Exploring Sensor and Simulation Raw Data with RasDaMan

Peter Baumann, Andreas Dehmel, Maximilian Höfner, Norbert Widmann

FORWISS (Bavarian Research Center for Knowledge-Based Systems)
Orleansstr.34, D-81667 Munich, Germany
Dial-up: voice +49-89-48095-207, fax - 203
E-mail: {baumann,dehmel,widmann}@forwiss.de, hoefner@in.tum.de

1 Motivation

Satellites and other sensors, supercomputer simulations, and experiments in science and engineering all generate arrays of some dimensionality, spatial extent, and cell semantics. While such data differ in aspects such as data density (from dense images to sparse physics data) and distribution, they share the property of extreme data volumes. A major bottleneck today is fast, user-centric evaluation of such *Multidimensional Discrete Data* (MDD).

The goal of the RasDaMan DBMS is to provide database services on general MDD structures in a domain-independent way. To this end, RasDaMan offers an algebra-based query language which extends SQL with declarative MDD operators. Server-based query evaluation relies on algebraic optimisation and a specialised storage manager.

Usually, research on array data DBMSs focuses on particular system components, such as multidimensional data storage [8] or data models [6, 7]. RasDaMan, on the other hand, is a fully operational[1] generic array DBMS. This makes it a unique opportunity to study all aspects of MDD management in a holistic way, thereby augmenting research done elsewhere.

2 System Overview

2.1 Conceptual Model

The conceptual model of RasDaMan centers around the notion of an n-D array (in the programming language sense) which can be of any dimension, size, and array cell type (for the C++ binding, this means that valid C++ types and structs are admissible). Each dimension's lower and upper bound can be fixed at data definition time, or can be left variable. Type definition is done through the RasDaMan definition language, RasDL, which is based on ODMG ODL [4].

Based on a specifically designed array algebra, the RasDaMan query language, RasQL, offers MDD primitives embedded in the standard SQL query paradigm [2]; as usual, a SELECT statement returns a homogeneous[2] set of items. Array expressions can be used in the SELECT part as well as in the WHERE part. The expressive power of RasQL allows to state operations up to, e.g., the complexity of the Discrete Fourier transformation [3]. Recursive operations (such as matrix inversion) are not supported to obtain a safe language. Nevertheless, this enables a wide range of statistical, imaging, and OLAP operations [1].

2.2 Storage Management

RasDaMan employs a storage structure for MDD which is based on the subdivision of an MDD object into arbitrary tiles, i.e., possibly non-aligned subarrays, combined with a spatial index [5]. A choice of different tiling strategies under administrator control accommodates different query patterns. Support of arbitrary tiling is one of the distinguishing features of RasDaMan.

The RasDaMan server resolves the MDD semantics and stores tiles as blobs in a conventional DBMS, along with index and catalog data. As the conventional DBMS is only used as aka "dumb" persistent storage manager, any DBMS type can do, both relational and object-oriented.

2.3 Query Processing

[1] Actually, RasDaMan is used in international projects and is marketed commercially by Active Knowledge GmbH (www.active-knowledge.de).

[2] Queries on variably sized arrays may return arrays with different spatial extent.

During server-based query evaluation, a specialised rewriting heuristic based on about 150 algebraic transformation rules is applied. Rules are derived from MDD operations, relational operations, and their combinations to optimise expressions wrt. evaluation performance and memory usage. Examples for such rules are "*pull out disjunctions while aggregating cell values of an MDD using logical or*" and "*push down geometric operations to expression leafs*". The latter rule ensures that just the minimal amount of data necessary to compute the result of the query branch is read from the storage manager. Common subexpression detection has been extended to detect overlapping regions which then are inspected only once. The choice of physical algorithms, finally, is driven by indexing, tiling, and catalog information. For instance, if an operation does not prescribe any particular tile inspection sequence, iteration order will be in storage order. Tile-based execution pipelines the execution process on tile level whenever possible to reduce memory requirements for intermediate results. Associativity and commutativity of most cell operations yield a huge potential for further optimisation and parallelisation, some of the future research topics.

