LARGE-SCALE MULTIDIMENSIONAL COVERAGE DATABASES *

Peter Baumann
Active Knowledge GmbH
Kirchenstr. 88, D-81675 Munich, Germany; Email: baumann@active-knowledge.com

Erhard Diedrich
German Aerospace Center
Oberpfaffenhofen, D-82234 Wessling, Germany; Email: erhard.diedrich@dlr.de

Clemens Glock
Bayerisches Landesvemessungsamt
Alexandrastr. 4, D-80538 München, Germany; Email: clemens.glock@blva.bayern.de

Michael Lautenschlager, Frank Toussaint
Max-Planck Institut for Meteorology
Bundesstr. 55, D-20146 Hamburg, Germany; Email: {lautenschlager|toussaint}@dkrz.de

ABSTRACT

With GIS technology for Web-enabled vector and meta data access becoming mature, the next quest is towards fast, flexible services on extremely large raster data assets. Currently 2-D images (satellite imagery, ortho photos) prevail, but at the horizon higher-dimensional objects appear, such as 3-D time satellite image time series and seismic data, and 4-D climate simulations.

At the same time, database researchers are getting aware of the specific challenges such multidimensional coverages pose. Some promising recent work shows that database systems with optimising query languages actually are superior to file-based systems in performance, flexibility, and scalability.

We demonstrate three applications: a 2-D ortho photo maintained by the Bavarian State Survey, a 3-D AVHRR time series deployed as experimental Web service by the German Aerospace Association, and a 4-D climate simulation database implemented by Max-Planck-Institute for Meteorology. All of them use the identical underlying database technology, namely RasDaMan / Oracle with the RasDaMan raster query language.

WHY RASTER DATABASES?

Increasingly raster data are complementing vectorial data in geo applications. On the one hand, storage facilities allow to keep substantial raster archives online, on the other hand acquisition of raster data is rapidly getting easier and less expensive. Additionally, timeliness of raster imagery is significantly higher as compared to vector data, due to the easier acquisition.

This advance in technology is paired by a rapidly increasing user demand for flexible, ubiquitous access to large archives of raster images and maps of various types. A broad range of raster imagery has to be considered in geo data management: grayscale and colour aerial images, multi- and hyperspectral satellite images, topographical raster maps, RADAR and laser scan, and Digital

* Work in part funded by the European Commission under grant no. IST-1999-11009 ESTEDI, www.estedi.org, and IST-2000-28766 EuroClim, http://euroclim.nr.no.

Elevation Models (DEMs). Sometimes it is desirable to keep older versions, resulting in 3-D time series data cubes.

Retrieval technology for large raster repositories, however, is lagging behind. Raster archives today commonly are implemented in a file-based manner; databases serve only for meta data search, but not for image retrieval itself. Hence, storage usually is driven by the data acquisition process rather than by user access patterns. Moreover, versatile retrieval as known from database query languages for alphanumeric data is impossible, not to speak of other services like transaction support, The following example we consider as representative for an important, although still basic class of queries: "*Overlay selected channels of a multi-band satellite image with cadastral maps; colour all areas in blue that would be flooded if water rose to level L, based on the DEM; do this for the geographic area selected; zoom the result into my browser window*". Notably, each of the map items usually has a distinct pixel resolution.

Aside from flexibility in task definition, there are several more arguments which advocate the use of database systems. Query languages allow to define complex tasks to the server rather than a small set of atomic steps as in procedural APIs; the consequence is that the query optimiser gains a lot of freedom to rephrase the query optimally for the particular situation[1]. Further, application integration is much higher because there is one central instance in charge of data integration and consistency. All in all, file-based solutions frequently re-invent all the features which have been developed by database technology over decades; using existing mature technology obviously is preferable.

A system offering all these classical database features on raster maps is RasDaMan[2] (Baumann 1999, Ritsch 1999, Widmann 1999). It is implemented as middleware running on top of a relational database system, offering a query language streamlined to semantically adequate raster retrieval.

In this paper, we present a series of cross-dimensional raster database case studies: a 2-D seamless aerial image map, a 3-D seamless map extended in the time dimension, and a 4-D database of climate simulation results. All use the same database platform, RasDaMan.

