

GRIDS – AN ARCHITECTURE FOR MANAGING VERY LARGE ORTHOIMAGE MOSAICS IN A DATABASE FRAMEWORK

Stephan NEBIKER

Fachhochschule beider Basel
Abteilung Vermessung und Geoinformation
Muttenz, Switzerland
E-mail: s.nebiker@fhbb.ch

ABSTRACT

This paper presents the investigation and implementation of concepts for the efficient management of very large raster and orthoimage mosaics in a database framework. The paper first identifies and discusses the key requirements of a data management solution for orthoimage mosaics. These requirements include efficient support for spatial partitioning, spatial partition indexing and access, multi-resolution support, georeferencing, storage management and mass storage support. The paper then presents the architecture of the prototype spatial raster management system GrlidS. GrlidS is based on a 'spatial raster middleware' approach which allows typical GIS database engines, e.g. Oracle, to be extended with a spatial raster management capability. Key features of the architecture include a tile-based spatial partitioning solution and a spatial access and query concept using a modified Morton code-based indexing mechanism. The presented multi-resolution concept for raster mosaics represents a novel approach. It uses a virtual resolution pyramid of constant-dimension tiles, combining the resolution access with the spatial access through a single spatial coding scheme. The paper also presents a series of tests and results. These results demonstrate that orthoimages and very large orthoimage mosaics can efficiently be managed with state-of-art relational DBMS technology. The paper concludes with an outlook on important DBMS trends as well as on ongoing and future investigations.

1 INTRODUCTION

Digital orthoimages and orthoimage mosaics are playing an increasingly important role throughout the entire geoinformation domain. Orthoimages offer an ideal combination of high information density, wide-area coverage, economical acquisition and production.

The use of orthoimages, however, is often limited due to the fact that areas of interest only rarely coincide with the spatial extents of individual orthoimages. Orthoimage mosaics efficiently address this problem by combining multiple orthoimages into 'seamless raster databases'. This liberates GIS-users from the demanding mosaicking process. Unfortunately, the use of orthoimage mosaics introduces new problems, namely in terms of storing, accessing and handling such large spatial objects with data volumes in the order of tens to hundreds of gigabytes per mosaic.

Today's data management solutions for orthoimage mosaics offered by photogrammetric, remote sensing or GIS products are typically file-based with considerable limitations in terms of size and functionality. These solutions, for example, offer little to no database support with the consequence that important database services (e.g. query support, concurrency, integrity and access control) are unavailable for this type of data.

The investigations documented in this paper are aiming at providing significant contributions towards a better integration of seamless orthoimage databases into geoinformation systems. This integration will be a key element in establishing orthoimagery as a central element in future geo-information warehouses. This paper presents the investigation and implementation of concepts enabling the efficient management of very large orthoimage mosaics within a database framework. As a

guiding principle, these concepts were to be database-model independent and they were to support an easy integration with state-of-the-art GIS technology.

In Section 2, the paper identifies the key requirements of a data management solution for orthoimage mosaics.

In Section 3, the architecture of the prototype system GrlidS (Geospatial Raster Image Database management System) is presented and discussed.

In Section 4, a series of tests and their results demonstrating the capabilities and limitations of the current prototype system are presented.

Finally, in Sections 5 and 6, conclusions and an outlook are presented.

2 SPATIAL RASTER MANAGEMENT REQUIREMENTS

Spatial raster data, and orthoimage mosaics in particular, can be characterised as '**large to very large objects with spatial characteristics**'. These characteristics heavily influence the design and implementation of an efficient management solution for this type of data.

This section provides an overview and a brief discussion of some of the key requirements:

- **DBMS Integration** – Existing solutions for managing orthoimagery and orthoimage mosaics, on the one hand, are mostly non-DBMS-based. These solutions are limited in terms of concurrent multi-user support, transaction and integrity control as well as information retrieval capability. DBMS-based data management solutions, on the other hand, either support large objects (e.g. video database servers) or spatial objects (e.g. vector-based spatial database extensions). However, there are no established solutions supporting and exploiting both characteristics simul-

taneously. Extending DBMS-support to orthoimagery will be essential in building large orthoimage databases capable of 'scaling to large numbers of users and large volumes of data' (Stonebraker and Moore, 1996).

