Lecture Notes in Computer Science 1288
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Markus Schneider

Spatial Data Types for Database Systems

Finite Resolution Geometry for Geographic Information Systems

Springer

Author

Markus Schneider
FernUniversität Hagen, Fachbereich Informatik
Postfach 940, D-58084 Hagen, Germany
E-mail: markus.schneider@fernuni-hagen.de

Cataloging-in-Publication data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Schneider, Markus:
Spatial data types for database systems : finite resolution geometry for geographic information systems / Markus Schneider. - Berlin ; Heidelberg ; New York ; Barcelona ; Budapest ; Hong Kong ; London ; Milan ; Paris ; Santa Clara ; Singapore ; Tokyo : Springer, 1997
(Lecture notes in computer science ; Vol. 1288)
ISBN 3-540-63454-1

CR Subject Classification (1991): H.2, E.1-2, H.4, J.2

ISSN 0302-9743
ISBN 3-540-63454-1 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author
SPIN 10546430 06/3142 – 5 4 3 2 1 0 Printed on acid-free paper

Foreword

Database research in the last decade has focused on developing support for so-called *non-standard applications*. One important area is the representation of *spatial* information, needed, for example, in *Geographic Information Systems*. Database systems extended by capabilities for managing spatial information are called *spatial database systems*. This book contributes to the design and implementation of geometry in spatial database systems.

The fundamental question in a spatial database is how to represent geometry. Attempts to use a relational database with the standard set of data types (integer, real, string, etc.) had serious shortcomings. Representing a simple polygon in a table of x and y coordinates is conceptually inadequate, leads to complex queries and is, last but not least, hopelessly inefficient. To determine which polygon in the database contains a given point, it is first necessary to reconstruct the polygon from the set of tuples representing it.

The fundamental idea is to introduce abstract data types for spatial objects with the corresponding operations. For example, in two dimensions, there may be types to represent points, (poly-)lines, or polygons in the plane with operations such as testing whether two polygons overlap or whether a point lies within a polygon, computing the intersection of a line and a polygon, etc. The relational model, or in fact any DBMS data model, can be extended by such types in the role of *attribute data types*. Hence we may now have, for example, a relation describing countries with an attribute of type *polygon* representing the country area.

This leads directly to the question how geometric constructions, as defined by the Euclidean axioms, can be represented with the finite approximations available in a computer system. Several systems of spatial data types (or *spatial algebras*) had been proposed in the literature, but these formal designs were based on exact Euclidean geometry. For example, one assumed that the intersection point of two line segments can be computed precisely. However, computers work with finite representations and can represent coordinates only approximately. It is usually necessary to round the coordinates of the intersection point to the nearest *grid point* where the grid corresponds to the resolution of the number system used. This introduces numerous types of errors, e.g., a subsequent test will tell you that the intersection point does not lie on either of the two line segments that created it in the first place! These errors were not only known theoretically, but hounded the daily practice of Geographic Information Systems: the most important "overlay" function of most commercial sytems failed sometimes even for simple inputs. It became crucial to find a solution for a robust implementation of geometric algorithms.

The essential new idea developed in this thesis work was not to base the design of a spatial algebra directly on Euclidean geometry but to use an underlying discrete basis, a so-called *realm*. A realm is a set of points and line segments defined over a discrete grid, with the additional property that no two realm segments intersect (except at their end points), and that no realm point lies on the interior of a realm segment. In other words, a realm is a planar graph embedded into the plane. It can be represented with finite coordinate values. A realm is intended to represent the entire geometry of an application, regardless of how the application is structured into different classes of geometric objects. The values of spatial data types are composed of points and line segments existing in the underlying realm. New geometries entered into the database are first put into the realm – so any intersections are computed at this level – and then propagated to the spatial data type values defined over the realm in the database. As a result, one can define a spatial algebra whose operations obey the algebraic laws that one would expect not only in theory, but also in the implementation. For example, computing the overlay of parcels with soil types gives the same result as the overlay of soil types over parcels. This fundamental law is generally violated in today's Geographic Information System software.

This book develops this strategy in detail. It describes the concept of realm and shows how updates can be performed in the realm layer so that the realm properties are maintained. This results in the design of the *ROSE algebra*, a comprehensive and coherent system of spatial types that can be implemented in (extensible) database systems without incurring the risk of computational problems due to the finiteness of computer algebra.

The algorithms and data structures designed are very efficient, in many cases even more efficient than the corresponding standard, floating point based algorithm. This is the rare case where the correct solution is also more efficient. The algorithms have been implemented and tested. They are available as a software package and we hope that they can be integrated by the industry into Geographic Information Systems. This book describes all the pieces of the design and implementation in a coherent manner, and should be a valuable resource for industry to help transfer these concepts into practice.

It has been a pleasure for me (RHG) to work with Markus, from the early days when he was a student member of a project group at the University of Dortmund that implemented a first prototype of a spatial database system called *Gral*, until now. His research results have been well received in the spatial database community and the concept of realms is well known now. The ROSE algebra, which is the central part of his Ph.D. thesis and described here, is used in some projects at other universities. It is a new approach to this challenging problem and has been an inspiration for others to focus on similar questions.

