DISSERTATION ETH No. 12374

Spatial Raster Data Management for Geo-Information Systems - A Database Perspective

A dissertation submitted to the

SWISS FEDERAL INSTITUTE OF TECHNOLOGY

for the degree of

Doctor of Technical Sciences

presented by

Stephan Nebiker
Dipl. Verm.-Ing. ETH Zürich

born 08 October, 1962
citizen of Itingen BL and Häfelfingen BL

accepted on the recommendation of

Prof. Dr. Alessandro Carosio, examiner
Prof. Dr.-Ing. Ralf Bill, co-examiner

1997

To Marianne, Rosmarie and Martin

ABSTRACT

In numerous geomatics domains, vast amounts of spatial raster data are being acquired by high-performance scanning devices, digital cameras and high-resolution remote sensors. The efficient storage, handling, querying and retrieval of this data is one of the main challenges facing producers and users of spatial information. Existing raster management solutions are generally restricted to the file-based handling of limited-size raster objects with limited support for spatial aspects. In GIS, spatial raster data, such as raster maps, raster imagery, digital terrain models and thematic raster data are traditionally not integrated within the overall data management concept. Thus, they cannot benefit from standard database services such as querying, integrity and concurrency control, features which are usually available for the other GIS data types.

This dissertation documents the investigations and developed concepts for the management of large spatial raster objects and raster mosaics in a DBMS framework. Furthermore, it discusses the design and implementation of a prototype spatial raster management system, which was developed for the evaluation of these concepts.

In the first part, the theoretical foundation for the subsequent investigations is established. This includes the examination and documentation of the fundamental principles and characteristics of spatial raster data. It also incorporates the evaluation of raster modelling concepts, database models and technologies as well as storage management concepts and mass storage technologies.

In the second and main part, a number of new or modified spatial raster management concepts are described. Most of them are data model-independent and can be implemented on any modern DBMS technology. The first concept is a tile-based spatial partitioning concept for raster mosaics with an indexing scheme based on the Morton ordering. This space ordering scheme provides efficient spatial access to mosaic tiles through linear indexing techniques available on all DBMSs. In combination with the presented query optimisation concept, this indexing scheme also supports efficient spatial range queries with a low oversearch ratio. Secondly, a novel approach is presented with the tile-based multi-resolution concept for raster mosaics. This concept uses a resolution pyramid of constant-dimension tiles covering different spatial extents. With this approach, the number of database objects to be retrieved can be kept roughly constant over the entire zoom range. By combining the resolution access with the spatial access, the concept allows the reduced-resolution tiles to be spatially indexed and accessed via the same indexing scheme. It also allows the resolution access to be implemented on the DBMS server as a relatively simple filtering operation. Thirdly, a multi-level georeferencing concept for raster objects is presented. This concept allows the spatial extents of raster objects to be represented and queried at different levels of approximation. It therefore supports coordinate system-independent global and system-specific local search scenarios.

The design and development of the prototype spatial raster management system GrIdS allowed a number of these concepts to be implemented and evaluated. The system architecture supports DBMS-internal and external raster data storage with a modular compression support mechanism. DBMS-internal storage currently uses the Binary Large Object (BLOB) concept, whereas DBMS-external uses a file-based solution. The prototype system was implemented on the basis of a relational DBMS (Oracle 7.3).

The system tests were primarily focused on the functionality and performance of the mosaic management solution, which incorporated the majority of new concepts. The tests were performed on a series of mosaics which had been generated from raster maps and orthoimages. The tests showed excellent and consistent tile access times. The spatial range query performance was generally very good but occasionally it was affected by an outlier with a large tile oversearch. It was shown that these outliers can be completely eliminated if the proposed query optimisation is applied. Extensive window retrieval tests were performed, which yielded good retrieval times. The import and export tests with large raster objects revealed a relatively moderate system performance. The main limiting factor was identified as the low performance of the available test platform. Typical database features such as concurrent mosaic retrieval and update were successfully verified.

On the whole, the system implementation and tests demonstrated that the developed concepts can be implemented relatively easily and rapidly. They are suitable for building robust, efficient and scaleable raster management solutions on the basis of standard DBMS technology. The concepts could also be integrated with spatial database extensions used in numerous GISs. The shortcomings of the test platform with respect to I/O performance are discussed and recommendations for future system configurations together with high-performance and high-capacity storage solutions are provided.

ZUSAMMENFASSUNG

Mit Hilfe von Scannern, digitalen Kameras und hochauflösenden Satellitenbildsensoren werden in verschiedenen Geomatikbereichen enorme Mengen an raumbezogenen Rasterdaten gewonnen. Sowohl für die Produzenten wie auch die Anwender dieser Daten, stellen deren effiziente Speicherung, Handhabung, Abfrage und Rückgewinnung eine der ganz grossen Herausforderungen dar. Existierende Lösungen zur Verwaltung von Rasterdaten beschränken sich grösstenteils auf die dateibasierte Verwaltung von Rasterdatensätzen begrenzter Grösse und ohne Unterstützung wichtiger räumlicher Aspekte. Im GIS-Bereich wurden raumbezogene Rasterdaten, wie zum Beispiel Rasterkarten, Rasterbilder, digitale Geländemodelle und thematische Rasterdaten bisher kaum in das Hauptdatenverwaltungskonzept einbezogen. Das hat zur Folge, dass die üblichen Datenbankdienste, wie Abfrageunterstützung, Konsistenzprüfung und Mehrfachzugriffsregelung für Rasterdaten nicht zur Verfügung stehen.

In der vorliegenden Arbeit werden Konzepte präsentiert, welche zur Verwaltung von sehr grossen Rasterobjekten und Rastermosaiken in einer Datenbankumgebung entwickelt worden sind. Zudem werden die Entwicklung und Tests eines Prototypsystems zur datenbankgestützten Rasterdatenverwaltung vorgestellt.

Der erste Teil der Arbeit enthält eine umfassende Zusammenstellung der grundlegenden Prinzipien und Eigenschaften von räumlichen Rasterdaten, eine Untersuchung von Konzepten zur Modellierung von Rasterdaten sowie eine Evaluation verschiedener Datenmodelle, Datenbanktechnologien und Speicherverwaltungskonzepte bezüglich deren Eignung für die Verwaltung räumlicher Rasterdaten.

Im Hauptteil werden neu entwickelte oder modifizierte Rasterdatenverwaltungskonzepte beschrieben. Diese sind weitgehend datenmodell-unabhängig und können mit jeder modernen Datenbanktechnologie umgesetzt werden. Das erste Konzept besteht aus einer räumlichen Unterteilung von grossen Rasterobjekten mittels regelmässiger Kacheln in Kombination mit einem räumlichen Zugriffsmechanismus basierend auf der Morton-Codierung. Dieser Mechanismus ermöglicht einen effizienten räumlichen Zugriff mit

den herkömmlichen linearen Datenbankindizierungstechniken. In Kombination mit dem vorgestellten Abfrageoptimierungsverfahren erlaubt die Methode eine effiziente Bearbeitung räumlicher Bereichsabfragen. Als nächstes wird ein neuer Ansatz zur Unterstützung von Mehrfachauflösung in Rastermosaiken beschrieben. Dieser basiert auf einer Bildpyramide mit konstanten Kacheldimensionen und erlaubt, die Anzahl zu transferierender Datenbankobjekte über den ganzen Zoombereich konstant zu halten. Der Zugriff auf die jeweilige Auflösungsstufe erfolgt zusammen mit dem Raumzugriff über den gleichen Indizierungsmechanismus, was eine einfache und effiziente Implementierung auf dem Datenbankserver ermöglicht. Als weiteres wird ein mehrstufiges Georeferenzierungskonzept vorgestellt, mit welchem Lage und Ausdehnung von Rasterobjekten mit unterschiedlicher Präzision und Detailtreue approximiert und abgefragt werden können. Dieses Konzept unterstützt sowohl eine globale als auch eine lokale räumliche Suche nach raumbezogenen Rasterobjekten.

Die Entwicklung des Prototypsystems GrIdS ermöglichte einen Teil der entwickelten Konzepte umzusetzen und in der Praxis zu testen. Die Systemarchitektur von GrIdS unterstützt eine datenbank-interne und -externe Rasterdatenspeicherung und bietet ein modulares Konzept zur Unterstützung verschiedener Kompressionsalgorithmen. In der implementierten Lösung kommen für die datenbankinterne Speicherung Binary Large Objects (BLOBs) zum Einsatz, die externe Speicherung ist dateibasiert. Das Prototypsystem wurde auf der Basis des relationalen Datenbanksystems Oracle 7.3 implementiert.

Die Systemtests konzentrierten sich auf die neuen Aspekte der Mosaikverwaltung. Dazu wurden aus den zu Verfügung stehenden Ortholuftbildern und Rasterkarten verschiedene Mosaiken erzeugt. Die Untersuchungen ergaben ausgezeichnete Ergebnisse für den Zugriff auf einzelne Mosaikkacheln. Die Ergebnisse für räumliche Bereichsabfragen waren im allgemeinen sehr gut, wiesen aber den einen oder andern 'Ausreisser' mit längerer Antwortzeit auf. Es konnte gezeigt werden, dass diese Ausreisser durch das präsentierte Optimierungsverfahrens eliminiert werden. Umfassende Tests zur Extraktion von Mosaikausschnitten ergaben zufriedenstellende bis gute Antwortzeiten. Die Untersuchung der Import- und Exportleistung für grosse Rasterobjekte ergab eine eher mässige Leistung. Es zeigte sich, dass die Hauptursache dafür bei der zur Verfügung stehenden relativ alten Hardwareplattform lag. Die Funktionalität und die Verfügbarkeit von Datenbankdiensten wurde, zum Beispiel mit der gleichzeitigen Veränderung und Abfrage im gleichen Rastermosaik, erfolgreich verifiziert.

Insgesamt zeigten die Untersuchungen, dass die neuen Konzepte relativ einfach und schnell implementiert werden können. Sie eignen sich für die Entwicklung von robusten, effizienten und skalierbaren Rasterdatenverwaltungslösungen auf der Basis von Standard-Datenbanktechnologien. Die Konzepte eignen sich speziell auch für die Integration in räumliche Datenbankerweiterungen, wie sie bereits in vielen GIS zum Einsatz kommen. Die Leistungsmängel der untersuchten Konfiguration sind zusammen mit Empfehlungen für zukünftige Systemkonfigurationen insbesondere bezüglich Einsatz von Hochleistungsspeichertechnologien dokumentiert.

1

INTRODUCTION

1.1 MOTIVATION

Visual perception of graphical information plays an increasingly important role in modern human life. Raster data is ideally suited for capturing, storing and presenting this type of information in analogue and digital form. The raster paradigm has a considerable tradition in information technology. Originally, it was mainly used for the output and presentation of graphical information by means of CRT displays and printers. Today it is encountered in an abundance of hardware and software technologies, ranging from digital cameras, scanners and printers to image processing and visualisation applications.

Raster data also played an important role in early geo-information applications where it was used, for example, for creating thematic maps on matrix printers. Early digital cartographic systems in the seventies, such as the SCITEX system, were raster-based and had their roots in the textile printing industry. In the late seventies and early eighties the trend in geo-information systems (GISs) changed towards vector data. Over the last ten years, raster data has regained some ground and is now growing in importance in the different fields of geomatics (geo-information systems, cartography, photogrammetry, remote sensing, etc.). This recent trend is likely to continue and raster data will play a key role in future geo-information management. This trend has a number of technical and economical reasons.

From a technical perspective, the main contributing factors are recent developments in the fields of sensor and information technology. Scanners, remote sensors, digital cameras and airborne laser scanners, on the one hand, provide cost-effective and powerful data acquisition tools for a wide range of applications. With these sensors enormous amounts of raster data are already being gathered today. With the forthcoming introduction of new high-resolution satellite imaging sensors the amount of data being acquired will continue to soar. Developments in the fields of information, computer and network technology, on the other hand, are gradually providing the basic means to handle and process these large volumes of data. Significant progress relating to the management of raster data has mainly been achieved in the areas of storage media, storage capacity, CPU performance, parallelisation, network technology, high resolution graphics adapters and output devices. In addition, theoretical methods and techniques required for analysing and processing raster data have progressed significantly and have resulted in numerous operational commercial solutions, such as digital photogrammetric workstations and satellite imaging systems.

From an economical perspective, one of the main reasons for the increasing popularity of raster data is the rapid and highly automated acquisition process. It avoids some of the labour-intensive acquisition and structuring tasks associated with vector data. This makes raster data an ideal alternative in cases where the acquisition of the corresponding vector data would either be too slow or too expensive. A unique feature of spatial raster data is the possibility to economically 'capture' large areas in a very short time frame. An often underestimated economical factor which certainly helped to accelerate the use of raster data is standardisation. The standardisation of peripheral devices and computer screens, for example, which allowed specialised hardware to be replaced by inexpensive standard components, was an important factor in making this technology affordable. From a customer point of view, an important argument for using raster data is the corresponding support provided by standard software products, such as word processing or slide presentation applications. These products have opened up a completely new, potentially very large customer base for spatial information, especially if one considers the potential combination with future spatial databases with on-line access via computer networks.

Status and Problems

In the advent of the 'digital photography age', numerous image management solutions are starting to appear, supplementing or replacing existing solutions in the desktop publishing sector. They provide handling features on the basis of image previews or 'thumbnail' images, which allow images to be graphically inspected, sorted, moved or deleted. However, these solutions hardly deserve the name 'management systems' and they usually cannot cope with normal-size spatial raster images, not to mention image mosaics. In addition to these consumer market solutions, specialised and very powerful image management solutions are being developed for medical image archives and databases or as part of document / workflow management systems. These projects are focused on the management of large numbers of relatively small images and, increasingly, on content-based information search and retrieval. Some of the main issues associated with spatial raster data, such as very large image sizes, spatial data access and image mosaics, are usually not addressed.

Spatial information management in GISs was traditionally focused on vector data. In many systems raster data support was only added over the last few years. In addition to these main-stream GISs, a number of specialised raster modules and raster GISs exist. They are primarily focused on processing and analysis aspects and provide only limited raster management capabilities. Overall, existing raster management solutions in GISs can be characterised as: system-specific, file system-based with little or no database support and with a low degree of system integration.

Some of the major problems related to the management of spatial raster data are:

- ***Data Volumes*** of typical individual spatial raster data sets are often very large, exceeding hundreds of Megabytes (see Table 1–1). These sizes by far exceed the amount of physical (and virtual) memory available in today's standard computers. Table 1–1 also shows that series of spatial raster images and future image mosaics are likely to exceed hundreds of GigaBytes. Data volumes of this size can no longer be stored on standard hard disk technology.

Spatial Raster Data Set	approx. Size
Scanned Map Layer (20 lines/mm, 14'000 x 9'600, black&white, bits packed)	16 MB
Scanned Map Sheet (50 lines/mm, 35'000 x 24'000 pixels, 256 colours (8 bits))	800 MB
Orthoimage Mosaic Switzerland (42'000 km^2, 1-metre resolution, 24-bit RGB)	117 GB
Orthoimage Mosaic Germany (360'000 km^2, 1-metre resolution, 24-bit RGB)	1'005 GB

Table 1–1: Data size of typical spatial raster data sets.

- The ***Spatial Characteristics*** of raster data in geo-sciences require concepts for georeferencing, spatial partitioning, spatial indexing and querying. Georeferencing is a key requirement for the management and transfer of spatial raster data which has long been neglected. Efficient spatial indexing and querying are features which have only recently been added to major relational DBMSs, which are the traditional database engines used by most GIS.

- ***Database Support*** for raster data is currently still limited. Most existing DBMSs were developed and optimised for efficiently managing structured data - and not for handling very large (raster) objects. There is a need for solutions combining the flexibility and sophistication of object-oriented concepts with the performance and standard services of relational database systems.

- The ***Exchange of Spatial Raster Data*** is a problem which the ongoing standardisation efforts will hopefully resolve in the near future. Distributed management, dissemination and multiple usage of spatial raster data also depend upon the availability of metadata concepts and standards which support spatial and thematic search and assessment of spatial raster data on an international scale.

Spatial Raster Data and Multimedia Image Data

Spatial raster data and multimedia image data share the same fundamental concepts and are closely related in many respects. Nevertheless, some of the management and storage requirements for typical spatial raster data are significantly different from those for multimedia images. A comparison between spatial raster objects and multimedia image objects is provided in Table 1–2:

Spatial Raster Objects[1]	Multimedia Image Objects
large image size (e.g. scanned aerial images > 100 MB)	relatively small image size (typically < 3 MB)
(mostly) time-independent	time-dependent (video and image sequences)
potentially very large image mosaics	N/A
lossless compression	lossy compression
support for georeferencing concepts	N/A
showing derivatives of reality (landscape, photo, map)	showing fictional reality (animation)

Table 1–2: Typical characteristics of spatial raster objects versus multimedia image objects.

The following investigations focused particularly on the special characteristics of spatial raster data. However, concepts and solutions used in the multimedia sector were incorporated wherever possible and appropriate.

1.2 RESEARCH OBJECTIVES

The overall goal of the investigations was to research and develop concepts for the DBMS-based management of spatial raster data and to verify these concepts by implementing a prototype system. The investigations should provide a significant contribution towards the integration of spatial raster data management in future geo-information systems (GIS).

The investigations were to address the following aspects:

- establishment of a systematical overview of the main concepts and characteristics of spatial raster data
- identification of relevant technologies and standards which could or should be used or supported in management solutions
- investigation of basic concepts for modelling spatial raster data and their applicability to different data models and the corresponding database technologies
- investigation and development of a framework of concepts for the DBMS-based management of spatial raster objects and mosaics of arbitrary dimension and size
- design and implementation of a prototype system on the basis of state-of-the-art commercial DBMS technology
- system evaluation and tests using real-world data

The following aspects which are also of great relevance to the management of spatial raster data were excluded from the investigations: physical database design, communication network technology and performance as well as legal aspects, such as copyright protection issues and supporting technologies (e.g. digital watermarking).

[1] The term 'spatial raster object' will be used throughout the text for individual instances of a generic spatial raster data type.

2

SPATIAL RASTER CONCEPTS

2.1 SPATIAL DATA MODELS AND RASTER CONCEPTS

In the geo-information domain, raster concepts have a much broader scope than in the traditional imaging context. In the following sections the spatial raster concepts are defined and positioned within the framework of spatial data models.

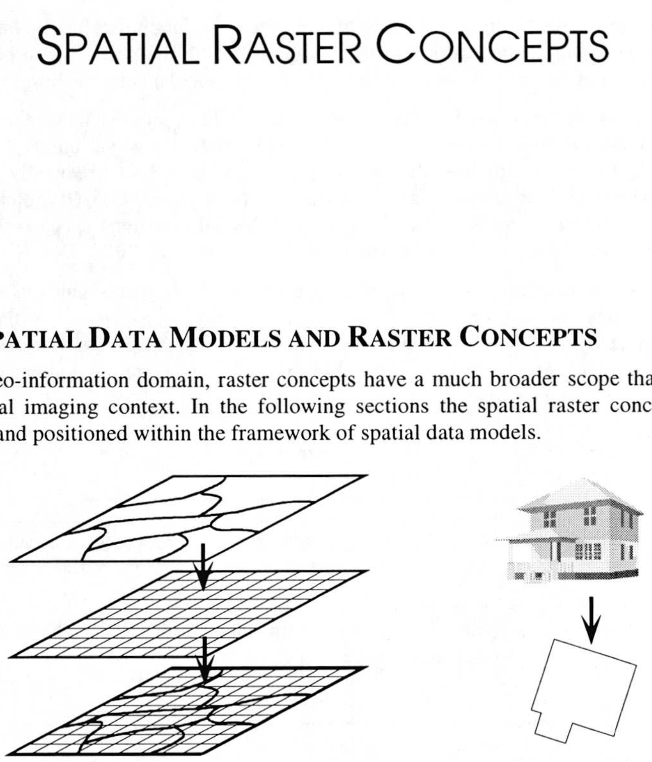


Figure 2–1: Examples of field-based (left) and object-based (right) spatial data models.

Field-Based versus Object-Based Models

Field- (or cell-) based and object-based models are the two main - and opposing - classes of geo-information (see Figure 1-1). In field-based models the space to be represented, e.g. the earth's surface, is considered a spatial or spatio-temporal continuum. In order to represent the continuum in a GIS, it needs to be converted to a discrete form of points or finite cells often in a regular grid or raster format. Attributes are treated as field functions which take a value at any position in a two-dimensional space and are evaluated for each point or cell. Object-based models assume that discrete spatial features can be identified, each having a geometric position and shape as well as several non-geometric characteristics (Molenaar, 1996). These two different models also result in the two opposing GIS implementation approaches, namely raster and vector. More detailed discussions of the two main classes can be found in (Worboys, 1995) and (Molenaar, 1996).

Tessellation and Rasterisation

Tessellation stands for the partitioning of a plane or portion of a surface as the union of a set of non-overlapping areal objects. It comprises both the general case of irregular subdivisions of the space as well as the special case of regular subdivisions.

Irregular tessellations consist of arbitrary 'network-like' subdivisions of an area. A typical representative is the tessellation of irregular triangles which builds the basis of the TIN (Triangulated Irregular Network) model, a model frequently used for representing DTMs (digital terrain models) (Worboys, 1995), (Bartelme, 1995). Another example for an irregular tessellation of the 2-dimensional space is the Region Quadtree (Nievergelt and Hinrichs, 1993), (Samet, 1990).

In regular tessellations a surface is subdivided into a regular grid of uniform elementary objects, such as squares, rectangles, triangles, hexagons which might also be represented by points. This regular tessellation is also called ***rasterisation*** and the resulting regular grid of cells is commonly referred to as ***raster***. Regular tessellations are the basis of the raster data model.

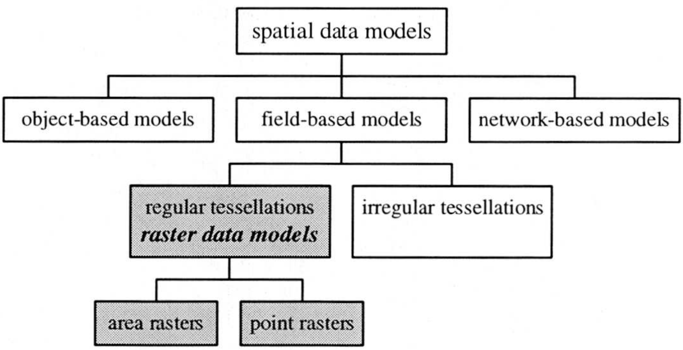


Figure 2–2: Raster concepts in the context of spatial data models.

Raster Data Model

The raster data model represents one of the main models of geo-information. It enables an abstraction of the real-world by modelling spatial variation over a spatial framework

(Worboys, 1995). This spatial framework is established by a regular subdivision of a region of interest into elementary objects referring to one or several attribute domains. The elementary objects of the raster model are called ***grid cells***, ***raster cells*** or ***pixels***. The term pixel originates from image processing and is typically used in a graphical context. However, it is now also frequently used in geo-informatics in a more generalised form (Bartelme, 1995). The raster data model can be characterised by a low level of logical structuring and by weak object links (Bill and Fritsch, 1991).

Area versus Point Rasters

It should be noticed that there exists a dual relationship between points and tessellations in general and between point and area rasters in particular. The elementary areal object of a tessellation can always be represented by the points defining these areal objects. The same applies for the inverse function, if bijective rules for the tessellation can be formulated. This is particularly the case with raster models where the attribute value of a raster element is either applicable to an individual grid position (point raster) or to the entire grid cell defined by that grid position (area raster). The example of digital terrain models where the same data set can be represented as set of spot heights or as an irregular tessellation based on these points (e.g. using a TIN representation) illustrates this dual relationship.

2.2 Spatial Raster Classes

The spatial raster data model as defined above goes beyond the traditional 'raster image' concept used in non-spatial applications. From a data management perspective four main classes of spatial raster data can be identified: raster images, raster maps, (regularly spaced) digital terrain models and thematic rasters. Each of these classes has its specific characteristics and functionality and each class has different data management requirements.

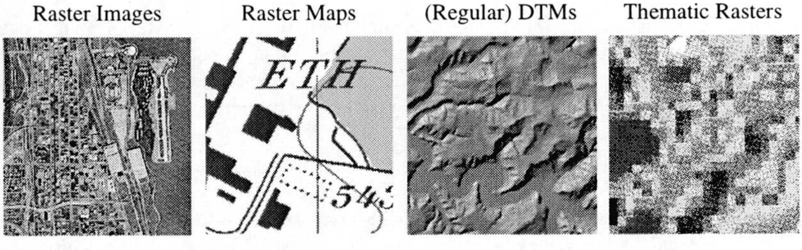


Figure 2–3: Overview of main spatial raster classes.

Raster Images

The raster image class represents the most important type of raster data. Raster images are used in a broad range of applications and can be economically acquired using a variety of sensors (digital cameras, remote sensors, scanners). Raster images are typically photorealistic grey-level or colour images with a ***large range of grey-levels*** or ***colours***. Spatial raster images, in contrast to standard raster images, have special data management requirements with regards to geo-referencing, creating mosaics of overlapping images and supporting metadata.

Raster Maps

The class of raster maps represents a specialisation of the raster image class. It is closely associated with cartography and the related domains of publishing and printing. Raster maps, for example, are obtained by scanning printed maps or original engraving plates. Alternatively, they can be directly generated by cartographic systems either on-demand or as part of a map production process. In that case, the data is often derived by rasterising vector-based or hybrid cartographic information.

In contrast to raster images, raster maps contain relatively ***few discrete colours*** which are typically associated with ***thematic*** or ***colour layers***. These layers are the result of cartographic design and production processes, which are both limited to a certain number of layers for reasons of good perception and economy. Such layer information consists to a good degree of background information (represented by 0 values) and is often stored as bilevel data. The limited number of colours favours the use of raster concepts such as colour maps and run-length encoding. Raster maps also have some special requirements with regards to georeferencing and associated metadata which are needed, for example, for handling map sheets and map series.

Regular Digital Terrain Models (DTMs)

This raster class comprises only a part of all conceivable DTMs. It covers the special case of DTMs represented by a ***regular elevation matrix***. The advantage of using a regular grid representation is the simplicity of modelling, storing and analysis. However, this type of DTM is neither suitable for modelling the landscape at varying degrees of refinement nor can it take into account prominent structures, such as ridges or spot heights. Today, DTM data are typically acquired by means of semi- or fully automated stereo-image correlation.

Many characteristics of DTM data are similar to those of grey-level image data. Depending on the application and the landscape to be modelled a high attribute resolution, real type attribute values or differential storage concepts might be required. DTMs are also often treated as thematic rasters, for example, as valuable additional layers in fields such as geostatistics.

Thematic Rasters

The class of thematic rasters represents field-based spatial information which are not directly related to the visual or radiometric properties of a surface (e.g. colour and intensity in aerial photographs). Another term for thematic raster is 'grid', which is often used in conjunction with thematic point rasters. Attribute values of thematic rasters could be of any data type, e.g. real or boolean. Typical examples of thematic rasters include statistical, census and meteorological data. Thematic raster data is often derived from other spatial data types by means of classification, re-sampling or binning. An important characteristic of the thematic raster class is the presence of ***layers*** and ***overlays***, which allows for relatively simple but powerful analysis.

	Image	Raster Map	DTM	Thematic Raster
no. of attributes (channels)	1, 3 or more	variable	1	variable
attribute resolution	high	low	medium-high	variable
geometric resolution	high	high	low-medium	low
raster type	area	area	point	area or point

Table 2–1: Typical characteristics of spatial raster data classes.

Raster Mosaics - a Special Case

Raster mosaics consist of lateral combinations of multiple raster data sets and are also referred to as ***seamless raster data sets***. Conceptually, raster mosaics can be generated from all the raster classes outlined above. And from an application point-of-view, mosaics of individual raster classes should retain their original characteristics. This means that, wherever possible, the mosaicking aspects should be transparent to the application. However, from a data management perspective, there are some fundamental differences, namely in ***spatial dimensions*** and in ***data volumes***. These differences are in the order of magnitude and make additional and specific data management concepts a necessity. Thus, from the data management point-of-view raster mosaics can be considered as extensions of the basic spatial raster classes. Raster mosaics will be covered in detail in Chapter 3.

2.3 RASTER DATA CHARACTERISTICS

This section provides a summary of the key characteristics of raster data. Most concepts and characteristics are generally applicable to all types of raster data. For some concepts a different terminology is used depending on the class of raster data in question.

2.3.1 Basic Raster Organisation

In all types of rasters, cells or pixels are conceptually organised in matrices with a ***row*** and ***column*** structure. Geometrically, a raster is defined by its origin, by the directions of two perpendicular axes, which define the orientation of rows and columns and by the cell or pixel spacing (Molenaar, 1996). The row and column values of a cell or pixel determine its position relative to the origin and vice versa. The ***height*** of a raster data set expressed in number of pixels is equal to the number of rows and the ***width*** to the number of columns accordingly.

In raster images, the origin is typically located in the upper left corner at row and column position (0,0) - in case for counting conventions. The rows are typically arranged in a top-to-bottom direction, however, bottom-to-top row organisation exists too. In thematic rasters, rows and columns are often aligned with the north-south and east-west axes of map projection systems, with an origin located in the 'lower left' corner at the minimum northing and easting coordinates of the data set.

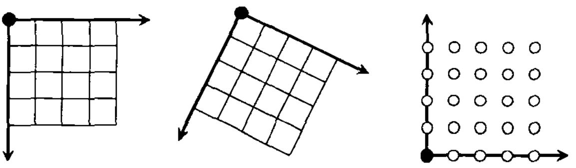


Figure 2–4: Basic organisation of raster data: area raster (left), area raster with arbitrary orientation (middle), point raster (right).

The concepts for encoding and possibly even compressing raster data in digital form are described in 2.4. A refined geometric definition and the concepts of georeferencing and geocoding raster data are presented in 2.5. Advanced raster organisation concepts, such as those used for efficient spatial data access, are also discussed in 2.4.

2.3.2 Geometric Resolution

The geometric or spatial resolution of a raster data set is particularly important in the geo-information domain. Depending on the type of application different definitions and measures for geometric resolution are in use, some addressing it from a more quantitative others from a more qualitative perspective.

In a ***quantitative*** approach the geometric resolution is defined as ***pixel spacing***, i.e. as the separation of neighbouring pixel origins in row and column directions. Pixel spacing can be defined on a physical level in device space and, in the case of spatial raster data, at a virtual level in model space. For numerous applications, such as document image processing, only the geometric resolution at the physical level is of interest. For most spatial raster data applications, however, the resolution at the virtual level is equally important, if not the determining factor. These quantitative definitions of geometric resolution are important factors in the management of raster data and are therefore outlined below in some detail.

A second, more ***qualitative*** approach for defining geometric resolution is commonly used in application areas such as remote sensing and digital photogrammetry. Here, geometric resolution is defined via the minimal object size which an imaging system is still able to detect. It is a measure for the capability of how well an entire physical imaging system can resolve the spatial details of the original scene. It comprises the quality of the imaging system's optics, photosensor and electronics as well as the pixel spacing. This second type of geometric resolution is mainly an image analysis aspect with no implications on the management of raster data. More information on the subject can be found in (Baltsavias, 1991), (Foley et al., 1992), (Baxes, 1994), (Williams, 1995).

Physical Geometric Resolution

Specifying geometric resolution in terms of physical pixel spacing is a very natural concept. It is used wherever pixels or raster cells are physically materialised. Examples for physically materialised pixels are: matrix elements of LCD displays, mask holes in CRT monitors, pigment dots on paper or film created by inkjet or laser printers. In all these cases pixels are represented by a single or multiple ***dots***. The physical geometric resolution is therefore directly related to dot size and spacing.

Geometric resolution is accordingly expressed in terms of dots (dpi = dots per inch) or lines (lines per inch or l/mm = lines per mm) or as actual pixel dimensions (usually in μm). The dot-based dpi measure is frequently used for specifying the supported geometric resolution of printers and scanners. The line-based measures correspond to the closest spacing at which adjacent black and white lines can be distinguished. If 20 black lines interleaved with 20 white lines can be distinguished across one millimetre, the geometric resolution is 40 lines per mm (also referred to as 20 line-pairs per millimetre) (Foley et al., 1992). This measure is frequently used in conjunction with scanners and particularly with scanned raster maps.

Virtual or Object Model-based Geometric Resolution

In many cases it is not practical, or not sufficient, to specify geometric resolution in terms of physical pixel spacing. This applies to typical spatial raster data, such as thematic rasters, DTMs or remote sensing images, where the primary interest focuses on the information density in object model space, e.g. on the surface of the earth. The cells or pixels of thematic rasters, for example, can be considered as virtual, as they are not materialised in metric space - unless, for example, they are sent to an output device. Virtual or object model-based geometric resolution is commonly referred to as 'pixel size' or 'on-the-ground' resolution. It is important to notice, that this type of geometric resolution is often not constant throughout the data set. Due to changes in platform height and orientation and due to the curvature of the earth the on-the-ground resolution of a raw satellite image might vary significantly. (Williams, 1995)

2.3.3 Attribute Count and Resolution

Attribute Count

Rasters are frequently used to represent multiple thematic topics by means of multiple attributes. A general distinction can be made between ***single-valued*** and ***multi-valued*** rasters, i.e. rasters in which a single attribute or multiple attributes are assigned to each raster cell or pixel. In the case of raster image data this is equivalent to the distinction between panchromatic (or achromatic) and chromatic image data. Achromatic images, such as black-and-white or grey-level images, are based on a single colour. In chromatic or colour images each pixel is assigned multiple colour components.

In the remote sensing domain the attribute count of raster images is referred to as number of ***spectral bands***. A band stands for a specific range of wavelengths within the electromagnetic spectrum. In general raster imaging applications, the attribute count is represented by the number of ***samples, primary colours*** or ***colour channels*** (see also Section 2.3.4 on colour models). In raster maps or thematic rasters the number of attributes is normally equivalent to the number of ***layers***.

Attribute Resolution

The attribute resolution specifies the domain of an attribute, i.e. the range of defined attribute values. In (Göpfert, 1991) it is called the ***quantitative resolution*** of a raster data set - as opposed to geometric resolution. Other terms used for attribute resolution are: thematic, radiometric, colourimetric or spectral resolution. The attribute resolution is usually specified by the maximum attribute value or by the ***number of bits*** required to store this value.

The attribute resolution of raster DTMs and thematic rasters is usually set to any arbitrary value suitable to store the information in as little storage space as possible. In the case of raster images there are a few 'standard' image types, which are commonly supported by most imaging applications. These standard types include: bilevel or black-and-white images, grey-level images, colour bitmap images and colour palette images. The characteristics of these standard image types are shown in Table 2–2. These standard image types are the result of the characteristics and limitations of typical sensors, data sources and of the human perception - a key factor in many applications.

A Word About 'truecolour'

The human eye is capable to discriminate between more than several thousand colours. Colour schemes, display or output devices with a radiometric resolution exceeding the colour-resolving power of the human eye are commonly called ***'truecolour'***. In computer graphics, this applies to radiometric resolutions of at least 24 bits - equivalent to 2^{24} (16.8 Million) colours. A radiometric resolution of up to 16 bits (65'536 colours) is sometimes referred to as 'hicolour'. (Murray and VanRyper, 1994)

raster type	**attribute count** [samples per pixel]	**attribute resolution** [bits per sample]	**total quantitative resolution** [bits per pixel]	**total quantitative resolution** [no. of grey-levels or colours]
bilevel or 'black-and-white' images	1	1	1	2
grey-level images	1 1 1	4 8 16	4 8 16	16 256 65'536
DTMs	1	16	16	65'536
bitmap colour images	3 3	5 8	15 24	32'768 ('hicolour') 16'777'216 ('truecolour')
colour images with colour table	3 (colour components in colour table)	8 (per colour table component)	8	256 (out of 16'777'216) depending on res. of colour table entries

Table 2–2: Summary of raster types with their typical attribute counts and resolutions.

2.3.4 Colour Models

The perception of colour by the human eye is determined by physiology, experience and viewing conditions (Murray and VanRyper, 1994). These factors make the accurate rendition of colour a very difficult task. In the pursuit to adequately model and represent colours in different environments using a wide range of output devices, several different mathematical systems have been developed. The terms colour model, colourimetric model, colour system and colour space are used synonymously for such systems. Conversion algorithms enable the conversion of colour values between different systems.

A raster data management system should support at least one of the basic colour models. Support for additional models could be provided by applying the corresponding conversion algorithms.

This section briefly describes the basic concepts and the most commonly used colour models, including conversion algorithms for the most important models. For a more detailed description, please refer to (Foley et al., 1992), (Murray and VanRyper, 1994) and (Baxes, 1994).

Additive versus Subtractive Colour Models

Additive colour models are created by adding colours to black. The more colour that is added, the more the resulting colour tends towards white. Additive colour environments are self-luminous, such as colour monitors.

Subtractive colour models work the other way round. Here, primary colours are subtracted from white to create new colours. In practice, this is achieved by adding colour pigments which absorb a certain part of the spectrum. Subtractive colour environments are reflective in nature, i.e. the colour perceived originates from a reflected light source. Colour prints are typical examples for the subtractive colour model.

RGB Colour Model

The RGB (red-green-blue) model is probably the most widely used colour system in digital imaging applications. It is an ***additive model*** in which varying amounts of the fundamental and undecomposable ***additive primary colours*** red, green and blue are added to black to produce new colours. Colours of individual pixels are represented by triplets of colour values in the form of (R,G,B). The black 'base colour' is normally represented by the triplet (0,0,0). The white colour is represented by a triplet holding the maximum values of each component. In case of a 24-bit (3 * 8 bit) radiometric resolution, the values for white would be (255,255,255). Some special cases of the RGB model which should be supported by a raster data management system include:

- different component orderings, such as GBR or BRG
- component domains of unequal sizes, e.g. a (5,6,5) bit RGB subdivision which assigns the additional bit of a 16-bit radiometric resolution to the green component, to which the human eye is the most susceptible

- finally, the rare inversion of the RGB colour definition, which assigns (0,0,0) to white and the maximum values to black

CMY and CMYK Colour Models

The CMY (cyan-magenta-yellow) model is a ***subtractive*** colour system. Cyan, magenta, and yellow are the complements of red, green and blue, respectively. When used as filters to subtract colour from white light, they are called ***subtractive primaries***. Colours are specified by what is removed or subtracted from white light, rather than by what is added to blackness. The CMY model is widely used by printers and photographers for the rendering of colours with ink or emulsion on a conceptually white surface. By applying pigment to a white surface, colours are subtracted from white light to create new colours. When illuminated, each applied colour component absorbs its complementary light colour: cyan absorbs red, magenta absorbs green, yellow absorbs blue. So, for example, by increasing the amount of cyan the amount of red reflected by the image is reduced, leaving blue plus green - in terms of additive primaries. Subtracting all of the CMY components creates black.

In practice, creating a perfect black colour using colour inks has proved to be very hard. As a consequence, an additional colour model, CMYK using black (abbreviated as K) as a fourth colour, was introduced. The CMYK model is, for example, used in many colour printers and in the commercial four-colour printing process.

	RGB	CMY	HSV
Red	255,0,0	0,255,255	0,240,120
Yellow	255,255,0	0,0,255	40,240,120
Green	0,255,0	255,0,255	80,240,120
Cyan	0,255,255	255,0,0	120,240,120
Blue	0,0,255	255,255,0	160,240,120
Magenta	255,0,255	0,255,0	200,240,120
Black	0,0,0	255,255,255	160,0,0
Shades of Grey	63,63,63 127,127,127 191,191,191	191,191,191 127,127,127 63,63,63	160,0,59 160,0,120 160,0,180
White	255,255,255	0,0,0	160,0,240

Table 2–3: Equivalent RGB, CMY and HSV values (24 bits) (Murray and VanRyper, 1994)

HSV Colour Model

The HSV (hue-saturation-value) model - also called HSB (hue-saturation-brightness) model - is user-oriented, in contrast to the hardware-oriented RGB and CMY models. It represents colours in a way which reflects the perception of the human eye. Instead of describing the primary colour components, HSV describes the intuitive components of colour. The ***hue*** component specifies "colour" in the common use of the term, such as red, orange, yellow, etc. The ***saturation*** (also called chroma) controls the purity of colour, i.e. it refers to the amount of white in a hue. For example, a full hue contains no white and appears pure. Finally, the ***value*** or ***brightness*** component describes the self-luminescence of a colour, i.e. how much light it emits.

Colour Model Conversions

The conversion of colour values between the RGB and the CMY models is straightforward. The following algorithms use the unit column vector (1,1,1) as the RGB representation for white and the CMY representation for black. In order to derive colour triplets as shown below the values for the individual colour components are scaled by the radiometric resolution of each colour component.

$$\text{RGB to CMY:} \quad \begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \qquad \text{CMY to RGB:} \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

The algorithms for converting between the RGB and the HSV models are slightly more complicated. Pseudo-code listings and descriptions of these algorithms can be found in (Foley et al., 1992).

2.3.5 Colour Maps

Colour maps are also referred to as colour palettes or colour look-up tables. They use an 'indirect' or 'pseudo-' colour representation which assigns an index value, instead of actual colour values, to each pixel. These index values represent addresses used for 'looking-up' the actual colour values in the previously established colour table.

Colour tables allow a significant reduction in the amount of data when applied to images with relatively few colours. This is because the index values can be kept much smaller than actual colour values, such as complete RGB triplets. Typically, index values are stored as 4- or 8-bit integer values in contrast to the colour map elements, which are usually stored as 24 bits (3 x 8 bits per colour). Colour tables are, however, not suited for storing images, such as photographs, which contain more than 256 colours. Here, a direct or 'literal' representation is usually preferable in view of the cost of establishing a colour table and of using larger index values. Finally, colour maps are particularly suited for changing colours in images.

