

Computational Geometry Algorithms Library

in

Geographic information Systems

Edward Verbree, Peter van Oosterom and Wilko Quak

TU Delft, Department of Geodesy, Thijssseweg 11, 2629 JA Delft, the Netherlands

e.verbree@geo.tudelft.nl

Introduction

Geographic Information Systems offers functionality to store and manipulate spatial data. In this, it brings together two different branches of computer science: Database Technology offers methods and structures to store data, while Computational Geometry offers the algorithms to efficiently analyze and manipulate Spatial Objects.

In this paper we will present one of the most important developments in applied Computational Geometry: CGAL (Computational Geometry Algorithms Library). The use of this library will be illustrated in two example applications. Finally, this paper concludes with a discussion regarding to the architecture of GIS and the role of the CGAL and spatial DBMS.

The CGAL project is a collaborative effort to develop a robust, easy to use, and efficient C++ software library of geometric data structures and algorithms. The CGAL library contains:

- Basic geometric primitives such as points, vectors and lines; predicates, e.g. relative positions of points; and operations such as intersection and distance calculation.
- A collection of standard data structures and geometric algorithms, such as convex hull in 2D/3D, (Delaunay) triangulation in 2D/3D, planar map, polyhedron, smallest enclosing circle, and multidimensional query structures.
- Interfaces to other packages, e.g. for visualization, I/O, and other support facilities.

CGAL example applications

In this section two CGAL example applications will be described. The first one uses a topology structure to represent a 2D planar map layer, and also applies the map-overlay operation to classify building types on the Cadastral map of the Netherlands. The second application is a 3D application based on the tetrahedron model and buffer operations.

Planar map overlay

There is a growing need for a map product that contains all buildings in the country (esp. houses), where every house has been assigned a specific type, such as 'apartment building' or 'free standing house'. One of the purposes of this application is the analysis of trends in house prices in the country, which depend on the house type. Currently, only the outer boundary of a building is stored, without any attribute data. This can be a large building (e.g. a row of houses), which must first be subdivided into individual units. The goal of this project is to find out whether it is possible to automatically classify the houses on the cadastral map, and whether the quality of the automatic classification is high enough.

The classification of the houses will take place in two phases: In the first phase, a map layer is created which contains both the boundaries of parcels and the boundaries of houses. In the second part, this map is used to automatically assign a class to all buildings.

In the first phase a map overlay of the building and the parcel layer will be made. For this step we will use the map-overlay class of CGAL 2.0. In order to use this C++ library we extract the layer data from the database and read it into the overlay software. Before the overlay is performed some pre-processing should be done on the data. For example, adjacent parcels with the same owner should be merged. The resulting overlay, together with the administrative data will be copied to a geo relational database.

In this resulting integrated database we will be able to pose the following queries:

- Is this building located completely within one parcel?
- How many mail addresses are connected to this building?

In the second phase of the process we are going to use the database created to classify all the houses. This will be done by writing rules for the different house types and applying these as SQL queries to the database (Oracle 8i). An example of such a rule might be: A building on two parcels with different owners, is a 'two-under-a-roof'. Other classification rules that can be applied are: A building with one owner with many addresses, probably is a building with rented apartments. Also the area of a building can be used to find a correct classification.

Tetrahedron model and buffer zones

The interest in applications and research on 3D-GIS is growing. The acronym 3D-GIS is often misleading in the sense that the datamodel is in fact 2D+1D. The third dimension (Z-value) is considered as an attribute value of the planimetric coordinates. This datamodel is appropriate for most applications (i.e. DEM-based), with the focus on visualization. But ‘real’ 3D-analyses needs a ‘real’ 3D-datamodel.

A 3D-object, which is given by its polygonal vector representation, can also be represented by cell decomposition consisting from the union of simple, non-overlapping, primitives. These primitives, or building-blocks, can be regular (cubes) or non-regular, i.e. tetrahedrons. We use the Tetrahedral Network (TEN), offered by the 3D-Triangulation class of CGAL. Now we can decompose complex operations on the geo-object (i.e. determine the buffer-zone) in the same manner. The buffer-zone of the 3D-object is the union of the buffer-zones of the decomposed tetrahedrons. We will show results of this method for convex 3D-objects. CGAL is used in conjunction with the World Toolkit (WTK) for visualization and Oracle 8i Spatial as geo-database.

GIS Architecture

Standard GIS architectures offer a fixed number of implemented geographic algorithms. In general the system offers an interface to a limited number of implementations of spatial algorithms, which can be combined to create the desired result. However, if the desired algorithm is not part of the GIS system, it is not possible to extend the system. A common solution to this problem is to extract the data from the GIS system, run the algorithm as a separate program, and copy the results back to the GIS.

With the advent of Object Oriented DBMSes (Jasmine) and extensible libraries for Computation Geometry (LEDA) we have the tools to build an algorithm extensible DBMS, which stores its data inside an Object Oriented database.

We consider the problem of devising external memory algorithms of which the memory allocations can change dynamically and unpredictably at run-time. The investigation of "memory-adaptive" algorithms, which are designed to adapt to dynamically changing memory allocations, can be considered a natural extension of the investigation of traditional, non-adaptive external memory algorithms. Our study is motivated by high performance database systems and operating systems in which applications are prioritized and internal memory is dynamically allocated in accordance with the priorities. In such situations, external memory applications are expected to perform as well as possible for the current memory allocation. The computation must be reorganized while running to adapt to the sequence of memory allocations.

References:

Andreas Fabri, Geert-Jan Giezeman, Lutz Kettner, Stefan Schirra, and Sven Schönherr. The CGAL kernel: A basis for geometric computation. In M. C. Lin and D. Manocha, editors, Proc. 1st ACM Workshop on Appl. Comput. Geom., volume 1148 of Lecture Notes Comput. Sci., pages 191-202. Springer-Verlag, 1996.

Internet links:

CGAL: <http://www.cs.uu.nl/CGAL/>

LEDA: <http://www.mpi-sb.mpg.de/LEDA/>

Jasmine: <http://www.cai.com/products/jasmine.htm>

WTK: <http://www.sense8.com/index.html>

Oracle: <http://www.oracle.com/database/options/spatial/>