CSE 252B: Computer Vision II, Winter 2014 – Final Project (suggested)

Instructor: Ben Ochoa

Due date: Friday, March 21, 2014, 11:59 PM

Automatic estimation of the camera pose (rotation and translation of a calibrated camera)

1. Download input data
   `http://vision.ucsd.edu/~bochoa/cse252b-wi14/final_points3D.txt`
   and
   `http://vision.ucsd.edu/~bochoa/cse252b-wi14/final_points2D.txt`
   The file `file_points3D.txt` contains the coordinates of 60 scene points in 3D (each line of the file gives the $\tilde{X}$, $\tilde{Y}$, and $\tilde{Z}$ inhomogeneous coordinates of a point). The file `file_points2D.txt` contains the coordinates of the 60 corresponding image points in 2D (each line of the file gives the $\tilde{x}$ and $\tilde{y}$ inhomogeneous coordinates of a point). The scene points have been randomly generated and projected to image points under a camera projection matrix. Noise has been added to the image point coordinates and false correspondences have been introduced in the data.

2. Camera calibration matrix and normalized coordinates. The camera calibration matrix was calculated for a 1280×720 sensor and 45° 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 $\hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x}$.

3. Outlier rejection. 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 $\hat{\mathtt{P}} = [\mathtt{R}|\mathbf{t}]$).

4. Linear estimation. For the set of inlier correspondences, estimate the normalized camera projection matrix $\hat{\mathtt{P}} = [\mathtt{R}_{\mathrm{DLT}}|\mathbf{t}_{\mathrm{DLT}}]$ 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.

5. Nonlinear estimation. Use $\mathtt{R}_{\mathrm{DLT}}$ and $\mathbf{t}_{\mathrm{DLT}}$ 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 $\hat{\mathtt{P}} = [\mathtt{R}|\mathbf{t}]$. Parameterize the camera rotation using the angle-axis representation $\boldsymbol{\omega} = \ln \mathtt{R}$ of a 3D rotation, which is a 3-vector. Note that the points in 3D are not adjusted, so they are not included in the parameter vector.

6. Results. Indicate the number of MSAC trials executed and the resulting number of inliers. Show the numerical values for the DLT estimate of the camera rotation $\mathtt{R}_{\mathrm{DLT}}$ and translation $\mathbf{t}_{\mathrm{DLT}}$. For the nonlinear estimate, show the initial cost (i.e., 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 $\mathbf{t}$. Provide all source code except for low-level functions (e.g., linear algebra operations).