The remainder of this paper is organized as follows. In Section 2, a brief overview of the RasDaMan model, query language, and architecture is presented. Sections 3, 4, and 5 describe the 2-D, 3-D, and 4-D databases. Section 6 reviews the state of the art, and Section 7 concludes the paper.

RASDAMAN AS AN EXAMPLE FOR MULTIDIMENSIONAL DATABASE SERVICES

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, spatial extent, and array cell type. Following the relational database paradigm, RasDaMan also supports sets of arrays. Hence, a RasDaMan database can be conceived as a set of tables where each table contains a single array-valued attribute, augmented with an OID system attribute.

Arrays can be built upon any valid C/C++ type, be it atomic or composed ("struct"), based on ODMG's (Cattell 1997) type definition language. Arrays are defined through a template `marray<b,d>` which is instantiated with the array base type `b` and the array extent (*spatial domain*) `d`, specified by the lower and upper bound for each dimension. Thus, an unbounded colour DOP can be defined by

```
typedef marray< struct{ char red, green, blue; }, [ *:*, *:* ] > RGBOrthoImg;
```

[1] Notably this optimisation step itself is not costly. Usually it takes less than a millisecond.
[2] See www.rasdaman.com

Array Retrieval

Like SQL, a RasQL query returns a set of items (in this case, MDD objects).

Trimming produces rectangular cut-outs, specified through the corner coordinates.

Example 1: "*A cut-out between (1000,1000) and (2000,2000) from all ortho images*":

```
SELECT OrthoColl[1000:2000,1000:2000]
FROM   OrthoColl
```

For each operation available on the cell (i.e., pixel) type, a corresponding *induced operation* is provided which simultaneously applies the base operation to all MDD cells. Both unary (e.g., record access) and binary operations (e.g., masking and overlaying) can be induced.

Example 2: "*Topographical map bit layer 3 oveölaid with the (grayscale) ortho image*":

```
SELECT Ortho overlay bit( Map, 3 ) * 255c
FROM   Map, Ortho
```

In general, MDD expressions can be used in the `SELECT` part of a query and, if the outermost expression is of type Boolean, also in the `WHERE` part. See (Baumann 1999) for further query constructs such as condensers (the MDD counterpart to aggregates).

The expressiveness of RasQL enables a wide range of signal processing, imaging, and statistical operations up to, e.g., the Fourier Transform. The expressive power has been limited to non-recursive operations, thereby guaranteeing termination of any well-formed query.

Physical Array Storage

Raster objects are maintained in a standard relational database, based on the partitioning of an MDD object into *tiles* (Furtado 1999). Aside from regular grids, any user or system generated partitioning is possible (Fig. 1). A geo index is employed to quickly determine the tiles affected by a query. Optionally tiles are compressed using one of various choices, using lossless or lossy (wavelet) algorithms; query results can be compressed for transfer to the client. Both tiling strategy and compression comprise database tuning parameters.

Tiles and index are stored as BLOBs in a relational database which also holds the data dictionary needed by RasDaMan's dynamic type system. Adaptors are available for Oracle, IBM DB2, and IBM Informix. A coupling with object-oriented O2 has been done earlier, showing the wide range of DBMSs with which RasDaMan can interoperate.

Query Evaluation

Queries are parsed, optimised, and executed in the RasDaMan server. The parser receives the query string and generates the operation tree. A number of optimisations is applied to a query tree prior to its execution (Ritsch 1999). Of the 150 heuristic rewriting rules, 110 are actually optimising while the other 40 serve to transform the query into canonical form. All rules are based on the algebra.

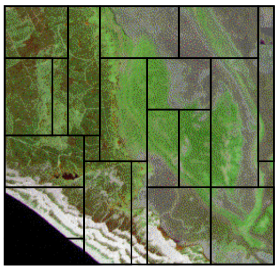

Fig. 1: arbitrary 2-D tiling

Execution of queries is parallelised. Right now, RasDaMan offers inter-query parallelism: A dispatcher schedules requests into a pool of server processes on a per-transaction basis. Current research work involves intra-query parallelism where the query tree transparently is distributed across available CPUs or computers in the network (Hahn 2002). First performance results are promising, showing speed-up / #CPU ratios of 95.5%.

For arrays larger than disk space, hierarchical storage management (HSM) support is being developed (Reiner 2002).

