Weekly Exercises 8

Room: 02.09.023

Wed, 22.06.2016, 14:00-16:00

Submission deadline: Tue, 21.06.2016, 23:59 to laehner@in.tum.de

Mathematics: LBO spectrum

Exercise 1 (3 points)**.** Consider a shape $S \subset \mathbb{R}^3$ that is described by coordinate maps $x_i : U_i \to S$ and $\alpha S \subset \mathbb{R}^3$ which is described by $y_i = \alpha \cdot x_i$. Show that

1. $$\text{area}(\alpha S) = \alpha^2 \text{area}(S) \tag{1}$$

2. $$\lambda_i^{\alpha S} = \frac{1}{\alpha^2} \lambda_i^S \tag{2}$$

3. $$\phi_i^{\alpha S} = \frac{1}{\alpha} \phi_i^S \tag{3}$$

where λ_i, ϕ_i denote the eigenvalues and functions of the LBO. Tip: Solve the exercises in the given order. Think about what you can say about mass and stiffness matrix of αS. You can use the lumped mass matrix for simplicity.

Programming: Multi Dimensional Scaling

Download the new supplementary material. It contains an outline for the whole exercise and code for several side tasks as well as visualizing the solution.

Exercise 2 (5 points)**.** Implement the Multi-Dimensional Scaling approach to find a correspondence between two shapes. `ex8_1.m` already contains an outline for the whole procedure and you can fill in code in `mds.m`, `alignpoints.m` and `extract_matching.m`.

1. `mds.m` should contain the algorithm as explained in the lecture. It takes a distance matrix $D \in \mathbb{R}^{n \times n}$ and a dimension to embed in $m \in \mathbb{N}$. It should return the coordinates of each point in $\mathbb{R}^m$ as a matrix $Z \in \mathbb{R}n \times m$.

   The parameters `epsilon` and `maxI` control the minimum relative progress (if you don't know how to compute it, just skip it) and the maximum number of iterations.

2. `alignpoints.m` should align two points clouds $Z_1, Z_2 \subset \mathbb{R}^m$ with rigid transformations (i.e. translation and rotation). The result should be two points sets $\hat{Z}_1 = \{R_1(z + t_1) | z \in Z_1\}, \hat{Z}_2 = \{R_2(z + t_2) | z \in Z_2\}$ that were aligned by the optimal rigid transformations.

   The translations t_1, t_2 can be found by computing the point clouds' mean. For the rotation we suggest to align the principal axes $a_1, ... a_m \in \mathbb{R}^m$ with the standard Euclidean axes. Note that since the signs of the principal axes are not uniquely determined, the visualization code includes sign parameters that can be altered manually.

3. `extract_matching.m` For each $z \in \hat{Z}_1$ find the $z' \in \hat{Z}_2$ that is its nearest neighbor. This will give you a matching (z, z') for each point in $\hat{Z}_1$. Notice that using this procedure will not necessarily give you a bijection between both shapes. You can use `visualize_matching.m` to visualize your results.

The Matlab functions `mean` and `pca` might be helpful, for nearest neighbor search you can use the function `knnsearch`.