3 Demonstration

We will demonstrate RasDaMan using the visual frontend rView to interactively submit RasQL queries and display result sets. The system will run in a WindowsNT/Linux client/server configuration. Demonstration will rely on real-life data sets taken from various areas, such as 1-D time series, 2-D Digital Elevation Models, 3-D human brain images, 3-D movie clips, and 3-D/4-D simulation data.

Demonstration will start by showing sample retrieval, thereby introducing basic RasQL concepts. Queries will encompass both search and array manipulation operations; examples will range from 1-D to 4-D, showing in particular dimension-spanning queries. Next, selected queries will serve to demonstrate the effect of several algebraic rewriting rules. This allows to discuss how they contribute to overall performance. Finally, implications of physical data organisation will be presented by applying various range queries to a 3-D MDD stored multiply with different tiling strategies applied.

References

[1] P. Baumann: A Database Array Algebra for Spatio-Temporal Data and Beyond. *Proc. NGITS '99*, Zikhron Yaakov, Israel, 1999, pp. 76 - 93.

[2] P. Baumann, P. Furtado, R. Ritsch, N. Widmann: Geo/Environmental and Medical Data Management in the RasDaMan System. *Proc. VLDB'97*, Athens, Greece, 1997, pp. 548-552.

[3] P. Buneman: The Discrete Fourier Transform as a Database Query. *Tech. Report MS-CIS-93-37/L&C 60*, U. of Pennsylvania, 1993.

[4] R. Cattell: *The Object Database Standard: ODMG-93*. Morgan Kaufmann Publishers, 1996.

[5] P. Furtado, P. Baumann: Storage of Multidimensional Arrays Based on Arbitrary Tiling. *Proc. ICDE '99*, Sydney, Australia 1999, pp. 480-489.

[6] L. Libkin, R. Machlin, and L. Wong: *A query language for multidimensional arrays: Design, implementation, and optimization techniques.* Proc. ACM SIGMOD'96, Montreal, Canada, 1996, pp. 228 - 239.

[7] A.P. Marathe, K. Salem: Query Processing Techniques for Arrays. *Proc. ACM SIGMOD'99*, Philadelphia, USA, 1999, pp. 323-334.

[8] S. Sarawagi, M. Stonebraker: Efficient Organization of Large Multidimensional Arrays. *Proc. ICDE'94*, Houston, USA, 1994, pp. 328-336.

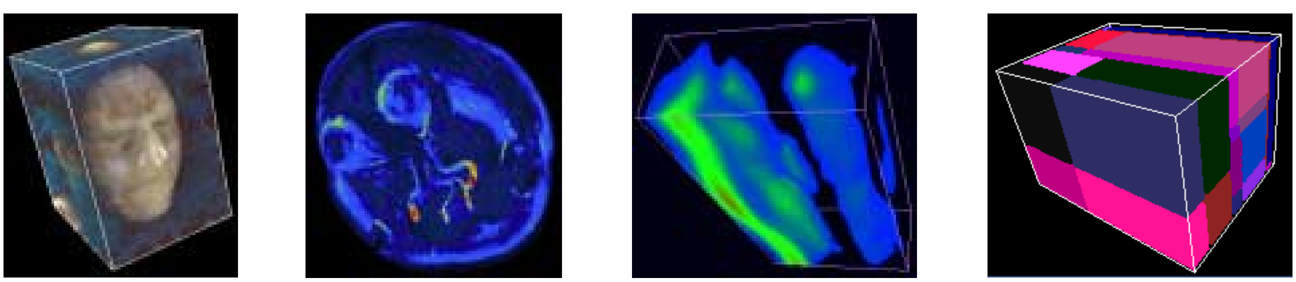

Fig.: Visible Human (3-D), human brain activity (3-D), a climate simulation (4-D), and visualisation of internal tiling structure (3-D). All visualisation done with rView.