THE 2-D CASE: SEAMLESS AERIAL MAPS IN BAVARIA

The Bavarian Land Survey Authority (Bayerisches Landesvermessungsamt, BLVA) maintains the official maps of the State of Bavaria which covers approximately 70,000 square kilometres, about 20% of Germany.

Ortho photos in the past could only be ordered by indicating the identifier according to the cadastral nomenclature. This has several shortcomings:

- Users have to know the nomenclature which frequently is meaningless to them and makes ordering error prone; actually, experience in several other German states shows that false orders happen not just occasionally.
- Images always cover a whole grid square. Sometimes users want only a subset but nevertheless have to order (and pay) the whole image; sometimes they need a larger area and get a set of files comprising the desired map plus excess data.
- Resolution is fixed, users can not indicate an arbitrary scale factor and resolution which is feasible in many situations (e.g., to get a quick overview).

Hence, it is desirable to have only one seamless map on which users can navigate interactively. As an additional benefit, such an approach requires considerably less human effort to satisfy customer requests. This is urgently needed in view of the anticipated wave of user requests – for example, in the state of Hassia where geo data access now is for free for public authorities, 7,000 regions – each involving a set of DOPs – have been requested in the first three months of 2002 alone, to be handled by one employee.

BLVA therefore has decided to switch from sets of aerial images to one seamless image with free area and resolution selection (Glock 2002). To this end, several raster management systems have been thoroughly evaluated.

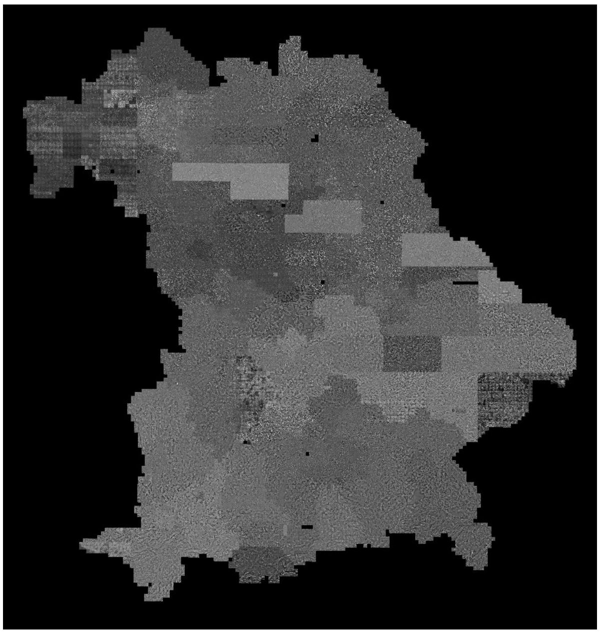

Fig. 2: overview image of Bavaria DOP map (zoom of the DOP map to browser window size)

Data Properties

The base data set consists of 13,500 images with a size between about 40 and 50 MB each, with a ground resolution of 40 cm. These are stored grouped by county and with duplicates to ensure that every county is covered completely. Including these duplicates there is a total of 16,500 images. While in principle the base set of 13,500 images is sufficient as data source, it would have been too tedious a task to sort out images not needed, hence simply all 16,500 have been imported. Seamless image size is 950,000 x

1,000,000 pixels (Fig. 2).

During the flight campaign, aerial images tentatively are taken in an overlapping way. This data redundancy has to be handled. Decision was made to neglect redundancy due to overlaps; during import a simple “last pixel wins” strategy was adopted. The slight loss of information is more than outbalanced by the easier handling for the users who can navigate now over a single, continuous, coherent area.

The null values in the rim of each file have to be removed during import. RasDaMan offers two ways to handle this: either by passing a polygon bounding box for each image which delineates meaningful values, or by defining a pixel value, in this case: 0, as transparent value. As the second alternative incurs small errors in situations where the actual intensity value within the DOP by coincidence is zero, the first strategy has been adopted.

Obviously radiometric correction is not complete in Fig. 2. Currently, up to thousand DOPs can be radiometrically balanced simultaneously. In general, radiometric correction is a photogrammetric task in the preprocessing phase and, as such, beyond the scope of the raster server itself.

