Geo/Environmental and Medical Data Management in the RasDaMan[1] System

Peter Baumann   Paula Furtado[2]   Roland Ritsch   Norbert Widmann

FORWISS (Bavarian Research Centre for Knowledge-Based Systems)
Orleansstr. 34, D-81667 Munich, Germany
{baumann,furtado,ritsch,widmann}@forwiss.tu-muenchen.de

Abstract

Multidimensional discrete data (MDD) – i.e., arrays of arbitrary size, dimension, and base type – appear in a variety of business, technical, and scientific application fields. RasDaMan is an effort to give comprehensive domain-independent MDD database support. Based on a formal algebraic array model, RasDaMan offers declarative array operators embedded in standard SQL; key DBMS components are an MDD query optimizer and a streamlined storage manager for efficient access to subsets of huge arrays. We present the RasDaMan approach to MDD management based on the medical and geographic application fields addressed in the project.

1 Introduction

In principle, any natural phenomenon becomes spatio-temporal raster data of some specific dimensionality once it is sampled and quantised for storage and manipulation in a computer system; moreover, a variety of artificial sources such as simulators, image renderers, and business data analyzers produce raster data. The management problems posed by different types of raster data have traditionally been treated separately. In the past, much work was focused on image databases, resulting in a large research body to draw from [Cha92]. Attention was also dedicated to management of statistical data in DBMSs. Recently, multidimensional business data from OLAP (on-line analytical processing) applications has become the most prominent case of raster data. In [Sho97], a comparison of these two application areas can be found.

Even though these domain-oriented approaches have led to important conclusions regarding requirements in specific application areas, we claim that all occurring raster data types pose similar data management requirements. Our investigation in the traditionally separate geo/environmental and medical raster data application areas further confirm that.

The common characteristic all raster data types share is that they correspond to large multidimensional arrays. In addition, raster data is to be distinguished from other multidimensional spatial data of vectorial nature (e.g. "geo data"), hence we use the term Multidimensional Discrete Data or MDD.

Regarding MDD functionality, the need to execute operations where operands and outcome can be of distinct cell types and dimensionalities (e.g. a 1-D histogram resulting from a 4-D spatio-temporal data object) urges for a unified approach. Furthermore, functions such as multidimensional subarea extraction, layer extraction (projection), and aggregation along specified dimensions play an important role in all application fields.

The paradigm of MDD management can satisfy all the domains mentioned, provided it is carefully designed and implemented. Design should be based on a formal model, and on clean semantics for declarative operators. Implementation requires storage support for multidimensional neighborhood and preaggregation.

In this paper, the RasDaMan approach to storage,

Proceedings of the 23rd VLDB Conference Athens, Greece, 1997

[1]sponsored by the European Commission in the ESPRIT Domain 4: Long-Term Research under grant no. 20073.

[2]PhD work sponsored by a PRAXIS XXI scholarship.

manipulation, and retrieval of MDD in databases and its application to the geographic, environmental and medical areas is presented. The MDD management approach follows the line first published in [Bau92] and [Bau94], which meanwhile has undergone substantial revision. Compared to other work in the field, the main distinguishing features are the following: domain-independence; comprehensive approach combining a conceptual MDD model based on well-defined mathematical semantics with a streamlined MDD storage management; and close adherence to the relevant standards, SQL-92 [Can93] and ODMG-93 [Cat96].

The sequel of this paper is organized as follows. Section 2 contains a brief description of the envisaged application areas. In Section 3, we outline the conceptual model followed by RasDaMan, and in Section 4 we focus on some aspects of the system. Section 5 presents the conclusions and plans for future work.

2 Application Areas

Among the aforementioned fields using raster data, in the project focus has been set on medical and geographic applications. Some investigation is also done into using an MDD DBMS for OLAP applications.

In the medical environment, digital archival of patient data is becoming more and more standard [Gar95]. Data is produced in a wide variety of forms such as 1-D curves (e.g. ECG), 2-D images (e.g. Ultrasound), and 3-D volumetric data (e.g. from Computer Tomography, CT, with typically 256 greyscale slices with 256x256 pixels each). For interactive computer-supported consultations, the examiner needs advanced retrieval mechanisms, such as projections along different axes in volumetric data, cutouts and zooms, and associative search support for querying the database for particular medical phenomena (content based retrieval). An application pilot providing PACS (Picture Archiving and Communication System) functionality will be evaluated under real-life conditions by the Spanish Hospital General de Manresa.

