Assignment 3:
Discrete Differential Properties and Smoothing

Handout date: 24.03.2017
Submission deadline: none
Demo session: none

In this exercise you will

- Experiment with different ways to compute surface normals for triangle meshes.
- Calculate curvatures from a triangle mesh.
- Perform mesh smoothing.
- Familiarize yourselves with the relevant implementations in `libigl`.

1. Normals

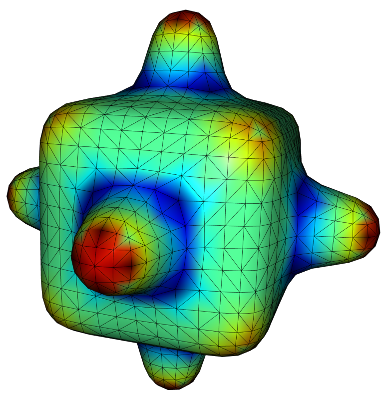
Starting from the `libigl` tutorial, experiment with different ways to compute vertex normals. A good starting point is tutorial # 201 of `libigl` and the documentation inside function `igl::per_vertex_normals`. Also have a look at tutorial # 205 for the calculation of the discrete Laplacian. Switch between these different normals:

- **Standard vertex normals** These are computed as the uniform average of surrounding face normals.
- **Area-Weighted normals** Same as above, but the average of the face normals is weighted by the face area.
- **Mean-curvature normals** These are the cotangent-weighted discrete Laplacian at every vertex.
- **PCA computation** At each vertex v_i, a plane is fit to the k nearest neighbours of this vertex using Principal Component Analysis. The vertex normal is then the principal component with the smallest eigenvalue (the normal to the plane). The neighbours can be collected by running breadth-first search.
- **Normals based on quadratic fitting** Using the local frame computed at a given vertex with PCA as above, the vertex and its k neighbours can be seen as a height function in that frame (the height axis being the principle component with the smallest eigenvalue). The derivative of that height function will then be normal to the surface. Thus the vertex normal can be found by (a) fitting a quadratic bi-variate polynomial to these height samples, (b) using the analytic expression of the polynomial derivative to compute the normal at the origin of the frame.

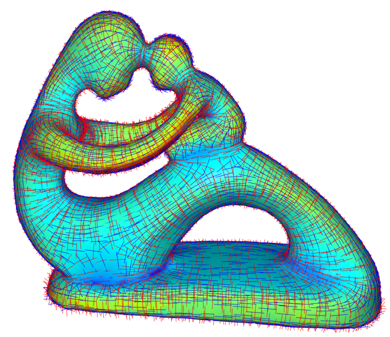
Relevant `libigl` *functions:* `igl::per_vertex_normals`, `igl::cotmatrix`, `igl::massmatrix`, `igl::fit_plane`, `igl::principal_curvature` (look inside for quadric fitting).

2. Curvature

Compute discrete mean, Gaussian, and principal curvatures (κ_{min} and κ_{max}) using the definitions in class. Color the mesh according to curvature by using a color map of your choice. Have a look at tutorials # 202, # 203 before you begin.



(A) Gaussian Curvature

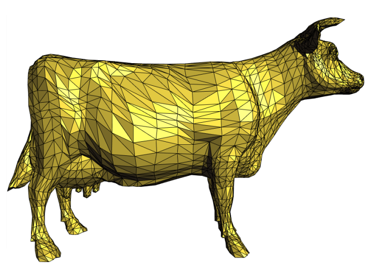


(B) Gaussian Curvature

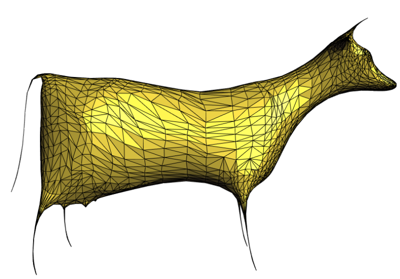
Figure 1. On the left: visualization of the Gaussian curvature. On the right, visualization of the mean curvature and principal curvature directions.

Relevant `libigl` *functions:* `igl::gaussian_curvature`, `igl::principal_curvature`, `igl::cotmatrix`, `igl::massmatrix`.

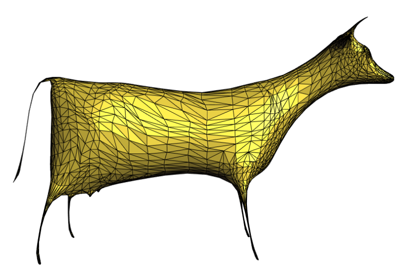
3. Smoothing with the Laplacian



(A) original



(B) explicit smoothing, $\lambda = 0.01$, 1000 iterations



(C) implicit smoothing, $\lambda = 20$, 1 iteration

Figure 2. Explicit and implicit smoothing on the *cow* mesh.

Perform explicit and implicit smoothing on the mesh. Experiment with uniform and cotangent weights. Have a look at tutorial # 205 before you begin, where implicit smoothing has been implemented.

Relevant `libigl` *functions:* `igl::cotmatrix`, `igl::massmatrix`, `igl::grad`, `igl::doublearea`.

4. Bilateral Smoothing

Implement bilateral mesh smoothing as described in the paper "Bilateral Mesh Denoising" by Fleishman et al.