

Matlab Exercises

Lecture 3: 3D Pose estimation from point correspondences

1) Linear LSE

A) Solve the linear regression problem : $f_i = a_1x_i + a_2$
for the set of points $(x_1, y_1), \dots, (x_N, y_N)$

x	1	2	3	4	5	6	7	8	9	10
y	2.6892	3.3476	3.8128	5.1878	5.6323	6.5843	6.7507	7.2644	8.5423	9.0124

(the “true” line has coefficients $a_1 = 0.7$, $a_2 = 2$)

B) Plot the result, by indicating with a cross ‘x’ the points from the table, and draw the line resulting from LSE, together with the “true” one.

C) Try to modify significantly one of the measurements y, and run again the estimation to see the sensitivity to outliers.

2) Weighted linear LSE

For the same set of points (x_i, y_i) , now introduce weights

x	1	2	3	4	5	6	7	8	9	10
w	1.0	0.5	0.4	0.3	0.1	0.8	0.6	0.8	0.9	1.0

and solve the weighted LSE problem by using the W matrix: $W = \text{diag}(w_1, \dots, w_{10})$.

Try again to see the sensitivity to each measurement, which now is different for each point: for example, it should be much more sensitive to y_1 than y_5 .

3) Nonlinear LSE

Given the following data set

x	1	2	3	4	5	6	7	8	9	10
y	1.7321	-0.0509	-1.4885	-1.8603	-0.7881	1.5522	2.2165	0.8167	-1.0226	-1.9651

Consider now the non-linear model: $f_i = a_1 \sin(x_i + a_2)$, where the state is $\mathbf{s} = [a_1, a_2]$, to be estimated. This is the problem of fitting a sinusoid to a noisy sequence of data.

A) Write a Matlab function that computes the SSD cost function $C(\mathbf{s}) = \sum_{i=1}^N (y_i - f_i(\mathbf{s}))^2$.

Input: the dataset (\mathbf{x}, \mathbf{y}) and the model hypothesis \mathbf{s}

Output: the SSD value $C(\mathbf{s})$

B) Derive the (Nx2) Jacobian matrix of $\mathbf{f} : J = \begin{bmatrix} \frac{\partial f_1}{\partial a_1} & \frac{\partial f_1}{\partial a_2} \\ \dots & \dots \\ \frac{\partial f_{10}}{\partial a_1} & \frac{\partial f_{10}}{\partial a_2} \end{bmatrix}$ at a given state $[a_1, a_2]$

C) Write a Matlab function that makes this computation, that is:

Input: the dataset (\mathbf{x}, \mathbf{y}) and model hypothesis \mathbf{s}

Output: the Jacobian matrix $J(\mathbf{s})$

D) Finally, write a script that does the Gauss-Newton optimization:

0. Start from a state guess $\mathbf{s} = \mathbf{s}_0 = [1, 0]$
1. while(true)
 - a. Compute the error vector $(\mathbf{y} - \mathbf{f}(\mathbf{s}))$
 - b. Compute the Jacobian matrix $J(\mathbf{s})$
 - c. Compute the Gauss-Newton increment $\Delta \mathbf{s} = [(J^T J)^{-1} J^T](\mathbf{y} - \mathbf{f})$
 - d. Increment the estimate $\mathbf{s} \rightarrow \mathbf{s} + \Delta \mathbf{s}$
 - e. If the increment is $\Delta \mathbf{s} < 0.001$, exit the loop
2. Output = final estimate \mathbf{s} .

E) Run the script, and display the results, like for the linear case (estimated sinusoid + measurements \mathbf{y})

F) Repeat step (D) using Levenberg-Marquardt instead of Gauss-Newton, starting with $\lambda=1$, and compare the results

4) 3D point correspondences - Cost function

Write a Matlab function that computes the SSD cost function for a set of point correspondences and a state hypothesis.

NOTE: Use the global transformation of Exercise (5) to project every point of the body to the expected position $f(\mathbf{p}_i)$, and then compute SSD with the measured points \mathbf{q}_i .

Input: A matrix of points in 3D space (Body): $P = [{}^B\mathbf{p}_1; {}^B\mathbf{p}_2; \dots; {}^B\mathbf{p}_N]$

A matrix of points in 2D image: $Q = [\mathbf{q}_1; \mathbf{q}_2; \dots; \mathbf{q}_N]$

The 6 pose parameters \mathbf{s}

Output: The cost function value, $C(\mathbf{s})$, i.e. the Sum of Squared Differences

10) Robust LSE – RANSAC

x	1	2	3	4	5	6	7	8	9	10
\mathbf{y}	2.6892	3.3476	7.0	5.1878	5.6323	6.5843	6.7507	7.2644	8.5423	9.0124

Solve the linear regression problem (exercise 6) with an outlier, in two steps:

A) Implement the RANSAC algorithm on the dataset above, to remove the outliers (it should find only one):

REPEAT 100 times:

- 1 - Pick two points at random
- 2 - Fit a line (a_1, a_2) through the points
- 3 - Set a tolerance around the line $(y \pm 1.5)$ to select outliers for this hypothesis

and keep the case with less outliers (if there is more than one, take just one).

B) After removing the outlier, do the standard LSE estimation (linear) with the remaining points

C) Compare the result with the (non-robust) standard LSE using the full dataset