Spatio-Temporal Retrieval with RasDaMan

Peter Baumann, Andreas Dehmel, Paula Furtado, Roland Ritsch, 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,furtado,ritsch,widmann}@forwiss.de

Abstract

Database support for multidimensional arrays is an area of growing importance; a variety of high-volume applications such as spatio-temporal data management and statistics/OLAP become focus of academic and market interest.
RasDaMan is a domain-independent array database management system with server-based query optimization and evaluation. The system is fully operational and being used in international projects. We will demonstrate spatio-temporal retrieval using the rView visual query client. Examples will encompass 1-D time series, 2-D images, 3-D and 4-D voxel data. The real-life data sets used stem from life sciences, geo sciences, numerical simulation, and climate research.

1. System Overview

Arrays of arbitrary size and dimension, so-called Multidimensional Discrete Data (MDD), appear in a multitude of database applications; natural sciences, OLAP and statistics, and multimedia comprise but a few representative fields. It is estimated that the larger part of digital data stored worldwide belongs to the MDD category. Nevertheless, MDD are not comprehensively understood by database research as of today, although important research has been accomplished in several subfields. In practice, BLOBs still prevail in multimedia, while statistics and OLAP have developed their own methods of MDD management.

The RasDaMan array DBMS has been developed by FORWISS in the course of an international project partly sponsored by the European Commission [Bau97]. The overall goal of RasDaMan is to provide classic database services in a domain-independent way on MDD structures. Based on a formal algebraic framework [Bau99], RasDaMan offers a query language [Bau98a], which extends SQL-92 [ISO92] with declarative MDD operators, and an ODMG 2.0 conformant programming interface [Cat96]. Server-based query evaluation provides several optimization techniques and a specialized storage manager. The latter combines MDD tiling with spatial indexing and compression whereby an administration interface allows to change default strategies for application-driven database tuning [Fur99]. Array sets resulting from queries are delivered in the client's main memory format or in some a data exchange format as selected by the application.

Research on array data management in DBMSs usually focuses on particular system components, such as storage of multidimensional data [Sar94], or query language [Mar97]. RasDaMan, on the other hand, is a fully implemented and operational generic array DBMSs. This makes it a unique opportunity to study all aspects of multidimensional data management in a holistic way, thereby augmenting focused research done elsewhere.

In [Bau98b] RasDaMan has been demonstrated in combination with the object-oriented database system O2 [Ban92]. This time, RasDaMan will be coupled with Oracle, thereby showing the interoperability capabilities with both relational and object-oriented systems and giving insight into the interworking of RasDaMan and Oracle. Additionally, advanced physical storage tuning, including various array decomposition techniques and transparent compression, will be demonstrated the first time.

Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, 1999.

1.1 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 or 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.

The RasDaMan query language, RasQL, consists of MDD primitives embedded in ODMG's OQL; as usual, a SELECT statement returns a homogeneous set of items. Array expressions can be used in the SELECT part to modify the selected elements, and they can be used in the WHERE part to search for arrays with particular properties. The expressive power of RasQL allows to state operations up to the complexity of the Discrete Fourier transformation [Bun93]. Recursive operations (such as determinants or matrix inversion) are not supported to obtain a language which is safe in evaluation. Nevertheless, this enables a wide range of statistical, imaging, and OLAP operations. The underlying array algebra is described in detail in [Bau99].

1.2 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 to accelerate access to the tile subset affected by a query. A choice of different tiling strategies under control of the database administrator or an application programmer serves to accommodate different query patterns. Support of arbitrary tiling is one of the distinguishing features of RasDaMan. The tiling strategies supported lead to performance increases of access operations to MDD objects, as they allow tuning of tiling to different types of retrieval. The following tiling strategies have been implemented: directional tiling which optimizes accesses along given dimension categories, tiling according to areas of interest which optimizes access to a given set of query regions, and statistical tiling which optimizes access given the statistics of access to an MDD object. These reflect different application requirements regarding types of access. The underlying tiling algorithms minimize the amount of data read for the most frequent accesses, thus reducing execution time. In [Fur99] these algorithms are described and performance comparisons against other tiling strategies are presented which are based on measurements on 3-D data cubes with different tile sizes. Average performance increases of 200% have been observed compared to the performance of regular tiling.

1.3 Query Processing

Demands on Array Query Processing (AQP) differ essentially from the ones on standard Relational Query Processing.(RQP): With RQP, tuples are very small compared to relation size and operations on single tuples (e.g., string comparison) are very inexpensive wrt. CPU costs. The main effort has to be spent on processing large sets of tuples. In contrast, single MDD objects already can reach the scale of Gigabytes, and MDD operations, such as consolidation in OLAP datacubes, become extremely complex and time consuming.

