Computational Geometry and Geometric Computing
Eric Berberich
Kurt Mehlhorn
Michael Sagraloff

Winter 2009/2010
Discussion on November 18th.

Exercise 4

Motivation

We practise linear kernels.

Basic Linear Kernel

In class we discussed the concept of a linear kernel and several models of it. The notes contain a `C++` implementation. Give an implementation in a programming language of your choice (preferably not `C++`).

Homogeneous Linear Kernel

Write the same kernel(s) with homogeneous coordinates. Also provide a floating point filter (formulation/implementation) for the orientation predicate for points homogeneous coordinates.

Install CGAL

Check `http://www.cgal.org`, download CGAL 3.5, and install it. It may be required that you also have to install CMAKE, BOOST, and gmp first. A detailed description of the installation process can be found at:

`http://www.cgal.org/Manual/3.5/doc_html/installation_manual/Chapter_installation_manual.html`

Next download:

`http://www.mpi-inf.mpg.de/departments/d1/teaching/ws09_10/CGGC/Software/kernel.zip`

Unpack the file. It contains a version of the kernels presented in class. List and explain the differences you can find - and suggest further "improvements". Use CMAKE (see CGAL documentation) to compile and link `kernel.cpp`. CGAL is required as the sources utilize one of its number types, namely `Gmpq`.

Extend Linear Kernel

Extend your linear kernel (or ours) by a type for a segment, that is, a straight line connection between two points and the following routines:

- x-comparison of points
- xy-comparison of points

- compute minimal/maximal end of a segment (lexicographic sorting)
- compute intersection point of two segments
- comparison of vertical order of two segments right of a common point
- vertical order of a point and a segment, that is, if a point in the x-range of a segment is below, on, or above a (non-vertical) segment

Smallest triangle of intersection points

Given a set $S = \{s_1, \dots, s_k\}$ of non-overlapping segments in the plane and let

$$\Gamma = \{\gamma \in \mathbb{R}^2 | \exists i, j \in \{1, \dots, k\}, i \neq j, s_i \cap s_j = \gamma\}$$

be the set of all intersection poins of distinct segments.
Describe and implement an algorithm that computes the triangle that minimizes the area among all triangles spanned by all segments' intersection points. Your implementation should rely on your extended linear kernel.

Have fun with the solution!