Astronomical Data Analysis Software and Systems XVI
ASP Conference Series, Vol. 376, 2007
R. A. Shaw, F. Hill and D. J. Bell, eds.

Spatial Indexing and Visualization of Large Multi-Dimensional Databases

László Dobos[1], István Csabai[1], Márton Trencséni[1], Géza Herczegh[2], Péter Józsa[2], Norber Purger[1]

Abstract. Scientific endeavors such as large astronomical surveys generate databases on the terabyte scale. These usually multi-dimensional databases must be visualized and mined in order to find interesting objects or to extract meaningful and qualitatively new relationships. Many statistical algorithms required for these tasks run reasonably fast when operating on small sets of in-memory data, but take noticeable performance hits when operating on large databases that do not fit into memory. We utilize new software technologies to develop and evaluate fast multi-dimensional, spatial indexing schemes that inherently follow the underlying highly non-uniform distribution of the data: one of them is hierarchical binary space partitioning; the other is sampled flat Voronoi partitioning of the data. Our working database is the 5-dimensional magnitude space of the Sloan Digital Sky Survey with more than 250 million data points. We show that these techniques can dramatically speed up data mining operations such as finding similar objects by example, classifying objects or comparing extensive simulation sets with observations. We are also developing tools to interact with the spatial database and visualize the data real-time at multiple resolutions at different zoom levels in an adaptive manner.

1. Relational Database Management Systems

Commercially available database management systems are optimized for financial data, but also can be used for scientific data analysis. However scientific data are usually multi-dimensional with a continuous numerical domain so tricks are needed to handle them efficiently. Microsoft SQL Server 2005 provides for running user programs within the server process thus allowing scientists to perform complex computationally intensive analysis of the underlying data while leaving the quite complicated task of disk and memory management to the database server.

2. Spatial Indexing

Spatial indexing techniques partition the parameter space using different geometric schema. Each cell of the partitioned space gets a unique identifier number. Data points are tagged with the ID of their encapsulating cell. By generating a clustered index over this cell ID the server partitions the database physically on

[1]Dept. of Physics of Complex Systems, Eötvös Loránd University, Budapest, Hungary

[2]Technical University of Budapest, Hungary

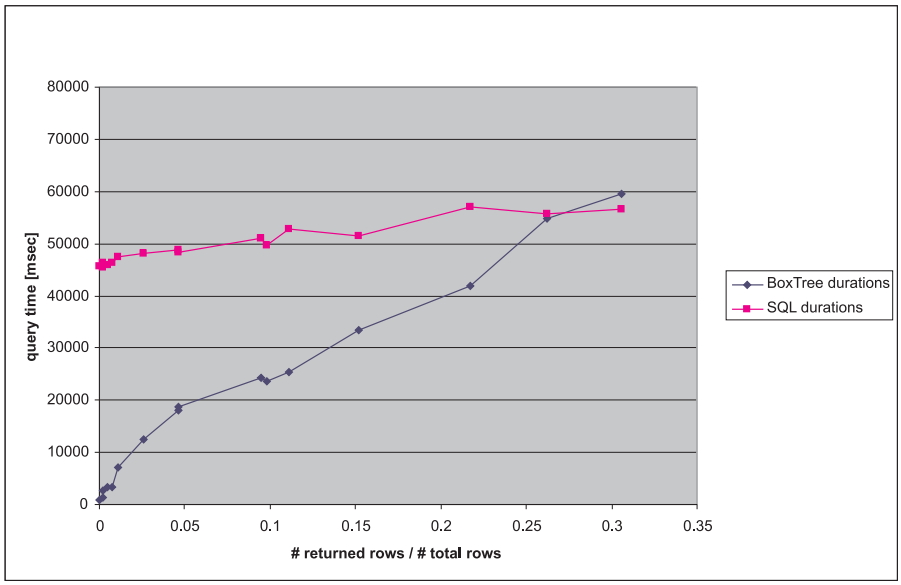

Figure 1. Kd-tree vs. standard SQL spatial query times.

the disk according to the chosen spatial indexing scheme; data points that are close to each other in the parameter space will be close to each other on the disk as well. This trick may dramatically speed up computation when random access to the data is required instead of systematic scanning of data tables. Geometric parameters of the cells of a partitioning, such as vertices, faces, volume etc. are stored within the database, but other precomputed quantities can also be stored, such as number of contained points, point density etc. The spatial partitioning of databases can not only help organizing data over the server's disk storage, but may also help to distribute them on a grid network in future applications.

3. Spatial Queries

Spatial queries are described by general multi-dimensional polyhedra using a set of linear inequalities. By intersecting the cells of the tessellation with the hyperplanes defined by the inequalities, a smaller set of cell IDs is determined. Intersecting multi-dimensional polyhedra is a computationally intensive task, so sophisticated algorithms are required. Based on the cell IDs, data points are looked up by a clustered index instead of running a table scan. Application of this technique leads to a significant gain of performance, see Figure 1. Nearest-neighbor and similar object searches can be conducted in a similar way, by looking up the cell ID of the sample point and querying data points within the surrounding cells.