Colour maps are well suited and frequently used for cartographic raster data with their limited number of colours often corresponding to individual colour 'layers'.

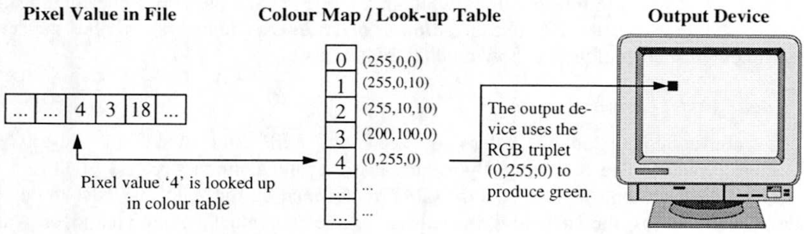


Figure 2–5: Using a colour map to specify a colour(Murray and VanRyper, 1994)

2.4 Raster Data Compression

The enormous ***storage space*** and ***communication bandwidth*** requirements of spatial raster data (e.g. satellite imagery) make data compression a very important issue in raster data management. The differing characteristics of the major spatial raster types (see Section 2.2) require the use of different compression techniques. Therefore a technique which compresses raster maps very efficiently may be of no use for satellite imagery. It is essential that a modern raster data management solution supports efficient compression techniques for all spatial raster types.

The following sections will introduce the main compression concepts. This is followed by the presentation and discussion of the main raster data compression techniques and standards. A more detailed discussion of the subject can be found in (Murray and VanRyper, 1994), (Bashkaran and Konstantinides, 1995), (Held, 1996) and (Pajarola and Widmayer, 1996).

A Word about Compression and Encoding

The terms encoding and compression are often used synonymously. Data encoding - in a very general sense - stands for the representation of data or signals in digital form. As shown in Figure 2–6, this is a multi-faceted issue. One issue is the actual encoding and decoding of data in a more narrow sense, for example, by means of a character set such as ASCII. Other issues include compression / decompression and also the efficient spatial access to data. Finally, some data encoding concepts even address the communication or the encryption of data.

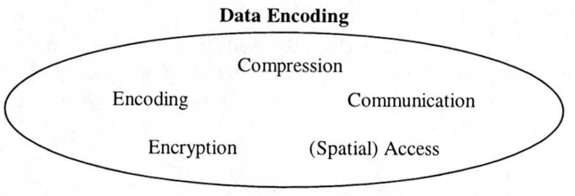


Figure 2–6: Different data encoding aspects.

For the management of spatial raster data other encoding aspects such as efficient spatial data access and spatial data partitioning are equally important. They are discussed in Sections 3.6 and 3.5.

2.4.2 Compression Concepts and Taxonomy

The Compression/Decompression Process

Figure 2–7 depicts a schematic view of a compression/decompression system. Such a system consists of an encoder and a decoder and is frequently referred to as a CODEC (COmpression / DECompression). It is, for example, common for advanced raster image formats to incorporate multiple CODECs.

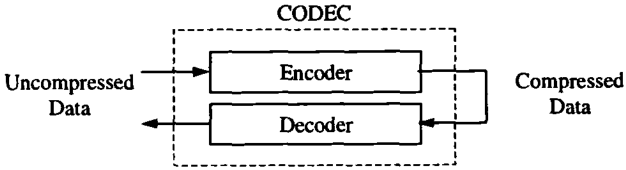


Figure 2–7: Generic data compression / decompression system.

Compression Ratio

The ***coding efficiency*** of a CODEC can be expressed by the compression ratio. It measures the quantity of compressed data in comparison with the quantity of original data (Held, 1996):

$$\text{Compression ratio} = \frac{\text{Length of original data string}}{\text{Length of compressed data string}}$$

Compression ratios higher than 1 indicate compression, values below 1 indicate deflation, accordingly. The compression efficiency for raster data largely depends on the raster type and on the information content. Typically achievable lossless compression ratios range from 20 to 50 for bilevel cartographic data to approx. 1.5 to 3 for satellite imagery (Pajarola and Widmayer, 1996).

Compression Taxonomy

The design of encoding/decoding methods always involves a trade-off among the coding efficiency, the encoder complexity and the coding delay. In cases where a certain amount of data loss is acceptable, there is even an additional aspect, the signal quality. In order to suit the requirements of different applications and data types a large number of methods have been developed. Figure 2–8 shows a classification of the compression methods based on some main characteristics. The following section focuses on those characteristics which are particularly important for the management of raster data.

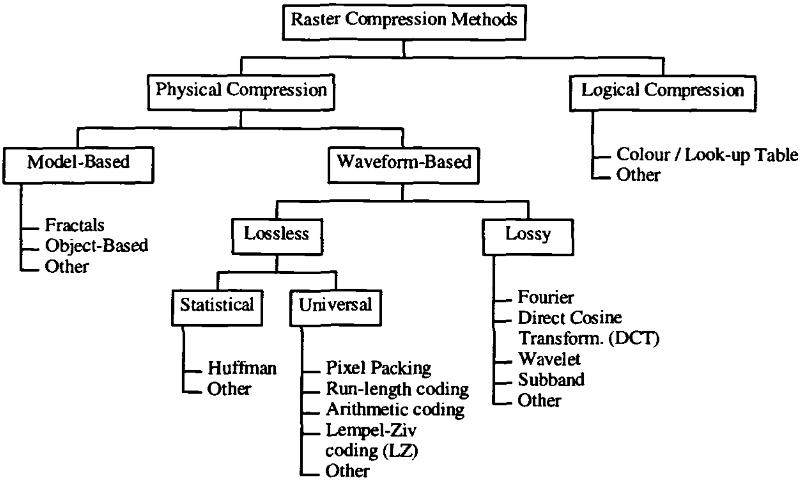


Figure 2–8: A taxonomy of image compression methods(Bashkaran and Konstantinides, 1995).

Logical versus Physical Compression

Logical compression schemes use external knowledge about the nature of the data in order to eliminate redundant fields of information and to minimise the remaining ones. A typical example of a logical compression scheme is the ***colour map*** or look-up table concept discussed in 2.3.5. Logical compression represents a special case in that it is not generic but largely data dependent, e.g. a RGB colour map CODEC only works on trichromatic colour image data. Logical compression schemes can be very efficient, if adapted to individual cases. However, the following example, sometimes referred to as the 'year 2000 problem', also illustrates the limitations and potential dangers of this method: the logically compressed two-digit representation of year numbers (e.g. 98 for 1998) leads to potential problems in arithmetics involving dates of both millenniums (e.g. (20)00 - (19)99 = -99).

Physical compression schemes exploit the statistical redundancy or any irrelevant information in the data. Depending on the type of information, the redundancy might be due to spatial, spectral or temporal correlation (Bashkaran and Konstantinides, 1995). In the case of ***spatial correlation***, the attribute values of neighbouring cells or pixels are correlated. Examples of data with high spatial correlation are raster maps which have relatively large areas of constant attribute value. High ***spectral correlation*** is typical to remote sensing image data. Here, the values of the different attributes (bands) of the same location are highly correlated. The third type, ***temporal correlation*** expresses correlation of pixel values over time. It is only important in image sequences, such as video, where it can lead to dramatic compression ratios.

Lossless versus Lossy Compression

A compression is called ***lossless*** if the decoder can reconstruct the original data from the compressed data without any loss (see Figure 2–7). In the case of raster data, this means that the original and reconstructed attribute values of each cell or pixel must be identical. Lossless compression is a frequent requirement in conjunction with spatial raster data or, for example, raster data in the medical sector.

In most other imaging applications, such as multimedia systems, identity of the original and the reconstructed data is often not a requirement. In these cases, ***lossy compression*** methods can be used. They achieve significantly better compression ratios than lossless methods, with the sacrifice that the compression process is no longer ***irreversible***.

Statistical versus Universal Compression Methods

Statistical compression methods take advantage of the probabilities of occurrence of single characters and groups of characters, so that short codes can be used for frequently occurring characters or character groups while longer codes are used to represent less frequently encountered characters and groups of characters. Statistical compression methods consist of a modelling component and a symbol-to-codeword mapping component. The ***probability model*** used in the modelling component can either be ***derived*** from the input data or from ***a priori assumptions*** about the data. The class of statistical compression methods includes the arithmetic encoding method and the important Huffman coding technique which is described in the following section.

The term ***universal compression methods*** is used for techniques which require no prior knowledge of the source to be compressed. The class of universal compression methods includes relatively simple methods such as byte-packing and run-length encoding as well as sophisticated methods such as the dictionary based string compression technique used in the LZW (Lempel-Ziv-Welch) algorithm (Held, 1996).

2.4.3 Compression Techniques

Literal Coding

In literal coding techniques each character or symbol of a data string is encoded and stored literally, i.e. using the individual symbol value. Literal coding results in no data compression but it also prevents data expansion. Some compression standards, such as PackBits, alternate literal coding with run-length coding. They use literal coding to represent random-like character sequences within the data string to be compressed. For the reasons of simple data handling and absence of a sophisticated encoding and decoding process literal coding is the basis of raster image bitmaps.

Pixel- or Bit-Packing

The byte or octet, consisting of 8 bits, represents the smallest directly accessible storage unit (data type) in most computer systems. However, the pixel information of raster data sets which have a low attribute resolution can often be represented with less than 8 bits ($2^8 = 256$ values). Bilevel raster data, for example, requires a single bit per pixel only. If such data is stored using a standard 'pixel-per-storage-unit' strategy a considerable amount of storage space is wasted. Pixel- or bit-packing schemes enable the storage of multiple pixel or cell values in a single storage unit. In the case of bilevel raster data, pixel packing results in a compression ratio of 8:1. Some raster data formats (e.g. TIFF) use pixel-packing in combination with other compression techniques (e.g. PackBits compression) to achieve even better compression ratios.

Run-Length Coding

Run-length coding reduces any type of repeating character sequences by replacing them with pairs of ***values*** and ***runs*** or run-lengths. Each value field contains the character to be repeated and each run field contains the number of repetitions (e.g. YYYYY ⇨ Y5) (see Table 2–4). Run-length encoding techniques are used in the PackBits compression scheme and, as a preprocessing operation, in certain CCITT fax data compression standards.

Run-length coding is very efficient on raster data containing relatively large regions with constant attribute values. Typical examples for this type of data are scanned topographic maps and bilevel fax data. Run-length coding is not suitable for raster data sets with a large attribute resolution and variability, such as encountered in aerial and satellite imagery and DTM data.

source string	**run-length encoded string**	**compression ratio**
AAAAACXXXXXXXXDDDD *(18 bytes)*	A5C1X8D4 *(8 bytes)*	2.25 : 1
XXXXXOOOOOOOOOXXXX *(18 bytes)*	X4O9X4 *(6 bytes)*	3 : 1
XYXY *(4 bytes)*	X1Y1X1Y1 *(8 bytes)*	1 : 2 (inflation !)

Table 2–4: Examples illustrating the principle of run-length coding and the resulting compression ratios.

Differential Coding

Raster data tend to have ***strong inter-pixel correlation***, this applies particularly to aerial and space imagery as well as to DTM data. Instead of encoding the original attribute values, differential coding allows the differences between neighbouring cell or pixel values to be encoded. These differences are usually much smaller than the original values and therefore immediately consume much less space. Differential coding is widely used as a pre-processing technique for a subsequent compression operation and is a key concept of many sophisticated encoding methods.

Huffman Coding

Huffman coding is a statistical data compression technique. It combines the modelling and symbol-to-codeword mapping function by determining and assigning codes to symbols according to their probabilities. The Huffman code construction procedure is discussed in (Bashkaran and Konstantinides, 1995). The Huffman code is a prefix-condition code, which means that no code can be the prefix of another code (e.g. code 01 is a prefix of 011). Such a prefix-condition code is uniquely decipherable (Held, 1996). Held also shows that the Huffman code is an optimum code in that it leads to the shortest average code length of all statistical encoding techniques.

Huffman coding is an important technique for encoding textual data, especially if a Huffman code table is used which matches the language of the textual data. The different CCITT fax compression schemes primarily rely on Huffman coding. The technique is also part of many advanced compression standards (e.g. JPEG).

Spatial Encoding Concepts

Most encoding techniques presented above are applicable to digital data and raster data in general. Specific concepts which take into account the spatial characteristics of raster data have been investigated more recently. In (Pajarola and Widmayer, 1996), for example, a spatial raster compression technique is presented which is based on the Hilbert space-filling curve (see also 3.4.2). This Hilbert compression technique uses a statistical prediction function which is based on the information from just a few spatially adjacent pixels. This small so-called encoder context enables good compression ratios to be achieved after encoding just a few pixels and without having to pre-calculate and store the entire image statistics. The Hilbert technique is therefore an interesting candidate for efficiently compressing spatial partitions of large raster data sets. This Hilbert compression particularly addresses the aspects of spatial data access and will therefore be discussed in Section 3.6.

Samet (Samet, 1990) and other authors propose quadtree structures for encoding and potentially compressing raster data. However, in practice, these structures are limited to bilevel raster data. They are much better suited as spatial access and partitioning techniques.

2.4.4 Compression Standards

The number of de jure or officially recognised (raster) image data compression standards is relatively small. This stands in contrast to the variety of available compression techniques and commercial / de facto standards.

The major compression standards applicable to raster data can be grouped into the following classes. A first class includes the ***general purpose lossless data compression standards*** such as gzip (gz), compress (Z) and Lempel-Ziv-Welch (LZW). All these standards are based on the original Lempel-Ziv coding algorithm. The second class comprises the ***lossless image compression standards***. It includes the PackBits compression and various CCITT versions for typically bilevel raster data and lossless version of JPEG for continuous tone image data. The third class comprises the ***lossy image compression standards***. The most important representative of this class is the international standard JPEG.

JPEG

The JPEG (Joint Photographic Experts Group) standard, officially known as *ISO/IEC international standard 10918-1, Digital compression and coding of continuous tone still images*, is the first international standard for compressing continuous tone colour or greyscale raster images. Bashkaran and Konstantinides (Bashkaran and Konstantinides, 1995) dedicate an entire chapter to JPEG and the various implementation issues.

It is important to notice that JPEG includes two basic compression methods: a DCT (Discrete Cosine Transform) -based ***lossy compression method*** and a predictive method for ***lossless compression***. Figure 2–9 illustrates the individual processing steps of a lossy JPEG compression. JPEG additionally provides progressive and hierarchical encoding extensions which are particularly useful for supporting multi-resolutions concepts.

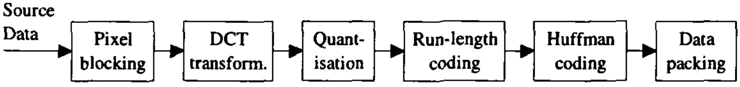


Figure 2–9: Basic lossy JPEG algorithm.

Kern and Carswell (Kern and Carswell, 1994) investigated the effects of lossy JPEG compression on the measurement accuracy in digital photogrammetric systems. They show that lossy JPEG compression yielding compression ratios of approx. 2.3-2.9 does not significantly affect the measurement accuracy of photogrammetric processes such as automatic DTM generation. However, the average compression ratios based on which the tests were carried out are close to the compression ratio of 1.6 to 2.6 achievable with lossless JPEG (Pajarola and Widmayer, 1996). It could therefore be argued whether the relatively small improvement in compression ratio would justify a transition from lossless to lossy data storage. In addition to these investigations the paper of Kern and Carswell also presents a summary of the gains in storage space, time and productivity achievable using (hardware-based) JPEG compression.

2.5 METADATA

In simple terms, metadata can be described as ***'data about data'*** or as Negroponte (Negroponte, 1995) puts it 'the second kind of bits – bits that tell you about the other bits'. The following section will introduce the subject and will show that comprehensive metadata support and management are key factors in spatial raster data management. The design and implementation of a metadata concept for spatial raster data management will be discussed in Chapters 5 and 6.

2.5.1 Introduction

The ICA Commission on Standards for the Transfer of Spatial Data defines geospatial metadata as 'data that describes the content, data definition and structural representation, extent (both geographic and temporal), spatial reference, quality, availability, status, and administration of a geographic data set' (Moellering, 1996). As indicated by this definition, the scope of metadata is very broad. The major tasks addressed by metadata are listed in Table 2–5.

Task	Metadata to provide information on ...
Data Discovery	What, where, when ?
Data Access	Cost, restrictions (copyright, security, etc.)
Data Analysis	Fitness for use: quality, intended application, completeness
Data Transfer	*Transfer mechanisms, media, formats*
Data Use	Parameters, settings

Table 2–5: Major tasks supported by metadata.

Generally, metadata can be divided into two different types:

- ***System-level or low-level metadata*** holds information pertaining to the implementation of the data management and storage solution. This, for example, includes information on the underlying object model, on data structures, storage locations and data migration paths. Usually, a large part of the system-level metadata is transparent to the user.
- ***Application-level or high-level metadata*** enables users to intelligently and efficiently find, evaluate, access and manage their actual data. A good part of this type of metadata is specific to the type of information to be managed and to the type of application.

2.5.2 Significance in Raster Data Management

The main reasons why metadata is particularly important in conjunction with the management of spatial raster data are:

- The ***volume*** of spatial raster data sets is generally very large. It is usually not feasible to access, retrieve and transfer entire data sets in order to assess their contents and suitability. This is due to limited network bandwidths and client resources and, last but not least, due to legal and commercial restrictions. Metadata concepts can help to overcome these problems by supporting suitable access, browsing and preview capabilities.
- ***Content-based search*** of very large raster data sets, if supported at all, is very expensive. It can typically not be performed for every query but needs to be pre-computed. Results of such queries could again become valuable metadata.
- ***Derived*** and ***interpreted data*** play an important role when dealing with spatial raster data such as satellite imagery. Metadata enables the lineage of raster data sets or parts thereof to be traced.
- Raster data is often acquired and organised as part of ***data series***. For example, cartographic map sheets usually belong to map series and aerial images to flight

paths and campaigns. This information is crucial, if raster data is to be efficiently managed in the context of large geo-information projects.

- The ***exchange*** of spatial raster data requires the inclusion of information on its context and semantics in order to enable the transfer to a target GIS without information loss.

2.5.3 Metadata for Spatial Raster Data

It is essential for spatial raster data management solutions which are to be used in an open, non-isolated context to support commonly used metadata categories. A first goal should be to include enough metadata to enable the generation of data sets in widely-used formats or data transfer standards (Anderson and Stonebraker, 1994). If the system should also support external on-line access and query capabilities a number of additional requirements and constraints are introduced. It is therefore advisable to base the design of the high-level metadata model on the contents of international spatial metadata standards and to complement it with application-specific requirements only where needed.

General Spatial Metadata

Based on (FGDC, 1994), (FGDC, 1995) and (ISO/TC 211, 1997) the basic categories and contents of spatial metadata can be summarised as follows:

- ***Identification Information*** includes information on the spatial and thematic extent of data sets and content summary information using descriptions, keywords or graphical overviews.
- ***Data Quality Information*** consists of information such as attribute accuracy, logical consistency, completeness, positional accuracy and lineage.
- ***Spatial Data Representation Information*** provides information on the spatial data models (e.g. raster), spatial data types and structures used.
- ***Spatial Reference Information*** covers the aspects of defining geodetic reference parameters (e.g. geodetic datum, ellipsoid and map projection), coordinate systems and units.
- ***Entity and Attribute Information*** defines domain types and ranges, units of measure, attribute resolution and value accuracy.
- ***Distribution Information*** provides information on the digital transfer including transfer mechanism, format, off-line or on-line options such as distribution media or network addresses.

Spatial Raster Metadata

The general spatial metadata categories presented above cover several aspects which are also applicable to raster data. These aspects include the one of spatial referencing and the provision of information on spatial extents. Additional generic raster metadata includes information on geometric resolution, photometric type and radiometric resolution. In addition to this generic raster metadata, additional information is required depending on the raster type and on the specific application. For the different spatial raster types his information includes:

- ***Raster Images:*** colour maps, sensor bands, on-the-ground resolution, previews, sensor type, sensor position and attitude, camera type and constants, flight campaign and line
- ***Raster Maps:*** map series, original map scale, map sheet numbers and names, production and revision information, scanning resolution
- ***DTMs:*** spatial quality information, acquisition information (e.g. interpolation function used), model type (terrain or surface)
- ***Thematic Rasters:*** content description, non-spatial quality information, acquisition information
- ***Raster Mosaics:*** overview, tile dimensions, completeness

2.5.4 Metadata Standards and Spatial Raster Support

Fortunately, the number of official spatial metadata standards is much smaller than the number of spatial data transfer standards. The main existing or evolving metadata standards recognised internationally are:

- The ***FGDC - Content Standards for Digital Geospatial Metadata (FGDC, 1994)*** is named after the Federal Geographic Data Committee of the U.S. Geological Survey, the organisation which is responsible for the standard. This metadata standard is well established and is particularly useful for high-level metadata, such as required in spatial data clearinghouse services. It includes all of the metadata categories listed above and provides a general support for different types of spatial raster data. Some of the characteristics and implementation issues of the FGDC standard are discussed in (Blott and Vckovski, 1995).
- The ***CEN/TC 287 Metadata Standard*** (CEN/TC 287, 1996) will be part of the CEN/TC 287 European standard on geographic information. Rather than explicitly specifying the data definition for spatial object types, such as different raster types, this standard refers to the corresponding definition in the Geometry part of the standard. This approach alleviates the common problem of synchronising the data definition in the metadata standard with the one used in the associated transfer standard.
- The ***ISO TC/211 - ISO Standard 15046-15*** will be part 15 of ISO 15046, a very comprehensive multi-part International Standard for Geographic Information / Geomatics. This latest international metadata standard uses many of the concepts of the standards listed above. Version 2.0 of the working draft (ISO/TC 211, 1997) contains a rich set of mandatory and optional raster metadata items, with a very comprehensive support for photogrammetry and remote sensing aspects. However, the level of detail at which application-specific raster metadata will be supported in the final standard is still debated.

Several spatial data modelling languages or transfer mechanisms such as the Spatial Archive and Interchange Format SAIF (Geographic Data BC, 1995) also provide comprehensive metadata support. A satellite imagery metadata concept based on SAIF is discussed in (Anderson and Stonebraker, 1994).

2.6 Georeferencing and Geocoding

Georeferencing and geocoding of raster data are two fundamental concepts which are used to distinguish spatial from non-spatial raster data. They enable spatial queries and operations and they are the key to the integration of raster data and other spatial data in Geo-Information Systems. The concepts of georeferencing and geocoding are occasionally mixed-up despite some clear distinctions. ***Georeferencing*** stands for the process of assigning spatial reference information and positional information to a raster data set which does not include the actual transformation process. ***Geocoding*** refers to the actual geometric transformation of a raster data set into a geodetic reference system, e.g. a map projection system.

2.6.1 Georeferencing

Georeferencing can be considered a ***spatial metadata concept*** which assigns spatial reference information and positional information to raster data sets. It is also referred to as direct image referencing (Williams, 1995) or as spatial registration (Anderson and Stonebraker, 1994). Georeferenced raster data enables spatial querying and access at the data set level (e.g. find all images based on WGS-72, UTM zone 3 in a latitude range between 30° and 35° north). They also provide the basis for transforming vector data and overlaying it with a georeferenced raster 'back-drop'. Figure 2–10 shows the main components of the typical georeferencing process. The aspects of spatial reference and positional information are discussed below. The actual design and implementation of a georeferencing concept will be discussed in Chapters 5 and 6.

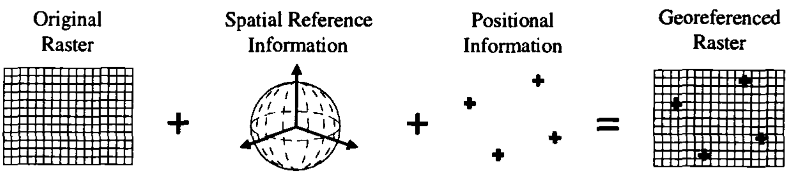


Figure 2–10: Schematic view of basic steps involved in georeferencing a raster data set.

Spatial Reference Information

Spatial reference information enables the logical assignment of raster data sets to a geodetic framework. This framework is typically defined by:

- a geodetic reference system (geodetic datum, reference ellipsoid, map projection type and zone, units)
- a coordinate system type (e.g. geocentric cartesian, geographic, map projection or local)

Due to the vast number of existing geodetic frameworks it is important to assure an unequivocal but non-restrictive specification of this type of information. The georeferencing solution presented in Chapter 5, for example, uses a coding system that is based on the EPSG list which is compiled and maintained by the Petrotechnical Open Software Corporation (POSC) (Ritter and Ruth, 1995), (POSC, 1996). This coding convention is also used in the GeoTIFF and OGIS standards.

Positional Information

The position and orientation of a raster data set within a spatial reference system can be defined using one or a combination of the following concepts:

- control or tie-points
- scale (on-the-ground resolution)
- transformation matrix

On the one hand, raster data sets with uniform pixel spacing (e.g. scanned topographic maps) are typically positioned by specifying a single tie-point, normally the origin of the raster, together with the pixel scale. On the other hand, raster data sets with non-uniform pixel spacing (e.g. aerial images) are typically georeferenced by a variable number of tie-points.

2.6.2 Geocoding

Geocoding deals with the actual ***geometric transformation*** of raster data sets into geodetic reference systems, e.g. a map projection system. This involves the resampling of an arbitrarily oriented raster data set onto a regular grid with uniform on-the ground cell or pixel dimension (see Figure 2–11). Geocoded rasters enable metric spatial operations such as the calculation of positions, distances and areas.

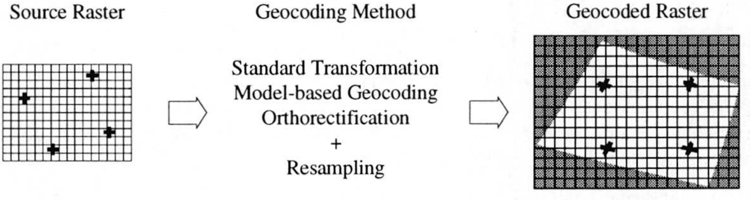


Figure 2–11: Geocoding principle.

The need for geocoding lies in a number of geometric distortions and errors introduced in the acquisition process. Based on (Williams, 1995) and (Frei, 1984) these errors can be grouped into the following categories:

- ground: topography, earth curvature, earth rotation
- sensor: sensor mechanics non-linearity, detector misalignment, channel or band-to-band misregistrations, lens distortions
- platform: variability in sensor attitude, satellite orbit or flight path variability

All these errors lead to a non-uniform geometric resolution in object-model space, e.g. a non-uniform on-the-ground resolution. This makes metric operations on such 'raw' raster data sets very difficult, if not impossible. The following geocoding methods have been developed to remove these errors.

Geocoding Methods

Commonly used geocoding methods range from relatively simple geometric transformations to very sophisticated orthorectification processes. The simple geocoding methods might be supported by a raster data management system. The more sophisticated methods are likely to be handled by specialised processes in external system (e.g. orthoimaging solutions on digital photogrammetric stations).

- ***Standard (global) transformations and interpolations*** such as similarity or affine transformations represent the simplest geocoding methods. They are particularly useful for map projection changes and for the compensation of regular distortions, e.g. the correction of maps scanned with precision scanners.
- ***Model-based geocoding*** methods are typically used in conjunction with remote sensing data. Depending on the type and resolution of data, a combination of orbit, sensor and ground models might be applied (Williams, 1995).
- ***Orthorectification*** is the most sophisticated geocoding method. It involves the determination and application of relief corrections for each pixel in the target raster. The orthorectification method is applied in the generation of high-resolution orthoimages.
- ***Resampling*** is not really a geocoding method on its own. It depicts the basic process of deriving target pixel values from a source raster using the geocoding methods presented above.

2.7 Raster Data Exchange

Raster data can be exchanged using mechanisms on different conceptual levels. These levels accommodate application- or domain-specific requirements in terms of transfer efficiency, data modelling power and complexity. The main data exchange mechanisms for raster data can be summarised as follows:

- General raster data formats
- Spatial raster data formats
- Spatial data exchange mechanisms

These data exchange mechanisms can be characterised by the type and extent of metadata provided 'internally' and by the metadata to be specified 'externally' in order to ensure a correct and complete raster data exchange. The main characteristics of the different data exchange categories are summarised in Table 2–6.

Exchange Mechanism	**'internal' metadata**	**'external' metadata**
General Raster Formats	• dimensions • radiometric resolution • compression scheme	• format • geodetic reference system • positional information
Spatial Raster Formats	+ • geodetic reference system • positional information	• format
Spatial Data Exchange Mechanisms	+ • non-spatial attribute information • quality information	• transfer mechanism

Table 2–6: Summary of different raster data exchange mechanisms.

2.7.1 General Raster Formats

General raster data formats represent the most widely-used mechanism for storing and exchanging raster data and raster images in particular. There are over a dozen raster formats which are generally supported by standard GIS and office software packages. In addition, there is even a greater number of vendor-specific raster formats which have not reached wide-spread acceptance. This does not include hybrid formats which support raster and vector data. General-purpose raster formats primarily include low-level metadata which is required to correctly read, decode and represent the data. Spatial metadata such as georeference information is typically not supported.

The spectrum of general raster formats ranges from very simple to highly sophisticated. The following selection of well-known raster formats covers most of this spectrum (Murray and VanRyper, 1994):

- PBM, PGM and PPM are simple 'bitmap'-based raster formats which were originally defined and created in conjunction with the Portable Bitmap Utilities (PBM+). These three formats are often used as an intermediate representation in the conversion between other less well known formats.

- BMP or Microsoft Windows Bitmap is a well-defined, relatively simple raster format with limited compression support that is in wide use – not only in the MS Windows environment.

- GIF - Graphics Interchange Format was originally designed by CompuServe as an image transfer and storage format in network environments. It has become a widely-used format in computer networks (e.g. Internet). It's main draw-back is the limitation of the radiometric resolution to 8 bits.

- TIFF - Tag Image File Format by Aldus Corporation (Aldus, 1992) is probably the most versatile and diverse raster format in existence. It is extensible, supports numerous data compression schemes and incorporates advanced features such as tiling and multiple images per file. TIFF has become widely used in a range of applications and on almost all platforms.

2.7.2 Spatial Raster Data Formats

The transfer and exchange of spatial raster data requires the inclusion of spatial metadata, a feature which the above-mentioned general raster image formats are lacking. Specialised spatial raster formats have existed in application fields such as astronomy or remote sensing for a considerable number of years. However, these formats have not yet reached a wide-spread acceptance and their support is only gradually being added to major GISs. Another, more recent trend is the emergence of general-purpose spatial raster formats which allow the essential georeferencing information to be stored together with the actual raster data. However, these general-purpose formats do not support specialised spatial metadata such as remote sensor attitude information or calibration values of metric cameras.

General-Purpose Spatial Raster Data Formats

- GeoTIFF (Ritter and Ruth, 1995) is a spatial extension of the TIFF format described above. GeoTIFF allows the georeferencing of a raster data set to be specified via a number of reserved TIFF tags. The georeferencing concept of GeoTIFF follows the convention described in Section 2.6.1. and it uses the POSC/EPSG database of geodetic reference systems (POSC, 1996). GeoTIFF-support has recently been added to major GIS and remote sensing products.
- NITF (National Imagery Transmission Format) originates from the U.S. military. Today, it is widely used in U.S. government agencies and it is also gradually starting to appear in the commercial GIS and remote sensing environment. NITF incorporates a basic 'georegistration' or georeferencing concept and it allows an abundance of other spatial and non-spatial information (e.g. vector overlays and text) to be stored and exchanged together with the raster image data.

Specialised Spatial Raster Data Formats

- GIS-GEOSPOT is the product delivery format of Spot Image. It consists of a collection of files with different file suffixes which hold information such as georeference, data source and image statistics. It is planned, that the georeferencing concept of GeoTIFF will be adopted in a new version of GIS-GEOSPOT (Angleraud, 1996).
- BSQ (Band Sequential), BIL (Band Interleaved by Line) and BIP (Band Interleaved by Pixel) are the prevalent traditional formats for storing remote sensing data (Jensen, 1996). They have a relatively simple structure consisting of a header followed by a number of data blocks which are separated by end-of-file (EOF) markers. The header holds a variety of metadata, including information on the map projection, reference ellipsoid and scene centre.
- HDF - Hierarchical Data Format is developed by the National Centre for Supercomputer Applications (NCSA). It is a very versatile, extensible format for raster and other (spatial) data types. It supports a large variety of metadata, including spatial reference information. HDF is frequently used in the astronomic and remote sensing communities (e.g. for the future Landsat 7 data to be archived and distributed as part of the U.S. Earth Observation System Distribution System (EOSDIS)).

2.7.3 Spatial Data Transfer Mechanisms

Currently, the evolution of spatial data transfer mechanisms and standards is in full flow. National and domain-specific standards (e.g. military or hydrography) which have been around for several years are in the progress of being implemented and further enhanced. At the same time there are very busy activities in the international standardisation of spatial data exchange, namely by ISO (International Standardisation Organisation) TC 211 and CEN (European Committee on Standards) TC 287. In parallel, there is a commercially driven effort towards defining and implementing the Open Geodata Interoperability Standard (OGIS). Summaries and classifications of national and international spatial data transfer standards can be found in (Bill, 1996b) and (Moellering and Hogan, 1997).

Many of the earlier data transfer standards were focused on vector data. Raster data was often given a much lower priority - especially in standards which originally targeted cadastral survey applications such as in the Swiss data exchange standard AVS/INTERLIS (Gnägi and Nebiker, 1997). The lower priority of raster data is illustrated by the fact that 30% of the standards assessed in (Moellering and Hogan, 1997) did not support a raster, grid or image data type. Examples of existing national or domain-specific spatial data transfer standards with spatial raster data support include:

- EDBS (Einheitliche Datenbankschnittstelle), Germany
- EDIGéO (Échange de Données Informatique dans la domaine de l'information Géographique), France
- SAIF (Spatial Archive and Interchange Format), Canada (Geographic Data BC, 1995)
- SDTS (Spatial Data Transfer Standard), USA
- DIGEST (Digital Geographic Information Exchange Standard), NATO

It can be expected that within a few years spatial raster data support will be available in all major standards which will not be replaced by the new ISO and CEN standards.

2.8 OPERATIONS ON SPATIAL RASTER DATA

Raster data processing and image processing in particular are booming research areas. As a consequence, the variety and sophistication of raster operations are continually increasing. With respect to operations on spatial raster data the driving application domains are: remote sensing, photogrammetry, geostatistics, raster GIS in general and visualisation. In each of these fields a number of very specialised and often domain-specific operations have been developed. Nevertheless, most of these specialised operations consist of sequences and combinations of basic raster operations. In (Göpfert, 1991) these basic operations are grouped into greylevel operations, frequency operations, geometric operations and (area-) correlation operations.

In addition to the largely domain-independent summary of raster operations provided in (Göpfert, 1991) and (Bartelme, 1995) there exists an abundance of literature on domain-specific raster operations. Remote sensing aspects, for example, are discussed in

(Jensen, 1996), (Campell, 1996) and (Williams, 1995). Typical cartographic raster operations are summarised in (Stengele, 1995) and (Carosio, 1996) and image processing operations are well covered in (Baxes, 1994) and (Ekstrom, 1984).

This chapter looks at raster operations from a data management perspective. It attempts to classify the different raster operations at a conceptual level without going into a detailed discussion of specific algorithms or implementations. The overview presented in Table 2–7 summarises typical operations and processing tasks applicable to spatial raster data and groups them into the following four classes:

- radiometric corrections
- geometric corrections
- raster combinations
- raster queries and manipulations

The first two classes primarily include 'base' functionality which is required to derive geocoded spatial raster products (e.g. orthoimages) from application-specific raw data. The third group includes operations which are frequently used for combining multiple data sets and for creating seamless raster databases. The fourth group encompasses operations which enable content-based querying, manipulation and analysis of spatial raster data. The individual operations which range from relatively simple 'nearest-neighbour resampling' to highly sophisticated 'feature recognition' greatly vary in their complexity.

In Table 2–7 the individual operations are also assessed with respect to their suitability for inclusion in a spatial raster management environment. Operations assigned to the left column are required as part of a basic raster management solution. Operations marked in the middle column would be suitable to be incorporated and would certainly be desirable in an operational system. Operations which are assigned to the right-hand column are typically part of a specialised and application-specific raster processing solution. Some of these operations are too specialised to be incorporated in general-purpose spatial raster management solutions. The integration of operations such as pattern matching or automated feature extraction, on the other hand, would allow applications to be extended to very large numbers of raster objects (e.g. matching of finger prints) or to very large data sets (e.g. extraction of roads in high-resolution satellite imagery).

Raster Operations:	Required Raster Management Functionality	Optional Raster Management Functionality	Specialised External Function-ality
radiometric corrections (& transformations)			
• radiometric transformation (e.g. grey⇨RGB)	✓		
• colour model transformations	✓		
• contrast enhancement and correction		✓	✓
• edge smoothing / edge correction		✓	✓
geometric corrections (& transformations)			
• resampling	✓		✓
• relative ('raster-to-raster')			✓
• trend functions			
– affine, Helmert, etc.		✓	✓
– projection ('raster-to-map')	✓		
• differential			
– sensor models		✓	✓
– orthorectification		✓	✓
raster combinations			
• image fusion		✓	✓
• band / layer combination	✓		
• mosaicking of:			
– raster imagery		✓	✓
– raster maps, thematic rasters and DTMs	✓		
raster queries and manipulations			
• image classification		✓	✓
• statistical operations		✓	✓
• geometric operations (e.g. area calculations)		✓	✓
• neighbourhood operations (e.g. DTM slopes)		✓	✓
• raster correlation			
– pattern matching and recognition		✓	✓
– auto-correlation (e.g. DTM generation)		✓	✓

Table 2–7: Overview of typical operations applicable to spatial raster data with a distinction between operations which are required or suitable for inclusion in a raster management solution and specialised application-specific operations.

Leer - Vide - Empty

3

RASTER DATA MANAGEMENT

3.1 INTRODUCTION

The previous chapter discussed the main spatial raster concepts in general. This chapter will focus on concepts and techniques for ***managing*** spatial raster data. It will address questions such as how can large spatial raster data sets be accessed and managed and it discusses what management functionality is required.

In the first section, the management characteristics of raster data are compared with those of other spatial and non-spatial data types. This is followed by an overview of typical spatial raster data management functionality and a discussion of the aspects of data representation, spatial partitioning, spatial data access and multi-resolution support. In addition, the major data models and the corresponding database technologies are introduced and evaluated with respect to their raster data support.

3.2 SPATIAL VERSUS NON-SPATIAL DATA MANAGEMENT

Raster data are inherently spatial, i.e. multi-dimensional. Traditional database management systems, however, are specialised in scalar data with linear characteristics. Specialised spatial data management solutions have been developed for specific spatial data types such as vector or network data.

Table 3–1 summarises some of the typical data management characteristics and requirements of several spatial and non-spatial data types. This summary shows that from a data management perspective there is no 'standard spatial data type'. In fact, the first four characteristics of raster and vector data are diametrically opposed. As a consequence, there is no universally applicable standard spatial data management solution. The following sections will look at the special requirements of spatial raster data in more detail.

	Spatial Data			Non-Spatial Data	
	Raster	Vector	Network	Multi-media	Attribute Data
Complex Objects					
Complex Object References					
Large Objects					
Large Numbers of Objects					
Spatial Access / Queries					
Thematic Access / Queries					
Sequential Access / Queries					
Georeferencing					

Table 3–1: Management characteristics and requirements matrix for different data types (grey = very important, white = less important or not applicable). The region marked with a circle points out the opposing characteristics of the two main spatial data types: raster and vector.

3.3 Basic Spatial Raster Management Functionality

In order to enable the subsequent evaluation and discussion of concepts and techniques for managing spatial raster data, the standard management functionality first needs to be identified. The main categories of functionality are listed in Table 3–2.

Basic Spatial Raster Data Management Functionality

General Raster Object Support	Large Raster Object Support
Data Handling and Manipulation Metadata Management Spatial and Non-Spatial Queries Visualisation	Spatial Data Access Data Partitioning Multi-Resolution Representation Storage Management
⇨ **'visible'** functionality	⇨ **'transparent'** functionality

Table 3–2: Basic functionality to be supported by a spatial raster management solution.

3.3.1 General Spatial Raster Object Support

This first group of features represents the main functionality required to handle spatial raster data objects irrespective of raster type, data volume or spatial extent. This covers the basic functionality which should be ***visible*** to the user.

Data Handling and Manipulation

A raster data management solution will have to support the basic tasks of creating, deleting and copying individual raster data sets and raster data mosaics. It should also provide some of the raster operations presented in Section 2.8. The minimally required mosaic management functionality includes the creation of seamless raster databases and the insertion and extraction of raster objects.

Metadata Management

Extensive metadata support is one of the key issues in spatial raster data management. The metadata to be supported ranges from general information about raster data sets (e.g. dimensions, radiometric properties) and georeferencing information to application-specific information. Examples for such application-specific metadata include information relating to cartographic map sheet series or photogrammetric flight campaigns.

Spatial Querying and Inter-Object Access

A spatial raster data management solution should support queries such as *'find all satellite images covering the region Lat 47° to 48° north and Lon 8° to 9° east with a geometric resolution better than 100 metres which were acquired in summer 1996'*. This query has a spatial (region specification), thematic (image type and resolution) and a temporal component. A spatial raster management solution requires a suitable spatial access concept in order to efficiently answer the spatial clause of the query.