On the logical level, RasDaMan applies a specialized rewriting heuristic based on about 150 algebraic transformation rules derived from MDD operations, relational operations, and their combinations to construct optimized 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 the expressions' leaves". 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. Further, the query tree is searched for common MDD subexpressions. Beyond conventional subexpression matching, the spatial domains are checked for overlapping regions which have to be loaded and computed only once. The choice of physical algorithms, finally, is driven by indexing and tiling information. For instance, if an operation does not prescribe any particular tile inspection sequence, iteration order will be chosen corresponding to storage order. The tile-based execution strategy pipelines the execution process on the level of tiles whenever possible in order to reduce memory requirements for intermediate results and to obtain a high pipelining degree. Due to associativity and commutativity of most cell operations, there is a huge potential for parallelization which will be incorporated in future versions of RasDaMan.

1.4 Architecture

The RasDaMan API consists of *RasQL* and the C++ Raster Library (*RasLib*) which serves for the integration of the MDD type into the C++ language. To make MDD persistent, RasDaMan follows the ODMG-2.0 standard through providing a smart pointer which behaves like a normal C++ pointer capable of managing transient and persistent data in a transparent way.

The cross-platform client-server architecture is realized through standard remote procedure calls. The server architecture consists of the modules *Query Evaluator*, *Index Manager*, *Catalog Manager*, and *Tile Manager*. The *Query Evaluator* parses the query and builds an operator based query tree. Then query optimization takes place in two steps. First, algebraic query rewriting is done, then physical optimization based on tiling and clustering information takes place. The

Query Evaluator is tile-based, operations on MDD items are decomposed into operations on tiles. To identify the tiles involved in a query and to calculate the costs to retrieve them, the *Index Manager* is consulted. The *Catalog Manager* takes care of schema information specified through RasDL. The final execution plan is evaluated by retrieving tile sets from the *Tile Manager* and applying elementary image operations, e.g. spatial or induced operations, on them. An interface layer between RasDaMan modules and the base DBMS, the *Storage Management Interface*, is responsible for the storage and access to all data in persistent storage. This prepares RasDaMan for easy portability between different base DBMSs and storage systems. RasDaMan is implemented in C++ and runs under several Unix versions as well as Windows NT; heterogeneous client/server environments are supported. The server interfaces with the object-oriented DBMS O2 and with relational systems.

2 Demonstration

We will demonstrate RasDaMan using the visual frontend rView, a C++ RasDaMan client, to interactively submit RasQL queries and display result sets containing 1-D to 3-D data. The system will run in a client/server environment with a Unix server and a Unix or Windows NT client. Demonstration will rely on the following data sets: 1-D time series, a 2-D Digital Elevation Model (DEM), 3-D volume CAT scans, a 3-D movie clip, the 3-D Visible Human [Nat90], a 3-D thermal flow simulation result, and a 4-D climate data set. If Internet access can be provided, wide-area queries to our Munich server will be shown.

2.1 RasQL Operations Overview

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 how cross-dimensional queries work.

2.2 Effectivity of query optimization

Next, selected queries will serve to demonstrate the effect of several algebraic rewriting rules. This allows to discuss how they contribute to overall performance.

2.3 Tiling strategies

Finally, implications of physical data organization will be presented. To this end, a sample 3-D volume tomogram and an animation sequence is stored with different tiling strategies. A particular administrator tool allows to visually inspect the tiling of MDD instances. Queries such as sub-cube extraction and cuts along the three different space axes clearly indicate strengths and weaknesses of particular tiling schemata. Various tiling strategies should be offered as a database tuning tool similar to indexes on tables in relational DBMS for optimal query performance.

References

[Ban92] F. Bancilhon, C. Delobel, P. Kanellakis: *Building an Object-Oriented Database System*. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[Bau97] P. Baumann, P. Furtado, R. Ritsch, N. Widmann: Geo/Environmental and Medical Data Management in the RasDaMan System. *Proc. of the VLDB'97 Conference*, Athens, Greece, 1997.

[Bau98a] P. Baumann: *The RasDaMan Array Algebra*. RasDaMan Technical Report for012, FORWISS, 1998.

[Bau98b] P. Baumann, A. Dehmel, P. Furtado, R. Ritsch, N. Widmann: The Multidimensional Database System RasDaMan. *Proc. ACM SIGMOD'98*, Seattle, USA 1998, pp. 575 - 577.

[Bau99] P. Baumann: An Algebra for Domain-Independent Multidimensional Array Management in Databases. RasDaMan Technical Report for016, 1999.

[Bun93] P. Buneman: The Discrete Fourier Transform as a Database Query. *Technical Report MS-CIS-93-37/L&C 60*, University of Pennsylvania, 1993.

[Cat96] R. Cattell: *The Object Database Standard: ODMG-93*. Morgan Kaufmann Publishers, 1996.

[Fur99] P. Furtado, P. Baumann: Storage of Multidimensional Arrays Based on Arbitrary Tiling. *Proc. ICDE'99*, Sidney - Australia 1999.

[ISO92] The International Organization for Standardization (ISO): *Database Language SQL*. ISO 9075, 1992(E), 1992.

