Web-enabled Raster GIS Services for Large Image and Map Databases

Peter Baumann
Active Knowledge GmbH
Kirchenstr. 88, D-81675 Munich, Germany
Dial-up: voice +49-89-458677-30, fax -39
E-mail: peter.baumann@active-knowledge.com

Abstract

While there is a rapidly increasing demand for flexible, ubiquitous access to ever-growing archives of raster images and maps of various types, data management technology lags behind, as raster structures (i.e., arrays) for a long time have been neglected in database research. File-based solutions prevail in products, and in research only prototypes with relatively narrow functionality have been tested.

The RasGEO database application provides interactive navigation on unbounded continuous raster maps as well as ad-hoc generation of compound map products. RasGEO integrates all kind of rasterized map data, up to user-defined hyperspectral imagery and DEMs. The tool is implemented as a servlet-based Web application on top of the RasDaMan multidimensional DBMS. RasGEO user input is transformed into a query sent to the DBMS server. The server's response is a JPEG image which RasGEO forwards to the client. Any Web browser can be used, as the front end is purely HTML based.

RasDaMan and RasGEO are commercial products operational at many sites, deployed with continuous aerial images in excess of .5 TB.

1 Introduction

More and more remote sensing data is continuously being generated by an ever-increasing number of sensors and platforms. Recent storage technology make it feasible to collect and keep these huge amounts of data over a long period. 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.

Retrieval technology for large raster repositories, however, is lagging behind. Commonly storage is done in a one-file-per-image manner, driven by the data acquisition process rather than by user access patterns. Consequently, retrieving the data needed for processing and analysis involves a lot of time and computing resources. This holds even for the extraction of a rectangular area with arbitrary bounding coordinates, maybe the most basic task of a raster map database. 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 has a distinct pixel resolution.

Ideally, one would expect to formulate raster image requests as queries which are answered by the database management systems (DBMS) by returning the desired result image. Traditional DBMSs cannot accomplish this, because multidimensional arrays or *multidimensional discrete data* (MDD) obviously form a separate, fundamentally distinct information category aside from sets, object nets, and text. MDD support, consequently, requires new techniques in all aspects of database architecture, such as modelling and query support, optimisation, storage management, and – last but not least – configurable front-end tools.

In this paper, we present the RasGEO database application for interactive navigation on unbounded continuous raster maps and ad-hoc generation of compound map products. This Web-based tool handles all kind of rasterized map data, such as topographical maps, aerial photos, satellite images (up to user-defined hyperspectral), and DEMs (both integer and float pixels). Data are stored in the RasDaMan[1] multidimensional DBMS [1,2,4,9,10]. Application fields are manifold, from geo e-commerce to disaster management.

The remainder of this paper is organized as follows. In Section 2, we review the state of the art. In Section 3,

[1] See www.rasdaman.com

we give a brief summary of the RasDaMan model, query language, and architecture in Section 3. Section 4 describes RasGEO and discusses the query techniques applied. Section 5 concludes the paper.

2 State of the Art

Image processing systems, such as ERDAS Imagine [3], are widely used in remote sensing. They offer advanced image processing algorithms, but lack data management capabilities such as sub-image extraction from a several hundred GB continuous image.

For fast zoom and pan on mosaicked image file sets, many products are available, e.g., [7]. 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 coloring. Most important, file-based solutions do not scale well.

Relational DBMSs, designed to scale well indeed, traditionally store multidimensional arrays as unstructured BLOBs ("binary large objects") which originally have been introduced by Lorie as "long fields" [6]. 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 [12]. Arrays, however, are not a data type, but a data type constructor ("template"), parametrized with cell type and dimension. 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, hence important optimization techniques like pipelining and parallelization of array query trees cannot be exploited. Finally, implementing the whole RasDaMan system as an Informix datablade or DB2 extender would mean a complete loss of portability. Hence, we feel that ORDBMSs currently are not an option in array data management, at least for short- and medium-term industrial deployment.

Some recent research focuses on aspects of database support tailored towards large, such as multidimensional data storage [11] or data models [5,8]. While there are interesting results, these have not (yet) made their way into operational systems. RasDaMan, conversely, is a complete multidimensional array DBMS product. It is operational at many sites and has been applied, e.g., to 2-D ortho image maps, 3-D satellite image time series, and 4-D climate simulations.