Visualisation

The visualisation of spatial data, including raster data, is a research field on its own. Today, there exist numerous powerful raster-based 2D- and 3D-visualisation solutions, either as stand-alone systems or as GIS modules. If suitable interfaces are provided, these external systems can be used for visualising the contents of a raster database, namely individual raster images. In addition to the standard raster visualisation a number of special visualisation requirements exist. These include the visualisation of spatial and thematic extents, the browsing of quick-look data as well as the graphical analysis of quality and completeness information.

3.3.2 Large Spatial Raster Object Support

For the efficient management of very large raster objects, an additional group of functionality is required. The sole purpose of this second group of features is to enable the scaling of the previous functionality to very large data sets without affecting the system performance. In an ideal case this class of functionality would be completely ***transparent*** to the user.

Spatial Partitioning

Large raster objects often exceed the amount of primary memory available on a computer system. They can therefore no longer be manipulated as a single contiguous data string and have to be divided into manageable pieces. A raster data management solution should support the transparent spatial partitioning and reconstruction of large raster objects involving secondary (and possibly tertiary) computer memory.

Spatial Intra-Object Access

The efficient access to partitions or to individual pixels or cells of large spatial raster objects requires the use of specialised spatial access methods. This is a fundamental requirement in the management of large raster mosaics.

Multi-Resolution Representation

Operating on large raster objects at full resolution is expensive. However, access at full resolution level is often not required or it might be restricted due to commercial or security requirements. Multi-resolution concepts should enable the provision of raster data at the desired or authorised resolution level.

Storage Management

The storage management functionality of a modern spatial raster data management solution should provide a largely automated support for different storage strategies (DBMS-internal or -external) and storage technologies as well as for the migration of data between different storage locations and hierarchies.

3.4 Raster Data Modelling

Data structures and data types are the tools for the digital representation of data, in our case raster data. They are the determining factors for the functionality and the efficiency of a data management solution. Data structures address the implementation aspects, i.e. ***how*** can data ***efficiently*** be organised and operated upon? Data types, and particularly abstract data types, on the other hand, specify only the functional properties of an underlying data structure, i.e. ***what*** is the ***functional behaviour*** ?

3.4.1 Data Structures

Data structures provide the means to efficiently organise, access and operate upon data. As shown in Figure 3–1 some types of data structures are generically applicable, others are dedicated to special types of data such as spatial data.

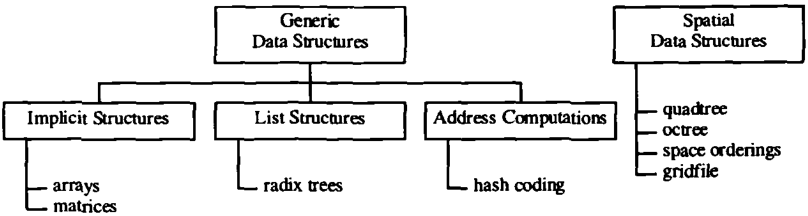


Figure 3–1: Taxonomy of data structures with typical representatives of the different classes.

Out of the variety of existing data structures a few are suitable for representing raster data. Each of these structures fulfils one or several of the following main (spatial) raster data management tasks:

- encoding of raster data at the pixel (raster cell) level
- partitioning of raster data (spatial and/or thematic)
- efficiently accessing raster data sub-sets or partitions (spatial and/or thematic)
- supporting content-based queries and operations

Optimal solutions supporting all these tasks will likely have to use combinations of the different data structures presented in the following sections.

Generic Data Structures

Generic data structures can be grouped into three major classes: implicit data structures, lists, and address computations (Figure 3–1).

- ***Implicit data structures*** use formulae or declarations to define the structural relationship between data elements. Relationships among data elements serve the purpose of a) defining the semantics and b) providing a means of accessing the data. In the case of implicit data structures these relationships do not need to be stored explicitly. Implicit data structures are particularly useful in cases where the structure of data is ***static*** and where it obeys certain ***patterns***. Typical representatives of implicit data structures are arrays and matrices.

  ⇨ Implicit data structures are frequently used for representing ***small raster images*** or parts of large raster data sets (scan-lines, strips or tiles). Relationship operations such as 'finding neighbouring pixels' are performed using extra information, such as image dimensions, number of colours per pixel and number of bits per colour.

- ***List structures*** are ideal for representing ***irregular*** or highly ***dynamic*** data. They allow not only the data values to be changed but also permit insertion, deletion and re-arrangement of the data elements. List structures are very important for managing the amount of memory available on a computer system by dynamically allocating and freeing memory for a given object.

⇨ List structures are particularly useful for organising and managing sub-sets or partitions of very large raster data sets. They allow allocation, access and management of multiple raster partitions on an as-needed basis.

⇨ For some special cases, such as bilevel image data, quadtree list structures can be used for encoding and compressing raster at the pixel level.

- ***Address computations*** or 'hash coding' refer to search techniques that assign an address of a storage cell to a key value x by means of a formula that depends on x only (Nievergelt and Hinrichs, 1993). The concept of address computation, if at all applicable to the specific problem, is potentially very fast. Space orderings such as the Morton order, discussed in Section 3.4.2, use the concept of address computation.

  ⇨ In raster data management, address computations can be used for an efficient spatial access to individual raster cells or raster data sub-sets such as tiles.

3.4.2 Spatial Data Structures

Spatial or metric data structures serve multiple purposes. They are used for storing simple spatial objects such as points, rectangles or raster cells and they at the same time organise the embedding multidimensional space (Nievergelt and Hinrichs, 1993). As such they are often capable of simultaneously addressing the problems of data encoding, spatial partitioning and spatial data access. This section focuses on the basic characteristics of different types of spatial data structures and on their capability to encode and store raster data. Their suitability with respect to spatial partitioning will be presented in Section 3.5. Their role in enabling an efficient spatial raster data access will be discussed in Sections 3.6. and 5.4.

Spatial data structures are often ***hybrid data structures***, which, for example, combine address computation and list processing techniques. The two main classes of spatial data structures, which are relevant to raster data management, are tree structures and space orderings. Their characteristics and use for raster data management are presented below. A detailed discussion of spatial data structures can be found in (Samet, 1989), (Samet, 1990), (Nievergelt and Hinrichs, 1993), (Bartelme, 1995), (Bill and Fritsch, 1991) and (Worboys, 1995).

Tree Structures

In computer science, the term tree stands for a directed, acyclic graph. Such a graph originates from a root node and branches out into further nodes 'without ever building any loops'. Tree structures belong to the generic class of 'list' data structures and are not limited to spatial applications. An example of such a generic data structure is the B-tree or binary search tree, which is the basic indexing and search structure for linear data implemented in almost any DBMS.

In our context, however, the primary interest is with spatial variants of tree structures. They enable a ***hierarchical*** and ***dynamic*** subdivision of space and can be used for point, line and region data. They also preserve locality, which means that local operations only affect the data of very few neighbouring leaf nodes and branches. Out of the great variety of existing tree structures only a few are of practical value for representing raster data. The most important one is certainly the region quadtree.

- ***Region quadtrees*** recursively sub-divide a region into 4 equal-sized quadrants, each represented by a leaf node. Therefore, in a quadtree each parent node has 4 child nodes, often referred to as a 'fan-out' of 4. The recursive subdivision of a particular quadrant is stopped, once all data elements inside this quadrant have equal values. Region quadtrees are an important component of several spatial data access methods and spatial data types, such as the Spatial Location Code (SLC) concept (Oosterom and Vijlbrief, 1996) and Hyperspatial Helical Code (HHCODE) of the Oracle Spatial Data Option (Oracle, 1995a).

  ⇨ Region quadtrees can conceptually be used to encode raster data as shown by (Samet, 1989), (Cheiney and Kerhervé, 1990) and (Nievergelt and Hinrichs, 1993). However, the practical use for encoding purposes is limited due to the restriction to relatively few colours or grey levels. These problems are outlined in (Worboys, 1995). Nevertheless, region quadtrees are potentially very useful for the spatial partitioning of very large raster objects and for implementing a multi-resolution support.

Space Orderings

The basic principle of space orderings or 'space-filling curves' is to organise the fields / cells of a regular tessellation in a sequential order according to a certain strategy. This is equivalent to the mapping between a typically 2-dimensional object space and the linear computer address space by means of an ordering function, typically a hash coding scheme. Most types of space orderings can be extended to 3 and more dimensions. Summaries of different space orderings and a comparison of their characteristics can be found in (Abel and Mark, 1990), (Bonham-Carter, 1994), (Nulty and Bartholdi, 1994) and (Worboys, 1995).

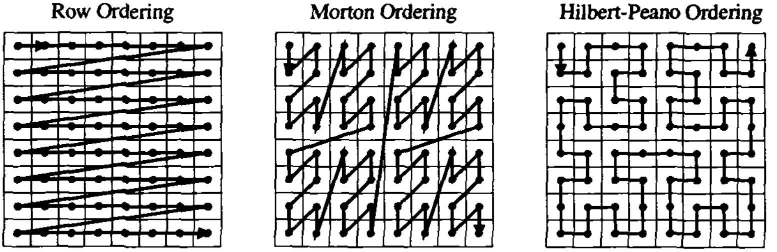


Figure 3–2: Different types of space orderings applied to a 8 x 8 raster (Samet, 1989).

The following space orderings are particularly relevant for representing raster data:

- ***Row orderings*** represent a special case among the space orderings. Row orders are often not even classified as space orderings. However, row orders are the underlying concept of most raster data formats. The hash function for determining the memory address of a pixel (or cell) in an uncompressed raster data set is:

$$MemOffset(row, col) = ((row * Width) + col) * (SamplesPerPixel * BitsPerSample)$$

$$MemAdress(row, col) = MemAdress(0,0) + MemOffset(row, col)$$

- ***Morton orderings***, sometimes also referred to as Z- or N-orderings are widely used in GIS (Abel and Mark, 1990). They organise raster cells in a quadrant-recursive fashion. This means that each sub-quadrant is visited (or filled) before moving to the next sub-quadrant. Morton codes or keys for raster cells are easily computed by interleaving the bits of their row and column values (see Figure 3–3).

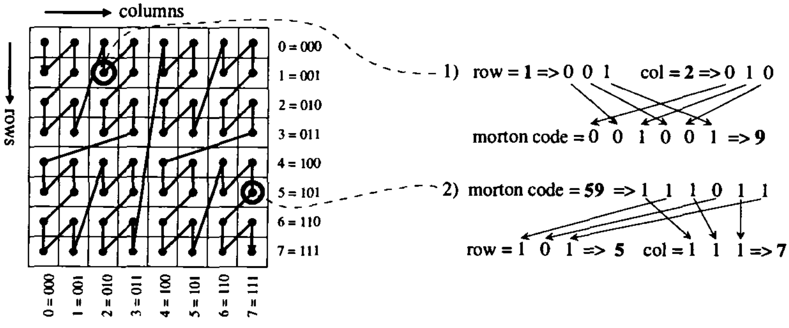


Figure 3–3: Morton ordering for a 16x16 pixel raster. Morton code computation by means of bit interleaving. 1) Computation of Morton code from on row/column position. 2) Computation of row/column position from Morton code.

- ***Hilbert orderings*** represent another important class of space orderings. They are based on the classic Peano curve. Hilbert orderings are in many respects similar to the Morton orderings. Hilbert orderings are better in preserving spatial contiguity but are poorer than Morton orderings in terms of oversearch for range queries (Abel and Mark, 1990). A major weakness of Hilbert orderings, as opposed to Morton orderings, is the complexity of the key encoding and decoding algorithm. A very interesting application which uses Hilbert orderings for encoding and compressing raster data is presented in (Pajarola and Widmayer, 1996).

3.4.3 Data Types

The concept of data types is one of the fundamental concepts in software development and database technology. Koshafian and Baker (Koshafian and Baker, 1995) give the definition

Data Type = Representation + Operation

whereby ***representation*** is equivalent to the description of a domain from which data elements are drawn (e.g. the domain of 32-bit integer values). The ***operation*** component refers to the definition of a set of operations applicable to this data type (e.g. addition, subtraction etc.).

Base Types

The number of basic data types or ***base types*** available in programming languages and DBMSs is relatively small. The following list summarises the base types supported by the database query language standard SQL-92.

- integer
- floating point numbers
- character strings
- day-time, time interval
- numeric and decimal

⇨ These base types are of limited use for storing raster data. An exception to this rule is the use of character strings for the representation of small raster images such as icon bitmaps (e.g. X BitMap format XBM). However, new base types such as BLOB ("binary large object") and CLOB ("character large object") which will be introduced by the SQL-3 standard (Date and Darwen, 1997) will lead to the standardisation and official recognition of two data types which have been available as 'built-in data types' in commercial DBMSs for many years.

Abstract Data Types

The concept of abstract data types (ADTs) takes the basic data type concept a step further by encapsulating or hiding the implementation from its external representation or interface. This enables the modification or replacement of the embedded data structures without affecting the interface of the data type. Abstract data types therefore define the functional behaviour of a data structure, i.e. they answer the question of 'what functionality is supported by a certain data type' regardless of the internal implementation.

Abstract data typing is one of the key ideas in object orientation and in object-oriented and object-relational database technology. Abstract data typing, for example, allows ***user-defined data types*** or base type extensions and the corresponding ***user-defined functions*** and operators to be defined and implemented (Stonebraker and Moore, 1996).

3.4.4 Selected Raster Modelling Concepts

From the variety of data types and structures which would (potentially) be suitable for representing raster data two very useful and powerful concepts were selected. They cover the range from very simple and generic (binary large objects) to potentially very complex and specific concepts (abstract data types). Both concepts can be used in a file-based and a DBMS-based environment.

Binary Large Objects

The concept of binary large objects (BLOBs) offers a very generic representation of raster data. Binary large objects are variable-length binary strings, with a typical size limitation in the excess of one Gigabyte. The concept is not bound to a specific data model or database technology. In fact, variable-length bit or character strings have been supported by commercial DBMS products for several years. The names used include BLOB, LONG RAW, IMAGE, VARGRAPHIC, etc. This variety indicates that the concept is currently lacking a certain level of standardisation. The evolving standard SQL3 defines BLOB and CLOB as new base types (Date and Darwen, 1997).

Stonebraker and Moore argue in (Stonebraker and Moore, 1996) that BLOBs are no data types due to the lack of supported operations. This statement is somewhat biased. It neglects the fact that most systems support operations to: create and delete BLOBs and to read from, insert into and append to BLOBs entire objects or pieces thereof (Koshafian and Baker, 1995). However, the greatest disadvantage of BLOBs is at the same time their greatest advantage: they do not specify the semantics and structure of the objects stored inside. On the one hand this always requires external information for encoding, decoding and interpreting the stored information, on the other hand this allows just about any conceivable data to be stored such as text, sound, video and last but not least ***raster data***.

Abstract Data Types

The modelling power and flexibility provided by abstract data types (ADTs) makes them the selection of choice 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.

The following simplified example shows the definition of an abstract data type 'RasterImage' in SQL3:

```
CREATE TYPE RasterImage
     WITH OID
     (PRIVATE
          Width     INTEGER,
          Height    INTEGER,
          ...
          DataString     BLOB,
     PUBLIC
     CONSTRUCTOR FUNCTION  RasterImage (I RasterImage,
          Path VARCHAR)
     FUNCTION ...
     ...
     );
```

This RasterImage data type defines the PRIVATE attributes Width, Height and DataString which are not visible and which cannot be accessed directly. They can only be modified through one of the PUBLIC functions defined in the lower part of the data type definition. In this example the attribute DataString, which stores the actual raster data, is defined as a BLOB (binary large object) data type. In the BLOB-based solution presented before, the knowledge about the semantics of data stored in the BLOB attribute had to be within the application. With ADT-based solutions the semantics can be encoded and encapsulated in the user-defined functions operating on the Data attribute.

HHCODE - An Example of a Proprietary Generic Spatial Data Type

In response to the limited support for spatial data in traditional DBMSs some manufacturers recently started to introduce generic spatial data types. Oracle, for example, is offering the Spatial Data Option[2] (SDO) as an extension to their relational DBMS (Oracle, 1995a). The SDO is based on a generic multi-dimensional data type called HHCODE (hyperspatial helical code), a set of associated operations for querying and manipulating this type of data and on a largely automatic data partitioning scheme.

The HHCODE is a proprietary coding scheme based on a recursive sub-division of space into discrete cells. The HHCODE closely resembles the standard Morton ordering and, in the two-dimensional case, the region quadtree. The dimensions of the cells are determined by the spatial extents of the area to be covered and by the required resolution, which in turn determines the length of the codes.

3.5 Spatial Partitioning

The encoding/decoding concepts described in 2.4 are usually sufficient for accessing and managing small- to normal-size raster objects such as usually found in multimedia applications. These objects are usually managed and accessed as a whole, e.g. loaded into a single contiguous primary memory area. In spatial raster data management where objects can be very large - often exceeding the amount of primary memory available - spatial partitioning concepts need to be introduced which supplement or even replace the basic encoding/decoding schemes.

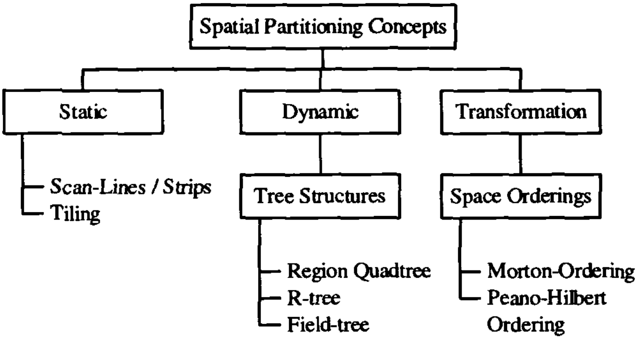


Figure 3–4: A taxonomy of spatial partitioning concepts and techniques.

Spatial partitioning techniques allow data sets, in this case very large raster data sets and raster data mosaics, to be sub-divided into manageable pieces or to be transformed to linear access structures. They enable the access to spatial sub-sets of a raster object without the need to retrieve and decode the entire object.

The ideal spatial raster data partitioning solution should:

- support efficient ***spatial access*** and querying
- support ***compression*** of the individual partitions

[2] With the release of Oracle8, the company renamed its Spatial Data Option (SDO), formerly called Oracle Multidimension, to Oracle Spatial Data Cartridge.

- ***cluster*** spatially adjacent raster elements
- be largely ***transparent*** for client applications and end-users
- ***minimise*** the ***computational cost*** and the amount of ***overhead data*** involved in decoding a raster data sub-set

A selection of partitioning concepts applicable to raster data is presented below:

Scan-Lines / Strips

The simplest - and at the same time most natural - spatial partitioning concept is the sub-division into ***scan-lines***, i.e. into individual rows of pixels. Scan-lines result from numerous scanning-based data acquisition techniques. They are the basis of most raster encoding/decoding concepts and formats (see 2.4 and 2.7.1). The ***strip***-based partitioning concept shown in Figure 3–5 extends the scan-line based approach by grouping several scan-lines into strips which can subsequently be addressed and retrieved individually. This partitioning concept is used in several data formats (e.g. TIFF).

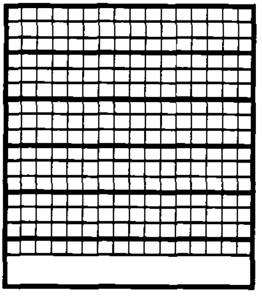


Figure 3–5: Strip-based raster partitioning (3 rows per strip).

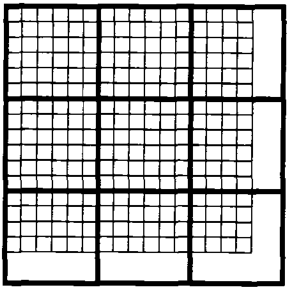


Figure 3–6: Tiling-based raster partitioning based on regular, square tiles (tile dimension = 6 pixels)

The characteristics of the scan-line based partitioning approach can be summarised as follows:

- low computational cost for encoding and decoding
- well suited for individual raster objects with limited dimensions
- well suited and often used for sequential data I/O, such as during data acquisition, output and transfer
- not suited for (very large) raster mosaics with dimensions which can dynamically grow or shrink
- the issue of spatial data access needs to be addressed externally

Tiling

The tiling approach sub-divides the raster space into a grid of sub-images or ***tiles***. It could be considered as an extension of the basic raster concept at a higher level of aggregation. Tiles are often quadrangular, usually even square, with constant dimensions. (Samet, 1989) summarises a number of tiling concepts, some of which are very complex and not suited for the partitioning of raster data.

The tiling-based partitioning concept:

- allows each tile to be treated as a raster data set (e.g. a raster image) on its own right
- supports dynamic growing and shrinking of raster data sets in all directions
- is well suited for raster mosaics and large raster images
- is supported by certain data formats (e.g. TIFF)
- requires the issue of spatial data access to be addressed separately

Tree Structures - Region Quadtree

Tree structures such as the region quadtree (see 3.4.2) enable the dynamic and hierarchical partitioning of a (raster) data space as opposed to the static partitioning provided by the previous concepts.

Most documented applications of region quadtrees on raster data (e.g. (Worboys, 1995), (Samet, 1990)) use the quadtree structure for encoding the raster data down to the level of individual raster cells or pixels. In this approach simple attribute values (e.g. colour values) are assigned to individual leaf nodes. Instead of simple values, leaf nodes could also hold pointers to more complex information, such as binary data strings. This approach allows the region quadtree to be used as the basis for a dynamic variable-size tiling approach. The criteria determining the depth of the quadtree could either be the spatial dimensions of the smallest tile or the maximum amount of data to be stored for an individual tile. This second, data-driven partitioning approach allows the partitioning process to be adapted to the system specific storage parameters, such as disk block sizes. One potential problem of the dynamic, quadtree-based partitioning approach is the data re-organisation involved in the splitting and merging of leaf nodes. In the case of raster data such re-organisations could be very expensive.

Space Orderings

The space orderings presented in Section 3.4.2 can also be used for partitioning raster data. The partitioning effect of the Morton or the Hilbert ordering is closely related to that of the region quadtree. These two orderings always completely fill a sub-quadrant (equivalent to a quadtree leaf node) before moving to the next quadrant.

The partitioning effect of space orderings is based on the fact that spatially adjacent objects (e.g. pixels) are stored in adjacent memory locations. Applying standard secondary storage management techniques which store the values of a range of adjacent memory locations on an individual disk page (or in a database column) automatically leads to a spatial partitioning effect. The advanced Hilbert raster compression and encoding technique presented in (Pajarola and Widmayer, 1996) uses the spatial partitioning effect of the Hilbert ordering. In this approach the individual spatial partitions are determined dynamically based on the amount of compressed data along the Hilbert curve and on the disk page sizes. The result of this dynamic spatial partitioning concept is an optimal disk page utilisation.

Summary and Comparison

The characteristics of four selected spatial raster partitioning concepts are summarised in Table 3–3.

Partitioning Concept / Characteristics	Scan-Lines / Strips	Regular Tiling	Region Quadtree	Space Orderings
partitioning concept		constant-size regular tiling	hierarchical tiling	transform-ation
partition dimensions	static	static	dynamic	dynamic
support for spatial extension	(yes) (along rows)	yes	no (sub-division)	yes (along order)
support for spatial partition access	external indexing	external indexing	included	included
support for partition compression	yes	yes	possible	yes (complex)

Table 3–3: Comparison of main spatial partitioning concepts applicable to raster data.

Figure 3–7 illustrates how different parts of a raster data set are retrieved when the same range query is run against different spatial partitioning concepts. It is clearly visible that the linear subdivision of the scan-line or strip-based partitioning approach (on the left) is not well adapted to a typical rectangular range query. As a consequence, a large amount of overhead data is retrieved in this scenario. The other three partitioning concepts are much better adapted to typical range queries. The different types of spatial queries will be discussed in the following section on Spatial Data Access.

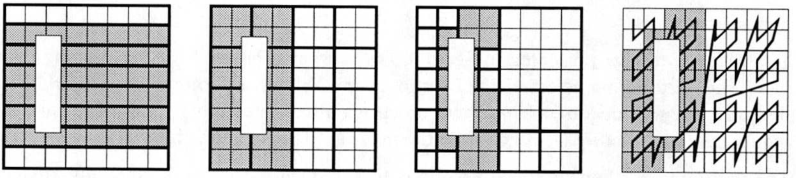


Figure 3–7: Schematic representation of the spatial partitions retrieved (grey) by a spatial range query (white rectangle) which was run against different spatial partitioning schemes.

3.6 SPATIAL DATA ACCESS

The purpose of spatial data access or spatial indexing methods is to provide an efficient access to spatial objects (or to parts thereof) on secondary storage. A spatial access method incorporates a spatial data structure (see Section 3.4.2) which allows spatial objects to be ordered and indexed based on criteria such as position and spatial extent. Dröge provides a classification and concise description of different spatial access methods (Dröge, 1995).

3.6.1 *Spatial Querying*

Efficient spatial data access can be expressed as the capability of the access method to efficiently process spatial queries. A short discussion of different spatial queries can be found in (Dröge, 1995). Table 3–4 provides a summary and short description of spatial queries which are applicable to spatial raster data. The listed *enclosure query* and *containment query* represent special cases which can always be answered by a more general ***intersection query*** and a subsequent filtering step.

Point Queries	
point query	search raster objects containing the query point ⇨ query predicate 'point inside'
Region Queries	
enclosure query	search raster objects enclosed by the query region ⇨ query predicate 'inside'
intersection query	search raster objects intersecting the query region ⇨ query predicate 'intersect'
containment query	search raster objects completely covering the query region ⇨ query predicate 'contains'
nearest neighbour query	search neighbouring raster objects ⇨ query predicate 'nearest'

Table 3–4: Summary of spatial queries applicable to raster objects.

3.6.2 *Spatial Intra-Object and Inter-Object Access*

In the case of spatial raster data management, spatial indexing should support efficient spatial querying and access for the following types of raster data:

- Pixels or cells of raster objects (primary index)
- Partitions of large raster objects (e.g. tiles of raster mosaics)
- Individual georeferenced raster objects (e.g. georeferenced aerial images)

These spatial access scenarios have largely different characteristics and requirements. The first and the second address the spatial access ***within*** a certain raster object, i.e. the intra-object access. The third scenario addresses the access ***to*** raster objects, which can be referred to as inter-object access.

Spatial Intra-Object Access

Spatial intra-object access addresses the spatial indexing of pixels or partitions of large raster objects (Figure 3–8, right). Spatial intra-object access should support an efficient processing of the following spatial queries:

⇨ Find the tile or pixel (of a given raster object) which contains the query point.

⇨ Find all tiles or pixels (of a given raster object) which intersect a query region.

Spatial Inter-Object Access

Spatial inter-object access deals with the efficient spatial access to all types of raster objects (Figure 3–8, left). This type of access does not require any knowledge of the internal structure of the raster object and is usually performed on a metadata basis. Spatial inter-object access should efficiently support the following generic queries:

⇨ Find all raster objects (e.g. orthoimages, raster mosaics) containing the query point.

⇨ Find all raster objects intersecting a query region.

Primary versus Secondary Indexing

Another important distinction, which is regularly used in database sciences, is made between primary and secondary indexing. Spatial access to individual raster cells or pixels is typically based on ***primary indexing***. Primary indexing or access structures directly store the actual objects as part of a spatial data structure. The space orderings presented in Section 3.4.2 and the raster encoding scheme presented in (Pajarola and Widmayer, 1996) are typical examples for this type of spatial indexing. In indirect or ***secondary indexing***, the access structure holds references to the storage location of each object together with information on the object characteristics (e.g. location and shape). Secondary indexing techniques are particularly useful for efficient spatial access to partitions of raster objects.

These investigations and the developed spatial access concepts will be presented in Chapter 5 (Investigations and Developed Concepts).

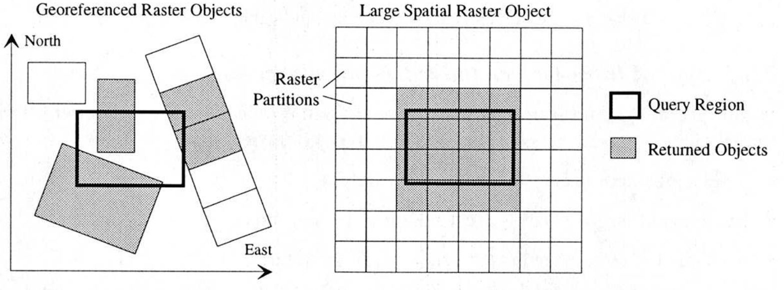


Figure 3–8: Spatial region queries run against georeferenced raster objects (⇨ inter-object access) (left) and against a large, partitioned raster object (⇨ intra-object access) (right).

3.7 File- versus DBMS-based Raster Data Management

Conceptually, raster data can be managed with file-based or DBMS-based concepts. In fact, many of the existing raster data management solutions, such as in typical digital photogrammetric systems, are exclusively file-based. However, with the establishment of spatial digital imaging techniques in a production environment these solutions are starting to reveal some of their limitations. One of the major draw-backs of file-based solutions is their lacking support for organising and querying large numbers of spatial raster objects, especially in a multi-user environment. A DBMS-integration of these objects was generally not attempted due to the limited support for large objects provided by traditional DBMS technologies. In view of recent and ongoing developments in the field of DBMS technology this situation is likely to change.

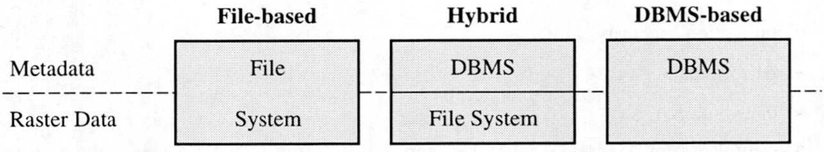


Figure 3–9: File-based, hybrid and fully DBMS-based raster data management concepts.

In the following sections the advantages and limitations of file-based, DBMS-based and hybrid raster data management concepts will be discussed and compared.

3.7.1 File-based Raster Data Management

In file-based or file-system-based data management concepts the applications are to a large degree responsible for the storage management (i.e. for writing and reading data to and from files on secondary or tertiary storage). File-based data management concepts are used in most traditional computing applications, including CAD- and digital imaging systems.

File-based concepts provide a great amount of freedom and flexibility. They allow the implementation of specialised data and access structures which are ideally adapted to the specific requirements of an application. File-based concepts are particularly well suited for the implementation of spatial data structures (e.g. r-trees or gridfiles).

Despite the apparent suitability for handling spatial data, file based data management concepts also have some serious limitations. For example, exclusively file-based applications cannot benefit from the standard database functionality described in the following section. Yet, features such as concurrent multi-user access, integrity, security, and querying are key requirements in modern (spatial) data management systems. In file-based systems they often have to be implemented separately at considerable cost.

3.7.2 DBMS-based Raster Data Management

DBMSs were originally designed and used for managing large numbers of simple objects. Support for large objects (e.g. spatial raster data objects) and for complex objects (e.g. topologically structured vector data) was only added recently.

The main benefits of DBMS-based data management solutions are gained from a series of standard features and services which any modern DBMS should provide, irrespective of its data model or query language. These standard capabilities can be summarised as follows (Stonebraker and Moore, 1996), (Koshafian and Baker, 1995):

- persistence
- transactions
- concurrency control
- crash recovery
- querying
- views
- protection / security
- performance
- distribution

Stonebraker and Moore (Stonebraker and Moore, 1996) sum up this collection of features by a single requirement: ***the ability of a DBMS to scale to large numbers of users and large amounts of data***.

In addition to these 'standard' features modern DBMSs have some characteristics which either particularly support or particularly limit the management of raster data. The main supporting and limiting features and characteristics are summarised below.

Supporting Features

- ***Parallelisation:*** One of the main problems with raster data is the volume of data to be accessed and transferred. Therefore, ***performance*** is a key issue in raster data management. Modern DBMSs achieve performance by supporting various hardware and software based parallelisation techniques and by supporting most high-performance hardware platforms. The supported parallelisation techniques include multi-processor systems, multi-workstation clusters and parallel secondary storage management techniques such as RAID (Redundant Arrays of Independent Disks).

- ***Distribution*** of data and operations is another key aspect in raster data management. The distribution of operations enables both client- and server-side activation of programs. Server-side program activation allows the execution of operations in the address space of the DBMS server. This avoids the expensive transfer of long function arguments, e.g. entire raster images, from server to client which itself results in a significant reduction of network traffic. Persistent Stored Modules (PSM) or stored procedures are typical examples of concepts supporting server-side program execution.

Limiting Features

- ***Lack of Data Modelling Power and Standardisation:*** Certain data models (e.g. relational) and the corresponding database technology provide a limited number of data types and operations. Other models offer a richer data modelling capability (e.g. object-oriented, object-relational) which would be more suitable for spatial raster data management. However, this increased modelling power is currently offset by the lack of standardisation in the fields of data manipulation and query languages.

- ***(Rollback) Overhead:*** The fundamental database concepts 'transactions' (rollback segments, locking, etc.) and 'recovery' introduce an overhead on certain database operations (e.g. update, delete). In order to ascertain consistency within a database system, a copy of the data to be updated or deleted is written to so-called rollback segments or redo logfiles. In the case of raster data where data throughput is one of the determining factors this additional overhead could have severe effects. It is therefore important to have a certain amount of control over this operation up to the level of disabling it for certain types of data and operations. For the affected data this would obviously involve sacrificing the aspects of integrity and recovery.
- ***2-Phase-Copy:*** The data flow in and out of current DBMSs is based on a 2-phase-copy approach: server storage ⇔ server memory and server memory ⇔ client memory. A significant gain in performance could be achieved, if unstructured data such as raster data could be transferred directly from client memory to the database server storage system and vice versa. This will require a better DBMS integration of the advanced tertiary and hierarchical storage technologies which will be discussed in Section 3.9. The direct transfer of unstructured data is the subject of ongoing database research (Relly et al., 1997).

3.7.3 Hybrid Solutions

Hybrid data management solutions usually differentiate between 'standard' data and 'specialised' data. Standard data or data which can be represented by means of standard data types is typically managed using a DBMS. Specialised data such as spatial data or unstructured data is often managed using proprietary file-based solutions. Logical links or references between the two different data spaces are maintained using common object identifiers (OIDs).

This approach reflects the underlying data management concept of ***most existing GIS solutions***. These hybrid spatial data management solutions use file-based spatial data structures such as gridfiles and r-trees for the spatial information and a typically relational database engine for the non-spatial or attribute data. A typical representative for this type of GIS architecture is the ARC/INFO™ system of ESRI Inc.. The ARC in the system name stands for the specialised geometric and topological information and INFO for the standard attribute data.

Hybrid solutions represent the current status among the existing (spatial) raster data management systems. Most of these systems use the combination of a relational DBMS for managing the metadata and a file-based solution for storing and managing the actual raster data (e.g. the HuS raster map management system by Hartmann & Sältzer (Hartmann, 1996)). Even in future raster management systems supporting DBMS-internal raster data storage, hybrid solutions will keep playing an important role. Namely, because they enable the integration of advanced mass storage technologies which might not directly be supported by the DBMS (see Sections 3.9 and 5.2.2).

3.8 DATA MODELS AND RASTER DATA SUPPORT

Over the last two decades several new data models and corresponding database techniques have evolved. They enabled a considerable extension of the traditional scope of database management systems towards complex data (e.g. spatial data) and unstructured data (e.g. imagery and video). However, 'newer' does not automatically mean 'better in every respect' for all types of data and applications. As shown by (Stonebraker and Moore, 1996) new functionality and increased sophistication usually have to be traded in for reduced performance and security.

The following sections will introduce today's three major data models and corresponding database technologies: relational, object-oriented and object-relational. Each of these models is analysed with respect to features supporting or limiting the management of (very) large spatial raster objects. The assessment characteristics used include issues such as supported data types and operations, spatial access, query languages, programming interfaces, standardisation, portability and security.

3.8.1 Relational Model and DBMS Technology

Relational Model

The relational data model is documented in countless publications. Standard publications include (Elmasri and Navathe, 1994) and (Lang and Lockemann, 1995). The use of the relational model and the corresponding DBMS technology for the management of spatial data is addressed in (Worboys, 1995) and (Singer, 1993).

The Relational Model (RM) was originally introduced in the early seventies by E.F. Codd. Since then it has undergone several revisions which are nicely illustrated by Camps in (Camps, 1996). This publication is an answer to 'The Third Manifesto' (Darwen and Date, 1995) which claims that the RM would be capable of supporting object-oriented features, namely user-defined data types and inheritance, if it were properly used and implemented. Camps, however, argues that the main problem is not caused by an incorrect interpretation of the RM but much rather by its requirement for the first normal form (1NF). The 1NF specifies that domains or attributes must consist of atomic, non-decomposable, values and as such is the basis for the simplicity of the original model – and maybe also its success. However, the 1NF concept was fiercely debated when the RM was introduced and many research projects abandoned the 1NF in the pursuit to integrate complex data. The resulting extended relational model is known as NF2 (Non First Normal Form) or Nested-Relational (Lang and Lockemann, 1995).

Relational DBMS Technology (RDBMS)

Relational DBMSs, which are mostly based on the 'standard' Relational Model, represent the main-stream of today's databases. They focus on transaction processing involving large numbers of relatively simple, concurrent queries on large numbers of relatively simple objects. Relational DBMS are commonly associated with the relational Structured Query Language (SQL) and are therefore often referred to as *SQL databases* (Stonebraker and Moore, 1996).

Raster Data Support and Application Interfacing

The SQL standard, like most other standards in the information technology, tends to lag behind the proprietary state-of-the-art technology available on the market. However, the different versions of SQL nicely reflect the evolution of the features supporting raster data. Pre SQL-92 versions supported fixed-length data types only, which is equivalent to constant column widths. It has been shown that fixed-length types can be used to store certain types of raster data, for example by using quadtrees, run-length or block encoding (Cheiney and Kerhervé, 1990), (Persson and Jungert, 1992), (Bill, 1996a). However, the fixed-length concept is not suitable for storing large raster objects with high attribute resolution. SQL-92 added the two variable length data types BIT VARYING and CHARACTER VARYING in the DDL (Data Definition Language) part of the standard. However, the standard does not specify the maximum length of these variable-length types and it does not include any specifications concerning their manipulation (retrieval and update) (Koshafian and Baker, 1995). The evolving SQL3 standard will introduce the standard or 'built-in' data types BLOB and CLOB for storing large binary or character objects and it will also specify a limited set of operations for these new types (Date and Darwen, 1997), (Demuth and Bruns, 1994).

The SQL standard defines four different binding styles with which applications can be interfaced with a RDBMS. These four binding styles are: module, embedded SQL, CLI (Call-Level Interface) and direct SQL (Date and Darwen, 1997). DDL operations for creating, modifying and dropping database tables with BLOB attributes are supported in SQL syntax through all these interfaces. DML (Data Manipulation Language) operations, however, are typically restricted to manufacturer-specific extensions of the Call-Level Interface. This means that BLOBs cannot be inserted, updated or selected by means of the current SQL-92 syntax.

3.8.2 Object-Oriented Model and DBMS Technology

Object Orientation

The central idea behind object orientation (OO) is to represent the real-world as closely (and intuitively) as possible. The origins of object orientation can be traced back to some of the early graphical user interface developments, for example, at Xerox PARC. Today, OO concepts are used in almost every domain of computer sciences, such as object-oriented programming (OOP), object-oriented analysis and design (OOA, OOD) and with a growing importance in object-oriented (OODBMS) and object-relational database management systems (ORDBMS) (see next Section).

Koshafian and Baker (Koshafian and Baker, 1995) define object orientation as:

Object Orientation = Abstract Data Typing + Inheritance + Object Identity

The key characteristics of each of these fundamental concepts can be described as follows:

- ***Abstract Data Types*** (ADTs) define the ***structure*** (appearance) and ***operations*** (behaviour, methods) of collections of similar objects. Object-oriented programming languages use the language construct ***class*** to define abstract data types. Conceptually, an abstract data type or a class represents the set of *all possible* objects with the corresponding structure and operations. However, in order to access and traverse the actual instances (objects) which exist at a given time, so-called ***class extensions*** or collection items (containers) are required. Class extensions are particularly important in object-oriented database management systems where large numbers of objects of the same type have to be processed (Koshafian and Baker, 1995).

- ***Inheritance*** is a powerful concept for achieving software ***reusability*** and ***extensibility***. It enables to construct new software modules on top of an existing hierarchy of modules. New classes can inherit both the ***behaviour*** (operations, methods) and the ***representation*** (attributes, etc.) from existing classes. In the standard case of ***single inheritance*** a subclass can inherit properties from exactly one superclass. In spatial information management it is often desirable for a class to inherit from more than one immediate parent. This case is referred to as ***multiple inheritance***.
- ***Object Identity*** is that property of an object which distinguishes it from all others. It is used to organise and access individual objects in object space - in contrast to the abstract data type and inheritance concepts which manage and organise classes of objects. Using object identity, objects can contain or refer to other objects. This is the basis for building ***complex objects***.

Object-orientation is often described by the ***object-message*** paradigm. A key concept of this paradigm is that each object responds to a class-specific collection of messages, which comprise the object's interface (Koshafian and Baker, 1995). A detailed discussion of different aspects of object orientation can be found in one of the following publications: Object-orientation in general (Lüscher and Straubinger, 1996) and (Breutmann and Burkhardt, 1992), OO and GIS (Fritsch and Anders, 1996), OO and DBMSs (Cattell et al., 1996) and (Stonebraker and Moore, 1996), OO and Multimedia (Koshafian and Baker, 1995).