[Mar97] A. P. Marathe, K. Salem: A Language for Manipulating Arrays. *Proc. of VLDB'97 Conference*, Athens, Greece, 1997.

[Nat90] National Library of Medicine (US) Board of Regents: *Electronic Imaging: Report of the Board of Regents*. US Department of Health and Human Services, Public Health Service, National Institutes of Health, NIH Publication 90-2197, 1990.

[Sar94] S. Sarawagi, M. Stonebraker: Efficient Organization of Large Multidimensional Arrays. *Tenth Int. Conf on Data Engineering*, pp. 328-336, Houston, Feb. 1994.

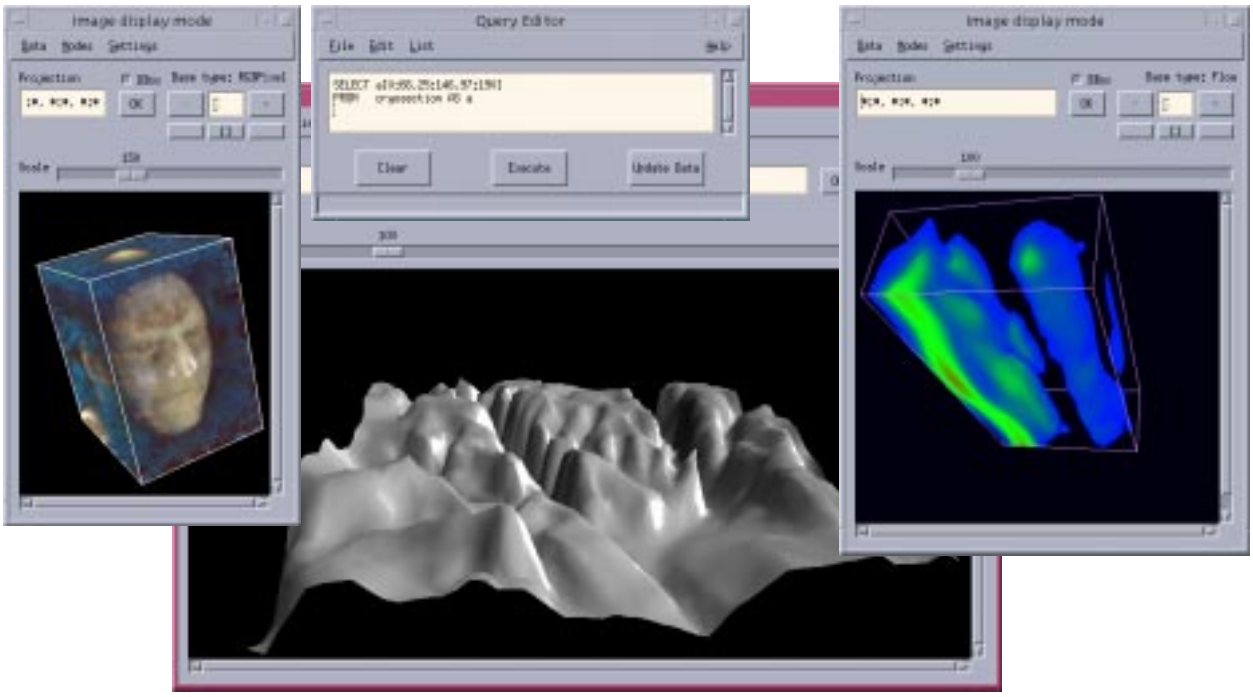

Figure 1: Sample MDD visualization using rView. The image to the top left shows a volume rendering of a section of the Visible Human obtained with the query displayed to its right. Top right and background images visualize a climate data set and a DEM of the Grand Canyon, respectively.

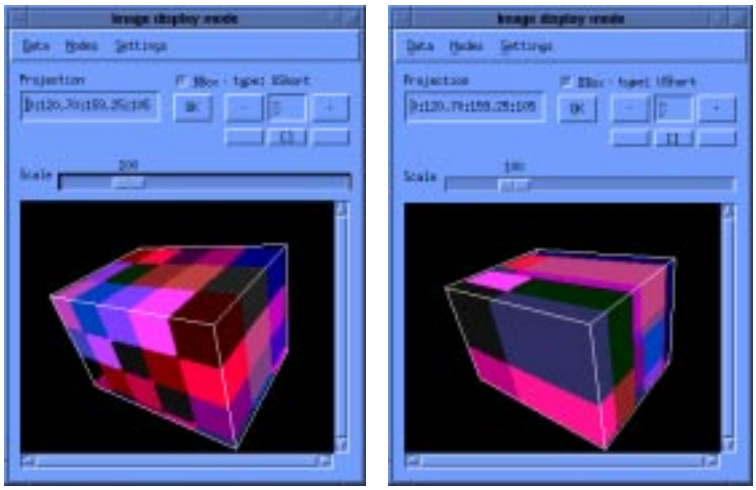

Figure 2: Visualization of different tiling strategies of a3-D movie sequence with rView: regular tiling (left) and areas of interest resulting in nonaligned tiles (right).