CSE 252B: Computer Vision II, Winter 2016 – Assignment 3
Instructor: Ben Ochoa
Due: Tuesday, February 16, 2016, 11:59 PM

Instructions

- Review the academic integrity and collaboration policies on the course website.
- This assignment only contains programming problems that must be completed using MATLAB.
- You must prepare a report describing the problems, and your solutions and results. The report must contain enough information for a reader to understand the problems and replicate your work without also having the assignment (i.e., this document).
- Your report will be a pdf file named `CSE_252B_hw3_lastname_studentid.pdf`, where `lastname` is your last name and `studentid` is your student ID number. The report must be prepared using LaTeX.
- All of your MATLAB source code must be included in an appendix of your report. You may find the `listings` package useful for this.
- You must create a zip file named `CSE_252B_hw3_lastname_studentid.zip`, where `lastname` is your last name and `studentid` is your student ID number. This zip file will contain the pdf file and a directory named `code` that contains all of your MATLAB source code.
- Submit your completed assignment by email to `bochoa@ucsd.edu` **and** `nkinkade@eng.ucsd.edu`. The subject of the email message must be `CSE 252B Assignment 3`. Attach the zip file to the message.
- It is highly recommended that you begin working on this assignment early to ensure that you have sufficient time to correctly implement the algorithms and prepare a report.

Problems

1. **Programming: Estimation of the camera pose (rotation and translation of a calibrated camera) (65 points)**

   (a) **Outlier rejection (20 points)**

   Download input data from the course website. The file `hw3_points3D.txt` contains the coordinates of 60 scene points in 3D (each line of the file gives the $\tilde{X}_i$, $\tilde{Y}_i$, and $\tilde{Z}_i$ inhomogeneous coordinates of a point). The file `hw3_points2D.txt` contains the coordinates of the 60 corresponding image points in 2D (each line of the file gives the $\tilde{x}_i$ and $\tilde{y}_i$ inhomogeneous coordinates of a point). The corresponding 3D scene and 2D image points contain both inlier and outlier correspondences. For the inlier correspondences, the scene points have been randomly generated and projected to image points under a camera projection matrix (i.e., $\mathbf{x}_i = \mathtt{P}\mathbf{X}_i$), then noise has been added to the image point coordinates.

The camera calibration matrix was calculated for a 1280×720 sensor and $45\,^\circ$ horizontal field of view lens. The resulting camera calibration matrix is given by

$$\mathtt{K} = \begin{bmatrix} 1545.0966799187809 & 0 & 639.5 \\ 0 & 1545.0966799187809 & 359.5 \\ 0 & 0 & 1 \end{bmatrix}$$

For each image point $\mathbf{x} = (x, y, w)^\top = (\tilde{x}, \tilde{y}, 1)^\top$, calculate the point in normalized coordinates $\hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$.

Determine the set of inlier point correspondences using the M-estimator Sample Consensus (MSAC) algorithm, where the maximum number of attempts to find a consensus set is determined adaptively. For each trial, use the 3-point algorithm of Finsterwalder (as described in the paper by Haralick et al.) to estimate the camera pose (i.e., the rotation $\mathtt{R}$ and translation $\mathbf{t}$ from the world coordinate frame to the camera coordinate frame), resulting in up to 4 solutions, and calculate the error and cost for each solution. Note that the 3-point algorithm requires the 2D points in normalized coordinates, not in image coordinates. Calculate the projection error, which is the (squared) distance between projected points (the points in 3D projected under the normalized camera projection matrix $\hat{\mathtt{P}} = [\mathtt{R}|\mathbf{t}]$) and the measured points in normalized coordinates (hint: the error tolerance is simpler to calculate in image coordinates using $\mathtt{P} = \mathtt{K}[\mathtt{R}|\mathbf{t}]$ than in normalized coordinates using $\hat{\mathtt{P}} = [\mathtt{R}|\mathbf{t}]$).

In your report, describe any assumptions, including the probability p that at least one of the random samples does not contain any outliers (used to determine the number of attempts to find a consensus set), and the probability α that a given data point is an inlier and the variance σ^2 of the measurement error (both used to determine the distance threshold; hint: this problem has codimension 2). Additionally, show the resulting number of inliers and the number of attempts to find the consensus set.

(b) **Linear estimation (15 points)**

Estimate the normalized camera projection matrix $\hat{\mathtt{P}}_{\text{DLT}} = [\mathtt{R}_{\text{DLT}}|\mathbf{t}_{\text{DLT}}]$ from the resulting set of inlier correspondences using the direct linear transformation (DLT) algorithm (with data normalization). Recall that this method is similar to the one used to estimate the camera projection matrix $\mathtt{P}$, but constraints must be enforced on the left 3×3 submatrix of $\hat{\mathtt{P}} = [\mathtt{R}|\mathbf{t}]$ such that it is a valid rotation matrix $\mathtt{R}$ and the translation vector $\mathbf{t}$ must be scaled to match these constraints. Include the numerical values of the resulting $\mathtt{R}_{\text{DLT}}$ and $\mathbf{t}_{\text{DLT}}$ in your report with sufficient precision such that it can be evaluated (hint: use `format longg` in MATLAB prior to displaying your results).

(c) **Nonlinear estimation (30 points)**

Use $\mathtt{R}_{\text{DLT}}$ and $\mathbf{t}_{\text{DLT}}$ as an initial estimate to an iterative estimation method, specifically the Levenberg-Marquardt algorithm, to determine the Maximum Likelihood estimate of the camera pose that minimizes the projection error under the normalized camera projection matrix $\hat{\mathtt{P}} = [\mathtt{R}|\mathbf{t}]$. You must parameterize the camera rotation using the angle-axis representation $\boldsymbol{\omega} = \ln \mathtt{R}$ of a 3D rotation, which is a 3-vector.

In your report, show the initial cost (i.e., the cost at iteration 0) and the cost at the end of each successive iteration. Show the numerical values for the final estimate of the camera rotation $\boldsymbol{\omega}$ and $\mathtt{R} = e^{\boldsymbol{\omega}}$, and the camera translation $\mathbf{t}$ in your report with sufficient precision such that it can be evaluated.