Architecture

Database set-up is as follows. In a first run, an empty overall map is generated. This allows, among others, to set global parameters such as compression and tiling schemes used. Notably, compression has to be lossless – as a public authority, BLVA must be able to deliver original, unmanipulated data. This excludes techniques such as ECW[3]. From the various compression schemes available in RasDaMan, `zlib` was chosen which is similar to well-known ZIP. The tiling scheme used was a simple one which only imposes an upper limit of about 250 kB on tile size. To speed up zoom access, an image pyramid (Burt 1983) is maintained by the RasDaMan server. After various tests of different physical database designs to get experience with issues such as backups of huge volumes on DLT tapes, an Oracle datafile size of 20 GB was chosen.

The raster part of the Oracle database occupies about 700 GB. At the time of this writing, work has started to add the colour DOP which will result in an RGB image of over 2 TB estimated size.

RasDaMan is operated with five parallel server processes for satisfying an expected clientele of about 5,000 users. The resulting performance (on a 533 MHz DEC Alpha server) allows to access any position at arbitrary zoom level in one second or less.

THE 3-D CASE: DLR SATELLITE IMAGE TIME SERIES

DLR (German Aerospace Assoc.) is a large trans-national provider of earth observation raw data and derived products. Image data files are provided through ftp archives for download, with extensive meta information to determine the files needed. Web-based navigation interfaces support meta data search. With DLR’s *EOWEB Interactive Data Service*, this navigation is extended to the image data itself, at the same time liberating them from file boundaries. With this front end, satellite data are treated as multi-dimensional discrete data (MDD) which are stored in full resolution in the database.

Each map object consists of a seamless map which extends into the time dimension as third axis (Fig. 3). Cell values have been prepared as 8-bit quantities, hence each object is a 3-D cube of integer pixels. The database currently contains, among others, AVHRR SST and LST as well as NDVI data,

[3] See www.gisdevelopment.net/technology/ic/ techip0003.htm

for Europe and the Mediterranean. By the end of 2002, 10,000 images have been imported into the time series cubes.

Extraction of relevant information is therefore performed via operations on the content of the pixels of the raster data. This results in flexibility and high performance. On the other hand, the data have to be loaded into the database and a data model appropriate to the physical parameters of the data as well as to the expected queries of the users has to be chosen. Since satellite data more and more become standard in GISs, data presentation by RasDaMan has to be compliant to GIS data formats.

Aside from the interactive interface, an OGC Web Map Service (WMS) and Web Coverage Service (WCS) are available, implemented as RasDaMan Java clients. Actually, the browser interface relies on WCS. Hence, as one of few data sets worldwide the same data set supports both WCS and WMS services simultaneously.

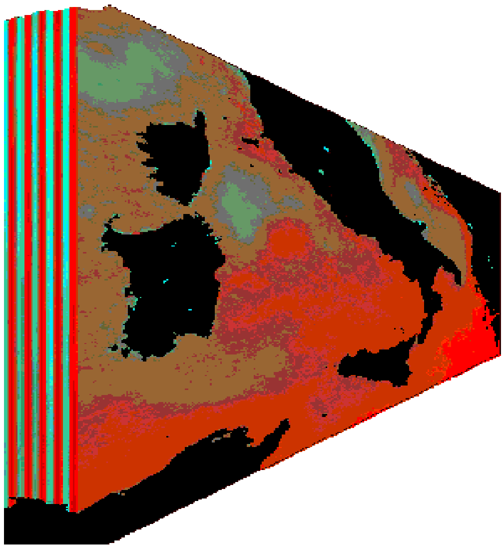

Fig. 3: 3-D cutout from DLR AVHRR SST time series database

As outlined, retrieval conveys very different patterns and access directions. Consequently, it is impossible to find a tiling strategy which is optimal for all cases. Retrieval of 2-D maps for particular time stamps suggests a “horizontal” tiling into slices as given by the input images. Extraction of time series, on the other hand, suggests a tiling into vertical tubes. Obviously each tiling strategy which is optimal for one case puts a penalty on the other case. The actual tiling pattern resembles aka honey comb structure, giving a compromise between the patterns; further work is necessary to determine the optimal sizing parameters of this structure.

The service was assessed by end users in a series of workshops. Users strongly supported the interface and contributed helpful suggestions for improvement of interaction and handling. One of the consequences was a clear separation of contents and representation using XML, which greatly enhances maintainability. In the current version 3 these have been incorporated.

The service is running under SUN Solaris with Oracle, RasDaMan, and Java WMS/WCS servlets. Interactive browser access is through http://eoweb.dlr.de/data-service/, for the WMS and WCS URLs please contact the authors.