Object-Oriented DBMS (OODBMS)

The key concepts of OODBMSs are very well summarised by the Golden Rules of the Object-Oriented Database System Manifesto (Atkinson et al., 1992). The Object Database Management Group (ODMG) furthermore defines an OODBMS (or ODBMS) as a DBMS that integrates database capabilities with object-oriented programming languages (Cattell et al., 1996). The ODMG also postulates that an OODBMS makes database objects appear as programming language objects.

This close integration with programming languages makes persistent DBMS operations almost as fast as operations of non-persistent object-oriented programming languages. An example supporting this claim is given in (Stonebraker and Moore, 1996). The use of low-level programming languages such as C++ also enables far more complex operations than would be possible with SQL. These factors make OODBMS well suited for complex objects and tasks.

One of the main shortcomings of current OODBMSs is the limited query language support. The Object Query Language (OQL) defined in ODMG-93 (Cattell et al., 1996), for example, is a superset of the query part of SQL supporting ***select*** statements only.

OQL does not support data manipulation. Data manipulation tasks, such as updates, have to be handled using a programming language specific Object Manipulation Language (OML) such as C++ OML.

Raster Data Support and Application Interfacing

The close integration of OODBMSs with programming languages enables powerful modelling of complex objects. The Object Definition Language (ODL) provides full support for the abstract data typing concepts outlined in Section 3.4.3, namely for user-defined types and operations.

The following example shows the interface definition of a simplified type RasterMap expressed in ODL:

```
interface RasterMap: RasterImage
(    extent rasterimages
     keys image_id): persistent
{
     attribute String image_id
     attribute Unsigned Short width
     attribute Unsigned Short height
     attribute Array<Octet> data_string
     relationship Set<MapSeries> belongs_to
     Short count_colours ()
}
```

This example models the state or properties of the type RasterMap by means of four attributes and one relationship. The behaviour of the type RasterMap is specified by the signature of the user-defined operation count_colours. It should be noted that in this example the actual raster data is stored as an array of octets (or bytes). Additional raster modelling examples on the basis of ODL can be found in (Baumann et al., 1997b). Unfortunately, the ODMG-93 standard (Cattell et al., 1996) is ambiguous concerning the collection literal *array*. In the ODMG object model section only fixed-length arrays are mentioned. In the C++OML section, however, a variable-length array class is listed, too. The latter would be required for modelling and managing variable size raster objects.

3.8.3 Object-Relational Model and DBMS Technology

Object-Relational Model

As to the knowledge of the author, there is currently no published formal definition of an Object-Relational Model (ORM) available. Based on (Stonebraker and Moore, 1996), (Jaedike, 1997) and (Pistor, 1994) it can be stated that the ORM extends the Relational Model (RM) by adding the concept of abstract data types and inheritance. This extension goes beyond that of the NF2 concept mentioned in 3.8.1.

Object-Relational DBMS (ORDBMS)

The ORDBMS technology is founded on the proven relational database concept. As a consequence ORDBMSs support traditional DBMS services, such as concurrency control, crash recovery, views, protection, replication, parallelism and data distribution (see Section 3.7.2). Stonebraker and Moore (Stonebraker and Moore, 1996) summarise the 4 main features of object-relational DBMSs as follows:

- support for base type extensions (in an SQL context)
- support for complex objects (in an SQL context)
- support for inheritance (in an SQL context)
- support for a production rule system

They also provide a list of additional features which a system must support in order to be fully object-relational in terms of base type extension. These features are:

- ***Dynamic linking*** enables user-defined functions to be loaded into the address space of a DBMS only when requested.
- ***Client or server activation*** provides the capability to activate and execute functions both on the server side and on the client side.
- ***Security*** is a critical issue in systems permitting server-side activation of user-defined functions. Such functions, if running in server address space, could accidentally or intentionally lead to a DBMS crash or could even destroy the entire database.
- ***User-defined access methods*** need to be supported in order to support efficient access to data of a user-defined type. In the case of user-defined spatial data types such as 'point' or 'line', suitable spatial access methods are required.
- ***Arbitrary-length types*** for storing large unstructured objects such as video sequences or raster data sets.

Raster Data Support and Application Interfacing

The object-relational technology is the first DBMS technology that was designed with a clear focus on unstructured and other specialised data. In (Stonebraker and Moore, 1996) (raster) graphics data and geographical information are listed among the prime target areas of ORDBMSs.

The support for arbitrary-length types provides the fundamental basis for storing very large spatial raster sets. The type extension concept enables the definition of different spatial raster data types and their specific operations. The client or server activation concept allows these functions to be executed either on the server or on the client. The server activation enables certain raster operations to be performed on the server, thus making the expensive transfer of entire raster objects to the client application unnecessary.

The emerging standard SQL-3 will incorporate features for defining types and for manipulating complex objects, which, for example, also includes the support for BLOB attributes. With SQL-3, not only DDL but also DML operations involving raster data will be supported. This support is likely to be available through all four binding mechanisms listed in Section 3.8.1. However, it is expected that the most recently defined CLI mechanism, which is closely related to the Microsoft ODBC standard, will be the interface of choice for most future database applications.

3.8.4 Summary

The main requirement for managing spatial raster data in a DBMS environment is the support of ***variable size, large objects***. Conceptually, the presented data models and the corresponding DBMS technologies now fulfil this requirement. The absence of a clear statement regarding large objects in the ODMG-93 standard (Cattell et al., 1996) leaves some questions about the priority of unstructured data in the development of OODBMSs.

The support of ***complex data*** is not such a firm requirement. The benefit of abstract data types, which support such complex objects, is primarily the availability of ***user-defined operations***. These become important when raster management solutions are to be integrated into GISs. Complex data and user-defined operations require the use of object-oriented or object-relational technology.

Another important aspect in the management of spatial raster data is the support of an efficient ***spatial data access***. This can be provided by several concepts ranging from space orderings, generic spatial data types and user-defined access methods. Each of the investigated database technologies supports at least one of these concepts.

Finally, ***performance*** is a determining factor in the efficient raster data management.

DBMS Technology / Criteria	REL	OR	OO
Base types (for raster data)	BLOB	BLOB	yes [3]
Type extension (ADTs)	no	yes	yes
User-defined operations	no	yes	yes
Spatial access methods	space orderings, generic spatial data types	space orderings, built-in (e.g. r-tree), user-defined	user-defined
Performance (parallelisation)	yes	yes	?

Table 3–5: A comparison of typical raster data management requirements and of the corresponding support provided by different DBMS technologies.

In addition to these key issues, the selection of an appropriate data model and database technology is mainly determined by the ***application type*** to be supported. The spectrum ranges from dedicated multi-user spatial raster data servers to single-user

[3] The object model of the ODMG-93 standard (Rev. 1.2) does not specify a base type for large objects or literals. However, the C++OML definition of the standard supports a dynamic array type.

photogrammetric workstations with their distinctly different characteristics and requirements. Despite some existing limitations, modern DBMSs now provide the basis for efficient spatial raster data management. An important commercial argument favouring DBMS-based solutions is that any functionality provided by a DBMS will considerably minimise the application development time and cost.

3.9 Raster Storage Management

The storage capacity and transfer bandwidth requirements of spatial raster management solutions make storage management an important issue. The storage management solutions of traditional DBMSs were originally developed for limited volumes of structured data. Large unstructured objects had not been foreseen and, as a consequence, cannot easily be accommodated. However, with the increasing number of DBMS applications using large unstructured objects (e.g. multi-media and video-on-demand databases) the demand for a better integration of DBMSs with modern mass storage technologies has risen sharply. This integration is primarily a low-level system and database design issue and as such it was not one of the primary objectives of these investigations. Nevertheless, a sound understanding of the technological status and trends is essential for the design and development of a DBMS-based spatial raster management solution. In the sections below, the following technologies and concepts will be discussed: storage media and storage systems, storage management, and mass storage support within DBMSs.

3.9.1 Storage Technologies

Efficient spatial raster management requires the combined use of several storage technologies. The main storage media and key technologies together with their particular benefits and drawbacks are described below. Special attention is given to the relatively new technologies of RAID (redundant arrays of independent disks) and optical jukeboxes as promising candidates in a spatial raster management environment.

Storage Media Overview

The different types of storage media are usually grouped into: primary memory, including RAM (Random Access Memory); secondary memory, traditionally consisting of magnetic disks; and tertiary memory, including removable media such as optical disks and tape cartridges. Another important distinction is made between in-line, on-line, near-line and off-line storage. In-line is used in (Ranade, 1991) for RAM-based storage devices such as Solid State Disks (SSDs). The on-line storage level applies to devices which are located in a drive unit and which are access-ready, e.g. magnetic disks or optical disks loaded in a disk drive. Near-line storage applies to devices which are not loaded but which are accessible to the system without human interaction. This applies to disks and cartridges which are stored on jukebox shelves. The fourth level of off-line storage applies to storage devices which are kept in external archives and libraries and which require human interaction in order to become accessible to the system. A comparison of the main storage media types is provided in Table 3–6.

Storage Media	Storage Level	Performance	Capacity	Cost	Persistence
RAM	in-line	very high	low	very high	volatile
magnetic disk	on-line	high	high	high	2-3 years
optical disk	on- & near-line	medium - low	very high	low	> 30 years
magnetic tape	on-, near- & off-line	medium - very low[4]	very high	low	< 10 years

Table 3–6: Comparison of main storage media types.

In a spatial raster management framework the following storage technologies are of particular interest (see Table 3–7):

RAM (Random Access Memory)

RAM with its volatile storage characteristics represents a special case of storage technology. Nevertheless, RAM plays a key role in high-performance DBMSs and in advanced storage solutions. The availability of RAM-based caches is one of the main performance factors for DBMS servers. Some servers, for example, even allow entire tables or databases to reside and be operated upon in primary memory, naturally with unbeatable performance. This approach is sometimes referred to as VLM (Very Large Memory) databases. RAM-based solutions such as Solid State Disks can be used to dramatically accelerate typical hard disk-based operations such as writing rollback information. RAM is also an integral part of advanced storage management technologies such as HSM (Hierarchical Storage Management) (see Section 3.9.2).

Magnetic Storage Systems

Magnetic Winchester disks or hard disks represent a mature storage technology which has experienced several decades of development. Magnetic disk technology, for example, has typically doubled in capacity and halved in price every two to three years (Koshafian and Baker, 1995). Modern magnetic storage systems have high capacities, good response times, a high read and write performance, a permanent accessibility and a reasonably good reliability. Current capacities of individual large server hard disks are in the excess of 20 GB. Entire magnetic storage subsystems with multiple disks and a total on-line storage capacity of 500 MB to 1 TB can now be fitted into a single desk-side storage cabinet. In contrast to these considerable storage capacities stand a relatively high price per MB storage and the static system characteristics which prevent magnetic devices to be easily added or removed from the system. Last but not least, the increase in performance has been considerably slower than the increase in storage capacity. As a result, the I/O bandwidth of single standard disks is too small for large-scale multi-user spatial raster management applications.

Due to their excellent response time, magnetic storage will remain the technology of choice for reading and writing large numbers of small data blocks and for storing limited amounts of structured data in DBMSs.

[4] if including random access to data blocks

RAID Systems

RAID (Redundant Arrays of Independent Disks) technologies address the insufficient performance and reliability of individual magnetic disks by grouping multiple disks into logical storage units and by addressing them in parallel mode. There exist seven levels of RAID, each directed at specific performance and fault tolerance aspects. RAID is typically implemented on the small computer interface (SCSI) which allows up to eight controllers and 56 logical devices to be interfaced to a 'daisy-chain'. For spatial raster data management, where high I/O performance is a primary requirement, RAID levels 0 and 5 are of particular interest.

In level 0 RAID, data is striped or declustered across multiple disk drives. Whenever a block of data is to be written to storage it is split into pieces and each piece is stored on a separate disk. Due to the high bandwidth of the RAID controller, the disk-write operation can be performed in parallel which dramatically improves the transfer rate. Reading a block of data works in the same way: the RAID controller issues a read instruction to the individual storage devices which are again serviced in parallel. RAID level 0 provides the best performance of all RAID schemes and can make full use of the available storage capacity. However, it provides no additional fault tolerance.

RAID level 5 is a more sophisticated scheme which distributes data blocks and parity information across all available disks. The additional parity information of RAID level 5 takes up approx. 10-20 % of the available storage capacity. The fault detection and recovery capability of RAID level 5 comes at the price of a slightly lower I/O performance.

Other RAID schemes such as levels 2, 3 and 4 have a limited write performance. This either due to the limitation that only a single I/O operation at a time can be performed (level 2) or due to the use of a single parity disk (level 3 & 4).

RAID technology is particularly suited for writing and reading large objects, whereby the reduced transfer time outweighs the additional overhead introduced by the technology itself. RAID systems are ideally suited for permanently storing spatial raster data which is frequently accessed and queried (e.g. raster mosaics) and for temporarily storing project-specific raster objects (e.g. raster images of a certain photogrammetric campaign).

Optical Storage Systems

Optical disk technologies comprise a bewildering variety of storage media and technologies, including WORM (Write-Once-Read-Many), MO (Magneto-Optical), CD-ROM, CD-R (Rewritable CD) and the emerging DVD (Digital Versatile Disk). The storage devices of all these technologies are removable and transportable. They have an expected archival life of over 30 years, far greater storage densities than magnetic disks and also a much lower price. The main shortcomings of optical storage technologies include moderate to low data transfer rates, a missing or still immature (re-)write capability on certain media, and a lack of integration into existing operating system file systems.

Optical disks are a cost-effective solution for storing large amounts of data. However, the manual handling and archiving of large numbers of optical disks can become an unacceptable burden. Keeping optical disks in off-line archives also prevents their integration into major information systems.

Magnetic Tape Systems

Magnetic tape technologies have large storage capacities and can be read or written with up to several MB/s. However, magnetic tapes are sequential-access oriented and have the disadvantage of a very poor random access performance. As such they also typically lack the feature of updating individual files.

The use of magnetic tape systems in a spatial raster management environment largely depends on the targeted type of application (see Section 4.2). Magnetic tapes play an important role in remote sensing archives and production systems. In query oriented applications such as spatial information servers, magnetic tapes are primarily used as back-up and data transfer media but not for on-line querying and processing.

Storage Technology	Advantages	Drawbacks
Magnetic Storage Systems	• *permanently available* • **good response time and performance for small object I/O**	• *insufficient performance* for large object I/O • large, but limited and static capacity • expensive
RAID Storage Systems	• permanently available • **high performance for large object I/O**	• system overhead for small object I/O • large, but limited and static capacity • expensive
Optical Storage Systems	• **high capacity** • **supports random access** • **archival life** • low cost	• low performance • requires special file systems
Tape Storage Systems	• high capacity • archival life • low cost	• **very poor random access performance** • requires special file systems
Jukebox Storage Systems	• **very high capacity in near-line mode** • archival life	• low performance • poor response time • low to medium cost

Table 3–7: Comparison of main storage technologies from a spatial raster management perspective (main aspects in bold letters).

Jukebox Storage Systems

Jukeboxes or robotic storage libraries allow a number of storage media or cartridges to be stored on shelves inside the jukebox cabinet. Robotic arms transport and load the removable media into the drive units, when requested by the jukebox controller. The main components of a jukebox storage system are: an import/export slot, robotic arm(s), storage shelves, (optical) drives and the jukebox controller (Ranade, 1992), (Koshafian and Baker, 1995). With the use of jukeboxes a large number of storage devices can be kept in near-line storage mode. Jukeboxes can contain optical storage devices, magnetic tape cartridges or a combination of the two. Existing optical jukeboxes, for example, can handle up to several thousand disks which is equivalent to a storage capacity of several TeraBytes.

Optical jukeboxes are ideal for storing large volumes of spatial raster data, particularly large raster objects (e.g. raw raster imagery) which are not permanently accessed or queried.

3.9.2 Storage Management Concepts

Storage management techniques ensure the optimal use of the storage technologies presented above. The following selection of storage management concepts are particularly important for managing data volumes which exceed the capacities of magnetic disk storage.

Clustering and Declustering

Clustering and declustering are two contradicting but important issues in the handling and storage of large objects. For example, declustering, i.e. splitting and distributing objects over multiple storage locations, is used in RAID techniques as a means to enhance the data transfer rate (Patel et al., 1997). In the RAID scenario the application has no control over the storage location of a specific object. This is perfectly acceptable with re-writable, non-removable magnetic disk storage but it is not desirable with optical jukebox storage. In optical jukeboxes clustering, rather than declustering, of objects onto individual storage devices is important for the following reasons. First, time-consuming storage device changes should be kept to an absolute minimum, especially in the case where multiple user-request are competing for a limited number of drives. Second, in applications such as spatial raster management, the application should be able to control the clustering and location of objects at the optical device level. This alone allows individual projects to be exported and transferred to off-line storage without a preceding time-consuming re-organisation of the data.

Volume Management

Managing large collections of optical and tape devices in jukeboxes and stand-alone drives is a task which is normally not yet handled by today's operating system (OS) file systems. Volume management techniques can treat a group of drives and jukeboxes as a single, hierarchical file system and map it into the standard OS file system. A key volume management feature is ***directory caching***. It is used to keep track of the storage state (e.g. near-line), location and contents of all storage devices registered within the system. Directory caching, for example, allows metadata queries to be performed on all storage volumes, including those in off-line storage. Volume management and specialised file systems for tertiary storage are further discussed in (Koshafian and Baker, 1995) and (Yu and DeWitt, 1997).

Hierarchical Storage Management

Hierarchical Storage Management (HSM) enables the integrated management of a hierarchy of storage media (Koshafian and Baker, 1995). HSM spans the storage levels RAM, magnetic drives (often with RAID), optical or tape jukeboxes with on- and near-line storage and optical or tape off-line storage (Ranade, 1991). The goal is to optimally utilise the resources available on each storage level in order to achieve the best possible performance. The key concept of HSM is ***data migration*** on the basis of usage patterns. High-water marks (HWM) and low-water marks (LWM) are used to control storage space usage on the different levels. Another concept which is of particular importance to applications such as spatial raster management is application- or DBMS-controlled **staging**. It allows series of raster objects to be promoted to and locked at higher hierarchy levels. Most HSM implementations are based on the IEEE Mass Storage System Reference Model (MSSRM) (Cole and Jones, 1995), (Koshafian and Baker, 1995).

HSM combines the high performance of RAID technologies with the very high storage capacities of (optical) jukebox storage systems. HSM is therefore the ideal storage management solution for spatial raster management systems with large volumes of data and with the requirement for high data transfer rates.

3.9.3 Mass Storage System Support in DBMSs

The integration of high-performance secondary and high-capacity tertiary storage subsystems into DBMSs is a key research topic addressed in several of the projects presented in Section 4.1.1 (Gennart et al., 1996), (Relly et al., 1997), (Baumann, 1995) and (Patel et al., 1997). Relly and co-authors (Relly et al., 1997), for example, postulate the two following main storage management system requirements for handling spatial raster data in DBMSs:

- good integration of secondary and tertiary storage management
- efficient partial raster object access (on tertiary storage) and effective caching strategies

The different DBMS-based architectures for storing large objects can be divided into the following three categories: DBMS-controlled secondary storage management, application-controlled tertiary storage management, and DBMS-controlled tertiary storage management (see Table 3–8).

DBMS-controlled Secondary Storage Management

In this storage architecture, all data is stored and managed on permanently available, secondary magnetic disk storage. Storage management is fully controlled by the DBMS which allows all data to be included in the transaction control mechanism. This storage option represents the traditional storage management solution in commercial DBMSs. More recent additions to this storage option are the support and integration of high-performance RAID subsystems.

Application-controlled Tertiary Storage Management

In this scenario, mass storage subsystems such as optical jukeboxes and HSM systems are not supported directly. Instead of directly managing the actual object on tertiary storage, a reference to the object is stored in the DBMS. The actual object is managed through the storage subsystem. Typically this DBMS-external storage management is under application control. This has the advantage of an improved flexibility with respect to controlling the storage location. However, this storage option has several drawbacks. For example, insertion and selection of externally stored objects usually has to be implemented within the application and is not available in the SQL(-3) context of the DBMS. Externally stored objects are also excluded from the transaction control of the DBMS.

DBMS-controlled Tertiary Storage Management

This storage option directly supports tertiary mass storage subsystems and as such represents the most advanced DBMS-based storage management concept. In this storage scenario, the DBMS knows about the performance characteristics of the different storage media. This information can be used to optimise the overall system performance. DBMS-controlled tertiary storage management in combination with HSM technologies enables the efficient management of TeraByte size databases in on-line and near-line storage mode. The integration of volume management concepts even allows off-line data to be managed.

DBMS-based Storage Scenarios	Supported Storage Technologies	Advantages	Drawbacks
DBMS-controlled Secondary Storage Management	• magnetic storage	• good performance on structured data (small objects)	• limited capacity
	• RAID	• good performance on large objects	• limited capacity
Application-controlled Tertiary Storage Management	• optical and tape jukeboxes • HSM	• very high storage capacity • flexible storage management	• proprietary integration of tertiary storage object I/O • objects not under transaction control
DBMS-controlled Tertiary Storage Management	• optical and tape jukeboxes • HSM	• very high storage capacity • integration in a SQL context	• limited application-controlled tertiary storage management and clustering

Table 3–8: Comparison of different scenarios for DBMS-based high-performance and high-capacity storage support.

4

SYSTEMS AND APPLICATIONS

This chapter sets the scene for the investigations and the prototype development described in Chapters 5 and 6. It puts the present investigations in perspective with other research and development activities and it discusses the spatial raster management requirements and shortcomings in various application domains.

In the first section, research projects with a partial or exclusive focus on spatial raster management are presented. It is shown that these projects originate from different scientific fields (e.g. GIS, databases, storage systems) and that they cover a broad range of different research issues. The projects have been summarised under the title 'spatial raster management engines' which reflects their common goal towards a generic, largely application-independent spatial raster management solution.

In the following section, an overview of application domains with a requirement for spatial raster management is provided. Subsequently, the specific characteristics and requirements within each domain are analysed by looking at selected application types. Examples of projects and services with a particular demand for spatial raster management are provided together with examples of projects or systems which address application-specific aspects.

4.1 SPATIAL RASTER MANAGEMENT ENGINES

Spatial raster management engines provide the core functionality for storing, handling, querying and manipulating spatial raster data either as stand-alone systems or as part of major spatial information systems.

4.1.1 Research Projects

The following research projects either partially or exclusively address issues related to the management of ***spatial*** raster data. Figure 4–1 illustrates the primary research direction and the scope of each of these projects.

- ***GrIdS*** (Geospatial Raster Image Database management System) is a prototype system designed and developed as part of the present research project. The objectives of the investigations and the characteristics of the GrIdS prototype system will be discussed in detail in Chapters 5 and 6.
- ***CONCERT*** of the Database Research Group at the ETH Zürich is a long-term prototype development effort supporting the group's research activities in open storage management (Blott et al., 1996). Recent investigations include physical database design issues, such as the externalisation of database functionality (Relly et al., 1997), which support the efficient management of raster data in DBMS-external repositories and archives.

The expertise and experience of the GrIdS and CONCERT projects contribute to the recently initiated interdisciplinary research project ***Raster-GIS*** at the ETH Zürich (Carosio et al., 1996). It is a joint effort of the GIS, Photogrammetry and Database groups towards the integration of raster images and 3-D objects in geodatabases.

- ***GigaServer*** is a project at the Peripheral Systems Laboratory, Ecole Polytechnique Fédérale in Lausanne (EPFL) (Gennart et al., 1996). The project focuses on parallel server and storage issues and on technologies for large raster images. It has also led to the development of the GigaView multiprocessor multidisk architecture. The original GigaView architecture has become part of the commercial Galore™ product line (Farine, 1995). The GigaView/Galore storage sub-system is accessible from host workstations through a standard SCSI interface and a set of API calls. Thus, it could relatively easily be integrated in DBMS- or file-based raster management solutions.
- ***RasDaMan*** (Raster Data Management in Databases) is an international research project sponsored by the European Commission. It focuses on database management system (DBMS) support for raster data of arbitrary size and dimension (Baumann, 1995), (Baumann et al., 1997a). The key issues addressed are the data-independent representation, storage and querying of Multidimensional Discrete Data (MDD) and raster data in particular. RasDaMan allows MDD or raster types to be defined and queried using RasQL (Raster Query Language) (Baumann et al., 1997b) which is embedded in the object-oriented DBMS standard ODMG (Cattell et al., 1996) (see also Section 3.8.2).

  RasDaMan does not particularly focus on spatial raster data. It rather provides a generic mechanism for efficiently managing collections of large individual raster data sets. In contrast to GrIdS, RasDaMan does not address special spatial raster features such as the management of image mosaics.
- ***Paradise*** is a major research project of the Computer Science Department at the University of Wisconsin with the objective of designing and implementing a parallel database system for handling GIS type applications (DeWitt et al., 1994). Paradise is based on an object-relational data model and provides base types for image data and

spatial vector data. The issue of efficiently managing large numbers of large – but individual – raster objects is addressed in Paradise by means of a tiling approach (Patel et al., 1997) together with a parallel client-server system architecture and by the integration of tertiary storage management (Yu and DeWitt, 1997).

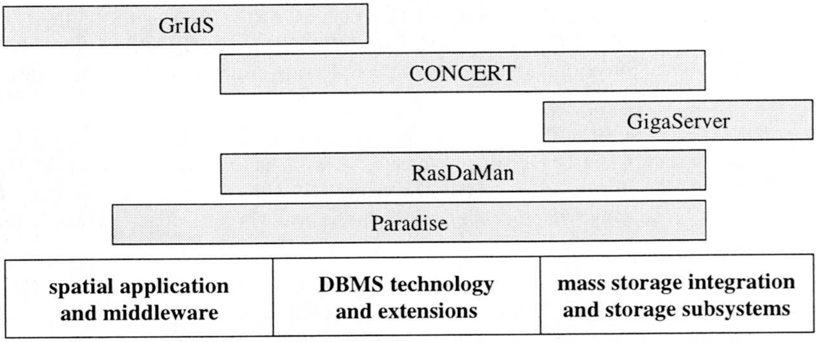


Figure 4–1: Comparison of topics addressed by a number of research projects which either partially or exclusively focus on spatial raster management.

4.1.2 Commercial Systems

To date, there appear to be no commercial DBMS-based spatial raster management engines available on the market. However, recent developments in two related database fields give rise to hope that this situation might change fairly soon.

On the one hand, there are a number of ***imaging and multimedia database engines*** (e.g. Oracle Media Server or Plexus XDP) (Koshafian and Baker, 1995) which provide powerful solutions for the management of non-spatial raster image data. These systems are designed and optimised for large numbers of (individual) multi-media objects and for multi-media streams. They also support important features such as tertiary and hierarchical storage subsystem integration and the concurrent access by a large number of users. Unfortunately, these systems lack a support for essential spatial aspects such as georeferencing, spatial partitioning, spatial access and querying or mosaic management.

On the other hand, there are a number of commercial ***spatial database engines*** such as Oracle's Spatial Data Option (Oracle, 1995b), ESRI's Spatial Database Engine (ESRI, 1996) and the Informix Spatial Data Blade. These systems cover many of the spatial aspects but limit the spatial support to vector and point data. Raster data support, if provided at all, is typically limited to bilevel imagery.

Despite the absence of 'ready-to-go' spatial raster management systems, the development tools and database services offered by the above-mentioned systems actually enable the implementation of spatial raster management solutions. An implementation on the basis of such standard state-of-the-art DBMS tools will be discussed in Chapter 6.

4.1.3 Raster Management Functionality in Raster GISs ?

Raster data support is a common feature in most of today's GISs. However, among the main GIS tasks (acquisition, management, processing, analysis and visualisation) raster data management is typically one of the weakest links in the chain. This statement applies to the raster modules of major GISs, such as ARC GRID of ESRI, IRAS of Intergraph, GenaCell of GenaSys, SPANS of TYDAC Inc. or GRASS of the USGS (U.S. Geological Survey). It also applies to the following examples of specialised raster GISs:

- ***IDRISI*** is a project of the Clark Labs for Cartographic Technology and Geographic Analysis at the Clark University in Worcester, Massachusetts, USA (Eastman, 1997). The ***IDRISI*** software is an inexpensive raster GIS with a very large user base. It offers a broad range of processing and analysis features but supports limited data volumes only.
- ***PCRaster*** is a system and research project of the Netherlands Centre for Geo-ecological Research (ICG), Department of Physical Geography, Faculty of Geographic Sciences, Utrecht University (ICG, 1997). The particular strengths of PCRaster are in environmental and geostatistical modelling, prediction and simulation.

These raster modules and raster GISs provide very powerful and sophisticated raster processing and analysis capabilities which stand in contrast to the relatively simple raster management solutions. These solutions are typically based on the file-based handling of individual raster data sets which are often restricted in dimension and size.

4.2 APPLICATIONS

4.2.1 Application Domains - Overview

Some of the main application domains of spatial raster management are shown in Figure 4–2. Despite the common need for a raster data management capability, the nature of the raster data to be handled and the management functionality to be provided varies greatly among these application domains. Applications in the photogrammetric and remote sensing domains, on the one hand, primarily deal with ***sensor-specific 'raw'*** raster data. Raster data used in the GIS and cartography domains, on the other hand, are typically of a ***generic*** nature, i.e. they are no longer sensor-specific. A typical requirement in these application domains is the combination of raster data with other spatial and non-spatial data types. Finally, applications in the spatial clearinghouse and digital library domains are primarily ***metadata-based*** and, as such, require less sophisticated spatial raster management support.

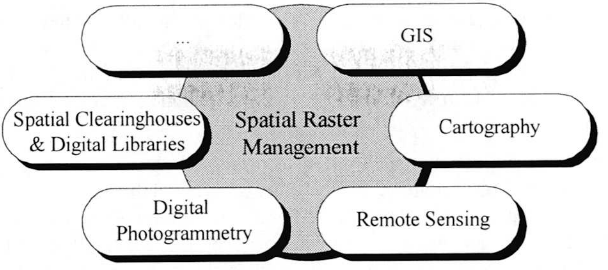


Figure 4–2: Application domains requiring spatial raster management.

In the following sections the respective characteristics and requirements in each of these main domains will be discussed by looking at some typical application types or tasks. Specific services and systems will be presented where appropriate.

4.2.2 GIS

The range of GIS applications using raster data is already very broad and with the emergence of new types of sensors it is likely to grow even further. A good overview of vector- and raster-based GIS applications is provided in (Bill, 1996b). An impressive selection of GIS applications which are primarily based on remotely sensed raster imagery is presented in (Morain and Baros, 1996). The issues discussed below represent a selection of some of the typical raster-based GIS application types which require an efficient spatial raster management support.

Orthoimage and Orthomosaic Databases

Digital orthoimagery derived from aerial photographs or from medium- to high-resolution remote sensing data is starting to become one of the prime data sources for almost any GIS application. The efficient handling of large numbers of orthoimages and the creation and management of very large image mosaics will thus become a key requirement in any major GIS.

Existing collections of orthoimages and the corresponding products and services include:

- ***Digital Orthophoto Quadrangles*** (DOQs) are a standard product of the U.S. Geological Survey consisting of 1-meter resolution, 8-bit greylevel orthoimages derived from aerial photographs (USGS, 1997). Each DOQ covers an extent of 3.75 minutes in latitude and 3.75 minutes in longitude is up to 55 MB in size. DOQs are available to the public at nominal cost on-line through File Transfer Protocol (FTP) or off-line on CD-R. DOQs are provided with overlaps at the edges in order to enable tonal matching of adjacent images.
- ***SwissPhoto*** is a similar but commercial product of SwissPhoto Vermessungen AG in Switzerland (Kersten and O'Sullivan, 1996). SwissPhoto consists of a collection of orthoimages (e.g.) and DTMs covering the area of Switzerland. The orthoimages are derived from aerial photographs. They have a ground sampling distance of 0.75 m

and have the property of large tiles. The tiles are tonally matched as part of the production process and are therefore provided without overlaps. Examples of SwissPhoto images include those provided for the system tests in Chapter 7: 3000x3000 pixel 24-bit RGB images covering an area of 4 km x 4 km with a size of 46 MB per image.

Most orthoimage collections, such as those presented above, currently still store and access orthoimages as individual raster objects. The relative ease of obtaining such orthoimages stands in contrast to the difficulty of combining or mosaicking multiple raster objects. This process is required whenever the area of interest is not fully contained within a single orthoimage. The mosaicking operation, whether it includes tonal matching or not, is a relatively complex and demanding task requiring specialised software and skills. The embedding of such orthoimage collections in an advanced spatial raster management solution would relieve users and client systems from these demanding tasks.

DTM Databases

It was pointed out earlier, that the management of regular matrix DTMs is largely equivalent to the management of raster images. Often, DTMs are provided on a 'per map sheet' basis which leaves the task of combining multiple data sets to the user system. Extra metadata required for managing individual DTMs or DTM mosaics includes information on the vertical datum, the height type (e.g. orthometric or ellipsoidal), the vertical offset (height of the 0-value) and scale.

Selected national DTM databases and services include:

- ***DEM*** (Digital Elevation Model) by the USGS
- ***Rimini*** and ***DHM25*** by the Swiss Federal Office of Topography

The following is an example of a dedicated system supporting such DTM services and databases:

- ***SCOP.DTM*** is a software package developed at the Institute of Photogrammetry and Remote Sensing, Vienna University of Technology (Hochstöger, 1996). It supports the DBMS-based management of country-wide DTMs not only as raster representation but also in the form of contour lines and spot heights.

Regional Planning and Census Databases

Regional planning and census databases often contain large raster data sets. These thematic rasters typically consist of large numbers of attributes covering a range of topics. Some major requirements for regional planning information systems are summarised in (Zimmermann and Sandoz, 1996). Among the postulated requirements with a particular significance on data management is a seamless operation allowing the selection of an arbitrary work area and the specification of a range of scales. Another requirement is the support for different administrative levels (community-county-state-nation) which, from a raster management perspective, could be treated as a special type of multi-resolution requirement. Other, more general, requirements concern standard GIS features such as raster-vector overlays and head-up digitising.

The following represents a selected example of a national census and planning database with a significant raster content:

- *GEOSTAT* is a spatial information service maintained by the Swiss Federal Office of Statistics (Zimmermann and Sandoz, 1996). It provides information on a broad range of topics in raster or vector format. A particularly important product of GEOSTAT is the so-called 'Hektar-Raster'. It consists of nation-wide thematic raster data at a 100m geometric resolution on topics such as land use, population and building density.

Spatial Information Servers

Government organisations and major companies have a strong requirement to provide their various departments with on-line access to spatial databases. Raster data in the form of raster maps, orthoimagery, DTMs and thematic rasters provide an ideal data source for many of these spatial information needs. As a consequence, a spatial information server which manages and provides this type of spatial information should be capable of handling all the above-mentioned raster-based GIS application types plus some of the cartographic aspects which will be discussed below. Spatial information servers with their priority on multi-user query and retrieval, in a potentially very heterogeneous and distributed environment, are ideal candidates for Intranet-solutions.

⇨ Spatial information servers represent a type of ***GIS application*** with a particularly strong requirement for ***efficient spatial raster management.***

4.2.3 Cartography

In the cartographic domain, map production and raster map databases are two application fields with a particular requirement for spatial raster management.

Map Production

In map production raster data primarily permitted a rapid transition from analogue to digital techniques. Beyond this simple transition, spatial raster management enables a seamless mode of operation, which liberates cartographers from the limitations of traditional map sheet boundaries. A seamless operation allows geometric and thematic discrepancies at map sheet boundaries to be resolved once and forever. It also removes the need for redundantly drawing and maintaining overlaps ('map collars') at the sheet boundaries. A seamless mode of operation furthermore provides a greater flexibility in defining new non-standard products and map sheets which, for example, cover specific areas of interest.

Raster Map Databases

Collections of scanned raster maps have been established by most mapping agencies. These collections were primarily established with the intention to economically capture the contents of the original map material in digital form. They are typically organised as databases holding individual map sheets or map sheet layers which are provided off-line, and in an increasing number of cases on-line, as individual raster files. It can be expected that raster map databases will play an increasingly important role, despite the current trend in digital cartography towards a vector-based mode of operation. One of the main reasons for this is that the raster representation allows an 'image' of the data to be provided or sold rather than the original data itself. In addition, the emerging watermarking techniques (Rink, 1997) are particularly well suited to control and protect the copyright on raster data.

A first and mandatory requirement for modern raster-based cartographic databases is a seamless appearance which, for example, allows arbitrary spatial extents to be selected. In cartography, a large part of the attribute information is traditionally represented by means of layers and ultimately colours. The possibility of selecting and combining certain thematic topics is a powerful and highly desirable feature in a cartographic raster management system.

Examples of raster map databases and the corresponding services with a demand for spatial raster management functionality include:

- ***Pixel Maps*** (Pixelkarten) of the Swiss Federal Office of Topography, a collection of scanned raster maps and map layers of the standard national topographic map series. Pixel Maps are available for all national map series from 1:25'000 through to 1:1'000'000 at a scanning resolution of 508 dpi (20 lines/mm). Standard raster map dimensions are 14'000 x 9'600 pixels which results in uncompressed sizes of 16 MB for individual thematic layers (packed bilevel data) and approx. 130 MB for colour combinations (8-bit colour-map).
- ***DRGs*** (Digital Raster Graphics) are scanned and georeferenced images of the U.S.Geological Survey (USGS) standard series topographic maps (1:24'000 to 1:250'000), including all map collar information. DRGs cover an area of 7.5' x 7.5'. They are scanned at 250 dpi and provided as 8-bit-colour compressed GeoTIFF files with sizes ranging from 5 to 15 MB.

The following systems were primarily developed for the specific tasks of archiving and retrieving raster maps:

- ***HuS Digitales Kartenarchiv*** by Hartmann & Sältzer GmbH (Hartmann, 1996) is a PC-based software package for handling individual raster maps. It combines DBMS-based metadata management (Microsoft Access) with file-based raster data management.
- The 'Rasterdatenverwaltungssystem of the Landesvermessungsamt Baden-Württemberg' (Graf, 1995) represents a traditional solution for managing raster maps within a mapping agency. It is an exclusively file-based solution in a UNIX environment primarily developed and operated on a script-basis.

4.2.4 Remote Sensing

In the remote sensing domain different spatial raster data management solutions are required to support the major tasks of data acquisition/production, archiving, cataloguing and processing. Depending on the organisation operating the remote sensor these tasks are either delegated to a number of distributed systems and sites (e.g. Distributed Active Archive Centers of the U.S. EOSDIS system) (Campell, 1996) or integrated into centralised solutions (e.g. Space Imaging) (Thurgood, 1997). The main characteristics and requirements of each of these tasks will be discussed below.

Data Acquisition and Production

The acquisition of remote sensing data is typically performed by a small number of highly sophisticated operation centres. These operation centres require tremendous data transfer, data storage and data processing capacities. With the existing medium- to low-resolution sensors (e.g. Landsat TM or NOAA/AVHRR) the production scenario consists of an automatic generation of standard products which are subsequently archived and catalogued. With the emerging high-resolution earth-observation sensors (e.g. Space Imaging or Earthwatch QuickBird) there is a paradigm shift towards the on-demand generation and delivery of a range of products (Fritz, 1996) and towards the archiving of raw data only (Thurgood, 1997). As part of the data acquisition process an abundance of specific metadata is acquired or generated. It includes information on: sensor type and number, track identifier and along-track position, sensor attitude, acquisition time, the spatial extent of the data set and cloud cover.

This metadata constitutes the key information for generating data catalogues and for querying and accessing remote sensing archives.

The example shown in Table 4–1 is based on the specifications for the Space Imaging system (Thurgood, 1997), (Space Imaging, 1997). It illustrates some of the requirements with regard to the type and amount of data acquired and produced by such a system. Raster management solutions supporting such data acquisition and production processes are typically custom-built according to the specifications of the specific remote sensing platform. As a consequence, these systems are usually not commercially available.

Geometric resolution	1 metre (panchromatic) 4 metres (multispectral)
Number of multispectral channels	4
Radiometric resolution	11 (bits per channel)
Swath width	11 km
Maximum download capacity	600 images / day (of 11 km x 11 km)
Uncompressed data volume (11x11 km image) - panchromatic (1-metre) - multispectral (4-metre) - fused pan and multi (1-metre)	 ≅ 158 Mbytes (11'000 x 11'000 pixels) [5] ≅ 40 Mbytes (2'750 x 2'750 pixels) ≅ 474 MB (11'000 x 11'000 pixels, 3 x 11 bits)
Raw data storage requirements - uncompressed - compressed, lossless (2 : 1)	 ≅ 200 MB (158 pan + 40 multi) ≅ 100 MB
Maximum amount of compressed raw data	60 GB / day

Table 4–1: Selected characteristics and figures of the Space Imaging high-resolution sensor expected to be launched in late 1997.

[5] Calculations based on contiguously packed bits, i.e. no Byte padding.

Data Archiving

Remote sensing data archives are typically operated by major national or international organisations. These archives are often associated or even integrated with the above-mentioned acquisition and production systems. However, decentralised archives focusing on certain regions (e.g. national archives) or on certain topics (e.g. snow and ice data) also exist. The primary data management issue in remote sensing data archives is the support for high-capacity, high-performance mass storage (see Sections 3.9.1and 3.9.2). Other important issues are the interfacing or integration with catalogue systems and with data ordering and delivery systems. Due to the bandwidth limitations of today's communication networks, remote sensing data is typically delivered to the end-users by means of portable storage media such as CD-ROM or tape.