- **Spatial Partitioning and Partition Indexing** – The size of individual orthoimages and orthoimage mosaics often exceeds the amount of primary memory available. Typically, only parts of these objects are accessed at any given time and operations affecting the entire object are relatively rare. A spatial raster data management solution should therefore support the transparent spatial partitioning and reconstruction of large raster objects. The efficient access to partitions or to individual pixels or cells of large spatial raster objects requires the use of specialised spatial access or indexing methods. Several of the spatial access methods described in Gaede and Günther (1995) are suitable for this type of intra-object or object partition access. An interesting approach combining the aspects of raster partitioning, spatial access and compression on the basis of a modified Hilbert ordering is presented in Pajarola and Widmayer (1996).
- **Spatial Object Access** – A basic requirement in the management of orthoimages and orthoimage mosaics is the support for spatial (inter-) object access. This involves the efficient processing of spatial queries on collections of such raster objects with arbitrary position and extent. For this type of query, raster objects can be substituted by vector objects, i.e. by bounding polygons or minimum bounding rectangles (MBRs). Querying, finding and retrieving such simple spatial objects is a standard feature of spatial database extensions (e.g. ESRI SDE, Oracle SDO) and GISs. A comprehensive overview of spatial access methods is provided in Gaede and Günther (1995).
- **Multi-Resolution Support** – The need for multi-resolution support in orthoimage databases is driven by the following two factors:
 - high cost (retrieval time, resource and CPU utilisation) of operations on full-resolution raster data
 - access restrictions to the full-resolution data (e.g. due to commercial or legal reasons)
 Existing multi-resolution techniques which are applicable to raster data include resolution or image pyramids (Shahin, 1994), interlacing schemes, spatial multi-resolution encoding (Samet, 1989) and simple concepts such as previews or 'thumbnails'. These concepts are well suited for individual raster objects with limited dimensions and size and they are often part of standard raster formats or compression schemes. However, they are not easily applicable to raster mosaics or seamless raster databases.
- **Metadata Support** – Metadata play a key role within the context of raster data management. They represent the primary means for any non-spatial queries. Generic raster attributes include information on radiometric resolution, geometric resolution, object identification, description, raster type, etc. Type-specific attribute information for orthoimagery includes:
 - channel / spectral band information
 - information on series of raster objects

- georeferencing information
- possibly colour palettes

In the light of the ongoing trend towards space-based imagery acquisition on a global scale, the aspect of **georeferencing** is rapidly gaining importance. It provides the basis for relating raster coordinates to real-world coordinates and for enabling the above-mentioned spatial access to individual raster objects. The major objective of a modern georeferencing concept should be its global usability. This requires a comprehensive support for geodetic reference systems and preferably the adherence to an international standard (see Section 0).

- **Raster Modelling and Compression** – An orthoimage management solution requires modelling concepts suitable for representing the actual raster data. Current DBMSs do not offer built-in data types for raster data. However, there are two generic concepts for modelling and storing raster data in a DBMS environment:

Binary Large Objects (BLOBs) offer a very generic representation of raster data in the form of variable-length byte strings. The concept is not bound to a specific database technology and has been supported by commercial DBMS products for several years. Fortunately, these mostly proprietary solutions will be standardised with the introduction of SQL3, where BLOBs will be defined as a new base type (Date and Darwen, 1997). Despite this standardisation effort, BLOBs will remain a data type with limited operations, which are restricted to the insertion, updating, selection and deletion of entire BLOBs or parts thereof.

Abstract Data Types (ADTs) provide a great modelling power and flexibility which makes them particularly suitable for the representation of spatial data and of other complex data such as multimedia objects. With raster data, the main advantage of using ADTs over BLOBs is not so much the superior modelling power but much rather the possibility of supporting user-defined operations.

- **Storage Technology Support** – A storage management concept supporting orthoimagery needs to address the following issues:
 - very large storage capacities versus high I/O performance
 - large raster objects versus small amounts of associated metadata

The first issue requires the combination or integration of a number of technologies:

Mass Storage – Large orthoimage databases require storage capacities in the range of hundreds of gigabytes to terabytes. Today, these capacities can only economically be provided by collections of tertiary storage media such as magnetic tapes and optical disks. Support for mass storage subsystems (e.g. optical or tape jukeboxes) is only gradually being added to commercial DBMS.

High-Performance Storage – Orthoimage databases with simultaneous multi-user access additionally demand data transfer bandwidths exceeding the I/O performance of individual hard disks or any of the mass storage subsystems available today. This requires the support for high-

performance secondary storage subsystems (e.g. RAID).

Storage Management – Large orthoimage databases will require the integration of near-line mass storage and on-line high-performance storage. Hierarchical Storage Management (HSM) concepts provide this type of integration, by transparently migrating data to a desired or pre-scheduled level within the storage hierarchy. Once again, DBMS support for this type of technology is still limited.

The second issue is the misbalance between metadata volume and actual raster data volume. It should be noted that an orthoimage database is predominantly accessed via its attribute or metadata information, i.e. not (yet) via its contents. This attribute information consumes little space and should be clustered for optimal performance. The actual raster data, however, is generally large and variable in size. As a consequence, attribute information should be stored separately on the highest hierarchy level, irrespective of the raster storage solution.

