GENESYS: A System for Efficient Spatial Query Processing

Thomas Brinkhoff, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger

Institute for Computer Science, University of Munich
Leopoldstr. 11 B, D-80802 München, Germany
e-mail: {brink,kriegel,ralf,bseeger}@informatik.uni-muenchen.de

1 Introduction

Recently, *spatial database systems (SDBS)* have gained increasing importance in various applications. Due to the high complexity of both spatial objects and spatial queries and also due to extremely large data volumes, SDBSs impose stringent requirements on efficient query processing. The goal for developing *GENESYS* was to design a system that efficiently supports a well-chosen set of *spatial queries*. These spatial queries serve as a basis for implementing other more sophisticated operations required in special applications. An efficient implementation of the basic spatial queries is most important for good overall performance of the SDBS.

2 Multi-Step Query Processing

In GENESYS, spatial queries are processed following the paradigm of *multi-step query processing* which is depicted in figure 1 [KBS 93], [BHKS 93]. The main goal is to reduce the time spent in expensive steps by using one or several filter steps.

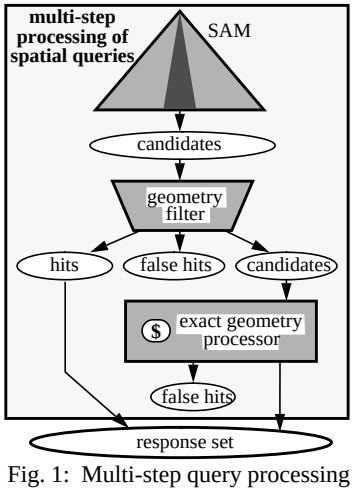

Fig. 1: Multi-step query processing

A spatial query is abstractly executed as a sequence of steps. In the first step, a *spatial access method (SAM)* is used for scaling down the search space. Due to the complexity of real spatial objects, the *minimum bounding rectilinear rectangle (MBR)* of the spatial objects is used as a geometric key. For a query, the SAM can only deliver a candidate set which includes all answers of the query. Obviously, some of the elements in the candidate set may not fulfill the query.

In a second step, all elements of the candidate set are examined using a geometry filter. This filter exploits more accurate approximations than the MBR. The approximations are used for determining elements of the candidate set as *hits* fulfilling the query, *false hits* not fulfilling the query and *candidates* possibly fulfilling the query.

The remaining candidates are eventually investigated in the last step using the exact description of the objects. Without an adequate object representation, this step is the most expensive one. Efficient algorithms from the area of computational geometry are essential to deliver a good response time.

3 GENESYS

GENESYS implements this type of multi-step query processing. Its architecture is illustrated in figure 2. The top-level of the system consists of a *graphical user interface* which provides an easy and natural way to specify spatial queries and to present their answers. The second level of the system corresponds to a module that contains a set of *spatial operations*. In particular, a well-chosen basic set of spatial queries is implemented such as *point and window queries*, *nearest neighbor queries* and *spatial joins* [BKS 93a], [BKSS 94]. The third level comprises modules which are the building blocks for the operations on the second level. The first module contains an implementation of the *R*-tree*, a SAM that efficiently organizes a dynamic set of multidimensional rectangles on secondary storage [BKSS 90]. The second module contains a *tool box for computational geometry* algorithms. This is an important component since such operations frequently occur in many spatial queries. The third module handles the *representation of the objects*. In our implementation, polygonal areas are decomposed into simple components such as *trapezoids*, *triangles*, and *convex polygons* [KHS 91]. The components of one object are organized by a spatial data structure *(TR*-tree)* [SK 91]. Eventually, the fourth module provides the computation of various *approximations* for spatial objects [BKS 93b]. Examples for *conservative approximations*, which completely include a spatial object, are the rotated rectangle, the convex hull, the n-corner, the circle, and the ellipse. A *progressive approximation* is completely inside of the object. Examples are the enclosed circle and the enclosed rectangle.

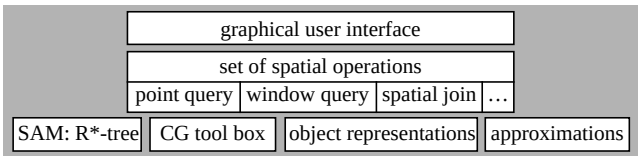

Fig. 2: Architecture of GENESYS

Acknowledgements

We would like to thank K. Ablaßmeier, M. Bauer, N. Beckmann, A. Braun, U. Droste, H.-G. Harms, P. Heep, S. Heep, H. Horn, M. Schiwietz, and E. Welzl for their contributions to the design and implementation of GENESYS.

References

[BHKS 93] Brinkhoff T., Horn H., Kriegel H.-P., Schneider R.: *'A Storage and Access Architecture for Efficient Query Processing in Spatial Database Systems'*, SSD'93, LNCS 692, pp. 357-376.

[BKS 93a] Brinkhoff T., Kriegel H.-P., Seeger B.: *'Efficient Processing of Spatial Joins Using R-trees'*, SIGMOD'93, pp. 237-246.

[BKS 93b] Brinkhoff T., Kriegel H.-P., Schneider R.: *'Comparison of Approximations of Complex Objects used for Approximation-based Query Processing in Spatial Database Systems'*, ICDE'93, pp. 40-49.

[BKSS 90] Beckmann N., Kriegel H.-P., Schneider R., Seeger B.: *'The R*-tree: An Efficient and Robust Access Method for Points and Rectangles'*, SIGMOD'90, pp. 322-331.

[BKSS 94] Brinkhoff T., Kriegel H.-P., Schneider R., Seeger B.: *'Multi-Step Processing of Spatial Joins'*, SIGMOD'94.

[KBS 93] Kriegel H.-P., Brinkhoff T., Schneider R.: *'Efficient Spatial Query Processing in Geographic Database Systems'*, IEEE Data Engineering Bulletin, Vol. 16, No. 3, 1993, pp. 10-15.

[KHS 91] Kriegel H.-P., Horn H., Schiwietz M.: *'The Performance of Object Decomposition Techniques for Spatial Query Processing'*, SSD'91, LNCS 525, pp. 257-276.

[SK 91] Schneider R., Kriegel H.-P.: *'The TR*-tree: A New Representation of Polygonal Objects Supporting Spatial Queries and Operations'*, 7th Workshop on Computational Geometry '91, LNCS 553, pp. 249-264.