3 The RasDaMan Array DBMS

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.

The RasDaMan query language, RasQL, offers an algebra-based query language [1,2] which extends standard SQL92 with declarative MDD operators. The expressiveness of RasQL enables a wide range of signal processing, imaging, and statistical operations up to, e.g., the Fourier Transform. Server-based query evaluation relies on algebraic optimisation [9] and a specialised array storage manager [10].

3.1 Array Definition

Arrays can be built upon any valid C/C++ type, be it atomic or composed ("struct"), based on ODMG's 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 ortho image can be defined by

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

3.2 Array Retrieval

The RasQL query language is based on standard SQL92. Like SQL, a query returns a set of items (in this case MDDs). For choosing elements from the collection as well as for tailoring each MDD element from the result set, multidimensional operators are available.

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

Example 2: "*Map bit layer 3 over (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 result type is Boolean, also in the `WHERE` part. See [1] for further query constructs such as condensers (the MDD counterpart to aggregates), etc.

3.3 Physical Array Storage

RasDaMan storage is based on the partitioning of an MDD object into *tiles*, i.e., sub-arrays [4]. Aside from regular grids, any user or system generated partitioning is supported (Fig. 1). A geo index (currently: R-tree) is employed to quickly determine the tiles affected by a query. Optionally tiles are compressed using one of various techniques; moreover, result data can be compressed for transfer to the client. Both tiling strategy and compression comprise database tuning parameters.

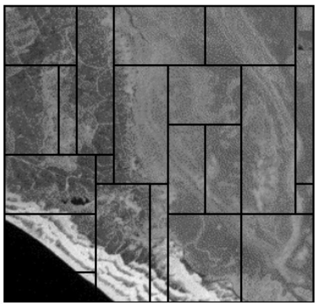

Fig. 1: arbitrary 2-D tiling

Tiles and index are stored as BLOBs in a relational database, which also holds the data dictionnary required by RasDaMan's dynamic type system. Storing MDD in the database instead of separate files does only eases administration, but also enhances data consistency considerably. Adaptors are available for Oracle and DB2, and further DBMSs will be added step by step. A coupling with object-oriented O2 has been done earlier, showing the wide range of DBMSs with which RasDaMan can interoperate.

A series of optimisation rules is applied to a query prior to its execution to achieve an optimal access and processing pattern [9]. Of the 150 heuristic rewriting rules, 110 are actually optimising while the other 40 serve to transform the query into canonical form.

4 RasGEO

RasGEO is a RasDaMan application for interactive navigation through large, heterogeneous raster maps. Application fields are in-house information services (e.g., networks planning), external information services (e.g., public maps, flood information), and, particularly, e-commerce with value-added geo products.

The user interacts with the database through Web browser windows for geo coordinate and map layer selection, zoom, pan, and layer coloring. RasGEO user input is transformed into a query sent to the DBMS server. The server's response is a JPEG image which RasGEO forwards to the client.

The central data structure is an image stack which the user builds up by selecting layers from the map types on hand. Stack sequence can be modified, and each layer can be individually colored where applicable. Geographic selection is done by zoom and pan or through direct input of geo coordinates. Visualization of the resulting image in a moderate size (e.g., 300 x 300) helps to adjust the desired product configuration. Once this has been determined, an ordering component allows to obtain a high-resolution image by adding resolution information, the desired data format, etc. This – usually large – image then is generated by a batch job and sent to the client by whatever means is appropriate (e-mail, ftp download, CD-ROM, etc.).

RasGEO is implemented as a set of Java servlets using the RasDaMan Java API, RasJ.

RasGEO itself concentrates on user interaction, all geo image processing is done by RasDaMan. In a small meta data table, RasGEO maintains information about the maps available, including their characteristics such as map name, geometric resolution, and image pyramid parameters. Based on this information, RasGEO generates a RasQL query from the user input. The RasDaMan server delivers the completely processed map as a single JPEG image which RasGEO passes on to the Web browser for immediate display. As the Web client receives pure HTML plus JPEG images, no plug-in is required.

