to worry about getting these points. What we needed to do was figure out a homography between the world and image points and get a 3D array of homography matrices. By following the derivations given in the homework reference, we calculated the parameters for the intrinsic matrix, K using the homography matrices. This K serves as an initial estimate for our calibration.

The reason why we need to solve an optimization problem at the end using all our initial estimates is that we are attempting to solve for an ill-posed problem. More specifically, the forward problem being defined as solving for data given a model, we are trying to solve the inverse problem(ill-conditioned) where we are attempting to solve for the model using the data. This is like trying to find a needle in the proverbial haystack and we need to use a non-linear optimizer to find the needle, i.e. the optimum model that allows us to fit the data correctly.

I shall talk mostly about inverse problems at the end of the pipeline because they are intimately connected to this homework and also because I find them fascinating because of the mathematical rigor involved.

1) Extrinsic Matrix,(Rs, ts)): Extending our previous section, we extend the equations after getting K to solve for Rs and ts. This was nothing great and was easy to code. At the end of this step, we are ready with our initial estimates and will now attempt to solve for the optimum parameters.

2) Non-Linear Opimization: We used the levenberg marquardt algorithm to solve the non linear least squares problem. The non linear least squares is different from normal least squares in that the linear term Ax is now a function in f. In other words, we are atempting to optimize $\|b - f(Ax)\|$.

II. Inverse Problems

I shall describe a little more about inverse problems. The forward problem is to try to find exact or approximate functions that describe various physical phenomena such as the propagation of sound, heat, electromagnetic waves, etc. In these problems, the media properties (expressed by the equation coefficients) and the initial state of the process under study (in the nonstationary case) or its properties on the boundary (in the case of a bounded domain) are assumed to be known. However, it is precisely the media properties that are often unknown. This leads to inverse problems, in which it is required to determine the equation coefficients from the information about the solution of the direct problem. Most of these problems are ill-posed (unstable and sensitive with respect to measurement errors). At the same time, the unknown equation coefficients usually represent important media properties such as density, electrical conductivity, heat conductivity, etc. Solving inverse problems can also help to determine the location, shape, and structure of physical phenomena.

The reason why inverse problems are ill-posed most of the times is because in a linear system, either it will have no solutions or it will have too many solutions in case the matrix is rank-deficient. In these cases, we need what is called a pseudo-inverse which is what is achieved by solving a least-squares problem. This is essentially what the optimization problem is doing in our homework.

III. Result

The screenshot in figure 1 shows that the **projection error is around 0.51**. Additionally, the code produces optimum parameters for the intrinsic matrix K, the rotation matrix Rs and the radial distortion parameter Ks. However, an error of 0.0072 is reported for the translational vector.