Geographic Information Systems (GIS) [Tom90] use information in different representations: e.g. vectorial data is needed for producing maps, while land usage is analyzed based on MDD from remote sensing. The full power of GIS can be utilized if vectorial and raster information are combined. While some systems store vectorial data in a DBMS, for raster data the file system is commonly used. A large part of geographic information is acquired or generated as MDD, such as output of climate or geological simulations (which typically produce 3-D or 4-D MDD) or remote sensing, e.g. the Landsat satellite images, which consist of 5760×7020 pixels over seven 8-bit bands resulting in a total size of 270 MB for one image. As this data is continually generated and historical information is indispensable for research (e.g. development of the ozone layer over time), a huge amount of data has to be stored online, resulting in archives for remote sensing data that will reach Petabyte size in the foreseeable future. Key operations needed on this data are content based queries (e.g. retrieve all pictures with less than 10% clouds), a declarative query language capable of specifying MDD operations like overlaying or arithmetic operations and transparent integration of tertiary storage. To assess the RasDaMan DBMS in this environment, an application will be implemented based on RasDaMan for the Spanish National Geographic Institute.

In all application areas mentioned, complex operations will be applied on huge amounts of data in the future. By providing these services close to the data source, i.e. the database, it is possible to efficiently execute such operations, to do optimizations and to minimize data transfer to the client application.

3 A Generic MDD 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 dimensionality, size, and array cell type (due to the C++ binding this means that any valid C++ type or struct is admissible). Each dimension's lower and upper bound can be fixed at data definition time, or can be left variable.

The formal semantics has been inspired by AFATL Image Algebra [Rit90] which turns out particularly suitable for a declarative query language in that it is abstract on a high level, mathematically rigorous, and powerful enough to specify image transforms in any number of dimensions. Studying Image Algebra also has helped in finding operations feasible for the model and the RasDaMan query language RasQL (*Raster Query Language*) which consists of MDD primitives embedded in ISO SQL-92[3]; as usual, a `SELECT` statement returns a homogeneous set of items. The general question on the desirable expressiveness of an MDD algebra currently must be considered as unsolved, though.

To outline RasQL, we will make use of two running examples taken from the above mentioned application fields. In the 2-D case, we look at Landsat satellite images with 5760×7020 pixels composed of seven-integer records for the seven sensors on board; we assume a class `LandsatImage` with an MDD-valued attribute `img`. For 3-D MDD, volume tomograms (VTs) are used which are gained from medical CT as sequences of 2-D slices; we assume a class `VolumeTomogram` containing an MDD-valued attribute `cube`.

[3] but with the more functional style of ODMG OQL [Cat96].

In the sequel, we present the currently available operation categories included in RasQL. First, however, an auxiliary meta information function is introduced. For some MDD `m` of dimension `d`, function `spatial_domain(m)` retrieves an array of integer pairs ($\texttt{lo}_i$,$\texttt{hi}_i$) containing `m`'s current lower and upper bound for all dimensions $0 \leq i < d$.

Example 1: "The number of x/y slices in volume tomogram v."

```
spatial_domain(v.cube)[2].hi -
spatial_domain(v.cube)[2].lo + 1
```

Operations on cell subsets ("geometric" or "spatial" operations). Their characteristic is that cell values are not changed, but a subset of cells is selected. This includes trimming (rectangular cutouts) and projection (extraction of lower-dimensional subarrays).

Example 2: "From all VTs, cuts along the three spatial axes through point (x_0,y_0,z_0)."

```
SELECT vt.cube[ x0,  *:*, *:* ],
       vt.cube[ *:*, y0,  *:* ],
       vt.cube[ *:*, *:*, z0  ]
FROM   VolumeTomograms as vt
```

Induced operations. For each operation available on the MDD cell type, a corresponding so-called induced operation is provided which simultaneously applies the base operation to all cells of an MDD. Both unary (e.g., record access and contrast enhancement) and binary operations (e.g., masking an image) can be induced.

Example 3: "Band 3 of all Landsat images, pruned with binary mask m."

```
SELECT l.img.band3 * m
FROM   LandsatImages as l, mask as m
```

Aggregation. The condenser statement, a second-order construct, iterates over a (sub)set of cells in an array, combining all values through the operation indicated. No particular iteration sequence is defined; to ensure a well-determined result, only operations which are associative and commutative are admitted. The syntax for condensing, through some operation `o`, expressions `e(a[x])` over all cells `a[x]` in array `a` which (optionally) fulfill some predicate `p(a[x])` is

```
CONDENSE o OVER x IN spatial_domain(a)
WHERE p(a[x]) USING e(a[x])
```

Condensers are useful not only for extracting scalar information from multidimensional data sets, but they also allow retrieval conditions to be stated on complete arrays or parts thereof; there they often are used to "collapse" Boolean MDDs resulting from induced comparisons. In general, condensers can dramatically reduce the amount of data to be transferred to the client application. The required commutativity and associativity of the condensing operations allows the optimizer to rearrange iteration sequence or parallelize condenser evaluation.

