Analysis of Three-Dimensional Shapes
E. Rodolà, T. Windheuser, M. Vestner
Summer Term 2014

Computer Vision Group
Institut für Informatik
Technische Universität München

Exercise Sheet 5

Room: 02.09.023

Tue, 17.06.2014, 13:45-15:15

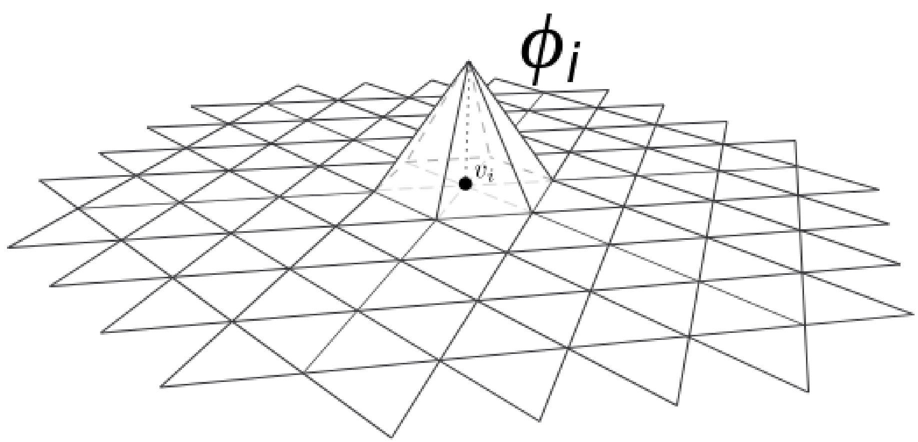
Submission deadline: Mon, 16.06.2014, 23:59 to windheus@in.tum.de

Mathematics: Laplacian

Exercise 1 (One Point). *In the lecture we have constructed a piecewise linear function $f : S \to \mathbb{R}$ by*

$$f(x) = \sum_i f_i \phi_i(x)$$

where the basis functions ϕ_i are linear in each triangle and $\phi_i(v_j) = \delta_{ij}$:



Calculate the inner products $-\langle \nabla\phi_i, \nabla\phi_j \rangle_S$ which are the elements of the stiffness matrix C.

Hint: Use the same technique we used in the lecture to calculate the elements of the *mass matrix* M.

Exercise 2 (One Point). *In this exercise we investigate the eigenvectors of the Laplace matrix $L = M^{-1}C$.*

1. *Show that ψ is an eigenvector of L with eigenvalue λ iff it is a solution to the generalized eigenvalue problem*

$$\lambda M\psi = C\psi$$

2. *Show that $\langle \cdot, \cdot \rangle_M := \langle \cdot, M \cdot \rangle$ defines an inner product.*

3. *Show that the Laplacian matrix L is symmetric with respect to $\langle \cdot, \cdot \rangle_M$.*

4. *What can you say about the eigenvalues of L?*

5. *Show that you can find eigenvectors $\{\psi_i\}$ of L such that $\Psi^T M \Psi = Id$. Here Ψ is the matrix with the eigenvectors as columns:*

$$\Psi = \begin{pmatrix} | & & | \\ \psi_1 & \dots & \psi_n \\ | & & | \end{pmatrix}$$

Programming: The Discrete Laplace Operator

Exercise 3 (Two points)**.** *Download and expand the file* `exercise5.zip` *from the lecture website. Modify the files* `cotanmatrix.m`*,* `massmatrix.m`*,* `heatsimulation.m`*, and* `hks.m` *to implement the functions as explained below. You can run the script* `exercise.m` *to test and visualize your solution.*

`cotanmatrix.m` *The function should compute the matrix $C \in \mathbb{R}^{n \times n}$ based on the cotangent scheme as defined in the lecture. The triangle mesh is given by matrices $V \in \mathbb{R}^{n \times 3}$ and $F \in \mathbb{N}^{m \times 3}$, where n is the number of vertices and m the number of triangles. C should be returned in sparse format.*

`massmatrix.m` *The function should compute the matrix $M \in \mathbb{R}^{n \times n}$ based on the scheme defined in the lecture. The triangle mesh is given by matrices $V \in \mathbb{R}^{n \times 3}$ and $F \in \mathbb{N}^{m \times 3}$, where n is the number of vertices and m the number of triangles. M should be returned in sparse format.*

`exercise.m` *Look at the code for the eigen decomposition. You see it is very easy to compute the generalized eigen decomposition $\lambda M \phi = C \phi$ by the matlab function* `eigs`*.*

`heatsimulation.m` *Given some initial heat distribution $u_0 \in \mathbb{R}^n$, the function simulates the diffusion of heat on the mesh. The function should display the distribution of heat at several given time points $t_1, \dots, t_T \in \mathbb{R}_+$.*

`hks.m` *The function should compute for each point on the mesh the heat kernel signature at time points $t_1, \dots, t_T \in \mathbb{R}_+$.*