The following examples show some query types used by RasGEO. To enhance legibility, symbolic constants have been used for values actually substituted from the meta data; hence, actual queries are a little bit more involved.

Query 1: "*From the grayscale ortho image, the same cut-out scaled to window size":*

```
SELECT jpeg(
          scale( Ortho[x0:x1,y0:y1], f )
          *
          {1c,1c,1c}
       )
FROM   Ortho
```

The "`c`" tag following the numeric constants enforces a `char` (i.e., 8-bit unsigned integer) pixel type.

Note that the induced RGB multiplication to transform the grayscale image to colour is quite expensive. Yet it is necessary for overlaying with coloured maps. Query response time obviously will become better once colour ortho photos will be used. We even have thought of (but not tested yet) storing gray-scale images as colour images redundantly, exploiting the server-builtin compression techniques.

Query 2: "*From the topographical map, the streets layer in red with background black; cut-out area (x0,y0) to (x1,y1), scaled to window size*":

```
SELECT jpeg(
          scale( bit( Tk10[x0:x1,y0:y1],
                      streets ),
                 f )
          *
         {255c,0c,0c}
       )
FROM   Tk10
```

From the single image stored in the `Tk10` relation, a cut-out is taken, then the `streets` bit layer is extracted, and scaling is applied. The binary result pixels are multiplied with the color value desired. Finally, JPEG is generated. The appropriate scale factor `f` is computed in the RasGEO servlets.

Query 3: "*The DEM, coloured in three steps: 0.0 to 5.0 in red, 5.0 to 100.0 in green, above 100.0 in blue":*

```
SELECT   (Dem <   5.0) * {255c,0c,0c}
       + (Dem >   5.0 AND Dem < 100.0)
                       * {0c,255c,0c}
       + (Dem > 100.0) * {0c,0c,255c}
FROM   Dem
```

Each comparison predicate results in a Boolean value which is multiplied with the color value desired. Obviously, dropping one of the addends suppresses the corresponding height step, thereby introducing transparency. This allows to combine DEMs with other map types, e.g., to display flood areas on top of recent satellite imagery.

The following table gives performance measurements for the above query types. They were taken on a Pentium III based Linux notebook (650 MHz) with 256 MB main memory. It ran an Oracle 8.1.7 server together with the RasDaMan server and the query client, plus the X server. Data sets used were a 36,000 x 36,000 ortho image and a 48,000 x 47,000 topographical map (1:10,000) containing nine thematic layers. The JPEG images generated by the queries varied between 10 kB and 25 kB in size. Ortho image zoom factors were chosen randomly, no difference was observed in response time.

Table 1: Performance measured on example queries.

Query type	Average elapsed time [sec]
Query 1	0.381
Query 2	0.360
Query 3	0.887

5 Conclusion

With the ever-increasing need to maintain growing sets of sensor data in their originally acquired form, geo information providers are faced with severe information management problems.

RasDaMan allows integrated access to multi-resolution heterogeneous map and sensor data, using a clear conceptual model and retrieval language, strictly server-based query evaluation, and extensive query and storage optimisation. Through its query-base retrieval, many value-added services based on the original "raw" data become available to geo content providers. RasGEO, as a RasDaMan application specifically designed for on-the-fly map generation, shows how geo query functionality can be embedded into a user-oriented Web service.

Although a relational DBMS is underlying RasDaMan as an additional system layer, performance turned out to be clearly superior to file-based products available in the market; in fact, a broad range of practically relevant queries performs in the sub-second area on PC servers. Mass tests conducted by Bavarian State Survey with a 600 GB ortho image and by German Aerospace Association (DLR) with a 500 GB satellite image showed scalability of the approach.

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 multi-user synchronisation, transaction support, and schema management to the area of large-scale raster data management.

Future work on RasDaMan encompasses parallel query evaluation, advanced optimisation for complex statistical queries, and tertiary storage support for multi-Terabyte databases. Among topics for RasGEO is vector/raster integration for geo portals. On the application side, WAP and PDA access to nation-spanning ortho image archives is being planned.