Example 4: In the Human Brain Database [Rol96], 3-D and 4-D VTs of human brains are maintained; during VT insertion, a normalized canonical representation is generated which ensures that, although brains come from subjects with individual brain shapes, brain regions always appear at the same position in the VT. Hence, brain areas can be defined through pixel masks for database search. The following example retrieves all brains where in the Hypothalamus a critical activation has been measured:

```
SELECT vt
FROM   VolumeTomograms as vt,BrainAreas as br
WHERE  CONDENSE or
       OVER x IN spatial_domain(vt.cube)
       WHERE br.mask
       USING vt.cube[x] > 230
       and br.name="Hypothalamus"
```

Partial updates. Because of the large size of the data items on hand, we feel that support for partial MDD updates is necessary. To this end, the SQL `update` statement has been included in RasQL with an extension to specify an MDD cutout on the left-hand side of the assignment clause.

Example 5: Add another scanner CT into a VT.

```
UPDATE VolumeTomograms
SET cube[ *:*, *:*,
    spatial_domain(cube)[2].hi+1 ]
    = <CT data>
WHERE  ...
```

Physical data independence. By default, RasDaMan delivers MDD in the client machine's and programming language's main memory representation for arrays, ready for further processing. On request, MDD are accepted or delivered using the particular data exchange format indicated in the query. Of course, the format must be able to host the structure to be encoded – a seven-band multispectral image cannot be cast into TGA, whereas a single band can.

Example 6: Store a Landsat image which has been read from tape in BIL format.

```
INSERT INTO LandsatImages
VALUE  data = invbil(<program variable>)
```

Example 7: Retrieve band 3 of all Landsat images in TGA format.

```
SELECT tga(l.img.band3)
FROM   LandsatImages as l
```

In general, MDD expressions can be used in the `select` part of a query and, if the outermost expression result type is scalar, also in the `where` part.

Among the list of further operations to be implemented are affine transformations[4] – especially scaling – and generalized convolution; for example, a Discrete

[4] Affine transformations are highly nontrivial if treated accurately. They involve interpolation in continuous space and subsequent resampling. It is an open issue how this can be modeled adequately without too much conceptual overhead.

Fourier Transform can then be expressed as a database query [Bun93]. Further, dimension hierarchies are on the agenda.

4 Architecture

The system is based on a client/server architecture with query processing on server side. In order to make objects of type MDD persistent and to achieve seamless integration of type MDD into the programming language, the client C++ API is ODMG compliant. It is extended with classes dealing with MDD and supports ODMG conformant creation and invocation of RasQL queries. These are evaluated by the Query Processor which performs query optimization based on indexing, clustering, and device information provided by the Storage Manager. The final execution plan is evaluated by retrieving MDD subareas from the Storage Manager and applying elementary image operations, e.g. spatial or induced operations, on them.

The modules of RasDaMan are based on the commercial ODBMS O_2 [Ban92]. An interface layer between RasDaMan modules and the base DBMS is responsible for the storage and access to all data in secondary and tertiary storage. This prepares RasDaMan for easy portability between different base DBMSs and storage systems.

4.1 Storage Management

Due to its potentially huge size, MDD requires specialized internal storage structures designed to minimize the number of pages accessed when an operation is executed on an object or part of it. The limiting factor for the overall performance will be the degree to which spatial proximity between multidimensional items is maintained.

The RasDaMan system adopts a storage structure for MDD which is based on the subdivision of an MDD object into arbitrary multidimensional rectangular subarrays (*tiles*)(possibly nonaligned), combined with adequate clustering and a spatial index to accelerate tile access [Fur93].

While there is no fixed optimal structure for all operations and all MDD objects, our research has shown that subdivision of an MDD object into arbitrary multidimensional rectangular tiles allows for higher optimization of MDD operations. A system supporting such generic tiling allows the definition of tiles corresponding to areas of interest, which is a means to preserve spatial proximity between contents of an MDD object. Performance can therefore be improved by optimizing for the most common access patterns and operations on MDD. Based on usage statistics, user provided information or data analysis, the most adequate tiling for an object can be adopted.

Such a generic subdivision of the object's domain and the need to support spatial access to the objects (typically access to a multidimensional rectangular area) requires a specialized index which supports efficient multidimensional access. In RasDaMan, a spatial access method provides fast selection of MDD parts. For each object, a spatial index is created which maintains the information about the object's tiles, including the coordinates for each tile.

4.2 Query Processing

A query statement of RasML is first translated into an operator based query tree which consists of MDD operations (e.g. induction or trimming) and relational operations (e.g. selection or cross product) as nodes, and data sources as leaves. After some equivalence preserving transformations for normalization purposes, heuristic optimization based on algebraic rules derived from AFATL Image Algebra (e.g. pull out disjunctions while condensing using logical `or`) and the relational algebra (e.g. push selections into the join operation) takes place.

