EX5: Nonlinear optimization

Dr. Haoang Li, Daniil Sinitsyn, Sergei Solonets, Viktoria Ehm
Computer Vision Group, TU Munich

Deadline: 12.07.2023 11:00 CET

Preamble

This task is a logical continuation of Exercise 3. In Exercise 3, you implemented a method to find camera calibration based on the observation of a known object. In this task, your objective is to refine a given calibration using non-linear optimization. Non-linear optimization is commonly used to estimate camera distortion, as it cannot be determined through closed-form solutions. For the sake of simplicity, we are not requesting you to estimate distortion in this task. Therefore, the solution of the non-linear optimization will coincide with the solution from Exercise 3. However, this task requires you to employ a more general approach that can be applied to any camera model.

Problem description

In this task, your objective is to refine a calibration of a pinhole camera using a unit cube. You are given projections of vertices of a unit cube as in EX3, followed by an initial intrinsic and extrinsic parameters. Your goal is to refine this calibration using Gauss-Newton optimization.

Let x_i and p_i be a known 3D point and it's observation on camera. Given a current pose estimate T_j and current calibration parametes fx_j, fy_j, cx_j, cy_j, we model our residuals as:

$$\begin{pmatrix} r_{2i-1} \\ r_{2i} \end{pmatrix} = \Pi_{fx_j,fy_j,cx_j,cy_j}(exp(\epsilon^\wedge) * T_j * x_i) - p_i$$

Then we are to optimize a loss function:

$$L = \sum_i r_i^2$$

Part 1

In the first submission you are asked to calculate the values of the residuals and their derivatieves w.r.t. ϵ and fx_j, fy_j, cx_j, cy_j at $\epsilon = 0$ You should report a vector $r \in \mathbb{R}^{16}$ and jacobian matrix $J \in \mathbb{R}^{16\times 10}$, where the i-th row of J is $(\frac{\partial r_i}{\partial fx} \; \frac{\partial r_i}{\partial fy} \; \frac{\partial r_i}{\partial cx} \; \frac{\partial r_i}{\partial cy} \; \frac{\partial r_i}{\partial \epsilon}|_{\epsilon=0})$ You have the option to calculate the Jacobian matrix numerically or analytically. For an analytical solution, please refer to the tutorial available here. You can find the relevant information in Section 10 and Appendix A.

Part 2

In your second submission, you are required to use the values calculated in the first submission and perform 10 steps of the Gauss-Newton algorithm. After calculating your step update δ, you should update T_{j+1} as $exp(\delta_{\epsilon}^{\wedge})T_j$, while keeping simple additive updates to camera intrinsics. You should report calibration matrix and camera transformation after each iteration.