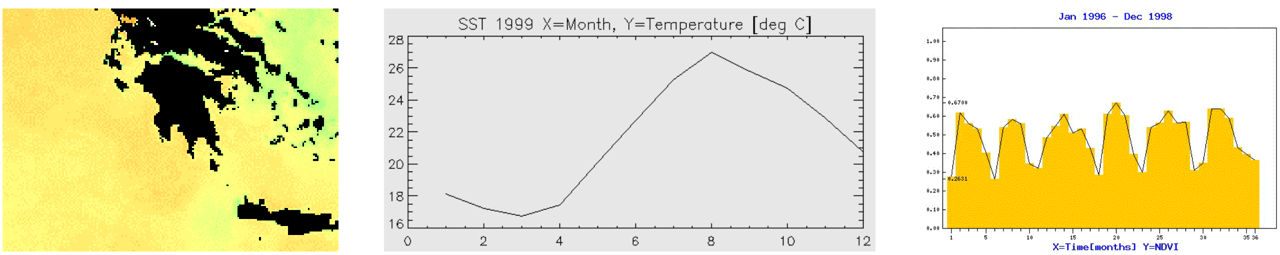

Fig. 4: 2-D and 1-D extraction from the AVHRR time series database (left to right): SST around Greece at some time; SST at some point in the Mediterranean; NDVI over Nile delta

THE 4-D CASE:
THE CLIMATE MODEL DATABASE
AT MAX-PLANCK-INSTITUTE FOR METEOROLOGY

Max-Planck-Institute for Meteorology in Hamburg/Germany serves as Data Distribution Centre (DDC) for the results of climate calculations according to the scenarios of the Intergovernmental Panel for Climate Change (IPCC). Furthermore, it distributes results of its own model calculations.

Climate model calculations are carried out as model run experiments. While these experiments simulate, e.g., two or three centuries of world time, they may well run some months of real time in the computer. During this runtime, the model can eject at an 6 hours interval its current state, i.e., the values of many physical variables at any grid point at the earth's surface. To every experiment, a set of parameters (=datasets) is affiliated. They represent the (up to about 50) physical values modelled in the calculations, like precipitation, wind speed components, and humidity.

In the course of the ESTEDI project, a database has been set up with monthly climate model data taken from the scenario calculations as well as from a control run that assumes constant (1990) values of greenhouse gases. The scenarios differ in the assumptions about CO2 and other greenhouse gases, sulphuric emissions, and ozone models. More recently, results of a coupled atmosphere/ocean model run have been added.

Most experiment data span over 240 years of T42 model output (i.e., longitude x latitude = 128 x 64 grid points) in about 8 physical parameters for either only the surface or 15 altitude levels. This amounts to a data volume of 128 x 64 x 2880 x 4 Byte = 90 MB (x 15 altitudes = 1.4 GB) for each parameter, having four-byte floating point numbers per data point.

Access to these 4-D cubes is through a Java interface which allows for metadata navigation and spatial subsetting (Fig. 5, 6, 7).

The climate database, which runs on SUN Solaris with Oracle and RasDaMan, is accessible by the public (free of cost) under the URL http://mad.dkrz.de/java/cera2browser_rasda/ .

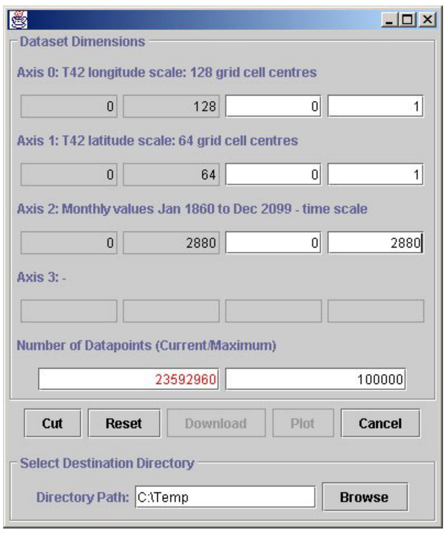

Fig. 5: spatial cutout selection window

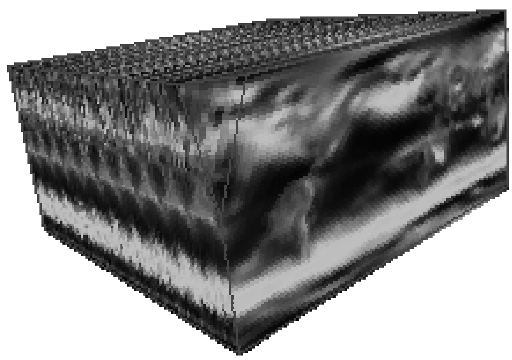

Fig. 6: 3-D cutout from 4-D climate model (projection to highest altitude layer, showing the jet streams; x/y axis top/right, time axis to the left)