These spatial raster management requirements can be summarised by:

The capability to efficiently handle, store, query and access large numbers of individual spatial raster objects (e.g. satellite or aerial images) and very large raster mosaics in a multi-user environment.

3 THE GrIdS SYSTEM

GrIdS (Geospatial Raster and Image Database management System) is a development platform and prototype system for implementing and testing DBMS-based spatial raster management concepts. The development of GrIdS was initiated at the Institute of Geodesy and Photogrammetry at the ETH Zurich as part of the interdisciplinary 'RasterGIS' project. GrIdS provides the basis of further investigations at the Institute of Surveying and Geoinformation at the Fachhochschule beider Basel.

The GrIdS platform enables the investigation and development of concepts and solutions addressing the requirements postulated above. In this section, the main concepts and features of the system architecture, the functional design and the user interface of GrIdS are presented.

3.1 System Architecture

3.1.1 Overview

GrIdS is based on a 'spatial raster middleware' approach which allows typical GIS database engines, e.g. Oracle, to be extended with a spatial raster management capability.

GrIdS consists of the following three sub-systems (see Figure 1):

- **client sub-system** – spatial raster management application and user interface (primarily written in Tcl/Tk)
- **middleware sub-system** – spatial raster middleware (written in C and Tcl/Tk)
- **server sub-system** – (object-) relational DBMS (currently Oracle Universal Server)

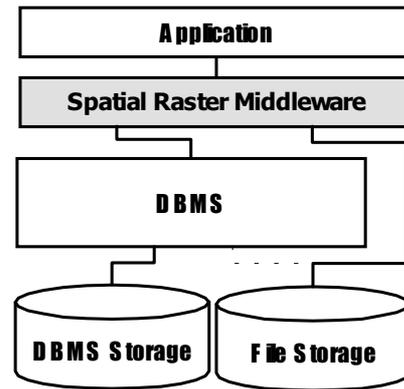


Figure 1 – Middleware-based spatial raster management architecture of GrIdS.

GrIdS supports a distributed environment by allowing each sub-system to be operated on a different node. The middleware - server communication is based on the CLI (Call-Level Interface) mechanism. The CLI utilises Oracle's SQL*Net interface which supports both local and distributed server sub-system configurations. The client sub-system is relatively light-weight. This favours its future implementation as an Intranet / Internet application.

The middleware sub-system received the main focus of our investigations and it incorporates the majority of the concepts presented below.

3.1.2 Tile-based Partitioning

GrIdS incorporates a spatial partitioning concept based on regular, square tiles. This concept enables a simple and unambiguous decomposition of the raster space. It also supports the use of simple and efficient spatial access methods (see below) and it allows efficient and proven compression methods to be applied. A potential drawback of regular tiles with fixed dimensions is the variable size of compressed tiles. This variability in object size could introduce inefficiencies into the secondary storage management.

3.1.3 Tile Indexing based on Morton Ordering

The Morton ordering (Oosterom and Vijlbrief, 1996), (Samet, 1989) in its N-ordering variant has been selected as the spatial indexing and access mechanism for the potentially very large number of mosaic tiles¹. The Morton ordering is ideally suited for indexing regular partitions and it provides an inexpensive code generation and evaluation by bit-interleaving the row and column numbers of a tile. It supports direct spatial access as well as spatial range queries. It also results in the lowest oversearch for range queries in comparison to all other major space orderings (Abel and Mark, 1990). The linear Morton codes can be stored and indexed in any DBMS by using standard data types and traditional built-in access methods (e.g. b-tree). A standard 32-bit Morton coding scheme is capable of indexing raster mosaics holding 2'500 terapixels² or more – depending on the selected tile

¹ 1-meter resolution raster mosaic for area of Switzerland (40'000km²) with 512x512-pixel tiles • 150'000 tiles

² Max. mosaic size with 32-bit Morton codes (4 Giga codes) and 768x768 pixel tiles = 2.5 * 10¹⁵ pixels

dimensions (Nebiker, 1997). This exceeds the size of any raster mosaic conceivable today.

3.1.4 Partition Queries and Query Optimisation

Orthoimage mosaics are typically accessed by means of spatial range queries which are defined by rectangular search windows. These range queries are used to fetch candidate tiles which are then subjected to further tests. The Morton ordering has the property that any tiles intersecting the search window also lie within the code range which is bounded by the codes of the tiles holding the upper left corner (minimum code) and the lower right corner (maximum code) (see Figure 2). Depending on the location and extents of the query window, this simple range query concept can result in a considerable oversearch (Figure 2, left). An effective query optimisation procedure evaluates and splits the search window along major sub-quadrant boundaries (Abel and Smith, 1984). It then substitutes the code range of the original query by multiple ranges. This approach was incorporated in GrdS and it is shown in Nebiker (1997) that this optimisation step dramatically reduces the number of excess candidate objects returned by the region query (Figure 2, right).