May 1997 *Ralf Hartmut Güting*

Andrew U. Frank

Preface

In various application fields there is a need to manage and process *geometric* or *spatial* data, that is, data related to *space*. For this purpose *spatial database systems* are designed which are full-fledged database systems with additional capabilities for storing, retrieving, manipulating, and querying spatial data. Such systems provide the underlying database technology needed to support applications such as *geographical information systems*. *Spatial data types* like *point*, *line*, and *region* provide a fundamental abstraction for modelling the structure of geometric entities, their relationships, properties, and operations. Their definition and implementation is probably the most fundamental issue in the development of spatial database systems and has influence on multiple disciplines like computer science, computer aided design, multimedia systems, VLSI design, (applied) computational geometry, geography, cartography, surveying engineering, linguistics, and psychology.

In this book we first give a comprehensive survey of the modelling approaches for spatial data types and operations so far discussed in the literature. We then introduce a system of spatial data types, or a *spatial algebra*, called *ROSE* (*RObust Spatial Extension*) *algebra*. The ROSE algebra avoids many of the deficiencies of current approaches and provides a satisfactory solution for a formal definition and robust implementation of a spatial type system in a single model. Current approaches to spatial data types primarily suffer from shortcomings regarding generality, formal definitions, closure properties, finite representations, geometric consistency, efficiency, extensibility, and data model independence. The development of the ROSE algebra consists of three steps.

The first step introduces the concept of a *realm* as a *discrete* geometric basis underlying one or more spatial data types. A realm is a finite set of points and non-intersecting line segments over a discrete grid and conceptually describes the complete underlying geometry of a particular application space in two dimensions. The idea is to construct the geometries of spatial objects by composing them from points or segments in the realm. As a basis for the design of spatial data types, very often Euclidean space is used or implicitly assumed. A point in the plane is then represented by a pair of real numbers. Unfortunately, in practice there are no real numbers available in computers but only finite approximations. This leads to many problems in geometric computation. The realm concept solves these problems of numerical robustness and topological correctness below and within the realm layer. In particular, it solves the line segment intersection problem over a discrete grid and enforces geometric consistency of related spatial objects. It also enables one to formally define general spatial data types or algebras on top of realms. These types enjoy nice closure properties not only in theory but also in computational practice.

The second step deals with the formal definition of the *ROSE algebra* itself which is defined on top of realms and offers *realm-based* spatial data types to represent point, line, and region features in the plane together with a comprehensive set of spatial operations. The ROSE algebra has a number of interesting features: it (i) offers (values of) data types of a very general structure, (ii) has a complete formal definition of the semantics of types and operations, (iii) has realms as a discrete geometric basis which allows for a numerically correct and robust implementation of types and operations in terms of integer arithmetic, (iv) treats consistency between distinct spatial objects with common parts, and (v) has a general object model interface which allows it to cooperate with different kinds of database systems.

The third step relates to the *ROSE system* as an implementation of the ROSE algebra. This system realizes the algebra's types and operations by providing efficient data structures and new *realm-based geometric algorithms* defined over a discrete grid. The main techniques used are parallel traversal of objects, plane-sweep, and graph algorithms. All algorithms are analysed with respect to their worst-case time and space requirements. Due to the realm properties, these algorithms are relatively simple and efficient, and numerically completely robust. In contrast to traditional work on algorithms, the focus is not on finding the most efficient algorithm for each single problem, but rather on considering a spatial algebra as a whole, and on reconciling the various requirements posed by different algorithms within a single data structure for each type. As an important result, a comparison with the algorithms of classical computational geometry reveals that realm-based geometric algorithms are much simpler and more efficient than their Euclidean counterparts.

This book is a revised and slightly extended version of my doctoral thesis (Ph.D. thesis) [Sc95a] submitted to the Computer Science Department of the Fernuniversität Hagen in Hagen, Germany, in December 1995. All parts of this book are worked out in considerable detail. This may be seen as a negative consequence of this text being essentially my thesis. However, it may also be interesting in its own right, in particular for readers who like to go into further detail or intend to construct and implement a spatial database system, a geographical information system, or a CAD system. Since detailed expositions are always preceded by conceptual overviews, it is left to the reader to omit detailed explanations and to focus on the book parts he or she is interested in.

Acknowledgments

I am indebted to several people who assisted me in one way or another to realize the original thesis which led to this book. First of all, I would like to express my esteem and my sincere thanks to my supervisor Prof. Dr. Ralf Hartmut Güting of the Fernuniversität Hagen, Germany, for his guidance and support throughout the development of the thesis. Furthermore, my gratitude extends to Prof. Dr. Andrew U. Frank of the Technical University of Vienna, Austria, for his interest in the subject of my thesis and his willingness to be its co-referee.

Prof. Güting suggested dealing with the subject of spatial data types for database systems and gave me the opportunity and the time to work on this topic and to pursue my ideas. He accompanied and supported my work by stimulating discussions and constructive criticism and helped me to structure my concepts and to focus on essential ideas. Prof. Frank made many valuable comments on a first draft and helped me to improve the contents and the presentation of my work.

Furthermore, I would like to thank Thomas de Ridder and Andreas Rohrbach for their excellent implementation work. Thanks also to the German Research Community (Deutsche Forschungsgemeinschaft (DFG)) for sponsoring a four-year research project entitled “Geo-Database Systems”. Without its financial support, neither my publications in this period nor my thesis nor this book would have been possible. Special thanks go to Alfred Hofmann of Springer-Verlag for his editorial help.

Last but not least, I would like to thank my wife Annette, my son Florian, and my parents Hans and Christel Schneider for their love, support, encouragement, and patience. This book is dedicated to them.

June 1997 *Markus Schneider*

Table of Contents