Acknowledgement

It is a team of outstanding ingenuity and spirit which has made possible RasDaMan. My special thanks goes to Paul Vacarescu for RasGEO implementation and Norbert Widmann for data loaders and geo referencing algorithms.

Thanks also to Bavaria and Brandenburg State Survey for permission to use their data sets for demonstration.

References

1. P. Baumann: A Database Array Algebra for Spatio-Temporal Data and Beyond. Proc. Next Generation Information Technology and Systems NGITS '99, Zikhron Yaakov, Israel, 1999, pp. 76 - 93.
2. Peter Baumann, Andreas Dehmel, Paula Furtado, Roland Ritsch, Norbert Widmann: Cross-Dimensional Sensor Data Management, 4th Int'l Airborne Remote Sensing Conference and Exhibition / 21st Canadian Symposium on Remote Sensing, Ottawa, Canada, 21-24 June 1999.
3. n.n.: http://www.erdas.com/

4. P. Furtado, P. Baumann: Storage of Multidimensional Arrays Based on Arbitrary Tiling. Proc. ICDE '99, Sydney, Australia 1999, pp. 480-489.
5. 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.
6 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, IFIP, 1982.
7. n.n.: http://www.lizardtech.com/index.html
8. A.P. Marathe, K. Salem: Query Processing Techniques for Arrays. Proc. ACM SIGMOD '99, Philadelphia, USA, 1999, pp. 323-334.
9. R. Ritsch: Optimization and Evaluation of Array Queries in Database Management Systems. PhD Thesis, Technische Universität München, 1999.
10. N. Widmann: Efficient Operation Execution on Multi-dimensional Array Data. PhD Thesis, Technische Universität München, 2000.
11. S. Sarawagi, M. Stonebraker: Efficient Organization of Large Multidimensional Arrays. Proc. ICDE'94, Houston, USA, 1994, pp. 328-336.
12. M. Stonebraker, D. Moore, P. Brown: Object-Relational DBMSs: Tracking the Next Great Wave (2nd edition). Morgan Kaufmann, September 1998

Appendix: Sample Screen Shots from RasGEO/RasDaMan Queries

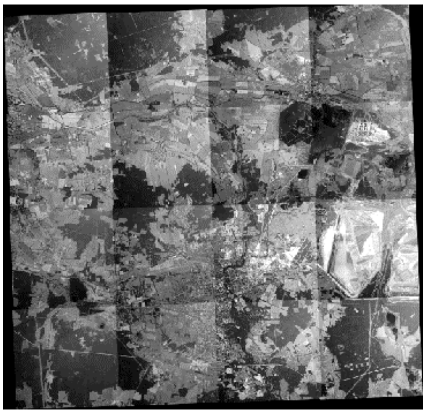

Fig. 2a: sample continuous ortho image (approx. 36,000x36000 pixels)

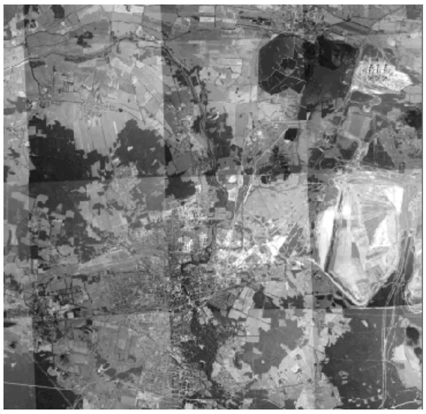

Fig. 2b: zoom into ortho image (Query 1)

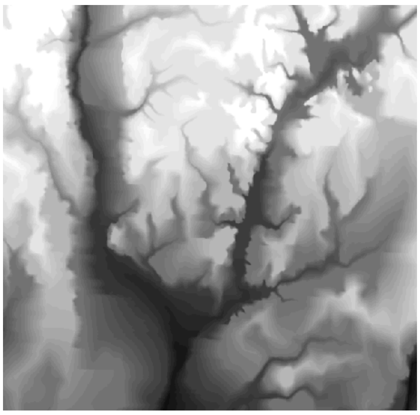

Fig. 3a: Original DEM

Fig. 3b: DEM colouring result obtained through RasGEO Query 3