Prof. Dr. Leif Kobbelt
Dr. Jan Möbius
Patrick Schmidt, Moritz Ibing

Basic Techniques in Computer Graphics

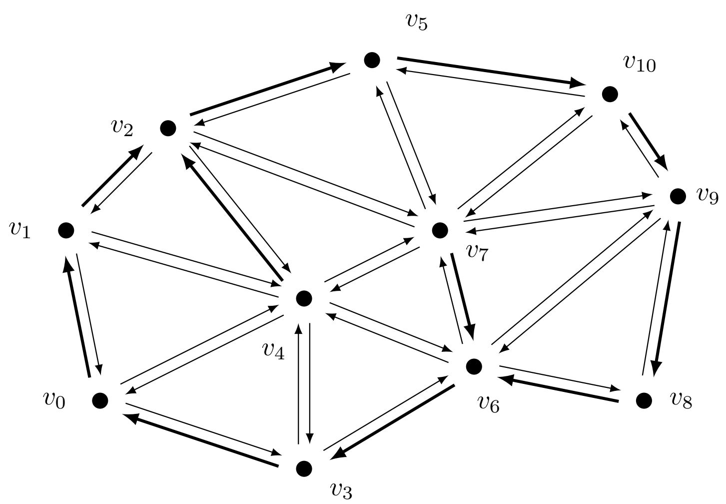
Assignment 10

Date Published: December 17th 2019, Date Due: January 14th 2019

- All assignments (programming and text) have to be completed in teams of 3–4 students. Teams with fewer than 3 or more than 4 students will receive no points.
- Hand in **one solution per team per assignment**.
- Every team must work independently. Teams with identical solutions will receive no points.
- Solutions are due 14:30 on January 14th 2019. Late submissions will receive zero points. No exceptions!
- Instructions for **programming assignments**:
  - Download the solution template (a zip archive) through the Moodle course room.
  - Complete the solution.
  - Prepare a new zip archive containing your solution. It must contain exactly those files that you changed. **Only change those files you are explicitly asked to change in the task description.** The directory layout must be the same as in the archive you downloaded.
  - Upload your zip archive through Moodle before the deadline. Use the Moodle group submission feature. Only in the first week (when Moodle groups have not been created yet), list all members of your group in the file `assignmentXX/MEMBERS.txt`. Remember, only one submission per group.
  - Your solution must compile and run correctly **on our lab computers** using the exact same `Makefile` provided to you. Do not include additional libraries and do not change code outside of the specified sections. If it does not compile on our machines, you will receive no points.
- Instructions for **text assignments**:
  - Prepare your solution as a single pdf file per group. Submissions on paper will not be accepted.
  - If you write your solution by hand, write neatly! Anything we cannot decipher will receive zero points. No exceptions!
  - Add the names and student ID numbers of all team members to every pdf.
  - Unless explicitly asked otherwise, always justify your answer.
  - Be concise!
  - Submit your solution via Moodle, together with your coding submission.

Exercise 1 Halfedge Data Structures **[14 Points]**

Consider the following triangle mesh represented by a halfedge data structure:



Navigation on a halfedge data structure is achieved by applying a sequence of the following elementary operations on the halfedges H and vertices V of the mesh:

- $n(h) : H \to H$: returns the **next** halfedge following h inside the same face (in counterclockwise direction).
- $o(h) : H \to H$: returns the **opposite** halfedge of h.
- $v(h) : H \to V$: returns the vertex that a halfedge h **points to**.
- $h(v) : V \to H$: returns an **outgoing** halfedge from a vertex v. In the above picture, the outgoing halfedge for each vertex is indicated by a bold arrow.

For example, in the given triangle mesh, $v(h(v_4)) = v_2$, and $v(n(h(v_4))) = v_1$.

(a) **[3 Points]**

Describe a sequence of operations that navigates from vertex v_0 to v_7.

(b) **[3 Points]**

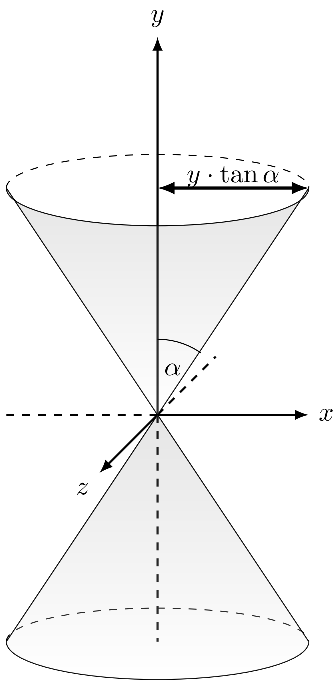
Using a sequence of the operations provided above, implement a new operation $p(h) : H \to H$ that returns the **previous** halfedge of h (i. e. the halfedge h' such that $n(h') = h$). You can assume that the given mesh is a triangle mesh without boundaries.

(c) **[8 Points]**

You are given an additional operation $p(v) : V \to \mathbb{R}^3$ returning the 3D position of a vertex v. Using this, give an algorithm in pseudo-code that computes a vertex normal for a given vertex v by averaging the normals of the incident faces. You can assume that the given mesh is a triangle mesh without boundaries.

Exercise 2 Implicit Functions **[8 Points]**

The two cones depicted below expand infinitely into the y and the $-y$ direction.



(a) **[4 Point]**

Derive a function such that $F(x, y, z) = 0$ is the surface of the cones depicted above. Their apex is located at the origin $(0, 0, 0)$.

(b) **[4 Point]**

Derive the matrix representation of the Quadric for the surface depicted above.

Exercise 3 Quadrics **[16 Points]**

In the lecture you have learned about quadrics as a way to specify certain objects in form of implicit functions. Let $Q \in \mathbb{R}^{4\times 4}$ be such a quadric defined as

$$Q = \begin{pmatrix} a & \frac{b}{2} & \frac{c}{2} & \frac{d}{2} \\ \frac{b}{2} & e & \frac{f}{2} & \frac{g}{2} \\ \frac{c}{2} & \frac{f}{2} & h & \frac{i}{2} \\ \frac{d}{2} & \frac{g}{2} & \frac{i}{2} & j \end{pmatrix}$$

then $F(x, y, z) = (x, y, z, 1)Q(x, y, z, 1)^T$ is the corresponding implicit quadratic function.

(a) **[8 Point]**

For rendering (e.g. by per-pixel ray intersection) one also needs to determine normal vectors (perpendicular to the defined surface) to perform lighting calculations. This can easily be done by computing the gradient of the function – a vector which points in the desired direction of the normal. Since our function is quadratic, its gradient (consisting of first-order derivatives) is linear, i.e. the normal can be expressed at some point $p = (x, y, z, 1)^T$ as

$$n = (n_x, n_y, n_z)^T = \frac{G \cdot p}{||G \cdot p||}$$

Specify the entries of matrix G that computes the gradient, respectively the normal, of F in this way.

(b) **[4 Points]**

An object C is specified by the following implicit function F:

$$F(x, y, z) = x(x-2) + y(y-4) + 3z(2\sqrt{3} - z) - 4 \ .$$

Derive the (unique) symmetric quadric such that it defines the same object C.

(c) **[4 Points]**

Using the matrix G, explicitly compute the normal at point $p = (1, 5, 2\sqrt{3}, 1)^T$ of object C.