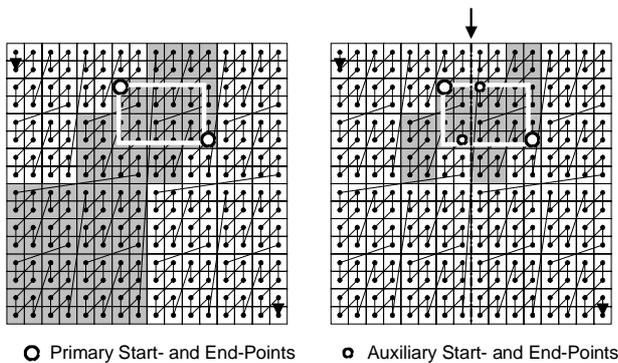


Figure 2 – Spatial range query (white search window): without optimisation (left), with optimisation (right)

3.1.5 Multi-Resolution Concept for Mosaics

The GrdS architecture supports a novel multi-resolution concept for raster mosaics which combines the resolution access with the spatial access through a single spatial coding scheme. The concept is based on a virtual resolution pyramid with constant-dimension tiles. The resolution access has been integrated with the Morton-based spatial tile indexing scheme described above. The main features of this solution are discussed below. A full description of the concept can be found in Nebiker (1997).

Resolution Pyramid based on Constant-Dimension Tiles

– In traditional resolution pyramids for individual raster objects, each level represents the same 'scene' or spatial extent. This has the desired effect that size and retrieval time of reduced resolution 'images' decrease roughly by the square of the reduction or scale factor. With raster mosaics, however, a typical operation consists of zooming a constant-dimension window in and out of the database. Applying the traditional resolution pyramid solution results in the retrieval of a roughly constant volume of data at each resolution level. At the same time, the number of objects (i.e. reduced resolution tiles) to be

retrieved increases by the square of the reduction factor. For example, at a reduction factor of 8 the number of objects to be retrieved is multiplied by a factor of 64 ! This example demonstrates that a multi-resolution concept, supporting efficient windowing operations on large orthoimage mosaics, should keep the amount of data AND the number of objects roughly constant throughout the entire scale range. A resolution pyramid based on constant-dimension tiles (Figure 3) fulfils both requirements.

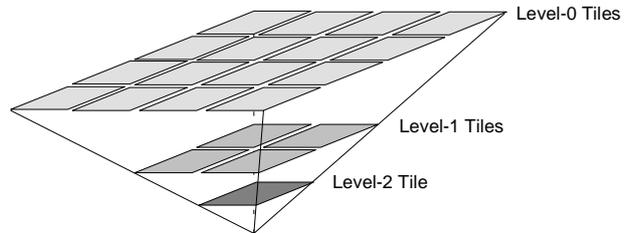


Figure 3 – Part of resolution pyramid for orthoimage mosaics with constant-dimension tiles.

Multi-Resolution Tile Indexing and Access

– The investigations of an efficient indexing and access mechanism for multi-resolution mosaic data resulted in a solution which can easily be integrated with the tile indexing method discussed above. The concept makes use of the duality between the Morton ordering and the region quadtree (Oosterom and Vijlbrief, 1996), (Samet, 1989). Full-resolution tiles can be considered as leaf nodes of the region quadtree. Reduced-resolution tiles correspond to higher-level nodes within the quadtree structure, each occupying the spatial extents of four child nodes. These properties allow the spatial and the resolution access to be integrated.

The reduced-resolution tiles intersecting a query region are determined as follows:

- determination of the full-resolution Morton code range by means of a spatial range query
- derivation of the reduced-resolution tile indices using the following algorithm:

$$SC_{(level\ n)} = SC_{(level\ 0)} - \text{mod}(SC_{(level\ 0)}, 2^{2n})$$

(level 0 = full resolution, n = reduction level)

In a database environment, this algorithm can be implemented on the DBMS server using Persistently Stored Modules (PSM). Additionally, the algorithm can be combined with a 'SELECT DISTINCT' statement which filters duplicate entries from the list of candidate tiles. These solutions avoid sending the full list of candidate tiles to the client application.

3.1.6 Georeferencing Concept

The requirement for a standardised and globally applicable georeferencing concept was accommodated by adapting the georeferencing mechanism incorporated in the GeoTIFF format (Ritter and Ruth, 1995). This mechanism makes use of the POSC / EPSG (Petrotechnical Open Software Corporation / European Petroleum Survey Group) georeference model and database. This is probably the most comprehensive publicly available list and it provides information on almost any geodetic datum, ellipsoid and map projection known in the world. The POSC/EPSC standard can easily be

(equivalent to an area of 50'331 km x 50'331 km at 1 meter pixel resolution)

implemented in a DBMS environment and has recently been adapted by the Open Geodata Committee (OGC) as part of their 'OpenGIS Simple Features Specification for SQL' (OGC, 1997).

