Analysis of Three-Dimensional Shapes
F. R. Schmidt, M. Vestner, Z. Lähner
Summer Semester 2016

Computer Vision Group
Institut für Informatik
Technische Universität München

# Weekly Exercises 5

Room: 02.09.023

Wed, 01.06.2016, 14:00-16:00

Submission deadline: Tue, 31.05.2016, 23:59 to laehner@in.tum.de

## Mathematics

**Exercise 1** (2 points)**.** Find a map $\varphi : T_{\text{ref}} \to \mathbb{R}^3$ that is

1. angle-preserving but not area-preserving
2. area-preserving but not angle-preserving

$T_{\text{ref}} = \operatorname{conv}((0,0),(0,1),(1,0))$ is the reference triangle as used in the lecture. The image should be a triangle in 3D.

**Exercise 2** (3 points)**.** The stiffness matrix $C \in \mathbb{R}^{n\times n}$ was defined in the lecture as $C_{ij} = \int_S \langle \nabla\phi_i(x), \nabla\phi_j(x)\rangle \mathrm{d}x$. Show that the entries are equal to

$$C_{ij} = \begin{cases} \frac{\cot(\alpha_{ij})+\cot(\beta_{ij})}{2} & \text{if } (i,j) \text{ an edge} \\ -\sum_{k\neq i} C_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

The derivation is similar to the one of the mass matrix shown in Exercise Sheet 3.

## Programming

**Exercise 3** (4 points)**.** This exercise will contain the first steps for implementing the gradient on triangle meshes $(\mathcal{V}, \mathcal{T})$. The gradient $\nabla f$ of a function $f : S \to \mathbb{R}$ can be calculated by taking $\nabla f = g^{-1} \cdot \nabla\tilde{f}$ where $g$ is the first fundamental form and, for a fixed coordinate map $x$ of $S$, $\tilde{f} : U \to \mathbb{R}$ is such that $f = \tilde{f} \circ x^{-1}$. Since $g = (Dx)^\top Dx$, we start with calculating $Dx$ and then $g$.

1. Remember we have a coordinate map for each triangle individually, but instead of being given the map $x_k$ for each triangle $k$ we only have the vertex coordinates. Think about how each $x_k$ and $Dx_k$ looks like. (Tip: They were already used in Exercise 3.) Implement a function `trimesh_differential` that takes a triangle mesh and returns a $\mathbb{R}^{3\times 2\times k}$ multi-dimensional array representing all differentials.

2. The first fundamental form is constant on each triangle, we can represent it as a $\mathbb{R}^{2\times 2\times k}$ matrix. Write a function `trimesh_fff` that takes the result of `trimesh_differential` and returns the first fundamental form as a tensor. In theory multiplication of matrices with more dimensions works the same way as with two, but it is not implemented in Matlab so you will have to simulate it with a for loop. The function `squeeze` will help to return to matrices when taking slices of the tensor.

3. Calculate the area of each triangle with the first fundamental form and compare your results to the areas you calculated in Exercise 3 (which were probably done with Heron's formula).