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 6

Room: 02.09.023
Wed, 03.06.2015, 14:15-15:45
Submission deadline: Tue, 02.06.2015, 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.

Laplacian

Exercise 1 (One Point). *In this exercise we investigate the eigenvectors of the Laplace matrix $L = M^{-1}C \in \mathbb{R}^{n\times n}$ as introduced in the lecture and last exercise. (In the last exercise the stiffness (or cotangent) matrix C was denoted by S.)*

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

$$\lambda M\phi = C\phi$$

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 $\langle\cdot,\cdot\rangle_M$, i.e. $\langle Lx, y\rangle_M = \langle x, Ly\rangle_M$.*
4. *Show that L has real eigenvalues?*
5. *Show that you can find eigenvectors $\{\phi_i\}$ of L such that $\Phi^T M\Phi = Id$. Here Φ is the matrix with the eigenvectors as columns*

$$\Phi = \begin{pmatrix} | & & | \\ \phi_1 & \dots & \phi_n \\ | & & | \end{pmatrix}.$$

6. *Let $f \in \mathbb{R}^n$, define coefficients $\alpha_i \in \mathbb{R}$ by $\alpha_i = \langle f, \phi_i\rangle_M$. Show that $f = \sum_i \alpha_i\phi_i$, i.e. $\{\phi_i\}$ is an orthonormal basis of $\mathbb{R}^n$.*

The Heat Equation

Let $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ be a continuous sequence of vectors $u(t) \in \mathbb{R}^n$, where we call t the time parameter. Since the eigenvectors $\{\phi_i\}$ of L form an orthonormal basis of $\mathbb{R}^n$, $u(t)$ can be written as $u(t) = \sum_i \alpha_i(t)\phi_i$, where the coefficients $\alpha_i : \mathbb{R}_{\geq 0} \to \mathbb{R}$

are real-valued functions of time parameter t. We say $u(t)$ is a discrete distribution of heat if it satisfies the *heat equation*

$$\frac{\partial}{\partial t}u(t) = Lu(t).$$

From the linearity of differentiation we can write $\frac{\partial}{\partial t}u(t)$ as

$$\frac{\partial}{\partial t}u(t) = \sum_i \phi_i \frac{\partial}{\partial t}\alpha_i(t).$$

Exercise 2 (One Point)**.** *Let* $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^n, u(t) = \sum_i \alpha_i(t)\phi_i$, *now be any such sequence of vectors that satisfies the heat equation.*

1. *Show that there exists coefficients* $c_i \in \mathbb{R}, 1 \leq i \leq n$, *such that*

$$\alpha_i(t) = c_i \exp(\lambda_i t).$$

2. *Let* $u_0 \in \mathbb{R}^n$ *and set* $u(0) = u_0$. *Show that the coefficients* c_i *can be computed by* $c_i = \langle u_0, \phi_i \rangle_M$.

The exercise shows that if we let some initial heat distribution u_0 diffuse over time t the resulting heat distribution $u(t)$ can be computed by

$$u(t) = \sum_i \langle u_0, \phi_i \rangle_M \exp(\lambda_i t)\phi_i.$$

If our triangle mesh has n vertices we can define the *heat kernel signature* $\mathrm{HKS}(v_i, t)$ for v_i at time t by

$$\mathrm{HKS}(v_i, t) = \sum_i \langle e_i, \phi_i \rangle_M \exp(\lambda_i t)\phi_i,$$

where $e_i \in \mathbb{R}^n$ is the vector that is 0 everywhere except in the i-th component. Now we can discretize the time line by taking a finite subset $\mathcal{T} = \{t_1, \ldots, t_T\} \subset \mathbb{R}_{\geq 0}$ and define the vector valued heat kernel signature of vertex v_i by

$$\mathrm{HKS}_{\mathcal{T}}(v_i) = \left(\mathrm{HKS}(v_i, t_1), \ldots, \mathrm{HKS}(v_i, t_T)\right).$$

Programming: The Discrete Laplace Operator

Exercise 3 (Two points)**.** *Download and expand the file* `exercise6.zip` *from the lecture website. Modifiy 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 $t_1, \ldots, t_T \in \mathbb{R}_+$.*

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