3.1.7 Raster Modelling and Compression

GrIdS currently utilises a BLOB-based raster modelling concept, whereby BLOBs can transparently be stored in DBMS-internal storage or in DBMS-external storage. In orthoimage mosaics, each BLOB corresponds to a mosaic tile. GrIdS incorporates a modular compression / decompression (CODEC) support which utilises the standard compression schemes of the TIFF format for the DBMS-internal (Figure 4, left) and -external storage (Figure 4, right). Currently, external file storage is performed under middleware-control (solid line in Figure 1). The LOB/BFILE mechanism now provided by Oracle8 will allow the DBMS-controlled external file storage option to be incorporated in GrIdS (dotted line in Figure 1).

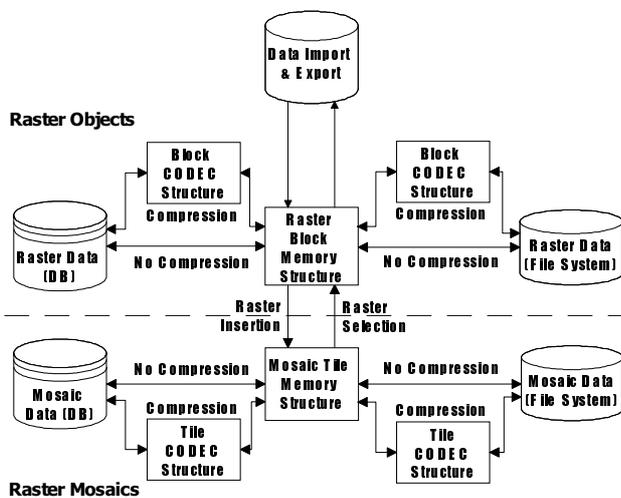


Figure 4 – Data flow diagram for individual raster objects (upper section) and raster mosaics (lower section).

3.1.8 Mass Storage Support

It was pointed out in Section 2 that (tertiary) mass storage support is only gradually being added to commercial DBMSs. For this reason, a middleware-controlled DBMS-external storage option was incorporated in GrIdS. This option uses storage management services offered by the operating system. This approach allows advanced storage technologies such as library systems and hierarchical storage management solutions to be incorporated and exploited almost immediately. The shortcomings of this 'file-based' solution, such as lacking support for transaction, integrity and security control were discussed earlier. Once DBMS-support for the specific storage technology becomes available, the modular storage concept of GrIdS permits a simple migration from middleware- to DBMS-controlled storage.

3.2 Functional Design

GrIdS incorporates modules for handling individual raster objects (e.g. orthoimages, raster maps, thematic raster objects), raster mosaics (e.g. orthoimage mosaics or raster map mosaics), series of raster objects and a number of utilities, such as an interactive SQL database

interface. For individual raster objects, the supported functionality includes import, georeferencing and export as well as comprehensive metadata management, such as the assignment of layer or series information. The raster mosaic management functionality includes the insertion and retrieval of raster objects or 'windows' of arbitrary dimension and size. It also supports the management and fusion of multiple mosaic layers.

3.2.1 Window Retrieval Operation

The window retrieval operation is the most frequently used operation on raster mosaics. It is subsequently used to illustrate the operational principle of GrIdS and consists of the following steps:

Tile Identification – In the first step, the minimum and maximum Morton codes are determined in middleware. Following an optional optimisation step (see Section 3.1.3), a standard query with one or multiple code-range predicates is used to retrieve a list of candidate tile headers from the server. This list is further evaluated to reject any remaining tiles outside the query window (see Figure 2).

Tile Retrieval and Data Projection – In the second step, the data of the requested tiles is retrieved from the server and decompressed in middleware (Figure 5). The tile data is subsequently mapped to the structure of the export raster object (Figure 4). This mapping operation involves geometric and radiometric transformations, such as clipping and an optional layer fusion. In order to support the export of arbitrary size raster objects, the export is performed on a sequential 'strip-by-strip' basis, whereby each completed strip is compressed and flushed to file or transmitted over the network.