4. Spatial Indexing Schema

We implemented the following schema over the SDSS 5-dimensional magnitude space (Stoughton et al. 2002):

1. The kd-tree index is a binary space partitioning technique which first separates the 5-dimensional magnitude space into two boxes parallel to one of the axes, then partitions each of the boxes into two smaller boxes parallel to the next axis, and so on, making sure at each step that each box contains an equal number of data points to keep the tree balanced.
2. The Voronoi index (Aurenhammer 1991) is constructed by taking a random 10K sample of the 250M magnitude points and building its Voronoi tessellation (Figure 2). Though the Voronoi tessellation is not a hierarchical structure, it can be refined hierarchically by running the tessellation again on the data points contained in each individual cell.

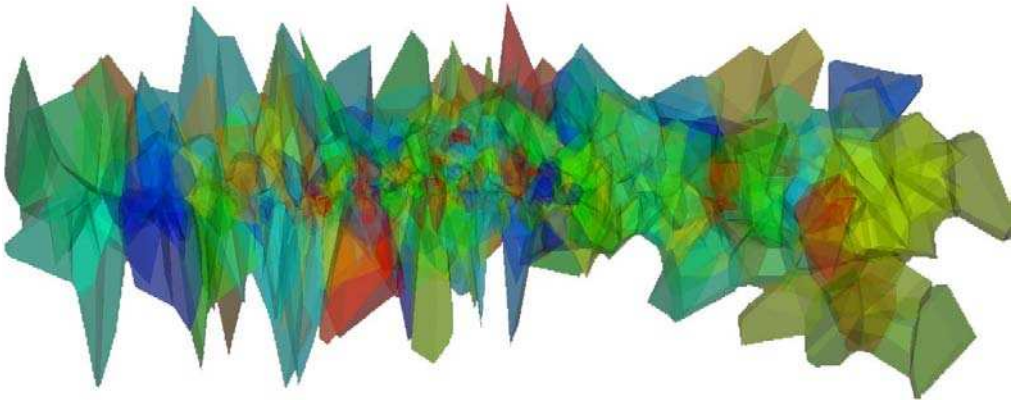

Figure 2. Voronoi tessellation of a sample dataset in 3 dimensions.

3. A Voronoi diagram (Figure 3) is constructed based on a set of n seed points. It partitions the data space into a set of n convex polytopes (Voronoi cells) around the seed points so that the inside of the cell is closer to the seed point of the cell than to any of the other seed points.

5. Visualization

We are developing an on-line visualization tool especially for multi-dimensional, spatially indexed databases. The visualization tool communicates with the database server interactively, so users may zoom in and get a detailed view of the dataset in real-time. To achieve this, the program either displays a coarse density map of the underlying data or retrieves only a random subset of data points. Users can select data points by adding cutting hyperplanes (n-dimensional inequalities) and the client then automatically generates the corresponding SQL query that can be sent to the server with a single click.

We use the managed DirectX technology to display 3D content. Platform independence was not an issue this time, since the database server extensions can

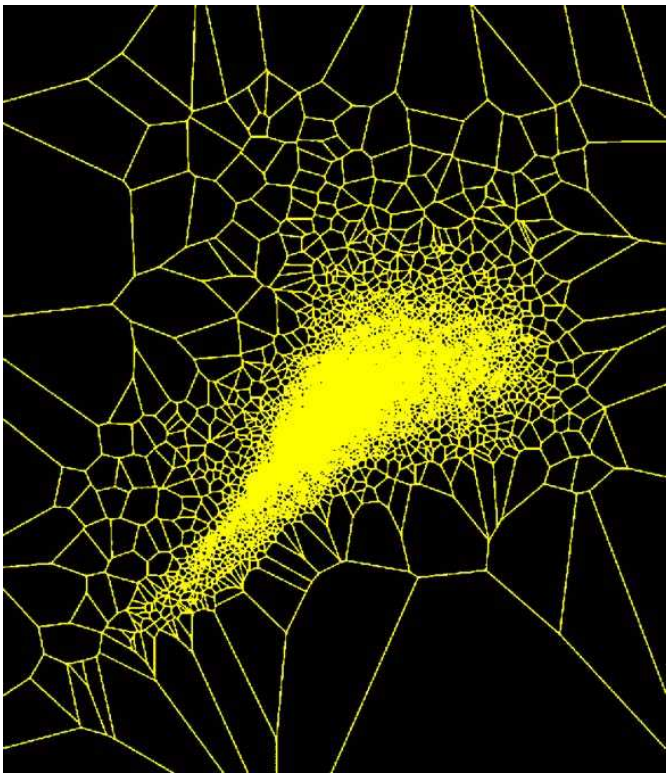

Figure 3. Voronoi diagram of SDSS data points.

only run on the Windows platform. However, Virtual Observatory compliance is a priority, so we are adding web service support to the program.

Acknowledgments. This work was partly supported by: OTKA-T047244, MSRC-2005-038, NAP-2005/ KCKHA005 and MRTN-CT-2004-503929.

References

Aurenhammer, F. 1991, ACM Computing Surveys, 23, 345
Stoughton, C., et al. 2002, AJ, 123, 485