Prof. Dr. Leif Kobbelt
J. Born, M. Ibing,
J. Möbius, P. Schmidt

Basic Techniques in Computer Graphics

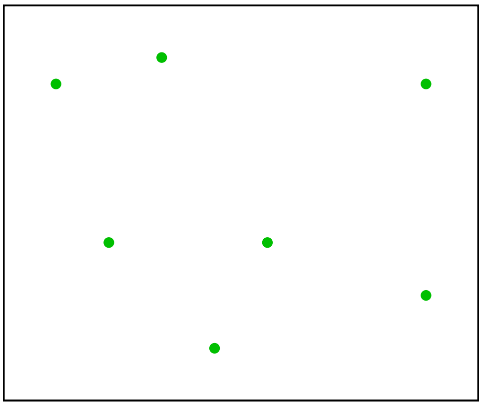
Assignment 9

Date Published: January 9th 2017, Date Due: January 16th 2017

- All assignments (programming and text) have to be done in teams of 3–4 students. Teams with less 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 18:00 on January 16th 2017. Late submissions will receive zero points. No exceptions!
- Instructions for **programming assignments**:
  - Download the solution template (a zip archive) through the L^2P course room.
  - Unzip the archive and populate the `assignmentXX/MEMBERS.txt` file. Any team member not listed in this file will not receive any points! (Also see the instructions in the file.)
  - 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. (At the very least it must contain the `assignmentXX/MEMBERS.txt`.)
  - Upload your zip archive through the L^2P before the deadline.
  - Your solution must compile and run correctly **on our lab computers** using the exact same `Makefile` provided to you. If it does not, you will receive no points.
- Instructions for **text assignments**:
  - Prepare your solutions on paper.
  - If you write your solution by hand, write neatly! Anything we cannot decipher will receive zero points. No exceptions!
  - If you hand in more than one sheet, staple your sheets together. (No paper clips!)
  - Put the names and matriculation numbers of all team members onto every sheet.
  - Unless explicitly asked otherwise, always justify your answer.
  - Be concise!
  - Put your solution into the designated drop box at our chair before the deadline. (1st floor, E3 building.)

Exercise 1 Voronoi Diagram and Delaunay Triangulation **[2 Points]**

Consider the following set of 2D points:



(a) **[1 Point]**

Draw the Voronoi Diagram of the given point set!

(b) **[1 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 2 Dual of Voronoi Diagrams **[2 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) **[1 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) **[1 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 3 Quadrics **[3 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) **[1 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) **[1 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.

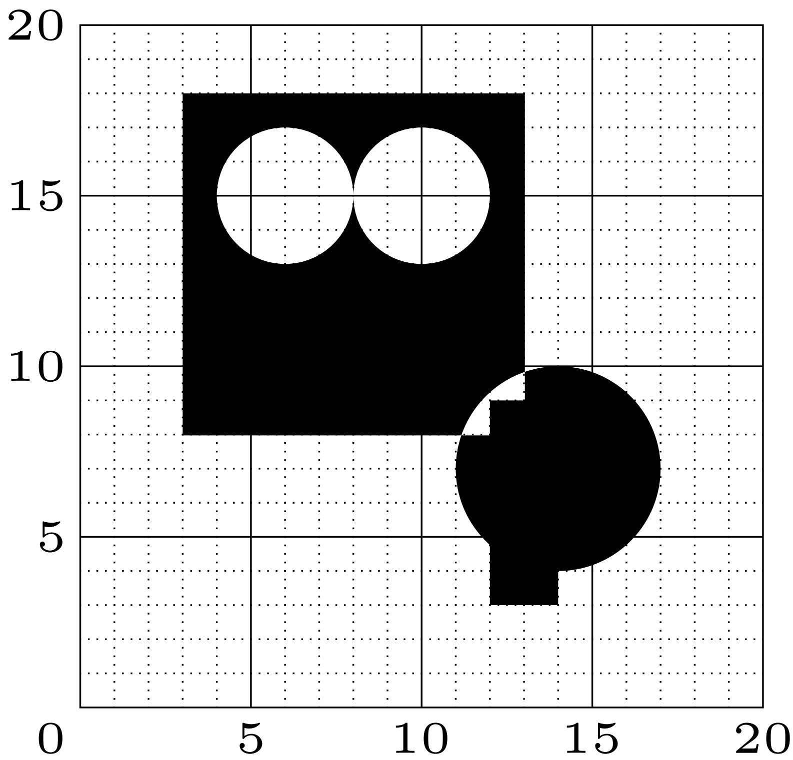
(c) **[1 Points]**

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

Exercise 4 Constructive Solid Geometry **[3 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) **[1 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) **[1 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) **[1 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.