Examples of different remote sensing archive services include:

- The ***Distributed Active Archive Centers*** (DAACs) (Campell, 1996) are part of the U.S. EOSDIS Program (Earth Observation System Data and Information System). There are currently 9 DAACs that are responsible for processing, archiving and distributing specific kinds of remote sensing data (e.g. "hydrology" or "land processes").
- ***Swiss National Remote Sensing Image Archive*** (RSIA) (Seidel and Therre, 1994), (Seidel and Therre, 1997). The RSIA is a joint project with the goal to establish a national archive using the infrastructure of the Swiss Centre for Scientific Computing (CSCS) in Manno. The benefits of such a central archive holding recent and historical remote sensing data pertaining to a certain area are undisputed. However, the use of a scientific super-computer centre for primarily archival purposes leaves some questions concerning the cost efficiency claimed in (Seidel and Therre, 1994).

Sensor-specific archive systems which are closely integrated with the respective acquisition systems and not available on the commercial market. However, in view of the wide-spread future use of high-resolution remote sensing data more and more users will be confronted with a need to archive their operational and historical data. An example of a commercially available product which addresses this market segment is:

- ***SeaShark*** developed by the VEGA Group PLC on behalf of the European Space Research Institute. SeaShark initially supports two different sensors (SeaWIFS and AVHRR) and performs the tasks of archiving, processing, cataloguing and distributing data from these sensors.

Cataloguing

Data catalogues provide the main means for querying, accessing and ordering data sets from the corresponding remote sensing archives. These catalogues primarily consist of metadata and, increasingly often, quick-looks of the actual data sets. This information allows these catalogues to be browsed and queried using spatial and thematic predicates. Specialised query functionality includes features such as the determination of stereo pairs or 'seasonal' queries. Today, all unclassified earth observation programs provide catalogues which can be accessed on-line (e.g. via WWW or Telnet) or off-line (e.g. on CD-ROM). Being ***primarily metadata-based***, cataloguing itself involves and requires only limited spatial raster data management functionality. Examples of two different types of remote sensing catalogue systems include:

- *DALI*, the central data catalogue of Spot Image, operated at their Tolouse premises (Spot Image, 1997). It contains metadata for all SPOT scenes acquired since February 1986 and quick-looks of scenes received since 1991. The DALI catalogue can be accessed on-line through a number of network services (e.g. WWW) and off-line using programs such as DIVA.
- *DIVA* is a software tool provided by Spot Image (Spot Image, 1997). It enables on-line queries of the central Spot Image catalogue DALI and off-line queries using the SPOT scene catalogue CD-ROM.
- *ISIS* (Intelligent Satellite data Information System) is the interface for accessing data archived at the German Remote Sensing Data Center (DFD), a division of the German Aerospace Research Establishment (DLR) (Lotz-Iwen et al., 1995), (Liebig, 1996). Data search and access are supported through different interfaces (ASCII, graphical and WWW). Features of ISIS include on-line catalogue retrieval, quicklooks, a geographical names database and a map browser with dataset footprints.

Data Processing and Interpretation

From a data management perspective the tasks of processing remotely sensed and photogrammetric data have very similar characteristics and requirements. Some of them are listed in section 4.2.5 below. The requirements specific to the remote sensing domain include the support for multispectral data with potentially large numbers of channels and the support for a range of domain-specific metadata such as sensor model information.

In photogrammetric and remote sensing system development the common ground between the two domains has led to an increasing functional convergence. Many traditional remote sensing image processing systems are starting to support typical photogrammetric features and vice versa. With the emergence of high-resolution remote sensors, the boundary between the two domains is likely to get even fuzzier.

4.2.5 Digital Photogrammetry

In the first ten years of civilian digital photogrammetric system development, the primary goal was to reach a level of functionality and performance which would allow analytical systems to be replaced by digital solutions. Today, most of the commercial digital systems presented in (Grün, 1996) have achieved that initial goal. However, the increasing use of digital photogrammetric stations in an operational environment has started to reveal some serious deficiencies concerning project and data management functionality in general, and raster management functionality in particular (Walker and Petrie, 1996), (Grün, 1996).

The raster management solutions in today's digital photogrammetric stations are almost exclusively file-based with proprietary formats, compression techniques and multi-resolution schemes. The abundance of metadata associated with photogrammetric data and processes is also primarily managed by means of control or header files. As a consequence, there is little to no query functionality available, both on a metadata- and on a content-based level. The tasks of organising and handling raster data are almost exclusively left to the operators.

Future raster data management solutions for digital photogrammetric systems will have to address the following requirements:

- tight integration with the entire photogrammetric workflow (including the steps of planning, acquisition, image measurement and the generation of DTMs and orthoimages)
- rapid access to multiple (small) image subsets (e.g. for control-point and multi-image measurements)
- efficient multi-resolution support
- high performance, supporting stereo-roaming in real-time
- transparent support for high-performance, high-capacity mass storage sub-systems

4.2.6 Other Applications

The examples of ***geospatial clearinghouses and data stores*** as well as ***digital libraries*** represent emerging application types with a certain requirement for spatial raster data management functionality and support. The scope of these application types can vary significantly from project to project which results in a significant variation in the type and extent of raster management functionality to be supported.

Geospatial Clearinghouses and Data Stores

The term of 'geospatial clearinghouses' is subsequently used for systems and services which serve as mediators between producers and users of spatial data. Geospatial clearinghouses are developed and operated with the goal of supporting the collection and dissemination of spatial information in a variety of representations on a broad range of topics. Geospatial data stores are similar in functionality but are much more focused on marketing and selling geospatial information. All these systems and services are primarily metadata-based which allows them to be operated in low-bandwidth WAN (wide-area network) environments. With respect to raster data, geospatial clearinghouses and data stores can be considered as generalised super-sets of the above-mentioned specialised catalogues for remote sensing or photogrammetric raster data. Hence, similar raster data management requirements apply.

The following is a typical example of a public geospatial clearinghouse service with quite a broad scope:

- *EWSE* (European Wide Service Exchange) of the European Commission's Centre of Earth Observation (CEO) at the Joint Research Centre (JRC) is a testbed system which currently covers information on topics such as earth observation products, educational courses and even job information.

An example of products which support the implementation and operation of geospatial data stores or clearinghouse services is:

- ***Sedona GeoCATALOG*** of Sedona GeoServices Inc. (Sedona, 1996). GeoCATALOG is a software product which enables the creation and distribution of product catalogues for geospatial information such as maps, aerial photography and satellite imagery. GeoCATALOG can be interfaced with a number of GISs and can be used to implement on-line querying and ordering solutions for these products.

Digital Libraries

There are some contrasting positions on the use and meaning of the term digital libraries. On the one hand, it stands for traditional digital library catalogues. On the other hand, it is used for the ultimate digital information system and data store with the capability of storing, finding, transmitting, viewing and manipulating complex information (Bhargava, 1996). This information could consist of text, video, audio, images, slides, maps, photographs, numerical data, software, etc. Building such comprehensive digital libraries is a very ambitious goal, especially if the support for all these types of information is to be extended beyond the metadata level.

In practice, it has proved to be more realistic to address domain-specific issues. The following research project, for example, primarily addresses spatial data:

- ***Alexandria Digital Library*** is a major research project of the Department of Computer Science at the University of California at Santa Barbara (Smith, 1997). The goal of this project is to build a distributed digital library for geographically referenced 'materials', which range from maps, images to text and multi-media. The research activities are divided into the following topics: library, interface design, information systems, image processing and performance. The latter three topics are largely equivalent to those investigated by some of the projects presented in Section 4.1.1.

4.3 SUMMARY

This chapter first presented the status and trends in the research and development of spatial raster management solutions. The subsequent survey presented a cross-section through a number of application types with their specific characteristics and requirements. A selection of domain-specific services and systems further illustrated the diversity of applications with a requirement for spatial raster management.

From a data management perspective these applications, services and systems can be classified on the basis of the two characteristics operational focus and operational scenario. The different types of operational focus of a system or application include: raster processing-oriented, raster management-oriented or metadata management-oriented.

- ***Raster processing-oriented*** applications and systems focus on the tasks of reading, manipulating and writing spatial raster data. Their typical characteristics include standard data flows and processing sequences with specialised, highly efficient processes. From a data management perspective, the primary requirements of these applications and systems are fast data access and high data transmission rates which usually requires custom-built solutions with a high level of system integration.

- ***Raster management-oriented*** applications and systems focus on the storage, handling and querying of spatial raster data. They can be characterised as having a strong requirement for services typically offered by DBMSs such as querying, concurrent multi-user access as well as consistency and security control. Management-oriented solutions usually have to support a range of different data types with their built-in or user-defined operations. System design priorities are query performance and flexibility and also the adherence to open standards which permit the interfacing and integration with other systems and modules.
- ***Metadata management-oriented*** applications and systems require a limited raster management support only (e.g. for handling and browsing quick-looks). The type and extent of supported raster metadata can vary significantly, ranging from minimal georeference information to very detailed sensor-specific information.

The second criteria is the system access scenario with the three categories Local/LAN (Local Area Network), MAN (Metropolitan Area Network) and WAN (Wide Area Network).

- *Local/LAN*-type systems and applications are typically operated locally on individual systems or on multiple systems which are interfaced by means of a local area network. They usually require very high data transmission rates ($\geq$ 100 Mbit/s) and are operated and used by specialised personnel which have access to the full functionality of the system.
- *MAN*-type systems and applications are often operated in distributed environments within individual organisations or companies. They can be characterised by heterogeneous platforms and user communities. These user communities typically share a common spatial information base which they wish to query, access and maintain efficiently and economically. For this type of access medium-bandwidth networks (128 Kbit/s - 100 Mbit/s) are usually sufficient.
- *WAN*-type applications usually address widely-distributed, heterogeneous and potentially very large user communities. WAN-type solutions are typically used to enable or facilitate the exchange of digital spatial information between producers and users. With the current bandwidth limitations in wide-area networks ($\leq$ 128 Kbit/s), these applications typically limit the accessible contents to metadata and data samples.

A classification of the presented applications on the basis of the above-mentioned characteristics is shown in Table 4–2. From a raster management perspective the current status and the perspectives in the different application categories can be summarised as follows:

- ***Upper-left*** applications typically incorporate custom-built, file-based and highly efficient raster management solutions. Future improvements can primarily be expected from spatial raster management functionality supporting project management and data handling tasks and from a better integration with raster management-oriented systems.
- ***Centre*** applications incorporate those applications with the greatest demand for spatial raster management support. However, current solutions in this category show

some serious deficiencies concerning database integration, query functionality and management of very large raster objects and raster mosaics.

System Access Scenario	**Local/WAN**	**MAN**	**WAN**
	full functionality	*mostly query and retrieval functionality*	*limited query functionality only*
	small number of expert users		large number of non-expert users
Operational Focus	few to no access restrictions	few restrictions for authorised user groups	tight access control and restricted access
Raster Processing-oriented	RS[6] Data Acquisition and Production RS and DP[7] Image Processing		
Raster Management-oriented	←	Orthoimage DBs	→
	←	DTM DBs	→
	←	Raster Map DBs	→
		Raster Map Production	
	RS and DP Image Archiving →		
	←	Digital Libraries	→
Metadata Management-oriented		←	Remote Sensing Catalogues
			Geospatial Clearinghouses

Table 4–2: Classification of applications with a demand for spatial raster management.

- ***Lower-right*** applications currently require only limited spatial raster management support. With the continuous increase in network bandwidths it can be expected that current catalogue systems will gradually transform into digital spatial data stores. These data stores will have a number of important spatial raster management requirements such as efficient multi-resolution and copyright protection concepts.

The following investigations will primarily address issues and problems related to the centre group of raster management applications in the raster management-oriented / MAN category.

[6] RS = Remote Sensing

[7] DP = Digital Photogrammetry

Leer - Vide - Empty

5

INVESTIGATIONS AND DEVELOPED CONCEPTS

5.1 INTRODUCTION

5.1.1 Objectives and Scope of the Investigations

The objective of the investigations presented in this chapter is to address and resolve some of the major issues and problems of spatial raster data management which were identified during the initial research. These investigations should result in a ***generally applicable DBMS-based concept*** which can be integrated within state-of-the-art GIS technology. In particular, the following aspects were investigated:

- supporting *DBMS concepts and technologies*
- *spatial partitioning*
- *spatial* raster data *indexing and access*
- *raster representation and encoding*
- *metadata* management and *georeferencing*
- *multi-resolution* support
- *visualisation* of large raster objects
- *system interfaces* to spatial raster databases

These investigations provided the basis for the design and development of the prototype system GrIdS. The key features and characteristics of GrIdS are described in Chapter 6.

5.1.2 Investigated Raster Types

The preliminary studies documented in Chapters 2 and 3, which looked at spatial raster data from a data management perspective, enabled the identification of two types of raster objects: ***individual raster objects*** and ***raster mosaics***. It will be shown that any type of spatial raster data can be stored and managed using these two raster types. In the following chapters, these key raster types will frequently be referred to. The following definitions serve as the basis for the subsequent discussions.

Individual Raster Objects

Individual raster objects represent raster data sets with limited dimensions and data size. Individual raster objects are typically ***acquired, manipulated, processed, visualised and exported as a whole***. They can nowadays be handled and operated upon using standard state-of-the-art computer hardware and software. Examples include aerial or remotely sensed raster images or scanned raster maps.

Raster Mosaics

Raster mosaics are composite raster objects, which enable the representation of 'seamless raster databases'. Raster mosaics are typically ***very large raster objects*** consisting of ***multiple sub-objects or partitions***. Their data volume usually exceeds the amount of memory available on current computer systems by an order of magnitude, and therefore they require the development of special data management and data handling concepts. Raster mosaics are typically built-up by combining multiple individual raster objects and they are only very ***rarely accessed or processed as a whole***. Raster mosaics can be used to represent large members of any of the spatial raster classes listed in Section 2.2.

Note: In certain application domains such as remote sensing and photogrammetry the term 'mosaic' is used for specific types of products which consist of lateral combinations of multiple orthorectified raster images. However, in the following text the term will (intentionally) be used in the more general sense outlined above.

5.1.3 Hypotheses

The management of spatial and unstructured data used to be typical domains for specialised data structures and storage solutions. Today, there is a clear trend towards integrating primarily vector-based spatial data in standard database management systems by means of spatial middleware or spatial database extensions. The same trend exists for the integration of unstructured data (e.g. sound and video). Despite the availability of the two technological key elements, the support for spatial raster data management provided by these DBMSs or by spatial middleware solutions is still limited to non-existent.

The fact that today's DBMS technologies incorporate the basic features for managing ***spatial and unstructured data*** suggests the following hypothesis:

Hypothesis 1: A spatial raster management concept can be developed and implemented on the basis of ***standard state-of-the-art (commercial) DBMS technology*** .

Hypothesis 2: Such a spatial raster management concept can efficiently be implemented on ***standard state-of-the-art computer hardware platforms*** .

5.2 DBMS-BASED MANAGEMENT CONCEPT

The main requirements of future spatial raster data management solutions can be expressed as

*the ability to scale to large amounts of **unstructured, spatial** data and large numbers of users.*

This is a modification of Stonebraker's (Stonebraker and Moore, 1996) original summary of standard database features presented in Section 3.7.2. In the case of spatial raster data these requirements imply the support for comprehensive metadata management, spatial and non-spatial querying and efficient storage management. Only DBMS-based solutions will permit to fulfil these requirements both efficiently ***and*** economically. The subsequent investigations will therefore focus on DBMS-based raster data management only.

In the following sections, the evaluation of a suitable DBMS technology, a raster storage concept and a system architecture will be presented.

5.2.1 DBMS Technology

The selection of a DBMS technology for the management of spatial raster data was influenced by the following key aspects. Firstly, from a data management perspective, the determining data characteristics are the ***enormous data volume*** followed by the ***spatial properties***. The complexity of the data is far less important than, for example, in vector-oriented spatial information systems. Secondly, the goal of investigating the integration of raster data management with GISs required that established and ***compatible DBMS technologies*** were investigated in particular. Last but not least, the selected DBMS technology should allow a spatial raster management system to be developed as a ***DBMS application***. This implied that only standard application programming interfaces (APIs) provided by database manufacturers were to be used. This application-oriented approach distinguishes these investigations from the database kernel-oriented research work of the associated database group at the ETH Zürich (Blott et al., 1996), (Relly et al., 1997).

'Core' Solution based on Relational Technology

The investigations were initiated on the basis of relational DBMS technology. Despite its well-known limitations for handling complex data, it turned out to be capable of efficiently supporting the majority of the spatial raster data management functionality listed in Section 3.3.

Most of the developed concepts, which are described in the following sections, are available or can be implemented on state-of-the-art relational technology. The prototype system GrIdS (see Chapter 6) represents a 'relational-only' (or 'fully-relational') spatial raster data management solution. It demonstrates the capabilities and some of the limitations of this database technology.

Object-Relational Extension

The first commercial object-relational DBMSs (ORDBMSs) started to emerge in the course of the investigations. Because the ORDBMS technology provides a super-set of the relational DBMS functionality, all the developed raster management concepts can easily be adapted. In addition, ORDBMSs provides some very useful new functionality such as content-based querying as well as actual raster management and manipulation in an SQL context.

These additional features were researched and documented. However, due to time and resource restrictions, such object-relational extensions could not be implemented in the GrIdS prototype system.

5.2.2 Storage Management Concept

Internal versus External Storage

Traditionally, any data to be managed by a DBMS first has to be imported into and owned by the system (Relly et al., 1997), which typically includes the transformation to a system-specific representation on non-removable secondary storage. This ***internal storage*** approach ensures the availability of the database services (e.g. transactions, security and concurrent access) for that data. Importing large objects into the DBMS is expensive. This is due to the 2-phase-copy-overhead and the rollback overhead outlined in Section 3.7.2. Additionally, the import process often involves copying of the data from one media to another, resulting in a space-consuming replication of the data. A comparison of DBMS-internal and -external storage is provided in Table 5–1.

	DBMS-internal Storage	**DBMS-external Storage**
controlled by	DBMS	application (future: DBMS)
storage space allocation	static	dynamic
advantages	supported by DBMS-services	advanced mass storage concepts supported
disadvantages	import overhead	no (or limited) DBMS services available

Table 5–1: Comparison of DBMS-internal and -external storage for large unstructured objects, such as spatial raster objects.

The selection of an optimal storage strategy depends very much on the type of raster data and on the type of application. The main storage scenarios supported by DBMSs were presented in Section 3.9.3. DBMS-controlled secondary or ***internal storage*** is ideally suited for geocoded raster objects which are frequently and possibly concurrently accessed by a large number of users. For this type of data, the additional data import overhead is outweighed by the benefits of the full DBMS support. Typical examples for this type of data are raster mosaics. ***External storage*** is better suited for raw raster data, which is accessed only sporadically and which is primarily used as a data source in production processes. The cost of importing this type of data into the DBMS is typically much higher than the potential benefits.

Dual Raster Storage Management Concept

On the basis of this assessment and of the results from some preliminary system tests, a raster storage concept with the following characteristics was developed (Figure 5–1):

- ***dual***, internal and external storage management
- storage solution ***transparent*** to raster management system users
- DBMS-controlled internal storage management
- ***application-controlled*** external storage management
- check-in/check-out mechanism for moving data between external and internal storage

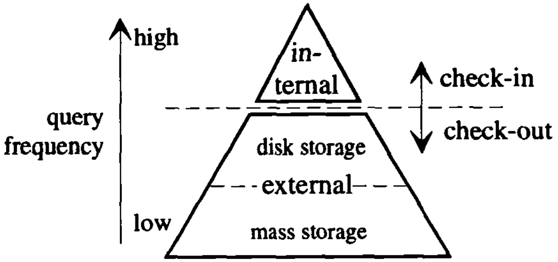


Figure 5–1: Conceptual view of the selected raster storage concept with DBMS-internal storage of frequently accessed data (e.g. orthoimage mosaics) and DBMS-external storage for 'raw' raster data (e.g. unprocessed satellite imagery).

This dual raster storage architecture combines the strengths of internal and external storage and at the same time avoids some of their shortcomings. The internal storage strategy, for example, ensures the consistency of regularly accessed data which is particularly important when performing partial updates on raster mosaics. Application-controlled external storage management enables the exploitation of the functionality and performance of advanced storage sub-systems, such as hierarchical storage management systems (HSM) and jukebox mass storage technologies. The application-controlled approach, on the one hand, means giving up some desired database features, such as data independence. On the other hand, this concept can be implemented in conjunction with any DBMS.

5.2.3 Conceptual System Architecture

Most of the raster management concepts discussed in this chapter are not dependent on a specific system architecture or DBMS technology. Thus, the selection of a specific system architecture is primarily an implementation issue. However, there are two important architectural issues which affect the data management concepts. The first aspect is the system level at which the raster management functionality is to be implemented and the second issue is the level of DBMS integration.

Implementation Level - Spatial Raster Middleware versus DBMS Extensions

Spatial raster management functionality can conceptually be implemented as part of the application, as DBMS functionality or as so-called middleware. The decision for one or several implementation levels determines the available development tools, affects the functionality which can be implemented and influences the interfaces through which this functionality is accessible. In commercial solutions for the DBMS-based management of primarily vector-based spatial data two main approaches have emerged. One is a ***spatial middleware*** approach (Figure 5–2a) as represented by ESRI's Spatial Data Engine (SDE) (ESRI, 1996). The other is a ***spatial database extension*** approach (Figure 5–2b) such as implemented on a relational basis in Oracle's Spatial Data Option (SDO) (Oracle, 1995a) or on an object-relational basis in the Spatial DataBlade of Informix (Informix, 1997). A slightly one-sided comparison of middleware- and extension-based approaches for spatial data can be found in (Stonebraker, 1997). The author particularly argues the advantages of an object-relational DBMS-extension approach for this type of complex spatial data.

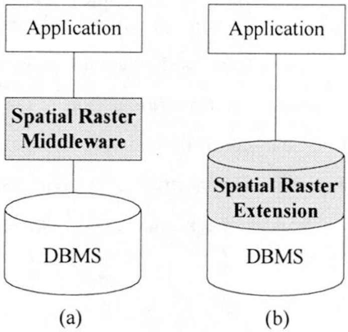


Figure 5–2: Main implementation levels for spatial raster management functionality.

The middleware and DBMS-extension approaches are also applicable to spatial raster data management. However, due to the special characteristics of spatial ***raster*** data which were outlined in 3.2, the arguments affecting the decision for one or the other approach are distinctly different. Table 5–2 summarises the characteristics of a spatial raster middleware solution (a) and a spatial raster DBMS extension (b).

Spatial Raster Middleware (SRM) (see Figure 5–2a)	**Spatial Raster DBMS Extensions (SRDE)** (see Figure 5–2b)
• enables the implementation of very complex functionality • largely DBMS-product independent • functionality primarily available through application programming interface (API)	• functionality available in a SQL context • user-defined extensions require object-oriented or object-relational technology • demanding and complex DBMS extensions required for raster mosaics • additional middleware layer required for external storage management (if not supported by DBMS)

Table 5–2: Comparison of two different levels for implementing spatial raster management functionality: a middleware-level solution (left) and a DBMS-extension solution.

Levels of DBMS Integration

Figure 5–3 shows three middleware-based spatial raster management architectures with different levels of DBMS integration. In early solutions DBMSs were used for metadata only (Figure 5–3a). With the added functionality for managing unstructured data the DBMS support could be extended to DBMS-internal raster data storage (Figure 5–3b). Finally, with the emerging support for DBMS-external storage concepts, bulk storage of raster data using high-performance mass storage sub-systems will be integrated, too (Figure 5–3c). The diagram illustrates, how the increasing level of DBMS integration reduces the number of interfaces required for the different data types and storage sub-systems.

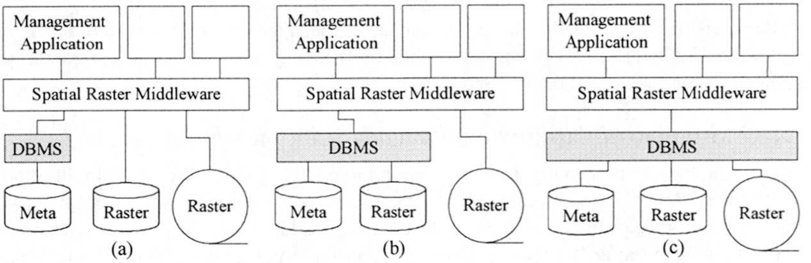


Figure 5–3: Conceptual overview of a DBMS-based spatial raster management architecture with different levels of DBMS integration: (a) metadata only, (b) partial DBMS integration, (c) full DBMS integration.

Selected System Architecture

The conceptual system architecture chosen for the investigations consists of a ***middleware solution*** as illustrated in Figure 5–2(a) and a partial DBMS integration shown in Figure 5–3(b) with DBMS-controlled internal data storage and middleware-controlled bulk raster storage.

Future System Architecture

In future, functionality will gradually be moved from middleware to the DBMS. However, in order to abandon the middleware approach the following requirements need to be fulfilled:

- standardisation of APIs and query languages (e.g. SQL3) in order to assure portability
- full DBMS-support for tertiary mass storage sub-systems
- availability of basic ‘raster’ DBMS extensions which can be used as a basis for the development of specialised spatial raster extensions

5.3 Spatial Partitioning Concept

The need for the spatial partitioning of large raster objects and the basic concepts available were outlined in Section 3.5. A comparison of the main spatial raster partitioning techniques can be found in (Nebiker, 1996a). The following two concepts were investigated in detail and subsequently implemented:

- a ***strip-based partitioning approach*** for raster objects with conceptually ***limited dimensions***, such as individual scanned aerial images
- a ***tiling-based partitioning approach*** for raster objects with conceptually ***unlimited dimensions*** such as raster mosaics

The decision for a DBMS-based raster data management solution introduces the additional issue of client- versus server-side partitioning. The two partitioning concepts and the latter issue will be discussed below.

5.3.1 Strip-based Partitioning of Standard Raster Objects

In a spatial raster data management system a large part of the raster objects will almost exclusively be imported, archived and exported as a whole. A major requirement will therefore be to ensure optimal i/o performance. Spatial access to parts of such raster objects will be required only occasionally. Additionally, the dimensions of such raster objects are typically limited by the acquisition process, e.g. by the maximum scanning dimension and resolution. For these types of raster objects, a simple spatial partitioning method had to be selected which did not require an expensive transformation of the original data.

Strip-based raster partitioning (see Figure 3–5) is well adapted to such ***stream-based i/o operations***. It requires minimal or no transformation of the typical scan-line-based raster data and it is commonly supported by different raster data formats (e.g. TIFF). Strip-based partitioning is a useful concept for reducing and controlling the amount of memory required on a client system. Any local raster operation, for example, requires a maximum of only two partitions to be available in memory at any given time. (This assumes that the height of such a partition is greater than the dimension of, for example, the coefficient matrix used in a filtering algorithm.) In the chosen solution, the maximum partition dimensions are determined based on a 'maximum amount of memory per uncompressed strip'.

Strip-based partitioning, on the other hand, is not well suited for typical spatial queries and selections as shown in Figure 3–7. Large raster objects which are frequently accessed and which should support efficient spatial queries are better partitioned using the tiling-concept described below.

5.3.2 Tile-based Partitioning of Large Spatial Raster Objects

Large spatial raster objects such as ***raster mosaics*** conceptually can have unlimited dimensions and size. They are frequently accessed by means of spatial queries but they are rarely manipulated as a whole.

A partitioning concept based on ***regular, square tiles*** was chosen for our investigations. This non-hierarchical tiling concept supports the spatial expansion of raster objects (in every direction). This distinguishes it from the approach of sub-dividing a pre-defined region which is the basic principle of recursive space decomposition methods, such as region quadtrees (Samet, 1989). The selection of regular and uniform tiles simplifies the process of decomposing the raster space and the reverse process. In particular, the concept enables the use of simple and efficient spatial access methods (see Section 5.4). The main drawback of regular tiles with fixed dimensions is incurred by the fact that compressed tiles will have variable sizes depending on the tile contents. This potentially introduces inefficiencies and problems in the secondary storage management. Finally, a square tile shape was selected because it is ideally adapted to the standard raster organisation based on rows and columns and because it makes it possible to treat each tile again as an individual ***raster object***. The influence of tile dimensions and compression schemes on the retrieval costs is outlined in (Lamb, 1994). The determination of optimal tile sizes for different types of raster data and applications will be discussed in Chapter 7.

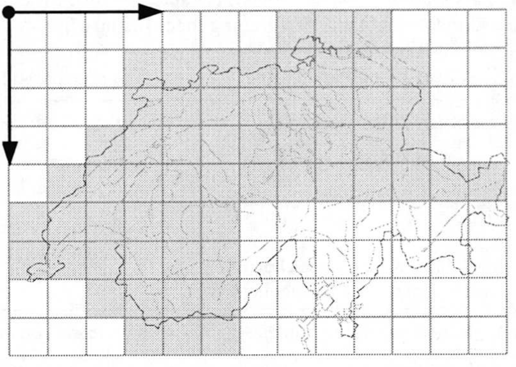


Figure 5–4: Raster mosaic consisting of regular, square tiles. Mosaic origin located at the upper left corner (tile row/column: 0/0). Populated tiles shaded in grey.

In summary, the regular tiling-based partitioning approach shown in Figure 5–4 fulfils the main requirements postulated in Section 3.5. Namely, it enables efficient spatial access to individual tiles and it supports the compression of the data held in each tile.

5.3.3 Client- versus Server-based Partitioning

In a DBMS-based client-server raster management architecture, such as that implemented in the GrIdS prototype system (see Chapter 6), raster data partitioning could conceptually take place on the client- or on the server-side.

In the ***client-*** or ***application-based*** partitioning scenario the data management application has to decide on the appropriate partition size (Relly et al., 1997). The application also has to perform the transformation operations involved in creating and joining partitions. The DBMS-server merely has to store and retrieve the partitions. The main advantage of client-based partitioning is that it can be implemented in combination with any DBMS technology. The main shortcoming of this approach is the potential susceptibility of the DBMS's storage management performance to varying object or partition sizes. The fact that commercial DBMSs are now used for large amounts of variable-size unstructured objects such as video sequences, leads to the assumption that they handle this issue quite well. The investigations presented in Chapter 7 will nevertheless test the respective robustness of the storage management solution available in a commercial DBMS.

In a ***server-*** or ***DBMS-based*** scenario, the raster partitioning is handled by the database server. In an ideal scenario, this process would be handled transparently and largely automatically by the DBMS's storage manager. A server-based partitioning strategy provides improved storage management and allows the partitioning aspects to be completely hidden from the client-system. Unfortunately, even emerging commercial DBMS products do not seem to address the issue of non-linear partitioning of very large unstructured objects. In the case of traditional spatial data such as points, lines and polygons server-based spatial partitioning concepts are starting to appear on the market. Oracle's Spatial Data Option (Oracle, 1995a), (Oracle, 1995b), for example, provides a hierarchical multi-dimensional table partitioning mechanism.

	Client-based Partitioning	**Server-based Partitioning**
advantages	• use of any DBMS technology • low server software complexity • enables transmission of compressed partition data • well suited for intra-object partitioning (e.g. large spatial raster objects)	• low client software complexity • server-controlled clustering and storage space optimisation • well suited for inter-object partitioning (e.g. vector-based spatial objects)
drawbacks / limitations	• high client software complexity • data overhead (if entire partition units are to be transferred)	• high server software complexity • import and export of very large objects requires additional application-based partitioning

Table 5–3: Comparison of client- and server-based partitioning concepts for large unstructured objects and spatial raster objects in particular.

Today, efficient server-side partitioning of large unstructured objects is still primarily a research issue (Relly et al., 1997). The introduction of ***table and index partitioning*** in new DBMSs, such as in Oracle8 (Bobrowski, 1997), marks an important step towards the efficient management of very large amounts of data. However, the implementation of a server-side raster partitioning concept on today's technology would require modifications of the DBMS kernel. As a consequence, the investigations were focused on ***client-based spatial raster partitioning*** concepts only.

5.4 Spatial Access and Query Concept

Efficient spatial access is a key issue in spatial raster data management. Some of the main aspects of spatial data access are summarised in Section 3.6. For this specific application a solution had to be found which was applicable to raster data and which could be implemented in a DBMS environment. This solution includes a tile indexing and tile query concept and a generally applicable approach for providing spatial access to different types of raster objects. The individual concepts will be presented in the following sections.

5.4.1 Mosaic Tile Indexing based on Morton Ordering

In raster mosaics not only the number of pixels but also the number of tiles can become very large. For example, a single high-resolution satellite image mosaic covering a small country such as Switzerland consists of more than 100'000 tiles[8]. In order to provide an efficient spatial access to these tiles an appropriate tile indexing mechanism had to be selected. The Morton ordering was chosen for the following reasons.

- It provides a simple and inexpensive code generation and evaluation by bit-interleaving the row and column positions of a tile.
- It results in the lowest oversearch for range queries in comparison with 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 (e.g. 32-bit integer) and traditional access methods (e.g. b-tree).

Tile Indexing Concept

The selected tile indexing concept is illustrated in Figure 5–5. Its key elements can be summarised as follows:

- ***N-ordering*** in column (prime) and row (secondary) direction.
- ***Origin*** located to the ***upper-left*** of the mosaic.
- Use of a ***variable-length*** Morton coding scheme.

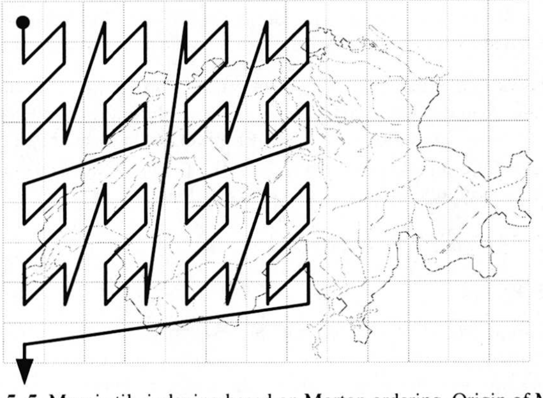


Figure 5–5: Mosaic tile indexing based on Morton ordering. Origin of Morton ordering located at 0/0 column and row position to the upper-left of the mosaic data. (Note: The diagram shows the first 64 Morton positions only.)

This specific configuration yields some interesting characteristics. It, for example, supports the expansion of the mosaic throughout the occupied quadrant. This is in contrast to approaches which are limited to the sub-division of a pre-defined region, such as documented in (Oracle, 1995a). The chosen approach also enables a relatively simple code re-organisation, which, for example, allows the origin to be moved without affecting the raster data. This is an important feature in situations where an expansion beyond the currently occupied quadrant should be required. A slight drawback of the chosen solution is that the evaluation of variable-length Morton codes is slightly more expensive than that of fixed-length codes.

Maximum Mosaic Dimensions

The maximum dimension of a raster mosaic is directly related to the domain of the Morton code. Therefore, it must be ensured that the tile indexing scheme does not impose any unreasonable restrictions. Ideally, the coding scheme would be based on existing integer data types which are particularly efficient in combination with the b-tree implementations found in many databases (Oosterom and Vijlbrief, 1996). The following calculations are based on standard ***unsigned 32bit integer*** types (e.g. the external type `unsigned(4)` in Oracle):

max. value:	2^{32} = 4294967296 = 4GBytes
max. number of tile rows and tile columns:	sqrt(2^{32}) = 2^{16} = 65'536
max. mosaic dimension: (1 metre pixel resolution)	⇨ 65536 tiles * 512 pixels/tile * 1 m/pixel = 33'554 km ⇨ 65536 tiles * 768 pixels/tile * 1 m/pixel = 50'331 km

Table 5–4: Summary of maximum mosaic dimensions for 32bit integer Morton codes. (Calculations based on 1 metre pixel resolution and on tile dimensions of 512 and 768 pixels).

Table 5–4 shows that a tile indexing scheme based on standard 32bit integer Morton codes is capable of managing and accessing high-resolution raster mosaics covering the entire globe. In the case of tiles with 768x768 pixels, this mechanism could be used to index mosaics, which might hold up to 2.5 * 10^{15} pixels (= 2'500 TeraPixels)! This by far exceeds the size of any raster mosaic conceivable today.

5.4.2 Spatial Mosaic Tile Queries

Spatial Point Queries

Point queries are used to find and retrieve the raster partition or tile containing a certain position and to subsequently access the corresponding pixel or cell. Using the Morton coding scheme, the key value or index of the tile containing the query point can be easily determined as shown in Figure 5–6. The bit-interleaving process used to

[8] Area: 40'000km^2 (Switzerland); resolution: 1 metre; tile dimension: 512x512 pixels = ca. 150'000 tiles

determine Morton codes was illustrated earlier in Figure 3–3. The resulting tile index can subsequently be used in a select query to retrieve the actual data from the database.

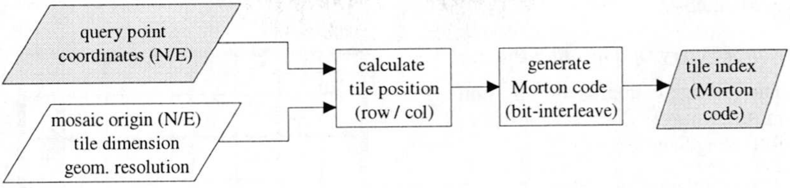


Figure 5–6: Process of determining the index of a tile, containing the query point.

Spatial Range Queries

Spatial range queries are typically performed by using rectangular search windows with sides aligned to the coordinate axes. These window queries are used to fetch candidate objects which are then subjected to further tests (Abel and Mark, 1990). In our case the range query is on the spatial codes of the tiles containing the upper left and lower right corners of the query window. The calculations for each point are illustrated in Figure 5–6. Due to a nice property of the Morton ordering, the codes of any tiles intersecting the query window lie in the range between the determined minimum code (upper left corner) and the maximum code (lower right corner). The results of a simple query consisting of this single code range are shown on the left-hand side of Figure 5–7. The candidate tiles returned by this query are shaded in grey. The figure illustrates the high oversearch which can be incurred by this basic approach.

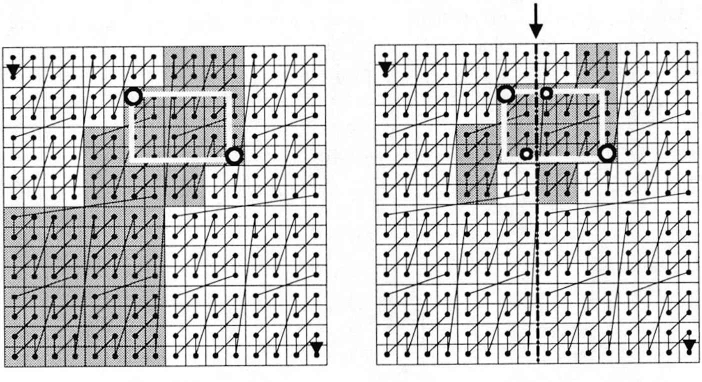


Figure 5–7: Spatial range query (white search window) on raster mosaic with Morton tile indexing. Candidate tiles returned (grey) by standard query using single code range (left) and by modified query with search window split along major quadrant boundary (right).

More advanced approaches are discussed in (Abel and Mark, 1990). The authors, for example, propose to use multiple ranges instead of just one, such that the ranges used include relatively few tiles outside the window. A still relatively simple but effective solution is to split the search window along major sub-quadrant boundaries (Abel and Smith, 1984). The effects of this approach are illustrated by the second, modified query in Figure 5–7.

Range Query Optimisation Concept

The selected approach for optimising mosaic range queries consists of the following features:

- specification of a threshold for lowest quadrant level above which range-splitting should occur (e.g. level ② in Figure 5–8)
- determination of the highest quadrant levels intersected by a query window in horizontal and vertical direction
- splitting of the query window along horizontal and vertical quadrant boundaries if quadrant level >= threshold

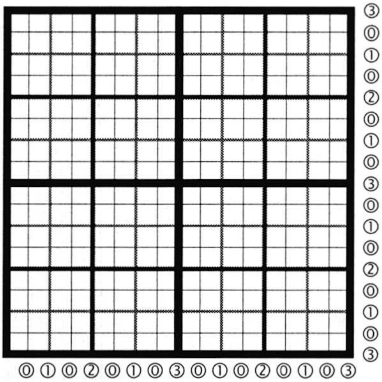


Figure 5–8: Quadrant hierarchy for mosaic tiles (Hierarchy levels labelled from ⓪to ③).

5.4.3 *Spatial Access to Raster Objects*

The spatial access concept presented above, addresses the issue of accessing spatial sub-sets of large raster objects. However, it does not enable the spatial search for individual raster objects with arbitrary position and extent (see Figure 5–9). For this type of spatial query, raster objects can be considered as vector-based areal objects defined by their bounding polygons. These polygons could consist of relatively simple minimum bounding rectangles (MBRs) or of multiple polygons which define the outline and possible holes of such objects.

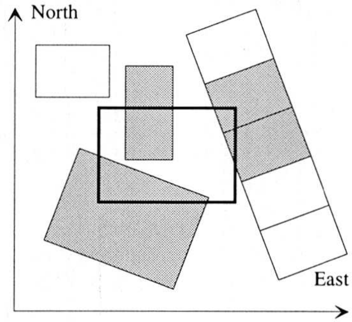


Figure 5–9: Spatial range query on raster objects. Raster objects intersecting the query window are shown in grey.

Querying, finding and retrieving such areal spatial objects is a standard feature in any GIS. A broad range of spatial access and query methods for such areal spatial objects have been described in literature (Frank, 1983), (Abel, 1983), (Abel and Smith, 1984), (Samet, 1989), (Oosterom and Vijlbrief, 1996). Several of these concepts have found their way into commercial spatial DBMS-extensions, which were summarised in Chapter 4. Any of these concepts or commercial products can be utilised to implement a solution for this type of spatial access.

As a consequence, the present investigations focused primarily on a concept for representing, storing and updating the georeference information required by such access methods. This information could then easily be used in the subsequent implementation of an appropriate access method. The developed georeferencing concept is discussed in Section 5.6.