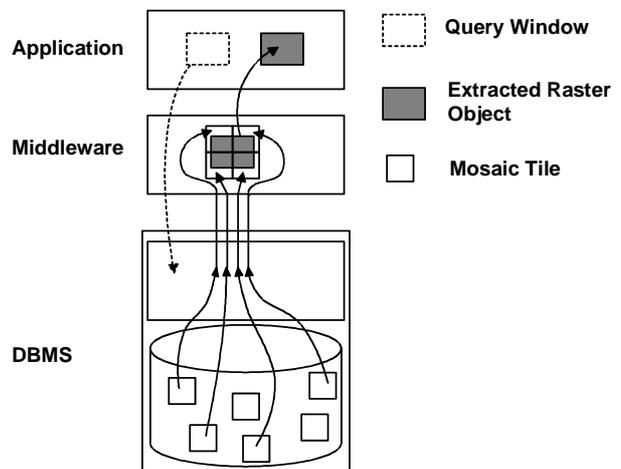


Figure 5 – Mosaic window retrieval – middleware solution (left) and server extension (right)

3.3 User-Interface

The user-interface of GrIdS was developed using the script-based Tcl/Tk programming language. Figure 6 illustrates an example of the screen interface of the full-functionality GrIdS version. This version also incorporates the system management functionality. It is foreseen to test the suitability of the Tcl/Tk web-browser plugin for developing a light-weight web-based client-interface with primarily query functionality.

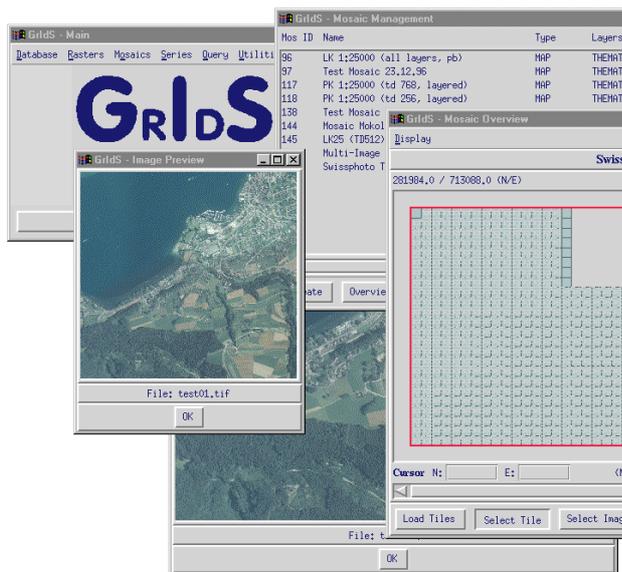


Figure 6 – User interface of GrIdS (only part of screen shown) with an exported orthoimage mosaic selection (centre, left), a list of available mosaics (top, right) and a graphical mosaic overview (centre, right)

4 TESTS AND RESULTS

The prototype system was tested with aerial orthoimages and raster maps. The orthoimage data set consisted of 28 colour orthoimages provided by SwissPhoto Surveys, Regensdorf. The orthoimage dimensions were 4'000 x 4'000 pixels with a 24-bit RGB radiometric resolution. The raster map data set consisted of 16 PixelMaps™ provided by the Swiss Federal Office of Topography. Each PixelMap was provided as 6 separately scanned and geocoded binary layers of 14'000 x 9'600 pixels, resulting in a total of 96 raster objects. From these test data sets a number of mosaics with different tile dimensions and compression schemes were generated. The size of individual mosaics was in excess of 1 GB and 20'000 tiles. The test objectives included the evaluation of the implemented concepts and the monitoring of the system performance. Particular attention was paid to the influence of tile dimension, tile size, compression scheme and raster type. A detailed description and discussion can be found in Nebiker (1997).

4.1 Import of Large Raster Objects

In a first test series, the import performance for large objects was measured. This scenario is typical for the mosaic creation process, whereby large amounts of orthorectified imagery or raster map sheets are inserted into the corresponding mosaics.

In these tests, the system demonstrated its capability to handle arbitrary size raster objects. However, the results showed a relatively low I/O performance of approx. 0.5 MB/s. The main limiting factors were identified as the DBMS's rollback overhead and the CPU performance³.

³ Test platform: IBM RS/6000 530 workstation (model 1990) with a CPU performance of approx. 10% of a 1998 high-performance PC workstation.

4.2 Window Retrieval

In a second test scenario, the mosaic window retrieval performance was monitored. This scenario represents the typical application of a seamless orthoimage database. In an operational window retrieval scenario, the window dimensions are guided by the dimensions and the resolution of display or printing devices. The window dimensions tested ranged from 512x512 to 1024x1024 pixels (see Table 1), which are representative for typical display windows in photogrammetric and GIS applications. The measurements included the time required to perform the tile query, to access, retrieve and decompress the tiles, and to compose a single contiguous raster object in main memory. The times measured did not include the subsequent display or export operations.

Mosaic:	RMM 512	RMM 768	RMM 1024	OIM 128	OIM 256	OIM 512
Window Dimens.						
512x512	0.8	0.6	1.0	2.8	3.0	4.3
768x768	1.4	1.1	1.0	7.0	5.5	7.0
1024x1024	1.1	1.0	1.5	9.0	7.8	8.0

Table 1 – Average window retrieval times (in seconds) from raster map mosaics (RMM) and orthoimage mosaics (OIM) with different tile dimensions (e.g. OIM256 = orthoimage mosaic with 256x256 pixel tiles).