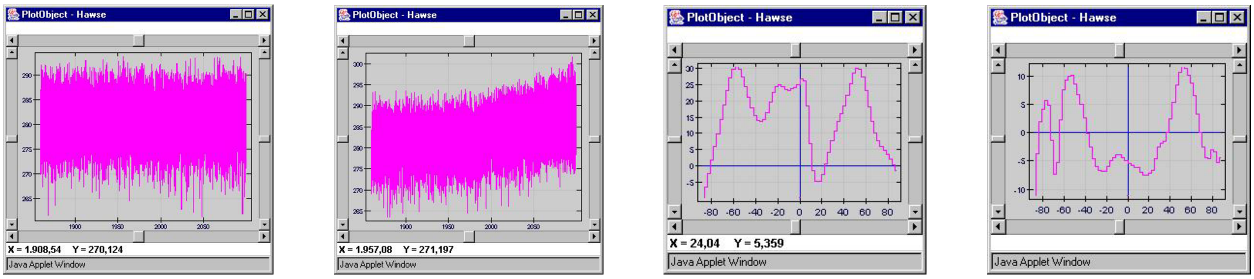

Fig. 7: 1-D time extracts from 4-D climate simulation data.
2m temperature in the control run - no change of greenhouse gases (far left) and with an increase of greenhouse gases (inner left); ground winds with westerlies and trade winds (inner right); jet streams appearing at higher altitudes (far right; note the different velocity scale)

STATE OF THE ART

Traditionally, multidimensional raster data have been stored in sets of files. As the file system has no idea about the semantics (pixel type, number of dimensions, etc.), all selection and processing is burdened to the application developer, leading to tedious, repetitive work and an ill-defined consistency state of the data. Moreover, file-based storage tends to favour particular access patterns (e.g., x/y selection on time series) while conveying disastrous performance on all others (e.g., z/t selection).

For fast zoom and pan on mosaicked image file sets, many products are available. Access is done through low-level API libraries instead of high-level, model-based query support with internal optimisation and without flexible image extraction functionality, such as hyperspectral channel extraction, overlaying, and *ad hoc* thematic colouring. Most important, file-based solutions *per se* do not scale very well. Optimisations done to speed up performance consist in adopting, not to say: re-inventing, one or the other of the set of techniques known in the database community since long – for example, spatial indexing, load balancing, preaggregation, and materialized views.

Image processing packages are widely used in remote sensing. They offer advanced image processing algorithms, but lack general data management capabilities with a quality of service comparable to database systems.

Relational DBMSs, designed to scale well indeed, traditionally store multidimensional arrays as unstructured BLOBs ("binary large objects") introduced by Lorie as "long fields" (Lorie 1982). This technique cannot give any support for operations beyond line-by-line access, something clearly not feasible for large archives.

Object-relational database systems (ORDBMSs) allow to add new data types, including access operations, to the server (Stonebraker 1998); examples for such a data type is Oracle Intermedia. Arrays, however, are not a data *type*, but a data type *constructor* ("template"), parametrized with cell type and dimension – see Section 3.1. Such templates are not supported by ORDBMSs, hence a separate data type has to be defined for 2-D grayscale ortho images, 2-D hyperspectral MODIS images, 4-D climate models, etc. Furthermore, server internal components are not prepared for the kind of operations occurring in MDD applications, therefore important optimization techniques like tile-based pipelining and parallelization of array query trees are difficult to implement. Finally, implementing the whole RasDaMan system as an Oracle cartridge, DB2 extender, etc. would mean a complete loss of portability. Hence, we feel that ORDBMSs currently are not an option in array data management, at least for deployment cycles measured in industrial timescales.

