Prof. Dr. Leif Kobbelt
Patrick Schmidt, Joe Jakobi

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Basic Techniques in Computer Graphics

Assignment 11

Date Published: January 17th 2023, Date Due: January 24th 2023

- All assignments (programming and theory) 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 on January 24th 2023 via Moodle. Late submissions will receive zero points. No exceptions!
- Instructions for **programming assignments**:
  - Make sure you are part of a Moodle group with 3-4 members. See "Group Management" in the Moodle course room.
  - Download the solution template (a zip archive) through the Moodle course room.
  - Unzip the archive and populate the `assignmentXX/MEMBERS.txt` file. The names and student ids listed in this file **must match** your moodle group **exactly**.
  - Complete the solution.
  - Prepare a new zip archive containing your solution. It must contain exactly the files that you changed. **Only change the files you are explicitly asked to change in the task description.** The directory layout must be the same as in the archive you downloaded. (At the very least it must contain the `assignmentXX/MEMBERS.txt`.)
  - One team member uploads the zip archive through Moodle before the deadline, using the group submission feature.
  - Your solution must compile and run correctly **on our lab computers** by only inserting your **assignment.cc** and **shader files** into the Project. If it does not compile on our machines, you will receive no points. If in doubt you can test compilation in the virtual machine provided on our website.
- 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.

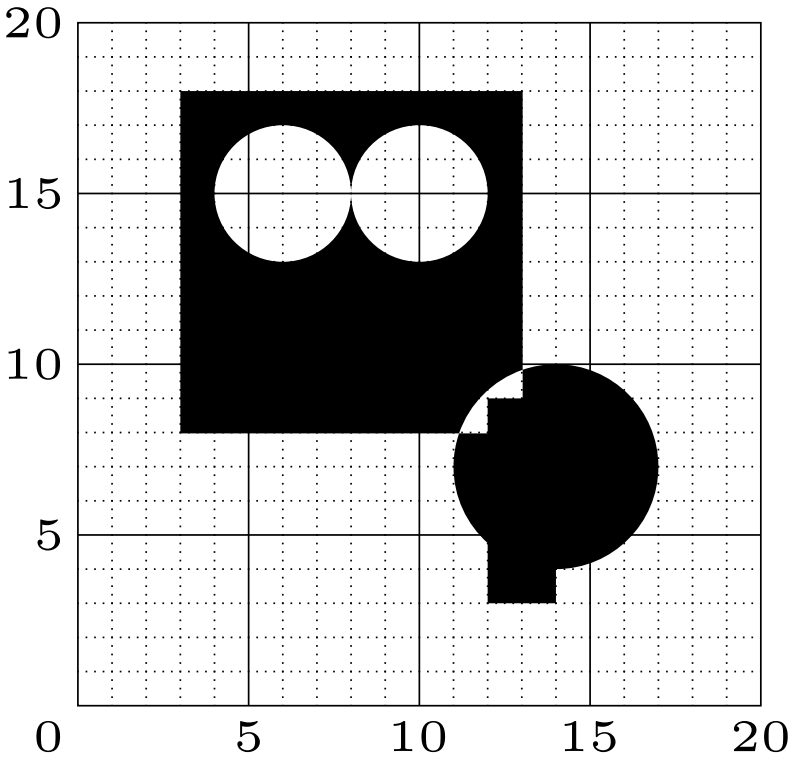
Prof. Dr. Leif Kobbelt
Patrick Schmidt, Joe Jakobi

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Exercise 1 Constructive Solid Geometry [10 Points]

Using Constructive Solid Geometry (CSG) complex solids can be obtained by performing Boolean operations (union, intersection, difference) on some simpler primitives.

The black object depicted below, for instance, can be constructed out of 5 or less primitives (i.e. rectangles and circles).



(a) [3 Points]

Give an abstract specification (e.g. "a rectangle of size 4×3 with its lower left corner at $(4, 2)^T$") of the 5 or less shapes required to produce the black object depicted above.