5.5 Raster Modelling and Encoding Concept

A number of data structures and data types and their suitability for representing raster data were discussed in Section 3.4. The main factors influencing the selection of a suitable concept for representing spatial raster data can be recapitulated as follows:

- The ***data volume*** of spatial raster objects is typically ***very large***. It normally exceeds the volume of other spatial data types by orders of magnitude.
- The ***number*** of raster objects and raster object partitions are often ***large***, too.
- The ***complexity*** of raster data is ***low*** compared to the high complexity of other spatial data types such as topologically structured vector or network data.

5.5.1 BLOB-based Raster Object Modelling

From the available concepts for representing raster data, the solution of Binary Large Objects (BLOBs) was chosen. They are used for the representation of ***entire small raster objects*** and of ***partitions of large raster objects***, such as mosaic ***tiles*** or raster object ***strips***. BLOBs provide a very generic and versatile concept which is applicable in a file- and DBMS-based data management framework. BLOBs are the only data types which support variable-length objects of arbitrary length and which are available in every modern DBMS (see Section 3.8.4). A shortcoming of the current BLOB concept is that it lacks the support for user-defined operations. However, with the emerging establishment of BLOBs as standard data types (Date and Darwen, 1997), they are ideal candidates for incorporating and encapsulating raster data in abstract data types (ADTs). As such, the developed BLOB-based concept provides a sound basis for a possible future transition to object-oriented or object-relational DBMS environments.

Future Extension of Current Concept

Object-oriented concepts, namely the underlying concepts of abstract data types and user-defined operations, enable a considerable extension of the current BLOB-based raster management functionality. The possibilities and the potential limitations of such extensions are discussed in Chapter 8 (Conclusions and Outlook).

5.5.2 Raster Encoding and Compression Concept

The necessity for efficient raster data compression, the basic compression concepts and techniques were discussed in Section 2.4. The encoding and compression concept to be developed should provide a generic interface for different types of operations and at the same time be easily extensible. The main characteristics of the developed concept are listed below (see also Figure 5–10):

- client-based compression and decompression
- conceptually symmetric architecture for DBMS- and file-storage compression with an uncompressed raster representation at the centre
- use of a limited number of formats and compression schemes for DBMS storage
- support of an arbitrary number of formats and compression schemes for file storage based on a modular approach for adding new format and compression schemes

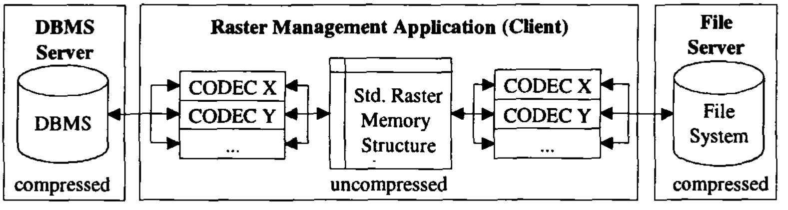


Figure 5–10: Schematic view of the raster encoding concept.

The client-based compression and decompression concept enables compressed data to be transmitted between the raster management application (client) and the DBMS- and file-servers. This is particularly beneficial if communication has to take place over networks with limited bandwidths. The standardised uncompressed data representation at the centre of the system has the function of a mediator interface between different formats and compression schemes and it at also serves as the standard representation for implementing raster operations. This standard interface enables the extension of the system, by facilitating the addition of new CODECs, new formats and new operations. The price for this flexibility is the additional cost of the regular compression and decompression operations. However, with today's CPU performance, on-the-fly compression and decompression is no longer a problem. In fact, overall compressed I/O performance is now significantly better than uncompressed I/O, despite the additional computational overhead.

5.6 Selected Metadata Aspects

The scope and importance of metadata in a spatial raster management context were discussed in Section 2.5. An important goal of the present investigations was to identify and specify the ***system-level metadata*** required for the efficient management of individual raster objects and raster mosaics. An exemplary metadata summary is provided by the logical schema of the prototype system GrIdS (Figure 6–7).

Generic raster metadata items comprises information such as raster type, radiometric resolution, geometric resolution, object identification and description. In addition, a number of issues were identified which primarily or exclusively apply to ***spatial*** raster data:

- series of raster objects
- layered or multi-channel raster data
- georeference information

The first two issues will be discussed below. The last issue, georeferencing, with its special importance and complexity will be addressed separately in Section 5.7. In addition to this system-level metadata, a considerable amount of application-level metadata might be required in order to support a specific application. An inexpensive and seamless integration of such additional metadata is certainly one of the biggest advantages of a DBMS-based raster management concepts.

5.6.1 Raster Series Management

Managing series of spatial raster objects should not be confused with the more general concept of managing arbitrary collections of objects (e.g. all X-ray images belonging to a certain patient). The term 'series' implies some regularity or repetitiveness (e.g. revision cycles, range of scales, sheet numbering) and a number of members with similar characteristics (e.g. dimensions, geometric resolution). The exploitation of series information is an important factor in the effort to increase the automation of spatial raster data acquisition and to facilitate the various management tasks.

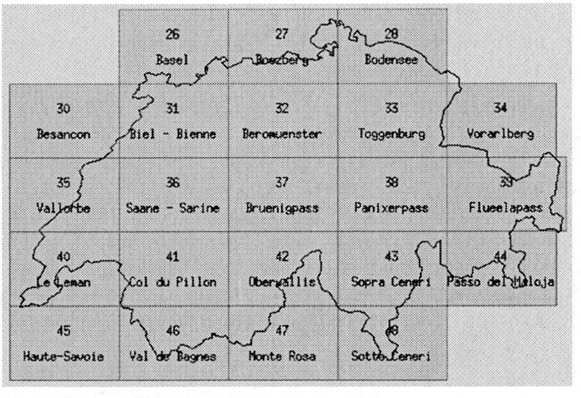


Figure 5–11: Overview of the Swiss national map series 1:100'000 showing sheet numbers and names. Sheet boundaries are identical for printed maps and scanned raster map products.

Figure 5–11 illustrates a raster map series. Other examples include series of raster images associated with aerial photogrammetric acquisition campaigns or orthoimage series, such as the Digital Orthophoto Quadrangle (DOQ) products provided by the United States Geological Survey (USGS). The solution implemented in the GrIdS prototype system demonstrates a basic approach for managing series of raster maps or orthoimages.

5.6.2 Raster Layer Management

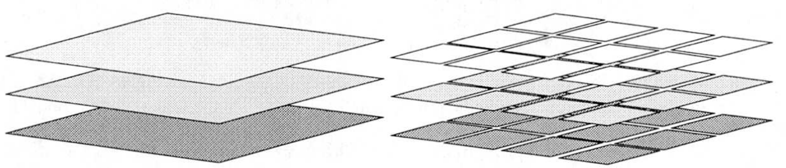
One of the typical characteristics of spatial raster data is the presence of ***multiple layers*** or ***channels***. A typical example for multi-layer raster data are raster maps, which often consist of several layers. Such layers typically represent different thematic topics and are represented by different colours. Another example is given by multi-spectral remote sensing data which often holds data from as many as one hundred spectral channels. This multi-channel raster data could be stored contiguously by using a large number of attributes or 'samples per pixel'. As an alternative, the data can be organised and stored in a planar approach, by assigning the data of each channel to a different plane or layer. A last example for the use of layered raster data are certain reproduction and printing processes which require colour separations. As indicated by these examples, the capability to manage raster data with multiple layers or channels is an important requirement in spatial raster management.

A multi-layer raster management concept should support the following functionality:

- addressing and selection of individual layers
- handling layers: creation, deletion, etc.
- assignment and editing of colour values
- support layer combinations ('image fusion') with the option to control layer priorities

The developed concept supports large numbers of layers for individual raster objects and for raster mosaics in particular (Figure 5–12). Individual layers can be assigned a name, a description, colour values and, if applicable, a priority value.

The data of multi-layered raster mosaics (see Figure 5–12(b)) is managed on the following basis. First, tiles on different layers cover the same spatial extent and can therefore be accessed using the original indexing scheme presented in Section 5.4.1. Second, the tile data for each layer is stored as an individual BLOB. In a DBMS framework this approach supports efficient access in horizontal (thematic selection) and vertical direction (spatial selection). However, it should be kept in mind that the concept being presented is a general purpose approach. Specialised applications such as multi-spectral image processing might require and provide storage and access concepts for multi-channel data which are much better adapted to the specific problem.



(a) (b)

Figure 5–12: Multi-layer support for individual raster objects (a) and raster mosaics (b)

5.7 GEOREFERENCING CONCEPT

As pointed out before, georeferencing or spatial registration, is an important metadata component. Its fundamental principles are outlined in Section 2.6. In short, georeferencing adds the direct spatial component to raster data and provides the basis for an efficient spatial access to individual raster objects, which was discussed in Section 5.4.3. The major objectives of the developed georeferencing concept were to ensure a global usability, to support map projection and geographic coordinates and to enable a modular and extensible implementation in a DBMS framework.

5.7.1 Global Geodetic Reference Database

A globally applicable georeferencing concept requires a convention for unambiguously specifying spatial reference information on a global scale. The geodetic database maintained by POSC/EPSG (Petrotechnical Open Software Corporation / European Petroleum Survey Group) was chosen for this purpose. It is probably the most comprehensive publicly available list and it provides information on almost any geodetic datum, ellipsoid, and any projected and geographic coordinate system known in the world. The responsible organisation maintains a mechanism for reporting back new or missing systems and for releasing revised versions to the public. The POSC/EPSG georeferencing database also supports the specification of 'undefined' or 'user-defined' systems in order to cater for special cases.

The POSC/EPSG database was first incorporated into the GeoTIFF format (Ritter and Ruth, 1995) and the Open Geodata Committee (OGC) are currently investigating how GeoTIFF can be supported by the Open Geodata Interoperability Standard (OpenGIS) (OGC Earth Imaging Committee, 1996). Selecting the same geodetic database ensures a high level of compatibility with these two important standards.

5.7.2 Multi-Level Georeference Information

The increasing 'globalisation' of spatial data acquisition (e.g. high-resolution remote sensing satellites) and data exchange (e.g. Internet) adds a new level of complexity to the processes of searching and retrieving spatial data. In contrast to the traditional 'local' mode of operation, which was limited to data on a single spatial reference (e.g. datum and projection), future spatial information will increasingly have to be queried and accessed internationally. This requires the support for multiple spatial reference systems, which was outlined in the previous section, and a hierarchical georeferencing concept which allows georeference information to be queried at different levels of abstraction and precision. The following concept is based on the outlines or enclosing polygons of raster objects. It is also applicable to other types of spatial objects, whose outlines can be defined by means of minimally bounding rectangles or enclosing polygons.

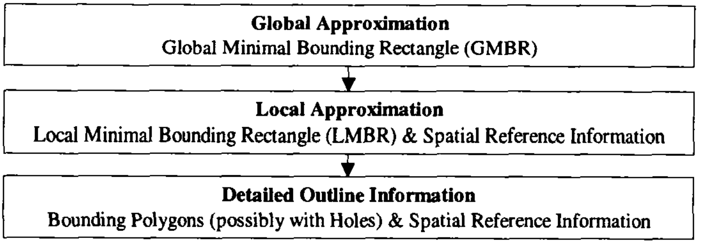


Figure 5–13: Hierarchical three-level representation of georeference information.

Level 1: Global Approximation

The global approximation or global registration (Figure 5–14a) enables spatial querying and retrieval of candidate raster objects irrespective of any associated spatial reference system. For example, a satellite image and a scanned map of Lake Zurich, of which the first is georeferenced based on WGS-84 and the second based on the local Swiss map projection system, will have completely different coordinates. Yet, a raster management system ought to have the capability to find both data sets.

The key elements of the developed global approximation concept are the following:

- global minimally bounding rectangle (GMBR) represented by geographical coordinates (Lat/Lon) of NW and SE corners of GMBR
- GMBR coordinates encoded using a suitable spatial data type (e.g. Oracle HHCODE)
- no geodetic reference system information required

An important aspect to be considered is the ***precision or resolution*** of the global approximation. It should take into account the typical inaccuracies in the traditional definition of geodetic datums, which can easily lead to positional differences in the order of several hundred metres. The resolution of the global approximation should therefore be coarser than this datum definition error. Spatial encoding techniques, such as Oracle HHCODE, enable to control the encoding precision by specifying the domain and resolution of each dimension. For example, the longitude domain of $-\pi$ to $+\pi$ in radians would need to be encoded at a resolution of $40'000/2\pi$ in order to obtain a precision of 1 km for the global approximation.

Level 2: Local Approximation

The local approximation (Figure 5–14b) is used in cases where the requested raster objects are to be based on a certain coordinate system and in cases where a coarse level-1-query is to be refined by a more precise second clause.

The key elements of the local approximation concept can be summarised as follows:

- coordinate system type: projected or geographic
- projected or geographic coordinate system code (from POSC/EPSG table)

- local minimally bounding rectangle (LMBR)
- represented by geographical or map projection coordinates for NW and SE corners (depending on coordinate system type)

Encoding the LMBR with two indexed pairs of geographic or map projection coordinates allows it to be represented at ***full precision***. This solution does not incorporate an efficient spatial access method and is therefore not suitable for 'global' spatial searches. However, in the developed concept local approximation is always used in clauses that follow a thematic pre-selection, e.g. the selection of a specific coordinate system. In that context it can be expected to perform very well. Spatial encoding solutions would require the explicit specification of domain and resolution for each new coordinate system. This would significantly complicate the operational use of such a solution.

Level 3: Detailed Outline Information

The third level of georeference information enables the representation of the complete outline of a raster object (Figure 5–14c). This could be a relatively simple rectangle for a raster image or a complex shape with multiple holes for image mosaics. This detail information is particularly useful for the ***graphical representation*** of raster object outlines and of the availability of data in partially populated raster mosaics in particular.

The main features of the current concept for representing precise outline information are the following:

- representation of object outline by means of polygons with provision for holes
- storage and representation by means of spatial data type (e.g. Oracle SDO type 'Polygon') or abstract data type

Managing and querying this type of information in a DBMS framework are typical tasks that will greatly benefit from spatial DBMS extensions or object-oriented DBMS concepts.

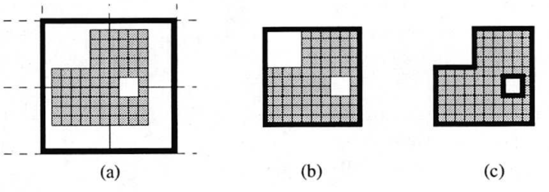


Figure 5–14: Three levels of raster object approximation illustrated using the example of a partially populated raster mosaic. Global approximation (a) spatially encoded at limited resolution, local approximation (b) and detailed object outline (c).

Implementation Issues

Aspects related to the implementation of a spatial raster georeferencing concept will be discussed in Chapter 6. However, it should be noted that the georeference information at level 2 can be omitted, if the underlying DBMS supports polygon window queries as efficiently as MBR window queries and if the bounding coordinates are accessible in a SQL context.

5.7.3 Georeference Query Matrix

The described multi-level georeferencing concept offers great flexibility for querying a spatial raster database (see Figure 5–15). The following features are supported:

- single-level and multi-level queries
- (top-down) hierarchical filtering
- (bottom-up) query window conversions and transformations

Single-level queries can be performed at any georeference information level. In very large databases this type of query will typically be limited to the global approximation level. In Figure 5–15 single-level queries are shown as horizontal query paths. In a multi-level query, several single-level queries are combined in hierarchical order, whereby inexpensive higher level queries are used as spatial filters for the subsequent more expensive queries. This improves the system performance and at the same time reduces the oversearch.

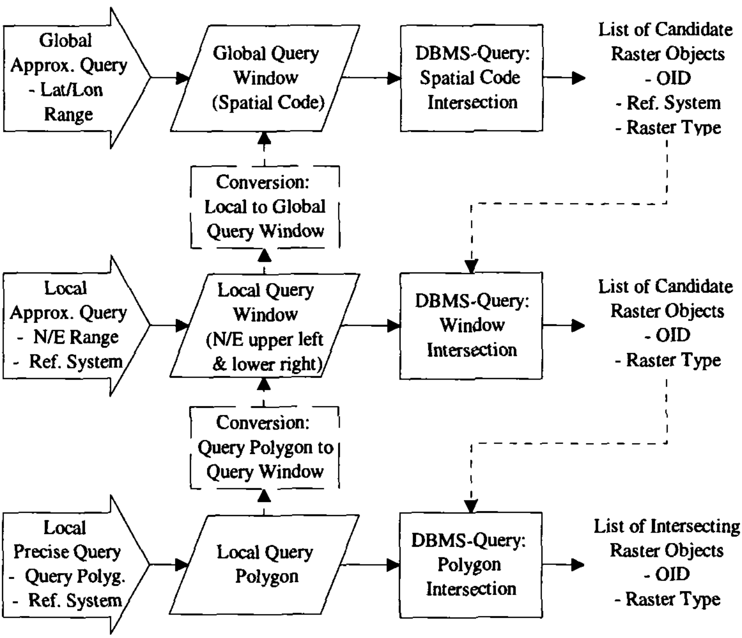


Figure 5–15: Multi-level georeference query matrix for spatial raster objects. Queries can be executed independently on each level (horizontal paths). Alternatively, combined queries can be performed by using higher level queries as filters for subsequent expensive detail queries.

A typical multi-level georeference query consisting of a global query (primary filter) followed by a more expensive local query (secondary filter) is shown below. This example uses the syntax and functionality of the Oracle Spatial Data Option (SDO). The primary, global query is performed using a spatial code (attribute 'sdo_code'). The secondary, detail query tests the candidate objects returned by the primary query for an intersection with the query window. It uses the SDO geometric operator 'RELATE' with the argument 'inside' to find any spatial objects fully enclosed by the query window.

```
SELECT  sdo_gid  gid1                                        (begin secondary filter)
  FROM(
    SELECT DISTINCT a.sdo_gid                                (begin primary filter)
    FROM rastermbr_sdoindex a, window_sdoindex b
    WHERE a.sdo_gid = b.sdo_code)                            (end primary filter)
  WHERE SDO_GEOM.RELATE('RASTERMBR', gid1, 'inside',
                        'window',1) = 'TRUE');               (end secondary filter)
```

5.8 Multi-Resolution Concept

Raster data is not suited to be visualised and used at arbitrary scales. The first reason for this is the degradation of the information content, which limits the range of scales within which raster data can be usefully exploited. This scale range largely depends on the type of raster data. For example, line-oriented cartographic raster data, such as contour lines or road networks, start to exhibit random appearance once viewed at a too small scale. The second reason is the cost involved in deriving an 'image' of an original raster data set at a specific scale. Computing a 1024x1024 pixel (1 Mega pixel) overview of a colour image mosaic at a scale of 1:10, for example, requires to access and process 300 MB of original raster data. With current computer technology, this task cannot be fulfilled on-demand and in real-time. Multi-resolution storage and representation techniques address these problems.

There exist several multi-resolution concepts and techniques for standard raster objects which are also applicable to individual spatial raster objects. It is recommended that a multi-resolution support for individual raster objects is implemented on the basis of these standard techniques. Quite often they are even supported by standard raster formats and their respective development libraries. The following section summarises some of the 'standard' techniques together with additional concepts which are specifically used in a spatial context. The subsequent discussions will focus exclusively on a multi-resolution concept for large raster mosaics.

5.8.1 Multi-Resolution Techniques

The available techniques can be classified on the basis of the following characteristics:

- discrete versus progressive
- redundant versus additive
- pre-computed vs. on-demand
- data-driven versus metadata-driven

In the following section several raster multi-resolution concepts are presented. Some of them are generally applicable, others are restricted to certain types of raster data.

- ***Previews*** or ***'thumbnails'*** represent a very basic multi-resolution concept. They are typically precomputed and stored together with the full resolution raster data. The use of previews is mainly limited to graphical inventories of collections of individual raster objects.

- ***Interlaced raster storage*** enables the gradual transmission and build-up of raster data sets. Most interlaced storage solutions, such as the one included in the GIF format (Murray and VanRyper, 1994), were primarily designed for this purpose. However, interlaced storage also provides a useful additive multi-resolution mechanism. In interlaced GIF, for example, a full resolution image is built-up in four consecutive passes, which have different offsets and row spacing. If the image build-up process is stopped upon completion of one of these passes, a complete image at one of four resolution levels is obtained.

- ***Resolution pyramids*** or ***image pyramids*** consist of a hierarchy of multiple raster objects representing the same spatial extent at different resolution levels (see Figure 5–16). There exist numerous different types of resolution pyramids and techniques for building them. A first, ***redundant*** pyramid type consists of the original raster object plus a series of reduced scale raster objects covering the same scene. Raster data at a specific resolution is obtained by reading the appropriate level of the pyramid. As an example, this type of resolution pyramid is used in Kodak's PhotoCD (Murray and VanRyper, 1994) and FlashPix image formats (Kodak, 1997). A second, ***additive*** pyramid type consists of the coarsest-level 'image' and a set of 'difference images' (Shahin, 1994). Here, raster data for a certain resolution level is built-up by adding the difference images to the low-resolution base raster object. Resolution pyramids are frequently used in modern digital photogrammetric workstations.

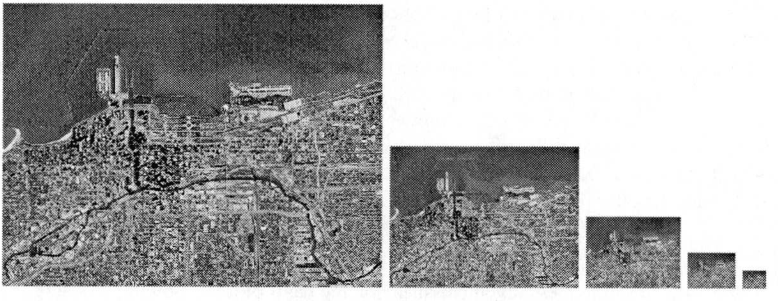


Figure 5–16: Five-level resolution pyramid with 50% reduction steps.

- ***Spatial multi-resolution encoding*** which is provided by several spatial data structures allows raster data to be represented and accessed at different resolution levels. With region quadtrees, for example, a reduced resolution representation is obtained by reducing the number of tree levels accessed (Samet, 1989). The same applies to hybrid spatial data structures, such as the Oracle HHCODE (Oracle, 1995a), which enable the specification of the resolution level at which the data is to be accessed. Direct multi-resolution encoding is typically limited to binary raster data.

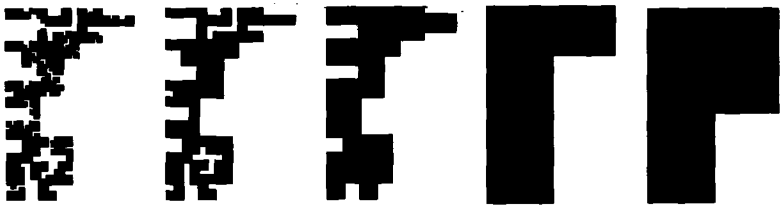


Figure 5–17: Spatial multi-resolution encoding of binary raster data shown at 5 different resolution levels (Oracle, 1995a).

- ***Metadata-driven / -based concepts*** provide multi-resolution support by relying on additional information about the availability of (raster) data sets at different scales. Mechanisms such as ***viewing or access profiles*** allow the access to different image or map series to be controlled based on the required scale or resolution. This approach is particularly suitable for frequently accessed, large spatial raster databases. It requires the availability of abundant information on the area of interest at the entire range of scales, e.g. mosaics of topographical maps, aerial and remotely sensed orthoimagery.

5.8.2 Multi-Resolution Concept for Raster Mosaics

The majority of the 'standard' multi-resolution concepts were developed for individual data sets with limited size and dimension. They are usually not suitable for large raster mosaics. As a consequence, a new multi-resolution concept had to be developed which could be applied to mosaics of arbitrary dimensions and which could easily be integrated with the full-resolution raster management approach outlined in the previous sections.

The main features of the multi-resolution concept for raster mosaics can be summarised as follows (see also Figure 5–18):

- regular tile-based approach
- resolution pyramid
- 'constant' tile dimensions (for all resolution levels)
- access to reduced-resolution tiles integrated in spatial tile indexing scheme
- supports DBMS-internal and external raster data storage

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 effect that the dimensions and the data volume of reduced resolution 'images' decrease with each additional resolution level. In a resolution pyramid with a 50% reduction step between adjacent levels a 1024x1024 image is represented by lower resolution images with the following dimensions: 512 (level 1), 256 (2), 128 (3), 64 (4), etc. At level 4, which corresponds to a reduction factor of 16 (2^4), the reduced-resolution image occupies roughly 1/256 ($1/16^2$) of the original storage space. This means that at resolution level 4 the original scene could theoretically be read and displayed 256 times faster than the original image.

With ***spatial raster mosaics***, however, this scenario of representing and visualising the same spatial extent hardly ever applies. Instead, it is much more common to operate with a constant size 'display window' irrespective of the current reduction level. This results in a variable spatial extent being covered by the display window: the larger the reduction factor, the larger the original area covered. A first idea might be to apply the standard resolution pyramid concept to the individual mosaic tiles. This approach of representing each tile as a multi-resolution pyramid addresses the data volume problem. It allows the amount of data to be read and displayed to be kept roughly constant for the entire range of reduction levels. However, this concept has a ***major shortcoming***: in contrast to the constant amount of data, the number of tiles or database objects to be accessed increases with the square of the reduction level (e.g. pyramid level 3 ⇨ reduction factor = 8 ⇨ multiplier for number of database objects = 64 !). Therefore, for a reasonable reduction factor of 8, instead of having to access 4-6 tiles at full resolution, this approach would require the retrieval of several hundred database objects!

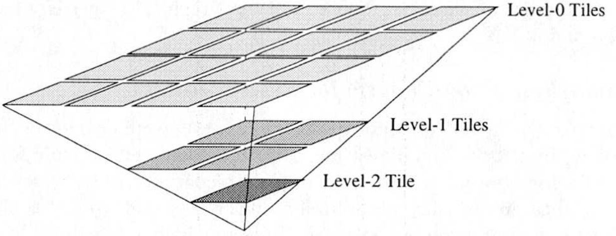


Figure 5–18: Resolution pyramid based on constant dimension tiles for multi-resolution raster mosaic representation. (Three top levels shown: level 0 = full resolution, level 1 = 50% resolution, level 2 = 25% resolution).

The chosen solution is shown in Figure 5–18. It is based on a resolution pyramid consisting of ***constant-dimension tiles***. The tiles in this approach cover different spatial extents depending on their level of resolution. For example, in the case of 50% reduction steps a lower resolution tile covers the area of four tiles on the next higher level (Figure 5–19). With this approach both the amount of data and the number of database objects to be retrieved can be kept roughly constant over the entire range of reduction factors. The chosen solution does not impose any restriction on the number of supported resolution levels. In most cases, however, a maximum of four or five resolution levels will be used.

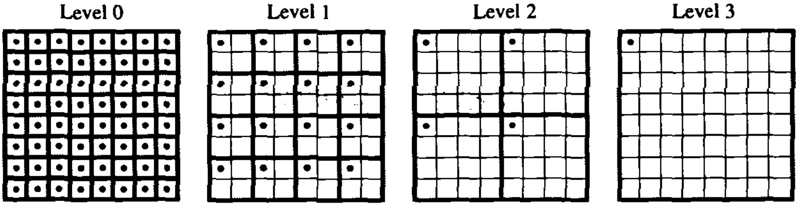


Figure 5–19: Conceptual view of spatial extents covered by tiles on different resolution levels. Tiles on each resolution level (bold rectangles) are shown in relation to the extents which are covered by the full resolution tiles at level 0 (indicated by the fine grid for levels 1-3).

Referencing and Indexing of Multi-Resolution Tiles

The investigations of an efficient access method for multi-resolution mosaic data led to a concept which could easily be integrated with the tile indexing method discussed in Section 5.4.1. The first requirement was that corresponding tiles on different resolution levels had to be assigned in order to establish unambiguous 'vertical' or ***resolution access paths***. In traditional image pyramids this assignment is straight forward. However, in the new tile-based approach, an appropriate convention had to be defined in order to avoid ambiguities. The decision was made to ***reference*** any lower level tile to the level-0 tile located immediately inside its upper left corner. In Figure 5–19 these ***reference tiles*** are marked with dots. As a consequence, all tiles with the same reference tile receive the spatial code or tile index of that level-0 tile. For example, the four tiles shown in Figure 5–20 which are located on four different resolution levels all have the same spatial code, which, in the case shown, has the value 0. As it turns out, this referencing concept enables an elegant access to multi-resolution tiles by integrating spatial and resolution access. This multi-resolution tile access will be discussed below. The selected referencing scheme also supports different concepts for storing multi-resolution mosaic data, which will be presented in Section 5.8.3.

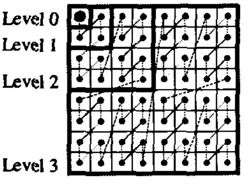


Figure 5–20: Hierarchical assignment of lower resolution tiles to higher resolution tiles based on position of upper left corner. Example shows four associated tiles at four different resolution levels, all with an identical spatial code of 0.

Multi-Resolution Tile Access

A ***multi-resolution point query*** is used to access a tile at a specified resolution level which contains the query point. Such a query consists of the following two main operations:

1. ***Spatial access operation:*** Determination of the full-resolution tile holding the query point. This is achieved by means of a spatial point query as described in Section 5.4.2. This query returns the index (Morton code) of the required tile at resolution level 0 (full resolution).
2. ***Resolution access operation:*** Based on the index of the full resolution tile which holds the query point, the corresponding tile on the desired resolution pyramid level is determined. This resolution access operation is described below.

A ***multi-resolution range query*** is used to return a list of candidate tiles on the specified resolution level. There are again two operations involved:

1. ***Spatial access operation***: Determination of the full-resolution tiles intersecting the range query window (see Section 5.4.2). This standard spatial range query returns a list of candidate tile indices (Morton codes).
2. ***Resolution access operation***: This operation applies a filter to retrieve tiles at the specified resolution level from the list of level-0 tiles returned by the spatial access operation. This access operation is also described below.

Resolution Access Operation

The main task of this operation is to determine the index of the reduced resolution tile at resolution level n which covers the spatial extent of the input tile at level 0. The characteristics of the selected tile assignment scheme (see Figure 5–20) makes the calculation of the spatial code at the required level an elegant and simple process:

$$sc_{(level\,n)} = sc_{(level\,0)} - mod\left(sc_{(level\,0)}, 2^{2n}\right)$$

The same calculation can be used in DBMS queries for filtering tiles on the requested resolution level from large numbers of candidate objects. The simplicity of the algorithm allows it to be implemented on the server side using Persistently Stored Modules (PSM).

Figure 5–21 illustrates the process of deriving the indices of corresponding reduced resolution tiles. The reference point of the full resolution tile which holds the query point is marked with a dark circle and labelled with ⓪. Based on this level-⓪ spatial code, the spatial codes of the reduced resolution tiles can be derived using the function presented above. Figure 5–21 shows the reference points and the coverage of the derived reduced-resolution tiles at the resolution levels ①, ② and ③. The corresponding calculations are documented in Table 5–5.

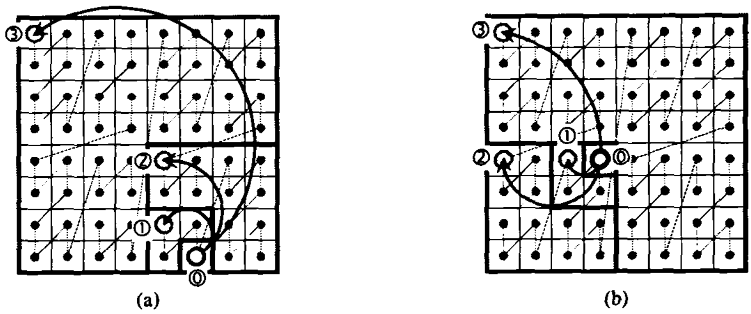


Figure 5–21: Examples illustrating the process of deriving indices or spatial codes of tiles holding reduced resolution mosaic data.

Level	Example (a)	SC	Example (b)	SC
⓪		**55**		**26**
①	55 - mod(55,2^2) = 55 - 3	52	26 - mod(26,2^2) = 26 - 2	24
②	55 - mod(55, 2^4) = 55 - 7	48	26 - mod(26,2^4) = 26 - 10	16
③	55 - mod(55, 2^6) = 55 - 55	0	26 - mod(26,2^6) = 26 - 26	0

Table 5–5: Computation of indices or spatial codes (SC) for tiles holding reduced resolution mosaic data. Calculations for examples (a) and (b) shown in Figure 5–21.

5.8.3 Storage Concept for Multi-Resolution Mosaic Data

The developed tile-based multi-resolution concept for raster mosaics can be used in conjunction with the DBMS-internal and -external storage concepts discussed in Section 5.2.2.

Single versus Multiple Tiles per Storage Unit

The requirement of managing multiple representation of the same data adds another level of complexity to the actual raster storage concept. The described multi-resolution approach which is based on several levels of discrete tiles leads to the following two basic concepts:

- storage of multiple resolution tiles in one storage unit
- storage of each resolution tile in an individual storage unit

The suitability and limitations of these two approaches are summarised in Table 5–6 below.

Multiple Tiles per Storage Unit (BLOB or File)	Single Tile per Storage Unit (BLOB or File)
• entire resolution pyramid can be treated as a single (database or file) object (e.g. creation, deletion, copying, moving) • does not increase number of objects • requires capability to manage and efficiently access 'sub-objects' or sub-files • creation of resolution pyramid involves (at least temporary) replication of original full resolution raster object • update operations typically require the 'replacement' of the original object • results in variable size objects	• tiles on each resolution level can be addressed individually (e.g. delete all data of a mosaic below resolution level x) • creation and destruction of resolution pyramid does not affect the original data • produces objects of roughly constant size • significantly increases the number of (database or file) objects

Table 5–6: Comparison of two approaches for storing multi-resolution tile data: 'multiple tiles per storage unit' versus 'single tile per storage unit'.

DBMS-Internal Storage Concept: Single Tile-per-BLOB

For the DBMS-internal storage the concept of writing each tile to a separate BLOB was selected. This solution allows each resolution tile to be accessed individually, regardless of the database technology used. It also enables BLOBs to be read and written as a whole, which simplifies the implementation. DBMSs are also well suited to handle the resulting increased number of objects. Last but not least, the chosen solution also allows object sizes to be kept at a fairly constant level, which is favourable for tuning the DBMS performance.

DBMS-External Storage: Multiple Tiles per File

The chosen solution for DBMS-external storage is that multiple resolution levels are stored in a single file. It uses the subfile mechanism of the TIFF file format (Aldus, 1992), which allows multiple raster images to be stored in the same file. Such images can, for example, be written and accessed using operations provided by standard development libraries such as libtiff (Leffler, 1995). The major advantage of this approach is that it keeps the number of files to a minimum and thus avoids potential file system limitations.

5.9 VISUALISATION CONCEPT

Visualisation of spatial data has become one of the topical issues in computer graphics and GIS. Numerous research projects focus on different visualisation aspects ranging from the design of efficient display algorithms to various applications for monitoring or simulating environmental change. The field of virtual reality which combines the visualisation of factual and fictional spatial information is the driving force in a highly dynamic hardware and software industry. Many of these projects are primarily based on vector data, they use spatial raster data for texture mapping only. However, any visualisation of spatial information has to rely on terrain information which is often represented by means of DTM rasters.

There exists a broad range of literature on the different aspects of spatial information visualisation. The basic visualisation concepts are discussed in (Foley et al., 1992) and (Baxes, 1994). A cross-section of different concepts and techniques for visualising cartographic data is presented in (MacEachren and Taylor, 1994). Particular aspects of visualising cartographic raster data are discussed in (Zanini and Carosio, 1995), (Zanini, 1996). A useful introduction is also provided by the on-line and off-line documentation of some of the standard raster image visualisation and editing software packages (e.g. PaintShopPro, ImageMagick, XView).

The investigations presented below focused on a number of visualisation aspects that are specific to the representation of large numbers of (very) large spatial raster objects. No attempt will be made to cover or discuss any of the standard visualisation issues in detail. The targeted use of the following visualisation concepts covers the range from system management to (remote) user access.

5.9.1 Raster Visualisation Techniques - Overview

The techniques for visualising spatial raster data can be grouped by their abstraction or aggregation levels into metadata visualisation techniques, reduced-resolution and full-resolution raster visualisation techniques. These techniques and their characteristics are described below.

Metadata Visualisation

Metadata visualisation is a key component in the management of large spatial objects and raster objects in particular. It supports rapid generation of graphical inventories and overviews of heterogeneous spatial data sets. It also enables their selection and inspection, using only a fraction of the original amount of data. The prime source of metadata for visualisation purposes is the georeference information outlined in Section 5.7. In addition, a multitude of thematic information can be represented either as graphical attributes or in textual form. Different aspects of metadata-based visualisation are discussed in (Evans et al., 1992) and (Walcher, 1996). Evans and co-authors describe a simple approach for interacting with spatial databases which is primarily based on spatial metadata. Walcher points out the benefits of metadata-based visualisation techniques in WWW-based interfaces of remote sensing image archives.

Typical metadata-based visualisation solutions for spatial data sets incorporate the following components:

- vector-graphics representation of spatial extents (MBR or detailed object outlines), possibly superimposed on a vector base map or low-resolution backdrop raster image
- representation of thematic information by means of colour-coding (e.g. different raster types, such as raster maps, aerial images, mosaics)
- textual presentation of additional attribute information

Reduced Resolution Raster Visualisation

For reasons of copyright and performance, the graphical representation of raster data will often be limited to reduced resolution levels. DBMS-based spatial raster management solutions are ideal for implementing ***resolution access restrictions*** applicable to individual users or specific groups of users. Most of the multi-resolution concepts presented in Section 5.8 are suitable for implementing such a reduced resolution visualisation concept.

Full Resolution Raster Visualisation

Visualisation of raster data at full resolution will rarely be required for performing actual raster data management tasks. In addition, for the copyright reasons mentioned above, full resolution will typically be limited to the actual process of data delivery and exchange. For special management tasks such as quality control and problem detection, raster visualisation at full resolution is nevertheless a useful tool.

5.9.2 Visualisation of Raster Object Metadata

Individual Raster Objects

The georeferencing concept presented in Section 5.7 enables the integrated visualisation of the outlines of all types of raster objects, including raster mosaics. This metadata-based visualisation concept displays the results of the spatial or non-spatial queries using georeference information at the appropriate representation level. For example, the objects returned by global-level queries, i.e. by queries with an unspecified geodetic reference system, are displayed as global minimal bounding rectangles (GMBRs) (see Figure 5–13 and Figure 5–14). The results of local-level queries can be displayed using local minimum bounding rectangles (LMBRs) or, where required and available, the detailed outline information.

Raster Object Series

At first sight, series of raster objects and their members in particular could be visualised using the same approach described above. However, they primarily represent a logical concept which might have no actual raster objects associated with them. Managing and displaying the outlines of existing raster objects together with possibly virtual members of raster series would cause conflicts and confusion. Thus, it was decided to store, access and visualise such series information separately.

The chosen visualisation solution for series information uses a vector representation of the series member outlines. This information can easily be overlaid with other information as shown in Figure 5–11. Such series layers enable a simple graphical selection of standard spatial extents from raster mosaics (e.g. 1x1km sub-sets). The graphical representation of series information also serves as a quality control tool. Figure 5–22 and Table 5–7 demonstrate the superiority of the graphical over the textual representation for inspecting the correct spatial registration of a series of raster objects.

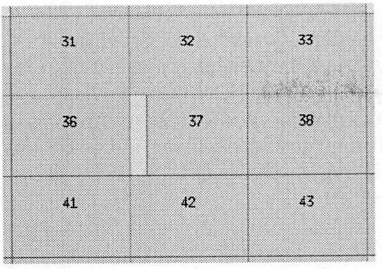


Figure 5–22: Visualisation of raster map series highlighting incorrectly registered map sheet 37.

Nr.	N(UL)	E(UL)	N(LR)	E(LR)
31	254000	550000	206000	620000
32	254000	620000	206000	690000
33	254000	690000	206000	760000
36	206000	550000	158000	620000
37	206000	630000	158000	690000
38	206000	690000	158000	760000
41	158000	550000	110000	620000
42	158000	620000	110000	690000
43	158000	690000	110000	760000

Table 5–7: Textual representation of the information displayed in Figure 5–22. (Incorrect coordinate highlighted)

Raster Mosaic Overviews

For the purpose of outline visualisation, raster mosaics can be treated like standard individual raster objects (see above). Some special characteristics of raster mosaics, such as their large dimensions and the possible presence of multiple layers, require the introduction of an intermediate visualisation approach which lies between a simple outline visualisation and the demanding visualisation of the raster contents. This intermediate metadata-based concept will be discussed below in Section 5.9.4 on raster mosaic visualisation.

5.9.3 Visualisation of Raster Object Contents

Individual raster objects can efficiently be visualised using commercial software packages as long as their data volume does not exceed the amount of physical memory available on a system. Most of these systems use one or several of the following data management concepts to ensure efficiency when representing relatively large raster objects over a wide range of scales:

- resolution pyramids
- thumbnails or quick-looks (as an alternative to complete resolution pyramids)
- image partitioning (e.g. tiling)
- concentric partition retrieval and image build-up

In cases where individual raster objects are stored externally (see 5.2.2) in their native formats, standard visualisation software can be used or even integrated into a spatial raster management system.