The results listed in Table 1 show window retrieval times from raster map mosaics from 0.6 to 1.5 seconds and from orthoimage mosaics in the range of 2.8 to 9.0 seconds. These results compare well with those of earlier investigations on BLOB storage performance (Lamb, 1994).

4.3 Summary

The tests demonstrated that very large orthoimage mosaics can efficiently be managed with state-of-art relational DBMS technology. In summary, the tests yielded the following results:

- the capability to efficiently handle (i.e. store, import and export) large raster objects
- a very good mosaic tile query performance with tile query times consistently below 0.1 s (CPU time) and 0.2 s (elapsed time) (query optimisation enabled)
- a dramatic reduction of the tile oversearch (Figure 2) by the query optimisation step (max. observed reduction factors > 1000 for std. query windows)
- optimal mosaic tile sizes in the range of 100-200 Kbytes (for the tested configuration)
- a very good scalability of the tile query and raster retrieval performance with no evidence of a size-related performance degradation (> 3GB and > 77'000 tiles in a single database table)
- the capability to handle single- and multi-layer mosaics and different compression schemes
- a compression-related I/O performance improvement on platforms with modern CPUs (performance improvement ≈ compression factor)
- a relatively low import performance on the available test platform (due to the inclusion of the raster data in

the transaction and rollback mechanisms of the Oracle DBMS in combination with a low system performance)

- the full availability of DBMS services such as concurrent multi-user access and recovery
- an excellent robustness and reliability of the DBMS-based solution

The tests primarily demonstrated the feasibility of the proposed and implemented solution. Major performance improvements, compared to the results presented above, can be expected on a modern platform. Initial tests on a modern UNIX workstation⁴, for example, indicate that performance improvements by a factor of at least 5 to 10 can be expected.

5 CONCLUSIONS

The presented investigations were motivated by the growing gap between the production rate of spatial raster data and the almost complete absence of suitable management solutions. The primary objectives of these investigations were to identify the key requirements and to develop concepts for the implementation of spatial raster management solutions on the basis of standard, state-of-the-art DBMS technology.

For the management of individual raster objects, a number of proven concepts, e.g. TIFF-based compression schemes and GeoTIFF-based georeferencing, could be used and integrated. For the management of large orthoimage mosaics, however, a number of concepts had to be developed or adapted. The key elements of the chosen mosaic management solution can be summarised as follows:

- regular tile-based raster partitioning
- Morton code-based tile-indexing
- spatial tile query and query optimisation
- tile-based multi-resolution support for raster mosaics
- BLOB-based DBMS-internal and -external raster storage

These concepts were implemented as part of the middleware-based spatial raster management system GrIdS. With this system it could be shown that very large orthoimages and orthoimage mosaics can efficiently be managed in a DBMS framework. It was also shown that such a solution can economically be implemented on standard DBMS and hardware technology. Thus, the concepts presented could relatively easily be integrated with existing middleware-based spatial data management solutions for GIS (e.g. ESRI's SDE or MapInfo's SpatialWare).

The system tests demonstrated a good scalability and performance. The tests also revealed some shortcomings and limitations which have to be attributed to the mandatory use of Oracle's rollback mechanism during data import and to the limited performance of the available test platform.

6 OUTLOOK

Future spatial raster data management solutions will benefit from a number of trends in DBMS technology, on

the one hand, and from the integration with other data types and technologies, on the other.

6.1 DBMS Trends

The following trends in DBMS technology are likely to have a particular impact on raster data management.

Mass Storage Support – The future direct support for Hierarchical Storage Management (HSM) solutions by standard DBMSs will permit the seamless integration of high-capacity tertiary jukebox storage systems with high-performance RAID technology (Yu and DeWitt, 1997). The presented middleware-based approach allows both technologies to be exploited immediately by sacrificing some of the database functionality.

Abstract Data Types (ADTs) for Raster Images – ADTs permit the definition and implementation of (complex) user-defined data types and user-defined operations. They are a fundamental and established concept in object-relational and object-oriented DBMSs. The main advantage of ADTs over BLOBs is not so much the superior modelling power but the possibility of supporting user-defined operations. In the future, these user-defined operations will enable content-based queries on spatial raster databases. First product-specific type extensions for raster images have started to appear on the market. However, most of these image ADTs are designed for individual, relatively small and simple raster objects and they cannot easily be adapted to the requirements of spatial raster data management.

