Spatial Data Management for Computer-Aided Design

Hans-Peter Kriegel[1] Andreas Müller[2] Marco Pötke[1] Thomas Seidl[1]

[1]University of Munich, Institute for Computer Science
{kriegel, poetke, seidl}@informatik.uni-muenchen.de

[2]Volkswagen AG
andreas5.mueller@volkswagen.de

Abstract

This demonstration presents a spatial database integration for novel CAD applications into off-the-shelf database systems. Spatial queries on even large product databases for digital mockup or haptic rendering are performed at interactive response times.

1 SPATIAL QUERIES

In mechanical engineering, three-dimensional Computer-Aided Design (CAD) is employed throughout the entire development process. From the early design phase to the serial production of vehicles or airplanes, thousands to millions of CAD files and associated documents are generated. Recently, a new class of CAD tools has emerged to support virtual engineering on this data, i.e. the evaluation of product characteristics without building a physical prototype. Typical applications include the digital mockup (DMU) [BKP 98] or haptic rendering of product configurations [MPT 99].
Virtual engineering requires access to the product data by geometric predicates, such as “find all parts in the immediate spatial neighborhood of the disk brake” (cf. Figure 1). We present the DIVE architecture (Database Integration for Virtual Engineering), a proposal to embed interactive spatial queries into off-the-shelf object-relational database systems. The DIVE prototype contains filters for the following spatial queries:

- *Volume query*: Determine all spatial objects intersecting a given rectilinear box volume.
- *Collision query*: Find all spatial objects that intersect an arbitrary query region, e.g. a volume or a surface of a query part.
- *Clearance query*: Given an arbitrary query region, find all spatial objects within a specified Euclidean distance.

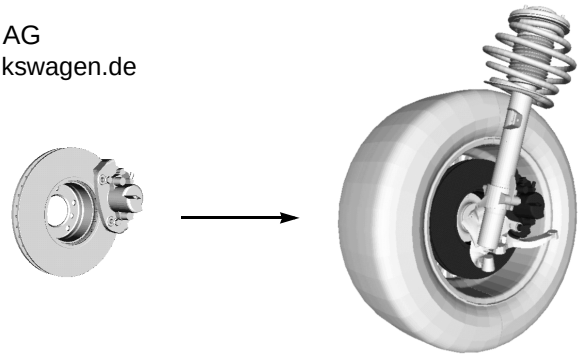

Figure 1: Querying the spatial neighborhood of a disk brake.

The DIVE server also supports the ranking of query results according to the intersection of the query region with the found spatial keys. Thus, the attention of the user is immediately guided to the most relevant problems in the current product design.

2 EXTENSIBLE INDEXING

Only few spatial access methods have been designed to operate on the object-relational data model without any intrusive modifications of the database server. The Relational Interval Tree (*RI-tree*) [KPS 00] is a light-weight access method that efficiently manages extended data on top of any object-relational database system while preserving the built-in transaction semantics and recovery services. In [KPS 01] we have shown that its spatial application using space-filling curves significantly outmatches Linear Quadtrees and Relational R-trees with respect to usability and performance. We therefore use the RI-tree as spatial engine for the DIVE system.

References

[BKP 98] Berchtold S., Kriegel H.-P., Pötke M.: *Database Support for Concurrent Digital Mock-Up*. Globalization of Manufacturing in the Digital Communications Era of the 21st Century, Kluwer Academic Publishers, 499-509, 1998.

[KPS 00] Kriegel H.-P., Pötke M., Seidl T.: *Managing Intervals Efficiently in Object-Relational Databases*. Proc. 26th Int. Conf. on Very Large Databases (VLDB), 2000.

[KPS 01] Kriegel H.-P., Pötke M., Seidl T.: *Interval Sequences: An Object-Relational Approach to Manage Spatial Data.* Submitted for publication.

[MPT 99] McNeely W. A., Puterbaugh K. D., Troy J. J.: *Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling*. Proc. ACM SIGGRAPH Int. Conf. on Computer Graphics and Interactive Techniques, 401-408, 1999.