Analysis of Three-Dimensional Shapes
E. Rodolà, T. Windheuser, M. Vestner
Summer Semester 2015

Computer Vision Group
Institut für Informatik
Technische Universität München

Weekly Exercises 8

Room: 02.09.023

Wed, 17.06.2015, 14:15-15:45

Submission deadline: Tue, 16.06.2015, 23:59 to windheus@in.tum.de

Please send in only Latex-PDF. If you have hand-written solutions, please hand them in during the lecture.

Functional maps

We will in the following denote by $\{\psi_i\}$ the standard basis (hat functions) and by $\{\phi_i\}$ the basis of Laplace Beltrami eigenfunctions.

Exercise 1 (One Point)**.** One way to convert a functional map T_F to a point to point mapping is to apply T_F to special functions on the first shape, associated with the vertices (*indicator functions*) and then look for the indicator function on the second shape that is most similar (in the L^2-sense) to the image.

1. Calculate the L^2-norm of the function $f = \sum a_i \phi_i^M$ in terms of the coefficient-vector a.
2. Show that if we choose the indicator functions to be Diracs $\delta_j = M^{-1} e_j$ we can find the most similar indicator functions by comparing the columns of the matrix $C\Phi_M^T$ with the columns of Φ_N^T. Your calculations should start with

$$\left\| T_F(\delta_i^M) - \delta_j^N \right\|_{L^2(N)} = \dots$$

Programming: Functional Maps

Exercise 2 (Two points)**.** Download and expand the file `ex8.zip` from the lecture website. Modifiy the files `spectralFM.m`, `fmToVertexMap.m` and `optimalFM.m` to implement the functions as explained below. You can run the script `exercise.m` to test and visualize your solutions.

`spectralFM.m` The function `C = spectralFM(P)` accepts a functional map `P` and returns an equivalent functional map `C`.

Functional map `P` is given as an n-by-n matrix and transforms the coordinates $\alpha \in \mathbb{R}^n$ of a real-valued function on mesh `M1` with respect to the basis of hat functions $\{\psi_i^{\mathsf{M1}}\}_i$ into the coordinates $\beta \in \mathbb{R}^n$ of a real-valued function on mesh `M2` with respect to the basis of hat functions $\{\psi_i^{\mathsf{M2}}\}_i$. I. e. let $f = \sum_i \alpha_i \phi_i^{\mathsf{M1}}$ be a function on `M1` and let $g = \sum_i \beta_i \phi_i^{\mathsf{M2}}$ the image of `P` at f, then $\beta = \mathsf{P}\alpha$.

Functional map `C` should be an (almost) equivalent representation of `P`, where the functions are given as coordinates with respect to the eigen basis of the Laplace-Beltrami operator. I. e. `C` should be a k-by-k matrix.

What additional arguments will functions `spectralFM(...)` need? Please modify the script `exercise.m` accordingly.

`fmToVertexMap.m` The function `c = fmToVertexMap(C, Phi1, Phi2)` takes a functional map $\mathtt{C} \in \mathbb{R}^{k\times k}$ between the two eigenspaces represented by matrices $\mathtt{Phi1}, \mathtt{Phi2} \in \mathbb{R}^{n\times k}$ and should return a point-to-point map represented by vector $\mathtt{c} \in \{1,\ldots,n\}^n$ that assigns to each vertex on mesh 1 a corresponding vertex on mesh 2 by the following optimization problem:

$$\mathtt{c}_i = \arg\min_j \left\| \mathtt{C}\ \mathtt{Phi1}_i^\top - \mathtt{Phi2}_j^\top \right\|_2^2,$$

where $\mathtt{Phi1}_i^\top$ is the transposed of i-th row of matrix `Phi1` and $\mathtt{Phi2}_j^\top$ is the transposed of j-th row of matrix `Phi2`. Keep in mind that this point-to-point map is not necessarily injective.

Use matlab functions `KDTreeSearcher` and `knnsearch` to implement the function efficiently.

`optimalFM.m` Given some K vertex correspondences between two shapes the function `C = optimalFM(c, Phi1, M1, Phi2, M2)` should compute an optimal functional map $\mathtt{C} \in \mathbb{R}^{k\times k}$ in the least-squares sense as described in the lecture. The correspondences are given by matrix $\mathtt{c} \in \{1,\ldots,n\}^{K\times 2}$, such that the i-th correspondence associates the $\mathtt{c}_{i,1}$-th vertex on shape 1 to the $\mathtt{c}_{i,2}$-th vertex on shape 2. `Phi1, M1, Phi2, M2` are the eigenbasis and mass matrices for shape 1 and shape 2.

You can assume $K \geq k$. How can you construct vectors $a_i, b_i \in \mathbb{R}^k$ from the given correspondences?