Some research focuses on specific raster database aspects, such as data storage (Sarawagi 1994) and data models (Libkin 1996, Marathe 1999). While there are interesting results, these have not (yet) made their way into operational systems, and they are not supported by an appropriate overall architecture. RasDaMan, conversely, is a complete multidimensional array DBMS product. It is being used in 12 countries over Europe, Russia, and USA for, 2-D ortho image maps, 3-D satellite image time series, 4-D climate simulations, and others.

SUMMARY AND OUTLOOK

Management of geo imagery in databases is an emerging area receiving growing attention. Technology is at a point where databases as a substitute to homegrown file-based image management are considered, albeit sometimes still with doubts among users and content providers. The BLVA case clearly demonstrates feasibility and scalability of large image databases. Actually, we believe that the seamless ortho image of Bavaria is the largest image that has been accessed by query language up to now.

Although RasDaMan is implemented as an additional tier on top of a relational DBMS, performance turned out to be clearly superior to file-based approaches; in fact, a broad range of practically relevant queries on near-Terabyte databases performs in the sub-second area even on PC servers.

Beyond performance, however, we see further advantages in the enhanced functionality provided by the query language approach, and in general by bringing standard database benefits such as multiuser synchronisation, transaction support, and concise, explicit schema modelling to the area of raster data management. Finally, storing geo images in the database together with meta and vector data not only eases administration, but also enhances data consistency considerably.

BLVA is going to establish a digital geo data infrastructure where the RasDaMan database will form a main component at the back-tier for the online retrieval of DOPs.

On the architectural side, next steps include introduction of load balancing techniques to accommodate heavily parallel access and failover hardware/software redundancy towards 7x24 service availability.

Current research work of FORWISS on RasDaMan encompasses further work on parallel query evaluation, advanced optimisation for complex statistical queries, and tape silo support.

ACKNOWLEDGEMENTS

The authors acknowledge and highly value the work of the outstanding RasDaMan team. Bruno Roder from BLVA provided valuable and supportive counterpart in all discussions. Bernhard Buckl, Peter Seifert, and Heike Rausch have implemented the DLR satellite image time series database. The climate database GUI was done by Michael Kurtz.

REFERENCES

Burt, P.J., Adelson, E.H., 1983, The Laplacian pyramid as a compact code
IEEE Transaction on Communications, vol. COM-31, April 1983, pp. 532-540

Baumann, P., A Database Array Algebra for Spatio-Temporal Data and Beyond
Proc. Next Generation IT and Systems (NGITS), Zikhron Yaakov, Israel, 1999, pp. 76 - 93

Cattell, R.G.G., The Object Database Standard: ODMG 2.0
Morgan Kaufmann Publishers, California, 1997

Furtado, P. , Baumann, P., Storage of Multidimensional Arrays Based on Arbitrary Tiling
Proc. ICDE, Sydney, Australia, 1999, pp. 480-489

Glock, C., Hettmer, W., Web-based management of raster data at the Bavarian Land Survey Authority (in German). In: Schilcher, Matthäus (ed.): 7. Münchner Fortbildungsseminar Geoinformationssysteme, Dept. of Geographic Information Systems, Technische Universität München, Munich, 2002

Hahn, K., et al, Parallel Query Support for Multidimensional Data: Inter-object Parallelism
Proc. DEXA, Aix-en-Provence, France, 2002.

Libkin, L., Machlin, R., Wong, L., A Query Language for Multidimensional Arrays: Design, Implementation, and Optimization Techniques, Proc. ACM SIGMOD, Montreal, Canada, 1996, pp. 228 – 239

Lorie, R.A., Issues in Databases for Design Transactions. in: Encarnaçao, J.; Krause, F.L. (eds.): File Structures and Databases for CAD, North-Holland Publishing, 1982

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

Reiner, B., et al, Hierarchical Storage Support and Management for Large-Scale Multidimensional Array Database Management Systems. Proc. DEXA, Aix en Provence, France, 2002.

Ritsch, R., Optimization and Evaluation of Array Queries in Database Management Systems.
PhD Thesis, Technische Universität München, 1999.

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

Stonebraker, M., Moore, D., Brown, P., Object-Relational DBMSs: Tracking the Next Great Wave (2nd edition). Morgan Kaufmann, September 1998

Widmann, N., Efficient Operation Execution on Multidimensional Array Data.
PhD Thesis, Technische Universität München, 2000