DBMS Server Extensions – Extensible DBMSs rely on the concept of ADTs for the implementation of new data types and specialised operations. This allows operations currently located in middleware to be integrated into the database server. An extension concept performing physical database design solely based on ADT operations has been implemented as part of the CONCERT prototype (Relly et. al., 1997). This externalisation approach of CONCERT represents a new DBMS paradigm which exports DBMS functionality to external data as opposed to the traditional paradigm which is based on importing data into the DBMS. In case of raster data, this new approach will provide full database support, including query capabilities, for data stored outside the database in external storage. Externalisation of DBMS functionality is a relatively new research topic which will gradually influence future versions of commercial DBMSs.

6.2 Ongoing and Future Work

The investigations presented have been carried out as part of the RasterGIS project. This project is a joint effort of the Database group, the GIS group and the Photogrammetry/Remote Sensing group at the ETH Zurich. The research objectives of the project include the investigation of the integrated management of spatial raster and 3D objects in GIS.

Ongoing and future related work at the ETH Zurich include:

- new DBMS concepts supporting spatial raster data (Relly et. al., 1997)
- the modelling and integration of 3D-objects

Ongoing and future investigations at the Fachhochschule beider Basel include:

⁴ Sun Enterprise 450, single UltraSPARC-II processor, 256MB RAM, UltraSCSI Disks

- the closer integration of high-performance, high-capacity storage technologies, in particular hierarchical storage management (HSM)
- the extended integration of photogrammetric and remote sensing support in order to enable the DBMS-based management and exchange of imagery and metadata throughout the entire production process

Acknowledgements

The author would like to thank SwissPhoto Surveys, Regensdorf and the Swiss Federal Office of Topography, Bern for generously providing the real-world test data sets. Thanks are also due to the project partners at the Institute of Geodesy and Photogrammetry and the Institute of Information Systems at the ETH Zurich, namely Lukas Relly, for the very interesting discussions on numerous DBMS design issues.

References

- Abel, D.J. and Smith, J.L., 1984. A data structure and query algorithm for a database of areal entities. *The Australian Computer Journal*, 16(4): 147-154.
- Abel, D.J. and Mark, D.M., 1990. A comparative analysis of some two-dimensional orderings. *International Journal of Geographical Information Systems*, 4(1): 21-23.
- Date, C.J. and Darwen, H., 1997. *A Guide to the SQL Standard - Fourth Edition*. Addison-Wesley Longman, Inc.
- V. Gaede and O. Günther, 1995. *Multidimensional Access Methods*. Humboldt Universität zu Berlin, Wirtschaftswissenschaftliche Fakultät, Institut für Wirtschaftsinformatik, Berlin ISS-16.
- Lamb, P., 1994. Tiling Very Large Rasters, *Advances in GIS Researches, Proceedings of the Sixth International Symposium on Spatial Data Handling*, Edinburgh. Taylor and Francis, London, pp. 449-461.
- S. Nebiker, 1997. *Spatial Raster Data Management for Geo-Information Systems – A Database Perspective*. PhD dissertation. Blue Series Report No. 63, Institute of Geodesy and Photogrammetry, ETH Zürich.
- OGC (Open GIS Consortium, Inc.), 1997. *OpenGIS Simple Features Specifications for SQL*, Revision 1, <http://www.opengis.org>.
- Oosterom, P.V. and Vijlbrief, T., 1996. The Spatial Location Code, *Proceedings 7th International Symposium on Spatial Data Handling*. 1. Faculty of Geodetic Engineering, Delft University of Technology, Delft, pp. 3B.1-3B.17.
- Pajarola, R. and Widmayer, P., 1996. Spatial Indexing into Compressed Raster Images: How to Answer Range Queries Without Decompression, *Proceedings of the 1996 Int. Workshop on Multimedia DBMS*.
- Relly, L., Schek, H.-J., Henricsson, O. and Nebiker, S., 1997. *Physical Database Design for Raster Images in CONCERT*, *Proceedings of the 5th International Symposium on Spatial Databases (SSD'97)*. Springer, Berlin.
- Ritter, N. and Ruth, M., 1995. *GeoTIFF Format Specification Revision 1.0*, Jet Propulsion Laboratory, Cartographic Application Group.
- Samet, H., 1989. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley Publishing Company.

Shahin, F., 1994. *A Hierarchical Approach to Digital Image Compression*, ASPRS/ACSM..

Stonebraker, M. and Moore, D., 1996. *Object-Relational DBMSs - The Next Great Wave*. Morgan Kaufmann Publishers, Inc., San Francisco, California.

Yu, J. and DeWitt, D.J., 1997. *Query Pre-Execution and Batching in Paradise, A Two-Pronged Approach to the Efficient Processing of Queries on Tape-Resident Data Sets*. Department of Computer Sciences, University of Wisconsin, Madison, Wisconsin.