5.9.4 Visualisation of Raster Mosaic Contents

The visualisation of raster mosaics or seamless raster databases poses new problems and makes some heavy demands on the underlying raster management solution. The data volume of raster mosaics normally prohibits the generation of graphical representations at arbitrary scales on demand. For example, scanning a multi-GByte raster mosaic on state-of-the-art hard disk technology is an operation which takes minutes to hours. In addition, navigating through a mosaic resolution pyramid (see Section 5.8.2) is considerably more complex than navigating through the resolution pyramid of a single raster object residing in a single file. Finally, the actual task of managing large raster mosaics itself, requires special graphical representations of information such as spatial and thematic completeness. Below, a number of concepts will be discussed which support and facilitate the visualisation of spatial raster data in a DBMS framework.

Raster Mosaic Overviews

For the reasons explained above, large raster mosaics cannot directly be visualised. However, there is a need for information on the detailed spatial and thematic extents and completeness of raster mosaics. A mosaic management module should, for example, allow the following questions to be answered: 'are there any data gaps or holes in the mosaic' or 'do all mosaic layers or channels cover the same spatial extents'. These questions could theoretically be answered by numerically querying and analysing information on member tiles which is held in the database. However, the enormous size, the use of multiple layers or channels, the partial population and the irregular outline of future raster mosaics, make such a numerical inspection to a 'mission impossible'.

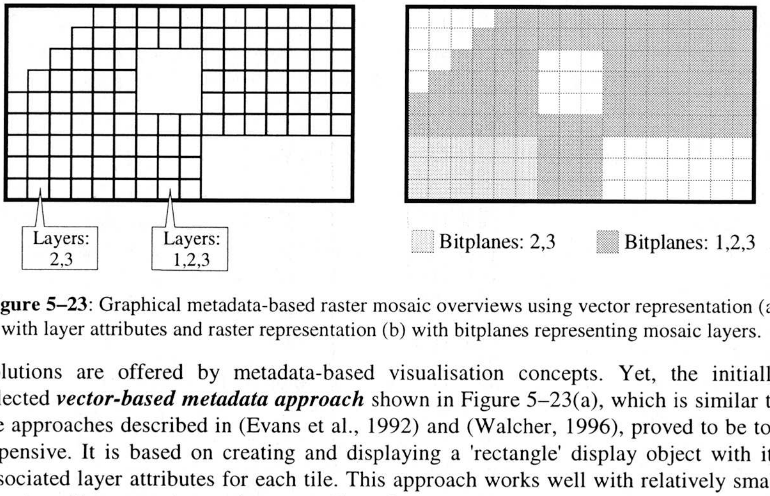


Figure 5–23: Graphical metadata-based raster mosaic overviews using vector representation (a) with layer attributes and raster representation (b) with bitplanes representing mosaic layers.

Solutions are offered by metadata-based visualisation concepts. Yet, the initially selected ***vector-based metadata approach*** shown in Figure 5–23(a), which is similar to the approaches described in (Evans et al., 1992) and (Walcher, 1996), proved to be too expensive. It is based on creating and displaying a 'rectangle' display object with its associated layer attributes for each tile. This approach works well with relatively small mosaics. However, its performance degrades to an unacceptable level with large mosaics which hold several ten to hundred thousand tiles.

As an alternative a ***raster-based metadata*** concept was developed (see Figure 5–23(b)). The characteristics of this new concept which could also be referred to as 'raster-based mosaic overview' or 'bitmapped mosaic index' can be summarised as follows:

- each tile is represented by one overview pixel (or cell)
- each mosaic layer is represented by a bit-plane of the overview raster
- overview rasters can be generated on demand or pre-computed and stored

The approach of overview rasters exploits the regular characteristics of the tile-based mosaic partitioning. The use of a simple raster structure instead of large numbers of complex and expensive vector objects dramatically reduces the cost of generating and managing large mosaic overviews. The chosen concept enables the complete overview information to be pre-computed and stored in a single raster object.

5.10 SUMMARY

This section summarises the key elements of the developed spatial raster management concept. For the management of individual (relatively small) raster objects traditional and established concepts were used and integrated where possible and appropriate. For the efficient DBMS-based management of very large raster mosaics a number of specific problems had to be resolved and several new concepts had to be developed.

Spatial Raster Middleware Architecture

The investigations led to the decision to develop a concept which could be implemented on a middleware layer. This solution combines some essential features, such as the capability to exchange compressed raster data between client and server with a great amount of flexibility. This middleware approach supports the delegation of a maximum of functionality to the DBMS, for example, by exploiting standard spatial DBMS extensions. It also enables DBMS limitations, such as the lacking tertiary mass storage support, to be circumvented by putting it under direct control of the middleware.

Tile-based Partitioning of Raster Mosaics

For raster mosaics and large raster objects a spatial partitioning approach on the basis of regular square tiles was chosen. This approach is relatively simple but at the same time very flexible and powerful because it can easily be combined with efficient spatial access, multi-resolution and compression concepts.

Tile Indexing based on Spatial Code

The regular characteristics of the underlying partitioning concept enabled the selection of a tile indexing method which is based on the Morton ordering. This solution enables a very efficient spatial access to mosaic tiles and it can easily be implemented on standard DBMS technology.

BLOB-based Raster Data Storage

The investigations showed that binary large objects (BLOBs) provide an efficient and flexible concept for storing any unstructured information in a DBMS framework. Conceptually, BLOBs allow spatial raster objects of arbitrary size to be stored.

Globally applicable Multi-Level Raster Georeferencing Concept

Among the investigated metadata issues, georeferencing is probably the most important. The developed georeferencing concept combines the global POSC/EPSG-database of geodetic reference system information with a multi-level georeference representation. This concept supports the registration, querying and visualisation of raster georeference information on multiple levels with different levels of detail and precision.

Tile-based Multi-Resolution Concept for Raster Mosaics

A concept supporting the representation of raster mosaic data at different resolution levels was developed. It modifies the traditional concept of resolution pyramids by combining it with multiple levels of constant-dimension tiles and the tile indexing concept outlined above. This new approach enables an integration of the spatial and resolution access within a DBMS framework.

6

GRIDS - A PROTOTYPE SYSTEM

6.1 INTRODUCTION

This chapter discusses the prototype system GrIdS[9] which enabled the implementation of some of the spatial raster management concepts presented in the previous chapter. With the development of GrIdS these concepts could be verified and, where required, also refined.

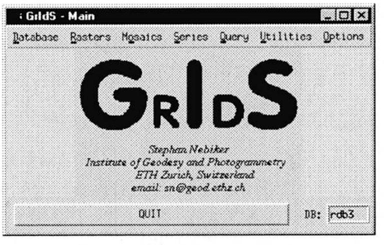


Figure 6–1: Main menu of the prototype spatial raster management system GrIdS.

In the first section of this chapter, the chosen development environment is introduced. The next section presents the system architecture of GrIdS. This is followed by the presentation and discussion of the database design and the functional design of the prototype system. Finally, several important implementation aspects will be highlighted and discussed.

6.2 DEVELOPMENT ENVIRONMENT

One of the objectives of the concepts presented in the previous chapter was their general applicability. This also implies the possibility to use standard tools and technologies for

[9] GrIdS = Geospatial Raster and Image Database management System

the implementation of these concepts. The different components of the chosen development environment are presented below. It should be noted that many of these components are generic and could quite easily be substituted by others.

6.2.1 Database Management System (DBMS)

Oracle Universal Server 7.3

The investigations and the implementation were initiated on the basis of state-of-the art relational DBMS-technology. According to an investigation in the June 1996 issue of GIS Europe, relational DBMS technology at that time (still) accounted for the vast majority of database engines which were either supported by or even integrated with current GISs (Nebiker, 1996b). The selected Oracle Universal Server is one of the main commercial relational database engines which supports all the standard DBMS services postulated in Section 3.7.2 (DBMS-based Raster Data Management). The object-relational extensions and DBMS technologies which started to emerge in the course of the project were closely followed and analysed. The present solution, however, is exclusively relational and, thus, maybe helps to demonstrate that the majority of issues involved in spatial raster management can be addressed with this technology. The main benefits of using object-oriented or object-relational techniques in future extensions of the current system will be discussed in Section 8.2 (Outlook).

Oracle Spatial Data Option (SDO)

Managing the spatial extents of large numbers of spatial raster objects requires database features which support efficient spatial queries. The Oracle SDO was chosen as a means to compare the spatial management features postulated in Sections 5.4.3 and 5.7 with the functionality offered by this spatial DBMS extension. Due to time and resource limitations, this functionality could not actually be coded and tested within the framework of this project.

6.2.2 Database Design Tools

The efficient design of a DBMS-based application such as GrIdS requires the availability of suitable database design tools. The initially selected product, Oracle's Designer/2000, proved to be far too sophisticated for such a small to medium scale project. The effort of maintaining such a highly integrated and complex database design system can only be justified in an environment with full-time database designers and developers. A better alternative was found in a stand-alone system called DB-MAIN (Hick et al., 1995) which is briefly presented below. Oracle's recent release of the relatively small and independent product Database Designer has confirmed the need for such a solution.

DB-MAIN

DB-MAIN is a database engineering CASE tool with a particular strength in reverse engineering (Hainaut, 1996), (Hick et al., 1995). It is being developed and maintained as part of a series of ongoing research projects at the Computer Science Institute of the University of Namur in Belgium. In order to support the various reverse engineering processes, DB-MAIN also provides powerful database design functionality at the conceptual and logical levels. The results of the database design process, namely the logical schema shown in Figure 6–7 can be exported as SQL scripts which contain all DDL (data definition language) statements required to generate the corresponding database. A demo/education version of DB-MAIN can be obtained on-line from the address quoted in reference (Hick et al., 1995).

6.2.3 Programming Languages

The GrIdS prototype system was implemented using a combination of the following three programming languages.

- Tcl/Tk
- C
- SQL

Each of these languages has its distinct characteristics with some specific strengths and limitations. Tcl/Tk is a scripting language which was used for the implementation of the graphical user-interface and of any non-time-critical tasks. Tcl/Tk is shortly described below. C was used for the implementation of complex or time-critical functionality. This included the reading, writing, encoding and decoding of binary strings as well as the partitioning, combination and transformation of raster objects. SQL was used from within Tcl/Tk and C for all metadata management tasks, i.e. all tasks not related to the management of BLOBs.

Tcl/Tk

Tcl/Tk is a high-level scripting language which originated from the University of California at Berkeley and which is now maintained and further developed by SunSoft (Ousterhout, 1994), (Welch, 1995) and (Ousterhout, 1997). Tcl/Tk is a full-featured programming language which also supports the development of graphical user-interfaces. It was originally developed for the UNIX and X-Windows environment but has since been extended to provide native support in the Microsoft Windows95/NT and Apple MacOS environments. One of the strong features of Tcl/Tk is the possibility to closely integrate it with other programming languages, particularly C. This feature was extensively used in this project.

6.2.4 Programming Libraries

For the implementation of the prototype system, a number of programming libraries were used. Some of them were essential (e.g. Oracle call interface library), others primarily helped to reduce cost and time by providing useful functionality which otherwise would have had to be developed at great expense. The different programming libraries used in the spatial middleware layer of GrIdS are shown in Figure 6–3.

Tcl/Tk Libraries

GrIdS uses the mechanism for extending the functionality of Tcl/Tk with C-code. This extension mechanism is based on linking the application object module(s) with the Tcl and Tk libraries. On some platforms these two libraries are part of the original distribution, on others they are built during the installation process.

Oracle Call Interface (OCI) Library

The OCI library provides the Call Level Interface (CLI) for Oracle DBMSs. It defines a set of application programming interface (API) calls through which the full functionality of Oracle DBMSs is made available to client programs, in our case a series of C-programs. The OCI library is available on all client-platforms supported by Oracle.

Oracle-Tcl Interface Library (oratcl)

The oratcl library provides a set of Tcl command extensions which allow Oracle DBMSs to be accessed and manipulated directly from Tcl/Tk scripts (Poindexter, 1997). Oratcl is also based on the Oracle OCI library. It provides a relatively simple mechanism for supporting DDL (data definition language) and DML (data manipulation language) operations in a SQL context. For example, it enables a SQL command to be defined, the results to be fetched (in case of a select statement) and transactions to be handled as shown in Figure 6–2.

```
set sql_str "select mosaic_id_seq.nextval from dual"   (1)
set dbret [catch {orasql $cur $sql_str}]                (2)
if $dbret==1 {
  oraroll $lda                                          (4)
  return TK_ERROR
}
orafetch $cur {                                         (3)
  set mos_id @1
}
```

Figure 6–2: Example illustrating the use of Oratcl commands in Tcl/Tk for selecting a new mosaic ID: definition of the sql command (1), execution of the query with the**orasql** command (2), fetching the result with **orafetch** and writing it to the Tcl variable mos_id (3) or, in case of an error, rolling back the transaction using the **oraroll** command (4).

TIFF Library (libtiff)

For the purpose of importing and exporting raster data a format had to be selected which supported a broad spectrum of raster types and which, at the same time, was commonly used and supported by other systems. The decision was made to use TIFF (Tagged Image File Format) (Aldus, 1992) as the standard import/export format and as the standard mechanism for DBMS-external storage. The TIFF software library ***libtiff*** is ***the*** standard C-library for reading and writing raster data in TIFF format (Leffler, 1995). It supports different lossless and lossy compression schemes and advanced features such as strips and tiles. The compression/decompression support mechanism of libtiff was also adapted and used as the basis for implementing compressed DBMS-internal raster storage.

GeoTIFF Library (libgeotiff)

The POSC/EPSG database was selected as the basis for the georeferencing concept discussed in Section 5.7. This choice and the selection of TIFF as the standard format made GeoTIFF a natural choice for the implementation of the georeferencing concept. The libgeotiff library (Ritter and Ruth, 1995) includes all the functionality required for reading and writing the corresponding georeference fields.

6.3 SYSTEM ARCHITECTURE

Middleware-based Client-Server Architecture

The spatial raster management system GrIdS conceptually consists of the following main levels and components (Figure 6–3):

- ***application*** level: spatial raster management application, primarily written in Tcl/Tk
- ***middleware*** level: spatial raster middleware, exclusively written in C
- ***DBMS*** level: Oracle DBMS *with stored procedures and/or DBMS extensions (note: features in italic are not implemented in present system)*

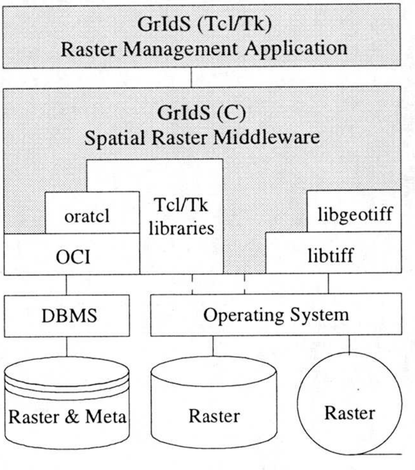


Figure 6–3: Conceptual view of the GrIdS system architecture.

The use of an Oracle DBMS and its associated development tools automatically enable a client-server architecture in a heterogeneous and distributed environment. From a DBMS perspective the current GrIdS system represents a client application.

Operating Systems and Platforms

The foundation on Tcl/Tk and ANSI-C ensures that the system can relatively easily be ported to any platform supported by Tcl/Tk and Oracle, namely almost any UNIX and Windows NT platform. It also ensures that the system can be operated in a number of possible scenarios. The three components shown in Figure 6–4 could either all be run on the same system or on three (or more) different platforms.

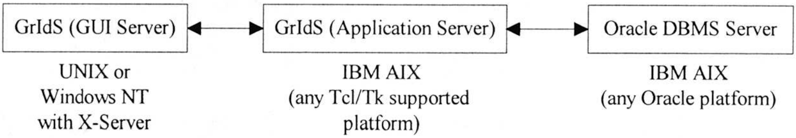


Figure 6–4: Operating systems and platforms supported by the different components of GrIdS.

Graphical User Interface (GUI)

Tcl/Tk provides a native-look window-based GUI on different platforms. An example of the GrIdS GUI is shown in Figure 6–5. It demonstrates the look of a UNIX-based Tcl/Tk application, whose GUI is displayed on a Windows NT client running a X-Windows server.

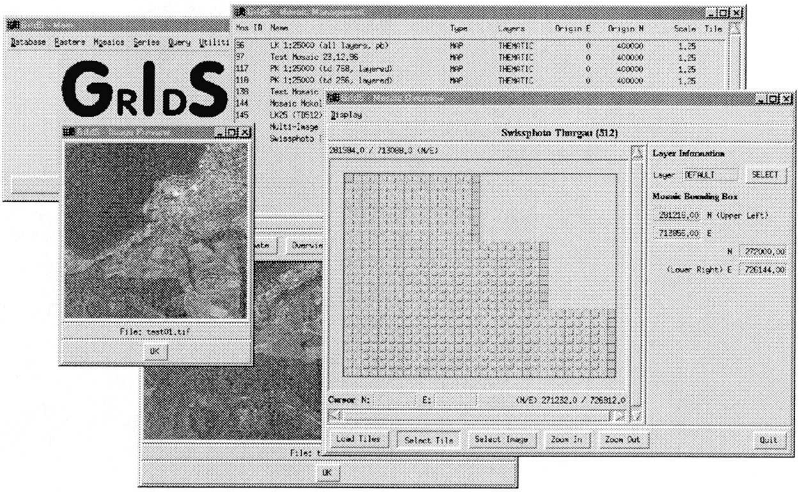


Figure 6–5: Tcl/Tk-based graphical user interface of GrIdS.

6.4 Database Design

6.4.1 Conceptual Database Design

The primary objective of the conceptual database design is the *ideal* modelling of the real-world, the quality of which might be measured by the completeness and expressiveness of the resulting conceptual schema (Batini et al., 1992). The conceptual design is independent of any database model or DBMS technology. The Entity-Relationship (ER) model is probably the standard tool for abstracting and modelling the real-world at the conceptual level. The basic elements of an ER model are entities, relations and entity attributes. A conceptual schema of GrIdS, expressed in the form of an ER model, is shown in Figure 6–6. The key entity types of this model are: raster, mosaic, tile and georeference.

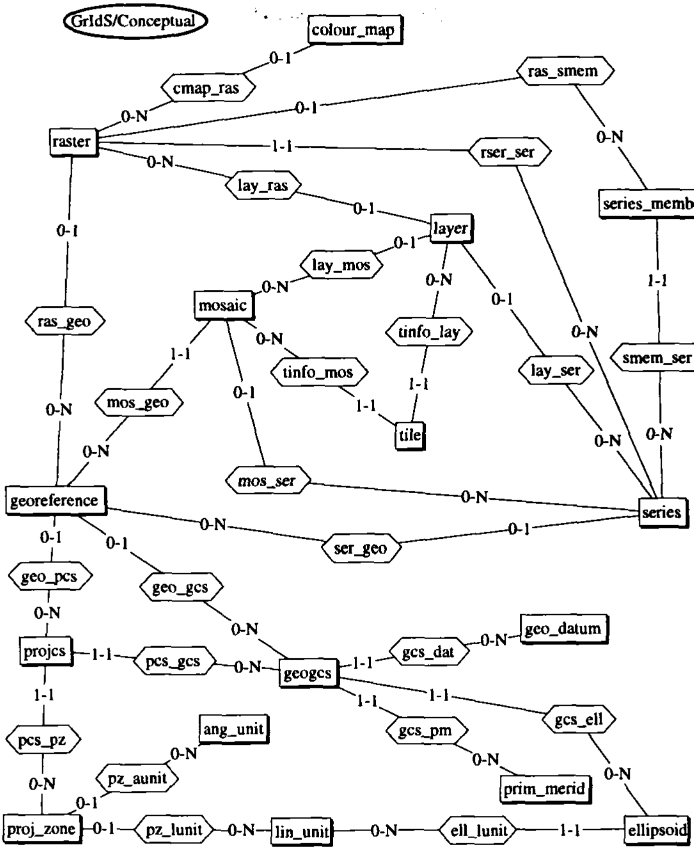


Figure 6–6: Conceptual schema of GrIdS based on an Entity-Relationship Model (ERM) with entity types (rectangles) and relationship types (hexagons) with their cardinalities.

6.4.2 Logical Database Design

The goal of the logical design is to translate the conceptual schema into a logical schema tailored to the database model and the DBMS to be used. In contrast to the conceptual design, logical modelling also needs to account for the aspects of efficiency within the framework of the given database model. In the case of the relational database model the process of mapping the conceptual model to the relational model is primarily directed by the four normal forms (Elmasri and Navathe, 1994). (Batini et al., 1992) summarise the corresponding mapping tasks as: elimination of composite and multi-valued attributes, modelling of internal and external identifiers, and elimination of relationships. The resulting logical schema of GrIdS is shown in Figure 6–7.

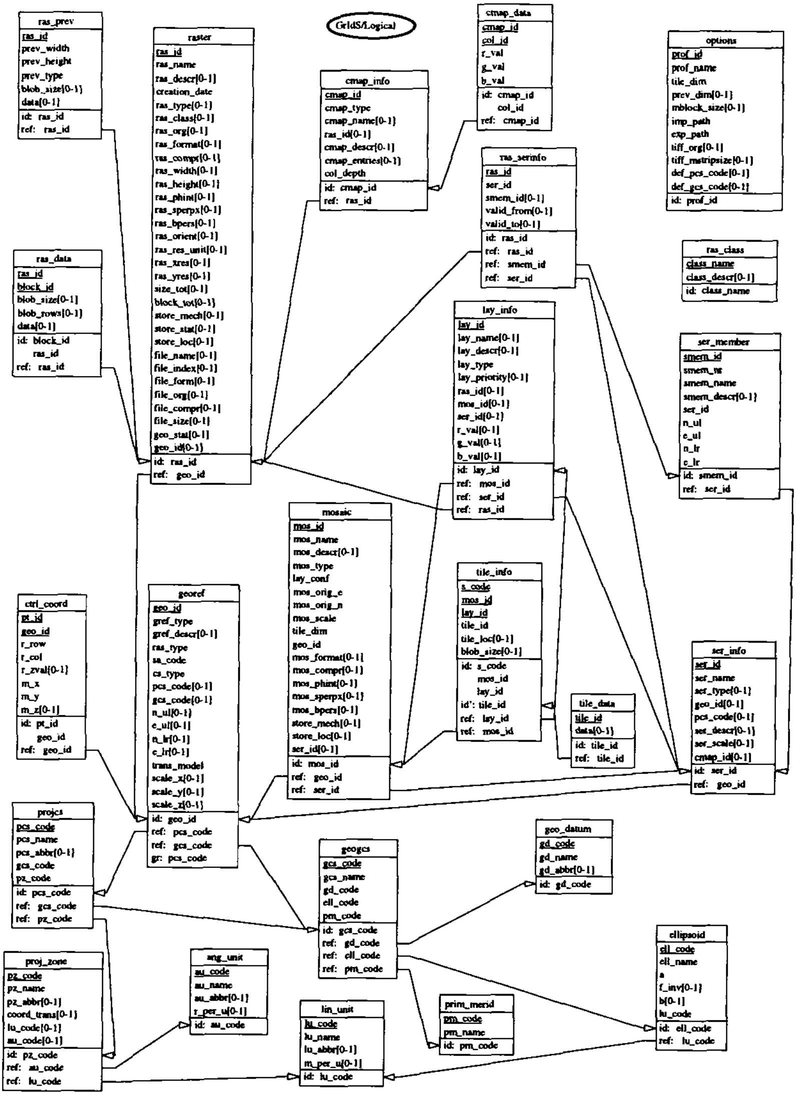


Figure 6–7: Logical schema of GrIdS. (Primary keys: underlined and redundantly listed with label 'id:'; Referential integrity constraints indicated by arrows and labelled with 'ref:'.)

6.4.3 Physical Database Design

The scope of this project did not include any low-level physical database design. These aspects were primarily addressed by a partner research group (Blott et al., 1996), (Relly et al., 1997). However, there are a number of essential high-level physical design issues which are typically accessible through the SQL/DDL interface.

The most important of these high-level physical design issues is the use of ***indices***. The typical indexing mechanism for scalar data types used in most commercial DBMSs is the B-tree (Koshafian and Baker, 1995), (Samet, 1989). An index basically consists of an ordered copy of the indexed database column with pointers to the storage locations. Modern commercial databases automatically create and manage indexes for primary keys and foreign keys. However, additional indexes need to be defined especially in conjunction with tables holding large objects. For example, searching for an object without an index requires a full-table scan, which in case of a multi-GByte table of BLOBS could require several hours.

Other important high-level physical design issues can be summarised under the term 'database tuning'. Database tuning is typically performed by the DBA (database administrator). Selected database tuning aspects will be addressed in Section 7.4.4.

6.4.4 Database Design for Raster Mosaics and Tiles

On a conceptual level, mosaics can be modelled using the entity types `mosaic` and `tile` (see Figure 6–6). However, for the implementation on a relational DBMS on the basis of the BLOB data type the `tile` entity needs to be split into two tables `tile_info` and `tile_data` (see Figure 6–8). This solution allows the actual raster data to be stored in the dedicated table `tile_data`. This table should exclusively be accessed through its indexed primary key `tile_id` in order to avoid any full table scans on this potentially very table. Any tile queries are performed on the tile metadata table `tile_info`. The `tile_info` table has a unique attribute tile_id which serves as foreign key for the access to the tile data held in the `tile_data` table. The basic concept of a single metadata and raster data table for all mosaics can easily be extended to a flexible storage concept in which the storage mechanism (DB or FILE) and storage location (e.g. table name) of each mosaic are stored in the store_mech and store_loc attributes of the mosaic table.

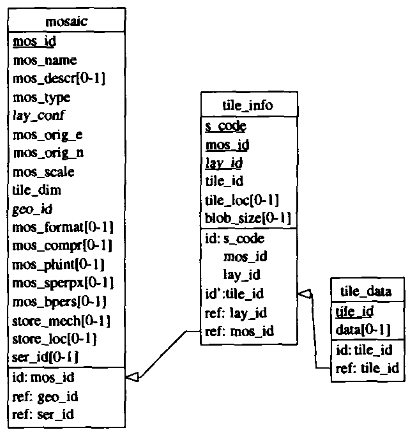


Figure 6–8: Logical schema illustrating basic mosaic data storage solution.

6.5 FUNCTIONAL DESIGN

6.5.1 Raster Data Flow and Main Data Structures

The key elements of the raster management structure and raster data flow within the GrIdS architecture are shown in Figure 6–9. The figure illustrates the conceptual subdivision into two sections corresponding to the two main raster object types: individual raster objects (upper part of Figure 6–9) and raster mosaics (lower part of Figure 6–9). The central elements of each section are the following two main memory structures:

- ***raster memory structure*** for handling raster object strips
- ***mosaic memory structure*** for handling multi-layered raster mosaic tiles

The symmetric raster storage and compression concept outlined in Section 5.5.2 was implemented for both raster types. It provides the choice between DBMS-internal raster storage and DBMS-external or file-based raster storage for raster objects and mosaic tiles. Compression / decompression modules (CODECs) can be implemented and interfaced using a standard mechanism. This mechanism allows the same CODECs to be used for internal and external storage paths and for both types of raster objects.

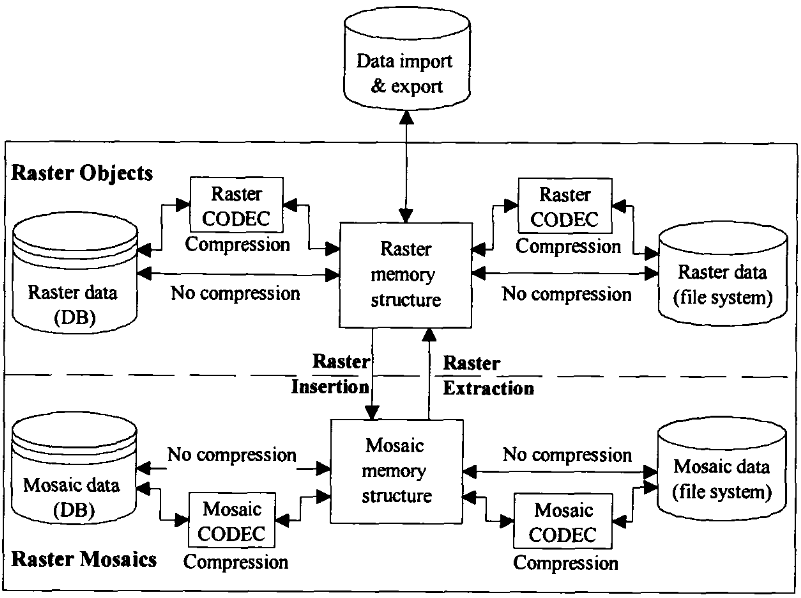


Figure 6–9: Data flow diagram for individual raster objects and raster mosaic.

6.5.2 System Functionality

As part of the functional design the basic functionality of a spatial raster management system had to be identified and suitably structured. This functionality was grouped into functional modules which correspond to the main menu items of GrIdS (see Figure 6–1). Each of these modules incorporates functionality related to a specific object type (e.g. raster or mosaic) or to a specific group of tasks (e.g. querying).

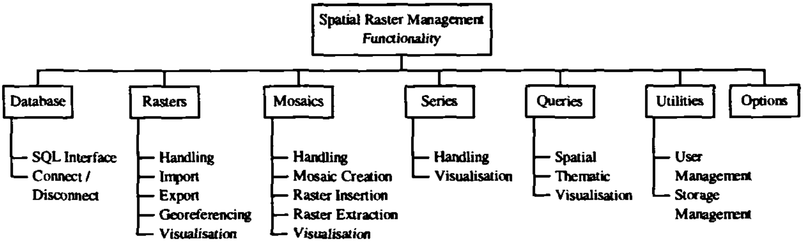


Figure 6–10: Summary of spatial raster management functionality grouped into different functional modules.

Figure 6–10 illustrates the functional module structure and lists the main functionality associated with each module. The presented structure is by no means imperative and could easily be re-arranged. In fact, most of the functionality was implemented in the form of largely independent program modules (e.g. georeferencing module) which can be built-in and accessed anywhere throughout the functional hierarchy.

The main functional modules and some of their key features will be discussed in more detail in the sections below. It should be noted, that the functionality presented is by no means exhaustive and that only some key aspects could be implemented as part of this project. Functionality not currently available in GrIdS is shown in italics in the diagrams below.

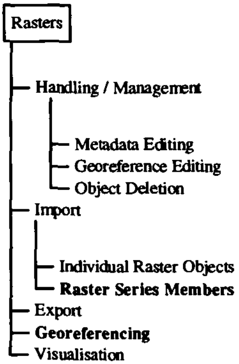


Figure 6–11: Raster object management functionality of GrIdS.

Raster Object Module

This module incorporates the functionality required to import, store, handle, export and visualise individual raster objects. The important features are:

- A georeferencing module which allows to spatially register individual raster objects. (The implemented georeferencing solution is further described in Section 6.6.)
- A program module which enables the (semi-) automatic import and georeferencing of raster objects belonging to raster series.

Individual raster objects can either be stored DBMS-internally in the form of strips or DBMS-externally in their original raster format.

- Mosaics
  - Handling / Management
    - Metadata Editing
    - Mosaic Deletion
    - Mosaic Layer Deletion
  - **Mosaic Creation**
    - Layer Configuration
    - Mosaic Georeferencing
  - **Raster Object Insertion**
  - **Raster Object Extraction**
    - Graphical Selection
    - Numerical Selection
    - File/Format Specification
    - *Selection Preview*
  - **Mosaic Visualisation**

Figure 6–12: Mosaic management functionality of GrIdS.

Mosaic Module

The mosaic management module incorporates the basic functionality for creating, handling and deleting raster mosaics. These can be of arbitrary dimension and size, can contain any kind of raster data (e.g. imagery or DTM) and can have a layered or multi-channel structure. The key operations supported by the mosaic management module are:

- Mosaic creation
- Insertion of raster objects
- Selection and extraction of raster objects
- Metadata-based mosaic visualisation

These operations will be discussed separately in Section 6.7.

- Queries
  - *Spatial (direct and indirect)*
    - *Coordinates (graph.)*
    - *Coordinates (num.)*
    - *Geographical Names*
    - *Series Information*
  - Thematic
  - Temporal
  - *Visualisation and Graphical Selection (MBRs and Object Outlines)*
    - *Global Approximation*
    - *Local Approximation*
    - *Detailed Object Outlines*

Figure 6–13: Querying functionality of GrIdS (features in *italics* were not implemented).

Query Module

This module incorporates the functionality required for querying the spatial raster database, for visualising and analysing the query results and for selecting raster objects for further inspection or processing. The query functionality is based entirely on metadata and particularly on the georeferencing information presented in Section 5.7.2.

An operational system might support the following different types of spatial queries:

- Direct, coordinate-based spatial queries (e.g. window query)
- Indirect spatial queries by means of geographical names (e.g. 'find all satellite images covering *Lake Lucerne*')
- Indirect spatial queries based on raster series information (e.g. 'extract spatial extents of standard map sheet no. 245 from the raster mosaic LK50').

The basis of all these spatial queries are the spatial extents or outlines of spatial objects in general and of raster objects in particular. Thus, standard spatial techniques and tools provided by most GISs can be used for implementing this functionality.

Within the framework of this project only thematic query functionality was implemented. No spatial query functionality was attempted

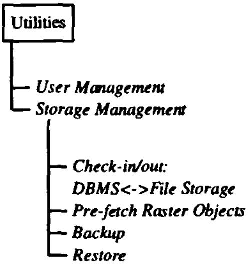


Figure 6–14: Selection of possible utilities in a spatial raster management system.

Utilities Module

The scope of a utilities module within a spatial raster management system could be very broad. Much of the functionality in such a module would furthermore be application dependent. The two major tasks of user management and storage management were selected and attributed to this utilities module.

User management is a key issue in an operational system. Fortunately, commercial DBMSs provide powerful user management concepts which are accessible from the standard SQL context. User management in a spatial raster management environment allows general access restrictions, accounting and invoicing to be controlled and enforced. Special limitations which may be enforced for certain groups of users might include the maximum resolution level accessible or maximum resource allocations (e.g. max. amount of memory per user).

Storage management is another aspect which will be particularly important in an operational environment. For certain backup and restore operations standard DBMS services can be used. However, for special storage management tasks such as raster check-in/check-out and for the pre-fetching of large raster objects, dedicated functionality will have to be implemented.

6.6 IMPLEMENTED GEOREFERENCING SOLUTION

As pointed out earlier, georeferencing is one of the key issues in spatial raster management. The two main issues in the implementation of a georeferencing solution are the georeferencing ***data model*** and a georeferencing ***user interface***. The basis of the georeferencing solution implemented in GrIdS is the GeoTIFF georeferencing mechanism and the corresponding POSC/EPSG database (Ritter and Ruth, 1995). This basis was combined and extended with some of the other georeferencing concepts presented in Section 5.7.

Georeferencing Data Model

The georeferencing data model of GrIdS is shown in Figure 6–15. It is divided into the following two groups of information: georeference information and global geodetic reference information.

The actual ***georeference information*** is represented by the tables **`georef`** and a **`ctrl_coord`**. These tables hold the information specified in the GrIdS georeferencing menu (Figure 6–16). The **`georef`** table holds a range of additional information which support some of the advanced georeferencing and spatial querying concepts described in Section 5.7:

- A spatial code attribute (`sa_code`) holding the spatially encoded global approximation (global bounding rectangle) *(currently not implemented - requires spatial DBMS extensions)*
- The coordinates (`n_ul, e_ul, n_lr, e_lr`) of the local approximation (local bounding rectangle).
- The attribute georeference type (`gref_type`) storing the type of raster object which the georeference record refers to. These types are: mosaic, raster and series. These raster objects can be accessed in their respective tables through the means of the foreign key constraints shown in Figure 6–7.

The ***global geodetic reference information*** is represented by a number of tables corresponding to the original spreadsheet files of the POSC/EPSG database.

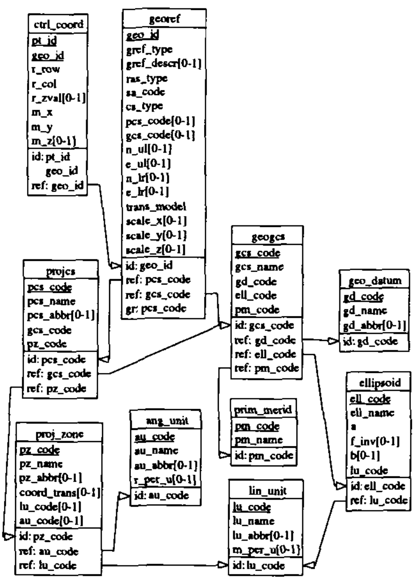


Figure 6–15: Logical schema of the GrIdS georeferencing concept which is based on the POSC/EPSG geodetic reference database.

Georeferencing User Interface

The implemented georeferencing user interface is shown in Figure 6–16. It supports the definition, editing or viewing of a raster object's georeference information. The process of georeferencing a raster object with this user interface follows the GeoTIFF convention which involves the steps listed below:

- selection of a raster type (area or point raster)
- selection of a coordinate system type (projected, geographic *or cartesian geocentric*) and subsequent selection of a projected or geographic coordinate system from the POSC/EPSG tables
- selection of a transformation model (tie-points only, tie-points and scale *or transformation matrix*)
- specification of raster-model scales (units per pixel) for the coordinate axes (opt.)
- specification of tie-points (opt.)

A detailed description of these georeferencing steps and of the underlying concepts and conventions can be found in (Ritter and Ruth, 1995).

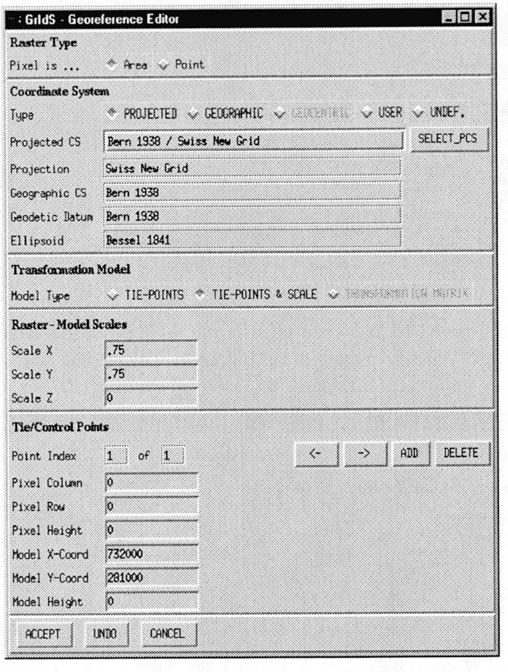


Figure 6–16: GrIdS menu for defining and editing the georeference of raster objects.

6.7 IMPLEMENTED MOSAIC MANAGEMENT SOLUTION

6.7.1 Mosaic Creation

The mosaic creation process sets up the logical structure defining a mosaic and inserts a metadata record into the database. Figure 6–17 shows the user interface for creating a raster mosaic in GrIdS. It supports the specification of the required mosaic metadata, including: name, type, coordinate system, origin, mosaic scale (pixel size), tile dimension, photometric type, compression, storage specification, etc.

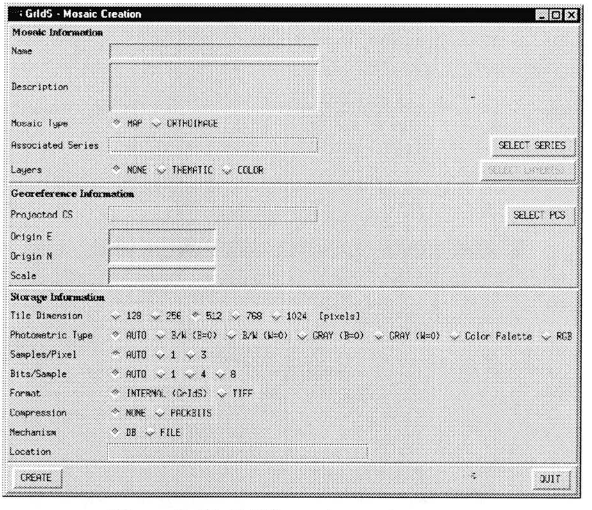


Figure 6–17: GrIdS mosaic creation menu.

A feature which proved to be very useful is the AUTO setting which is available for several parameters. Whenever this option is enabled the mosaic will inherit the value of the corresponding parameter from the first raster object inserted.

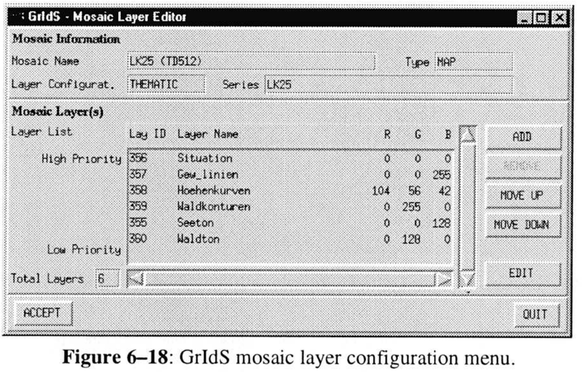


Figure 6–18: GrIdS mosaic layer configuration menu.

For multi-layered or multi-channel mosaics the additional layer configuration menu is provided (Figure 6–18). With this menu, layers can be created together with the required colour attributes and priorities.

6.7.2 Raster Data Projection between Raster Objects and Mosaics

Projecting or mapping raster data between raster objects and mosaics is the key operation in raster mosaic management. It enables the insertion of raster objects into mosaics and the opposite process, namely the extraction of raster objects.

The implemented solution is shown in Figure 6–19. It interfaces the strip-based partitioning concept for raster objects with the tile-based partitioning concept for raster mosaics. The key idea of the present solution is to process raster data on a '***one strip at a time***' basis. This has the consequence that only those tiles overlapping the currently active strip have to be loaded into primary memory. With this approach raster objects of arbitrary dimension and size can be inserted and extracted using a limited amount of primary memory.