(b) [3 Points]

For each shape described in the previous subtask, define an implicit function $p_i : \mathbb{R}^2 \to \mathbb{R}, \quad 1 \leq i \leq 5$ such that its value is negative on the inside and positive on the outside of the respective shape.

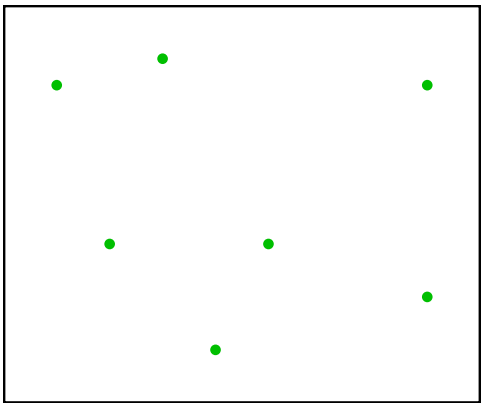
(c) [4 Point]

Construct an implicit function $p : \mathbb{R}^2 \to \mathbb{R}$ that combines the functions p_i above to describe the black shape depicted above. Again, this function should be negative for points inside the shape and positive for points outside the shape.

Note for all three subtasks: You may only use the operators $+$, $-$, $\cdot$, $/$, max(,), min(,), $|\ |$, $\sqrt{\ }$ in the implicit functions you define.

Exercise 2 Voronoi Diagram and Delaunay Triangulation [8 Points]

Consider the following set of 2D points:



(a) [4 Point]

Draw the Voronoi Diagram of the given point set!

(b) [4 Point]

Draw the Delaunay Triangulation of the given point set!
(You can draw your solution for tasks (a) and (b) in the same sketch, if you want.)

Exercise 3 Dual of Voronoi Diagrams [8 Points]

In most cases, dualizing the Voronoi Diagram of a set of 2D points immediately yields a Delaunay Triangulation of the given points. However, there are exceptional cases where the dual is not a triangle mesh: For example, four points in a square configuration have a Voronoi Diagram that dualizes to a quadrilateral instead of two triangles.

(a) [4 Point]

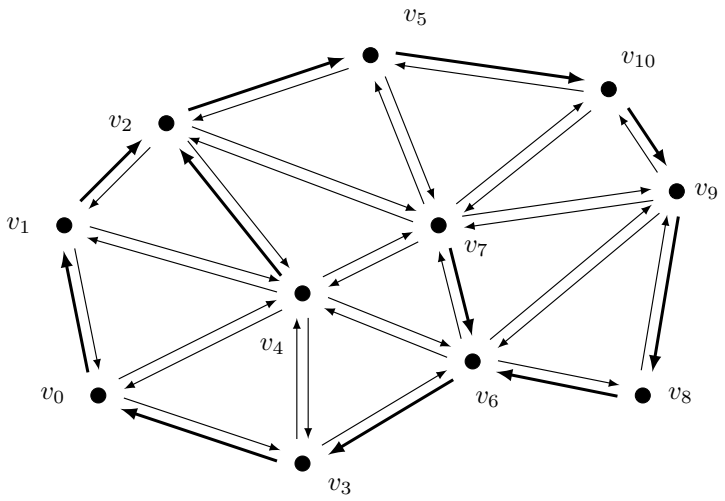
Similarly, it is possible to construct a point set such that the dual of its Voronoi Diagram is an n-sided polygon (for any $n \geq 3$).
Give instructions how to construct such a point set!
For $n = 5$, draw an example sketch of the input point set, its Voronoi Diagram and the resulting dual mesh.

(b) [4 Point]

You are given a point set constructed accordingly to task (a) and the resulting n-sided polygon (with $n > 3$). The resulting polygon is now triangulated by arbitrarily picking one vertex v and connecting it to all other vertices that aren't connected to v yet.
In general, is the resulting triangle mesh a Delaunay Triangulation? Explain why or why not!

Exercise 4 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.