In the next step, the tree is searched for common MDD subexpressions. In addition to lexical comparison, the spatial domains of MDD terms have to be examined. Two MDD expressions have some exploitable common components if their spatial domains are overlapping. After rewriting the query according to common subexpressions, an execution plan is generated. The choice of physical algorithms is driven by indexing, clustering, and device information.

We now discuss how tile based query execution potentially reduces disk I/O, memory requirement, and computation time. First, the Executor accesses an MDD, tile by tile. If the operation does not prescribe a particular tiles sequence, iteration order is chosen corresponding to storage order. Otherwise, direct access to single tiles is supported by the spatial index of the Storage Manager. If tiles are small in size, and regions of interest are divided into minimal tiles, retrieval of the minimal tile set covering the requested area delivers nearly no unnecessary data. Therefore, querying subimages becomes more efficient.

Second, tile based execution of condenser sub expressions resulting in a smaller MDD of some lower dimensionality leads to small intermediate result sets and reduces memory usage. Further, if a condensing operation supports "lazy evaluation", the result is computed, whenever possible, without reading and applying operations on the whole MDD tile set. Finally, tile based execution of operations, like induce and condense, provides high parallelization potential.

5 Conclusion and Future Work

The RasDaMan DBMS, which addresses cross-dimensional, domain-independent management of MDD, is being used in the development of both a geo/environmental information system and a medical image database management application. Using the RasDaMan C++ API, application developers can use the MDD type in an integrated way. RasQL provides MDD operations which otherwise would have to be implemented by the application. Compliance of these interfaces to the SQL and ODMG standards help for a smooth integration with conventional DBMS technology.

Several features in RasDaMan aim at providing applications with an efficient DBMS to be used as server. The declarative nature of RasQL allows the choice of optimal execution plans for queries. A specialized structure supports efficient management of large amounts of MDD on secondary storage. Data independence offers further possibilities for optimizing the mapping of queries to physical storage in a way transparent for the application[5]. The operations provided by the DBMS lessen demands on processing power and memory of the clients, as general MDD functionality is computed on server side and only the relevant preprocessed data is transferred over the network.

Even though the first version of the system is implemented on top of an ODBMS, the MDD concepts presented are not tied to object-oriented technology; experiments with a relational system have been made [Fur93]. The next steps include further implementation, and evaluation using the PACS and GIS application pilots at end-user sites; two project partners have started development of these applications. In parallel, we proceed with further investigation on conceptual refinement of the query language, as well as on query optimization and storage management, namely regarding tertiary storage and compression of MDD. In addition, detailed benchmarking to quantify the performance characteristics of multidimensional databases will be used to assess the concepts presented. The most current information on RasDaMan can be found on our WWW pages:
`http://www.forwiss.tu-muenchen.de/~rasdaman/`

Acknowledgements

The RasDaMan team at FORWISS welcomes Peter Zoller as a new member. We gratefully acknowledge the domain contributions of our end users CNIG and HGM.

[5]This and many of our concepts cannot be implemented using current object-relational DBMSs like Informix/Illustra.

References

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

[Bau92] P. Baumann. Language Support for Raster Image Manipulation in Databases. *Proc. Int. Workshop on Graphics Modeling and Visualization in Science & Technology,* Darmstadt, Germany, April 1992.

[Bau94] P. Baumann. On the Management of Multidimensional Discrete Data. *VLDB Journal, Special Issue on Spatial Database Systems,* 4(3), pp. 401-444, 1994.

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

[Can93] S. Cannan, G. Otten. *SQL - The Standard Handbook.* McGraw-Hill, 1993.

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

[Cha92] S.-K. Chang, A. Hsu. Image Information Systems: Where Do We Go From Here? *IEEE Transactions on Knowledge and Data Engineering,* 4(5), pp. 431-442, October 1992.

[Fur93] P. Furtado, J. Teixeira. Storage Support for Multidimensional Discrete Data in Databases. *Computer Graphics Forum - Special Issue on Eurographics'93 Conference,* 12(3), pp. 89-100, September 1993.

[Gar95] H. Garcia, D. Yun. Intelligent Distributed Medical Image Management. *Proc. SPIE Medical Imaging Conference,* pp. 80-91, February 1995.

[Rit90] G. Ritter, J. Wilson, J. Davidson. Image Algebra: An Overview. *Computer Vision, Graphics, and Image Processing,* 49(1), pp. 297-331, 1990.

[Rol96] P. Roland, K. Zilles. The Developing European Computerized Human Brain Database for All Imaging Modalities. *NeuroImage,* 4(1), pp. 39-47, 1996.

[Sho97] A. Shoshani. OLAP and Statistical Databases: Similarities and Differences. *ACM Transactions on Database Systems,* 1997.

[Tom90] C. Tomlin. *Geographic Information Systems and Cartographic Modelling.* Prentice Hall, 1990.