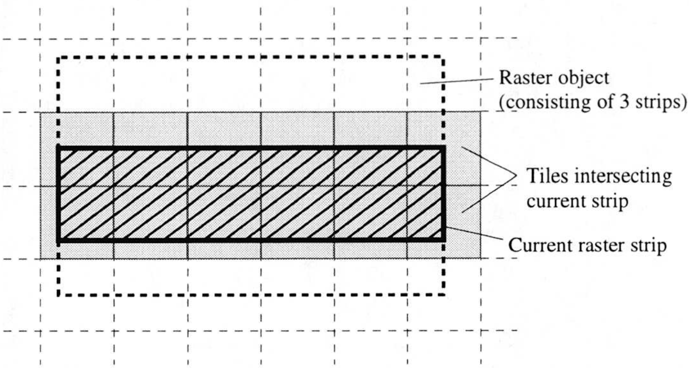


Figure 6–19: Illustration of concept used in raster object insertion and extraction for projecting raster data between raster strips and mosaic tiles.

The individual tasks involved in projecting raster data between raster objects and mosaics range from the simple copying of pixel values to the following more complex operations:

- radiometric transformations
- controlling of priorities (existing data has priority vs. inserted data has priority)
- simple geometric transformations (shifts, nearest neighbour re-sampling, etc.)
- *advanced geometric transformations (orthorectification, etc.)*
- *image fusion*

The more complex operations such as orthorectification or image fusion are typically better handled by specialised external systems than by a general purpose raster management system. For this reason, the current version of GrIdS supports basic transformation operations only and relies on external systems for more sophisticated pre-or post-processing.

6.7.3 Raster Insertion ('Mosaicking')

The raster insertion module of GrIdS allows multiple georeferenced raster objects to be imported into a raster mosaic. It could also be referred to as a basic 'mosaicking' module. In the current version raster objects are required to be georeferenced, orthorectified and aligned with the axes of the coordinate system. The necessary pre-processing tasks are typically supported by scanning modules of GISs or by orthorectification and mosaicking modules in remote sensing and digital photogrammetric systems. The user interface for inserting raster objects is shown in Figure 6–20.

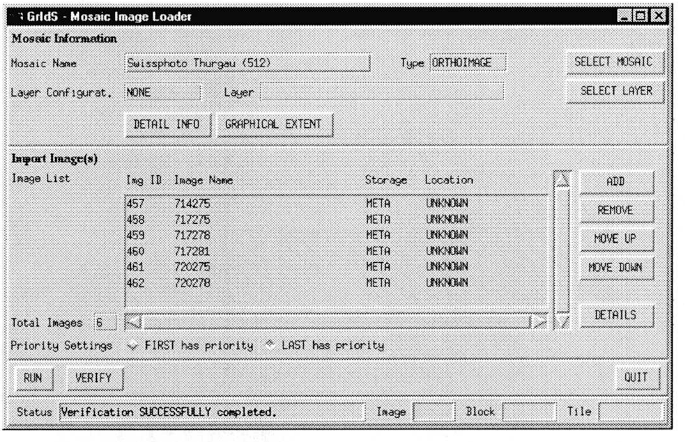


Figure 6–20: GrIdS interface for inserting a selection of raster objects into a raster mosaic or into a certain mosaic layer.

The raster insertion process itself is illustrated in Figure 6–21. The process is initiated with the selection of a raster object and a mosaic, plus optionally a mosaic layer. Based on the associated metadata, the range of spatial codes (SC-range) is determined and a linked list of the tiles inside the query window is established. A spatial range query such as shown in Figure 6–23 then returns the metadata of those tiles which exist in the database and adds this information to the tile list.

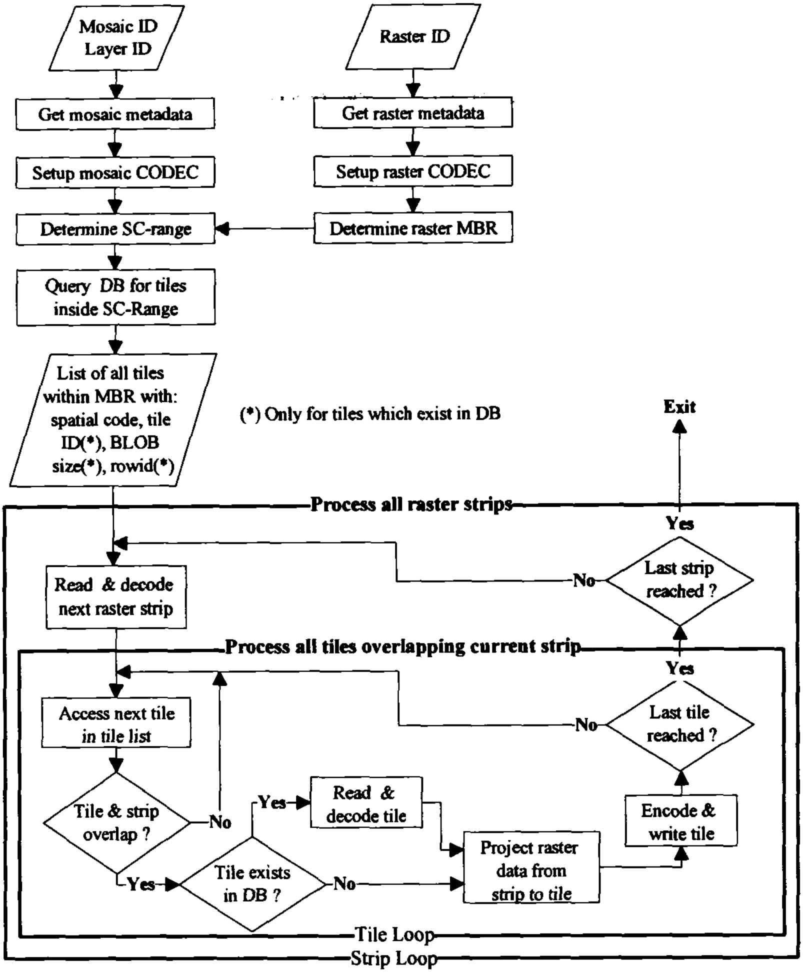


Figure 6–21: Flowchart summarising the process of inserting a raster object into a mosaic.

With the availability of all the required metadata, the actual raster data is subsequently inserted by the means of two loops. The external loop reads and decodes the data of the raster object in consecutive strips and writes it to the ***raster memory structure*** shown in Figure 6–9. The internal loop sequentially accesses each tile which overlaps the current strip. If the tile already exists in the database its data is retrieved, decoded and written to the ***mosaic memory structure*** also shown in Figure 6–9. If the tile does not yet exist, a new empty tile is created in the same structure. The raster projection operation subsequently maps the data from the present strip to the currently accessed tile (see

Figure 6–9 and Figure 6–19). Once a tile has been processed its data is encoded and written to DBMS-internal or -external storage.

6.7.4 Raster Extraction

The raster extraction process is in many ways similar to its opposite process described above. The actual extraction process is preceded by a spatial and thematic selection and by the specification of a number of output parameters. The user interface for selecting and subsequently extracting sub-sets of raster mosaics is shown in Figure 6–22. The features of the present menu are:

- Spatial window selection (numerically or graphically via a linked mosaic overview window)
- Thematic selection (layers or channels) in the case of multi-layered mosaics
- Editing of layer priorities and appearance (e.g. for correctly combining multiple cartographic layers)
- Photometric / radiometric options
- File format and compression
- Extraction operation: preview of write to file

Many of these parameters are proposed by the system on the basis of available metadata and default options.

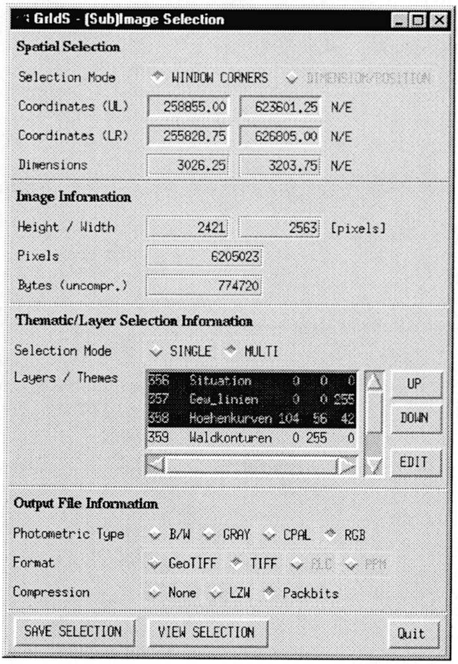


Figure 6–22: GrIdS interface for selecting and extracting a raster object from a multi-layered mosaic.

Once the selection is complete the extraction operation can be invoked. The process of extracting a raster object and writing it to file is illustrated in Figure 6–24. First, the spatial code range (SC-range) for the tiles overlapping the selection is determined. Then a linked tile list is established using the results of a spatial range query such as shown in Figure 6–23.

```
select s_code, ti.tile_id, blob_size, td.rowid
     from tile_info ti, tile_data td
     where mos_id = MOS_ID and lay_id = LAY_ID
     and s_code between SC_MIN and SC_MAX
     and ti.tile_id = td.tile_id
     order by s_code
```

Figure 6–23: Spatial range query returning information on tiles within the query window, i.e. tiles with spatial codes between *SC_MIN* and *SC_MAX* (see discussion in Section 5.4.1)

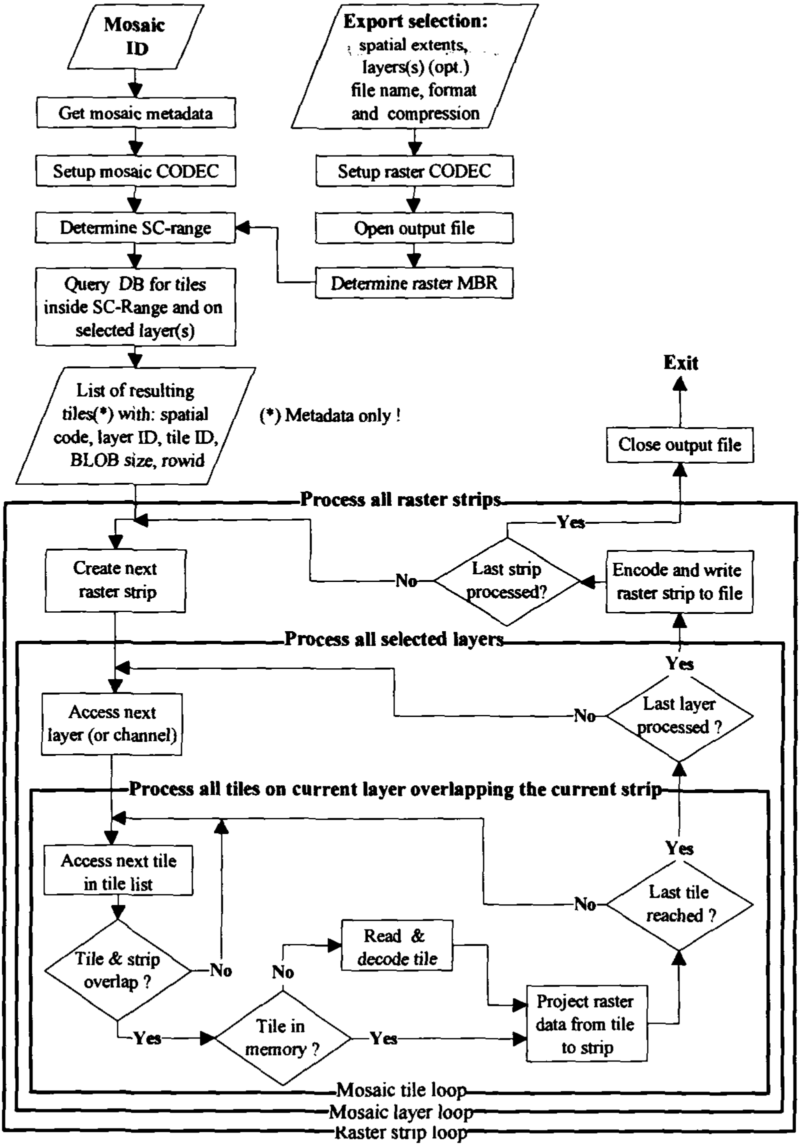


Figure 6–24: Flowchart illustrating the process of extracting a raster object from a raster mosaic and writing it to a file.

Based on the information held in the linked tile list the actual raster data can be extracted by the means of the three loops shown in Figure 6–24. The outer loop is again the one for processing consecutive raster strips. During each cycle of this loop a new strip is created in the ***raster memory structure*** shown in Figure 6–9. If data from multiple mosaic layers or channels is to be extracted, then a middle loop is initiated. Within each cycle of this layer loop, all the tile data which belongs to the active layer and which at the same time overlaps the active strip is processed. The processing of the actual tile data, including the raster projection operation which was presented in Section 6.7.2, occur in the inner or tile loop. Whenever the outer loop is reached again, i.e. whenever a strip in the raster structure has been completely written, the strip is encoded and written to raster object storage. This storage medium would typically be the file system.

It should be noted that the currently implemented strip-layer-tile order of the three processing loops is not mandatory. For data with large numbers of channels such as multi-spectral remote sensing data an alternative strip-tile-layer order could be added in the future. In such a scenario all layers corresponding to the current tile would be processed before moving to the next tile in the list.

6.7.5 Mosaic Visualisation

Different concepts for visualising raster objects were discussed in Section 5.9. This discusses some aspects of the mosaic visualisation solution currently implemented in GrIdS. It also points out the shortcomings and possible future improvements and extensions. The following list summarises the main approaches for visualising raster mosaics at increasing levels of detail:

- Mosaic bounding box representation
- Vector-based *or raster-based(*)* mosaic tile summary *(*) not implemented*
- Visualisation of individual tiles
- *Multi-resolution mosaic visualisation (*)*

The first approach uses georeference metadata, the next is based on tile metadata and the last two techniques are content-based. The second approach of visualising tile summaries is particularly interesting. It enables a metadata-based visualisation at an intermediate level of detail. The characteristics of this visualisation concept with its vector- and raster-based variants were presented in Section 5.9.4.

In GrIdS only the vector-based raster mosaic overview concept was implemented (see Figure 6–25). It was mainly the experience with this vector-based implementation, which triggered the investigations of the raster-based alternative which is also discussed in Section 5.9.4. The vector-based tile summary approach provides a great amount of flexibility in assigning attribute information to individual tiles and it performs well with relatively small mosaics. However, with large mosaics consisting of several tens of thousands of tiles the current implementation proved to be inefficient. Significant improvements could be expected by moving some of the display functionality from Tcl/Tk to C. However, the raster-based solution has some conceptual advantages which should be exploited in a future implementation (see Section 5.9.4).

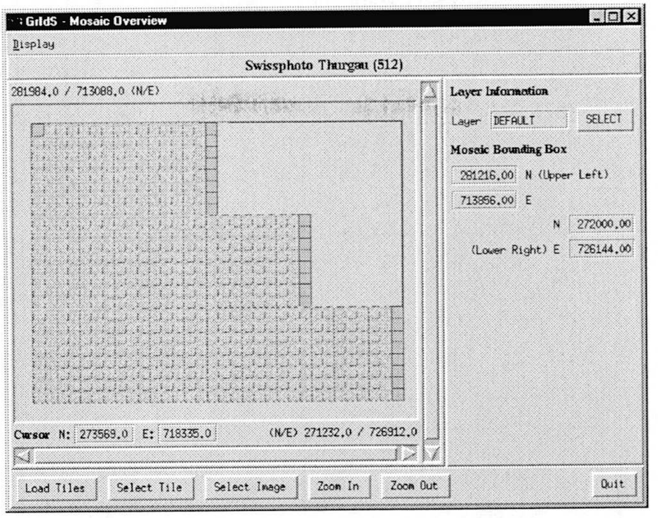


Figure 6–25: Grids interface for visualising overviews and contents of raster mosaics. In this example, the bounding box (outer rectangle) and the existing tiles (grey rectangles) of a partially populated raster mosaic are shown. They enable a vector-based metadata representation of raster mosaics at two different levels of detail.

Leer - Vide - Empty

7

TESTS AND RESULTS

The functionality and performance of the GrIdS prototype system were evaluated and measured by a number of tests. The tests were carried out on a standard, relatively old hardware platform using real-world data sets. The tests focused almost exclusively on the functionality and performance for handling raster mosaics. This chapter summarises and discusses the results of these tests. It particularly highlights some of the advantages and special features of the current solution and points out limitations and possible improvements.

7.1 TEST PLATFORM AND CONFIGURATION

The spatial raster management concepts and the prototype system presented in the previous chapters were designed to run on standard hardware and DBMS technology. This was put on trial on the hardware platform available at the time, whose technology was more than 6 years old at the time of the tests.

7.1.1 System Configuration

IBM RS/6000 Model 530

The platform available for the system implementation and for the subsequent tests was an IBM RS/6000 workstation Model 530 which represented the technological standard of 1990. The main system characteristics were a 25 MHz POWER™ RISC-CPU with

34 MIPS[10] and 10.9 DP-MFLOPS[11] (Stengele, 1995). In terms of the current SPEC95 performance benchmark this is roughly equivalent to 0.8 SPECint_base95[12] and 1.8 SPECfp_base95[13] or the rating of a 66Mhz Intel 486-based PC system. The test system was equipped with 96 MB of physical primary memory and 200 Mb of swap space and was running under the operating system AIX Version 3.

StorageWorks Fast-Wide-SCSI Hard Disk Array

The available data storage subsystem consisted of a StorageWorks fast-wide-SCSI disk array with five 4 GB hard disks and a total amount of storage space of 20 GB. The disk array was connected to the host system through a fast-wide SCSI controller with a theoretical bandwidth of 20 MB/s (10 MHz at 16bit) (Ehrig and Maier, 1995). No hardware- or software-based RAID[14] support and no disk-striping were available on the platform and under the operating system version used.

GrIdS on Oracle Universal Server 7.3

The tests were carried out using GrIdS on Oracle Server Version 7.3 in the configuration described in Chapter 6. The System Global Area (SGA) of the Oracle DBMS was set to a total size of 25MB. Data access was performed through the UNIX file system. The rollback tablespace with a total size of 200 MB was placed on a disk separate from the tablespace which held all the raster data and which was distributed across the other four hard disks.

Local Database Scenario

In the test configuration the database server (Oracle) and the client application (GrIdS) were running on the same workstation. With this arrangement any network delays could be avoided and a maximum data throughput between server and client could be ensured. However, running all processes on the same system increased the demands with respect to main memory and CPU utilisation.

7.1.2 Test Data

Provided Raster Data Sets

For the subsequent tests two major data sets were used, a set of raster maps and a set of orthoimages. A short description of these data sets is given below and their key figures are presented in Table 7–1.

- ***Pixel Maps*** or 'Pixelkarten' were provided by the Swiss Federal Office of Topography in Bern. The available data set consisted of 4 by 4 maps sheets at 1:25'000, covering the total area of 70 km x 48 km. Each map sheet consisted of 6 separately

[10] MIPS = Million Instructions Per Second

[11] DP-MFLOPS = Million FLOating Point OPerations in Double Precision

[12] SPEC CPU95 integer / non-floating point benchmark composite score

[13] SPEC CPU95 floating point benchmark composite score

[14] RAID = Redundant Arrays of Independent Disks

scanned and geocoded binary layers. The scanning resolution was 20 l/mm resulting in a geometric resolution of 1.25 metres per pixel.

- ***SwissPhoto*** orthoimages were provided by SwissPhoto Surveys in Regensdorf. A set of 28 orthoimages from an area of 36 km x 15 km were available. The provided SwissPhoto images were 24-bit RGB data sets with a geometric resolution of 0.75 metres.

The data sets were provided as TIFF files without explicit georeference information. They were georeferenced and registered in GrIdS using the Raster Import module.

	No. of Raster Objects	No. of Layers	No. of Files	Radio-metric Res.	Dimensions [pixels]	Size / File (uncompr.) [bytes]	Total Size (uncompr.) [MB]
Pixel Maps	16	6	96	1 bit	14000x9600	16'800'000	**1'575**
SwissPhoto	28	1	28	24 bit	4000x4000	48'000'000	**1'125**

Table 7–1: Summary of the test data sets used: Pixel Maps (scanned raster maps) and SwissPhoto (orthoimages).

Created Raster Mosaics

From the registered raster objects a number of raster mosaics with different tile dimensions and compression schemes were generated. These mosaics are summarised in Table 7–2 below:

Mosaic Name	Mosaic Tile Dimension	Compress. Scheme	Radio-metric Resolution	Average Tile Size [bytes]	Number of Tiles	Total Size [MB]
Raster Map Mosaics (RMMs)						
RMM 512 PB	512x512	PackBits	1 bit	6093	46509	270.3
RMM 768 PB	768x768	PackBits	1 bit	34501	962	31.7
RMM 1024 PB	1024x1024	PackBits	1 bit	56690	980	53.0
RMM 512 NC	512x512	None	1 bit	32768	4329	135.3
RMM 768 NC	768x768	None	1 bit	73728	962	67.6
RMM 1024 NC	1024x1024	None	1 bit	131072	1120	140.0
Orthoimage Mosaics (OIMs)						
OIM 128 NC	128x128	None	24 bit	49152	6946	325.6
OIM 256 NC	256x256	None	24 bit	196608	1039	194.8
OIM 512 NC 1	512x512	None	24 bit	786432	272	204.0
OIM 512 NC 2	512x512	None	24 bit	786432	1455	1091.3

Table 7–2: Summary of raster mosaics generated as part of the test campaign.
(PB = PackBits compression, NC = no compression)

⇨ At the time of the tests the database was holding over ***3GB of mosaic data*** consisting of more than ***77'000 tiles***.

7.2 PERFORMANCE TESTS

The practical performance tests focused on the novel aspects of the investigations, namely on the DBMS-integrated management of raster mosaics. The performance of the developed solution was tested for the three typical tasks: raster object import, raster object export and, most importantly, window retrieval. The tests were primarily based on measuring the ***system run-time*** for entire operations or sub-processes. For the present transaction-intensive environment with heavy disk access and I/O activity, run-time measurements yield more realistic results than CPU-timings which primarily reflect the processing load of a specific operation.

7.2.1 Raster Object Import

In a typical data import scenario, one or several relatively large raster objects such as orthoimages or scanned maps are inserted into a mosaic or into a mosaic layer. Prior to the import operation, these raster objects have to be georeferenced as well as geometrically and radiometrically corrected. GrIdS supports the georeferencing process and delegates the other tasks to specialised systems, e.g. orthoimage production systems.

The objective of this first series of tests was to determine the data import rate for different types of raster objects and for mosaics with different tile dimensions and compression schemes. The tests were performed by measuring the time required to import a selection of the Pixel Map and SwissPhoto test data sets into the mosaics listed in Table 7–2. The results are summarised in Table 7–3. The import rates derived from these results are illustrated in Figure 7–1.

Mosaics:	RMM 512 PB	RMM 768 PB	RMM 1024 PB	RMM 512 NC	RMM 768 NC	RMM 1024 NC	OIM 128 NC	OIM 256 NC	OIM 512 NC
Object Size[15] [MB]	16.0	16.0	16.0	16.0	16.0	16.0	45.8	45.8	45.8
No. of Tiles	570	247	148	570	247	148	1024	264	77
Avg. Time [s]	**240.0**	**143.0**	**121.0**	**296.8**	**232.1**	**261.0**	**663.0**	**674.0**	**953.0**
Imp. Rate [KB/s]	68.7	115.3	136.2	55.5	71.0	63.1	70.7	69.6	49.2

Table 7–3: Results summary of mosaic import performance tests.

The determined data import rate is between 50 and 140 Kbytes per second. At first sight this might appear rather low. It is therefore worthwhile to look at the individual tasks which are part of the entire import process and which are illustrated in Figure 6–21. The main and expensive operations are: reading (and possibly decompressing) data from file, projecting the raster object data to the tile data structure, and finally compressing and writing the tiles to the database. In the case of Oracle this last step involves copying the data from client to server memory and writing a copy of the data to the rollback segment.

[15] Uncompressed average raster object size with bits contiguously packed into bytes (i.e. 8 bits per byte).

Even though the import operation will often be performed in batch mode or as a background process, the data import performance of the tested configuration is probably not satisfactory for an operational environment. The reasons for the current limitations and performance improvements will be discussed in Section 7.4.

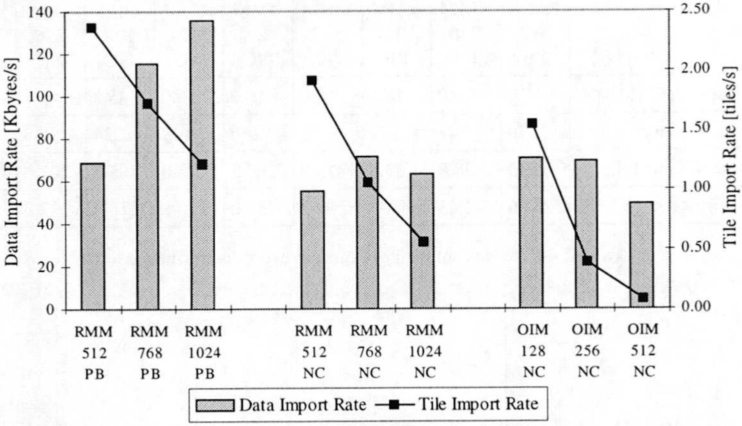


Figure 7–1: Mosaic data import performance for raster mosaics of different type (raster maps and orthoimages), tile dimension and compression scheme.

7.2.2 *Raster Object Export*

The export of (large) raster objects from raster mosaics is the reverse process of the import process described above. The significance and frequency of use of this operation largely depends on the type of raster management application. The export of entire map sheets from raster mosaics to file is a typical example for this type of operation. In our case, these raster object export tests permitted the analysis of the continuous or 'stream' export performance of the system. They also enabled the subsequent comparison of the import and the export performance. The process of exporting raster objects from mosaics involves the following steps and operations:

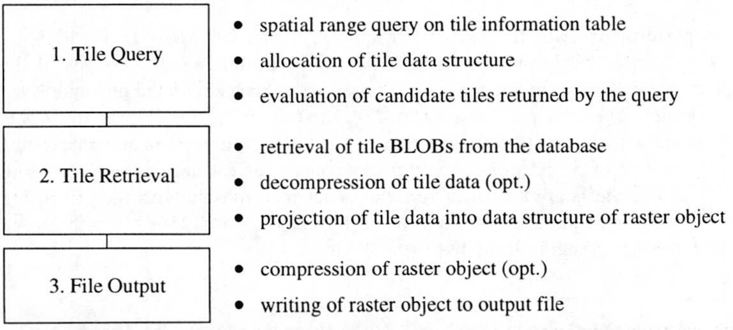


Figure 7–2: Main steps of the raster object export process.

The data export tests were carried out by extracting and re-exporting the raster objects, which had previously been imported as part of the import tests. The results of the export tests are listed in Table 7–4 and illustrated in Figure 7–3.

	RMM 512 PB	RMM 768 PB	RMM 1024 PB	RMM 512 NC	RMM 768 NC	RMM 1024 NC	OIM 128 NC	OIM 256 NC	OIM 512 NC
Object Size[16] [MB]	16.0	16.0	16.0	16.0	16.0	16.0	45.8	45.8	45.8
Tiles Retrieved	570	247	148	572	250	148	1024	264	77
Export Time [s]	**65.3**	**38.8**	**34.8**	**78.8**	**69.3**	**73.0**	**205.5**	**202.0**	**265.8**
Export Rate[17] [KB/s]	252.6	425.3	474.3	209.2	238.0	225.8	228.3	232.2	176.5

Table 7–4: Results summary of mosaic export performance tests.

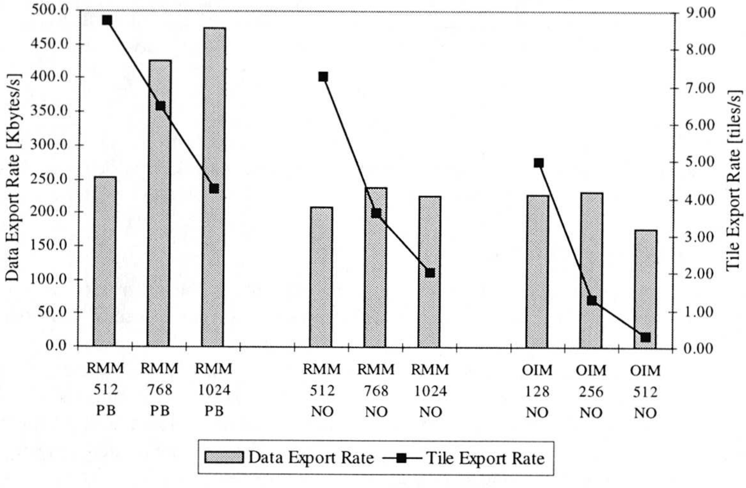


Figure 7–3: Mosaic data export performance.

The export performance which ranges from approx. 0.2 to 0.5 MB/s is significantly better than the import performance. As with data import, compressed versions of the raster map mosaics (left) yield a significantly better performance than the uncompressed versions (middle). The best performance of 474 Kbytes/s is achieved with the *RMM 1024 PB* mosaic with 1024x1024 tile dimension, PackBits compression and an average compressed tile size of 55 Kbytes. The optimal export rates for uncompressed mosaic data of 230-240 Kbytes/s are obtained from the raster map mosaic with tile dimension 768x768 and a tile size of 72 Kbytes and from the 128x128 and 256x256 orthoimage mosaics with tile sizes ranging from 48 to 192 Kbytes.

[16] Uncompressed average raster object size with bits contiguously packed into bytes (i.e. 8 bits per byte).

[17] Export rate = export time / raster object size.

7.2.3 Comparison of Import and Export Performance

The import and export performance for different mosaic tile dimensions and compression schemes are summarised in Figure 7–4. The comparison reveals a largely 'parallel' behaviour with a consistent ratio between export and import performance ranging from 3.2 to 3.8. On an average, the export data rate is 3.5 times quicker than the corresponding import rate. This difference is primarily caused by the fact that inserting data into a database is a data manipulation operation which involves the use of rollback segments. Rollback segments are a fundamental part of the transaction concept which allows the last consistent database state to be restored following a failed database operation or a system crash. As part of an insert or update transaction, a copy of the data is written to disk into one of the assigned rollback segments. This immediately at least doubles the amount of data to be written to disk. Other factors contributing to the lower import performance include: the requirement to retrieve and update tiles located at the border of the imported object, the additional database activities of tile 'header' record insertions and associated index updates, and the general difference between disk write and read performance.

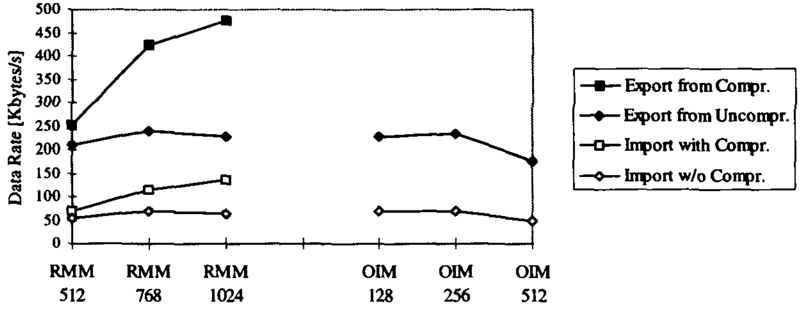


Figure 7–4: Import and export performance in function of mosaic type, tile dimension and compression scheme.

Another aspect investigated was the influence of the tile data size, rather than the tile dimension, on the import and export performance. For this comparison, the previous results for the mosaics without data compression were used. This allowed bilevel and RGB mosaic data to be included in this test. The import and export performance as a function of the tile data size are illustrated in Figure 7–5 below. The diagram shows maximum import and export rates for tile sizes between 50 and 200 Kbytes and a peak rate at a tile size of 72 Kbytes for the RMM 768 NC mosaic. This corresponds well with the results presented in (Lamb, 1994) which showed an optimal retrieval performance from an Oracle database for tile sizes between 16 and 160 Kbytes[18]. Figure 7–5 also contains trend curves for the import and export performance. However, they are to be treated with great caution due to the absence of observations for tile dimensions between 300 and 700 Kbytes.

[18] Results in (Lamb, 1994) are quoted in terms of tile dimension for mosaics with presumably 8-bit raster data (1 byte/pixel). For example: 128x128 pixels = 16 Kbytes.

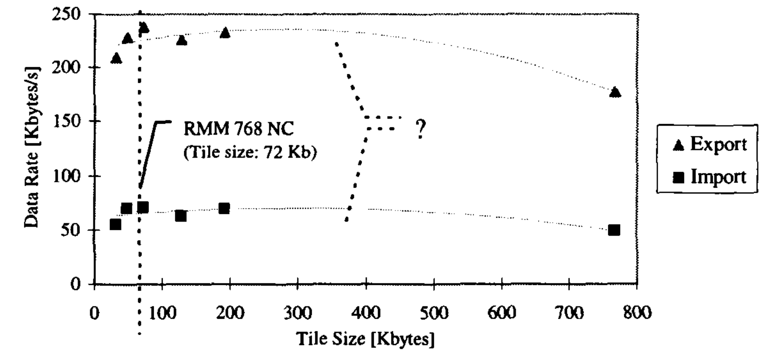


Figure 7–5: Data import and export performance for uncompressed mosaic data in function of the tile size.

7.2.4 Window Retrieval

An operation more important than that of importing and exporting large raster objects is the selection and retrieval of limited size windows. The dimensions of these windows are often guided by the dimensions and resolution of display devices. Today's typical medium- to high-resolution monitors have a resolution of roughly 1200x1000 pixels. However, quite often only parts of the screen are occupied by the window(s) displaying the actual raster data. The window retrieval process involves steps 1 (tile query) and 2 (tile retrieval) of the raster export process shown in Figure 7–2.

Mosaic: / Window Dim:	RMM 512 PB		RMM 768 PB		RMM 1024 PB		RMM 512 NC	RMM 768 NC	RMM 1024 NC	OIM 128 NC		OIM 256 NC		OIM 512 NC	
	T	No.	T	No.	T	No.	T	T	T	T	No.	T	No.	T	No.
512x512	**0.8**	4.0	**0.6**	3.2	**1.0**	1.5	**2.0**	**1.0**	**0.7**	**2.8**	25.0	**3.0**	9.0	**4.3**	4.0
768x768	**1.4**	6.4	**1.1**	4.0	**1.0**	1.8	[19]	[19]	[19]	**7.0**	49.0	**5.5**	16.0	**7.0**	6.0
1024x1024	**1.1**	9.0	**1.0**	4.7	**1.5**	4.0	**2.7**	**1.2**	**1.2**	**9.0**	81.0	**7.8**	25.0	**8.0**	9.0

Table 7–5: Average window retrieval times (T in [seconds]) (bold) and average number of tiles (No.) retrieved for different window dimensions (left column) and different mosaics (top row).

The timings of this third test included the time required to perform the tile query, to access and retrieve the tiles, and to compose a single contiguous raster object or 'image' in main memory. At this final stage the raster object would have been ready for display by a graphics system, for transmission over a network or for a subsequent file export. However, the times provided in Table 7–5 and illustrated in Figure 7–6 do not include these display or export operations.

[19] not measured

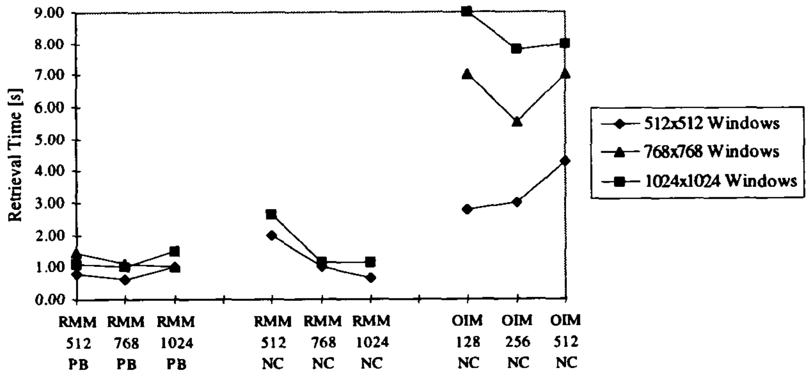


Figure 7–6: Retrieval times for different window dimensions and mosaic types.

The test results yield retrieval times for all ***raster map windows*** in the range of ***0.6 to 1.5 seconds***. The effect of different window dimensions becomes more evident with 24-bit orthoimage data. The window retrieval times for ***orthoimage windows*** range from ***2.8 seconds for 512x512 windows*** to ***7.8 seconds for 1024x1024 windows***.

A closer inspection of the higher retrieval values for the mosaics RMM 512 NC and OIM 128 NC reveals tile query times (see Figure 7–2) of 1.3 - 1.8 seconds instead of the normal 0.1 - 0.5 seconds. This anomaly is caused by a larger tile oversearch resulting from the missing tile query optimisation, which was not implemented in the prototype system (see Section 5.4.2). The tile oversearch problem will be addressed in detail in Section 7.4.3 below.

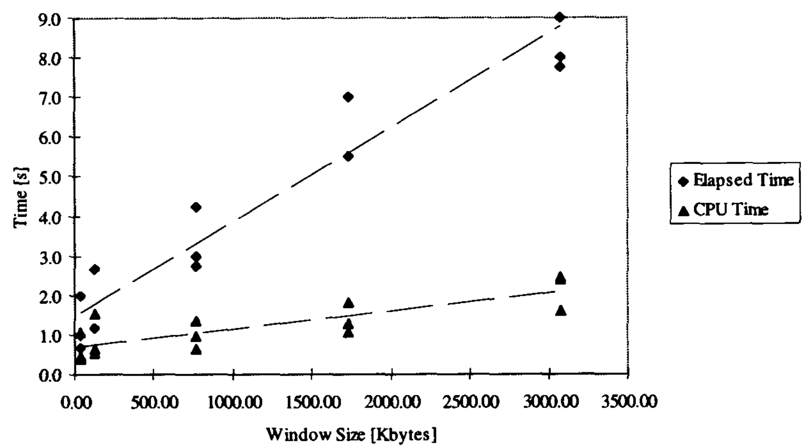


Figure 7–7: Window retrieval times (for all mosaic types) in function of window data size.

The amount of data to be retrieved for specific window dimensions largely depends on the type of raster data held in the mosaic, i.e. on the attribute resolution. Therefore, the window retrieval times shown in Figure 7–6 cannot easily be transferred to other types of raster mosaics. A generally applicable window retrieval performance summary using the same test data is provided in Figure 7–7. It shows the elapsed system and CPU times which are required to retrieve a certain number of bytes. In order to permit a direct comparison, only the timings for uncompressed mosaic data were used. The diagram contains the values for all tested tile dimensions. This allows the bandwidth of the achieved results to be illustrated and it also leads to conservative average performance figures. Using a linear regression model, the following equations for the window retrieval times and CPU time can be derived:

$$\text{Time [s]} = 1.47 + (0.0024 * \text{Window\_Size [Kbytes]})$$

$$\text{CPU [s]} = 0.69 + (0.0005 * \text{Window\_Size [Kbytes]})$$

It is evident, that these equations lead to too pessimistic results for small window sizes. However, they give a useful approximation for medium to large windows.

Example: Retrieval of 1024x1024 window from 8bit greylevel orthoimage mosaic: window size = 1024 Kbytes ⇨ estimated time = 3.9 s, estimated CPU time = 1.15 s.

The results of this example compare favourably with those presented in (Lamb, 1994) with CPU times[20] for the same window size ranging from 2.5 to 5.5 seconds, depending on the mosaic tile dimension.

7.3 System Capabilities

Layer Management

The implemented mosaic layer management functionality proved to be very useful. For example, raster map layers can be exported individually as bilevel or RGB raster objects or as RGB combinations with arbitrary colour assignment. The current solution also supports certain layer management operations on a SQL level, e.g. the deletion of mosaic layers:

```
delete from tile_info, tile_data
      where mos_id = XXX
      and lay_id = YYY
      and tile_info.tid = tile_data.tid;
```

Registration of Raster Objects

The system supports DBMS-external storage for raster objects and allows them to be georeferenced and registered in the database without requiring the import of the actual raster data. This was a very useful feature during the system tests where the same raster objects had to be selected and used several times.

[20] The times provided in (Lamb, 1994) are explicitly quoted as CPU times.

Mosaic Updates

The DBMS-based raster management solution makes the update of mosaic data a straight-forward and frequently used operation. Mosaic updates, for example, are required during the import process for adding data to partially occupied tiles and for the cartographic editing of areas along the original map sheet boundaries. The mosaic update process is approximately 20% faster than the insertion or import process.

Concurrency and Multi-User Support

Concurrent data access as one of the standard DBMS services was successfully verified by concurrently importing and retrieving raster objects into and from the same raster mosaic. These concurrent tasks were performed by multiple GrIdS client processes, which, for test purposes, were running simultaneously on the same workstation.

Transaction Management

The transaction concept allows sequences of database operations to be grouped together and to be undone in case of a problem. This, for example, would permit all tiles affected by a raster object to be inserted or updated, the results to be visually inspected and, finally, the transaction to be committed or rolled back depending on whether the results were OK or not. However, the large size of typical spatial raster objects requires the use of multiple large rollback segments, whose total size would be several times the total amount of the data inserted or updated in the course of such a transaction. Two transaction options were implemented and tested: a strip-based and a tile-based solution. In the strip-based approach a transaction is completed upon insertion or updating of a raster strip or an entire raster object, if it fits into the strip memory limit (see Section 6.7.2). In the tile-based approach a new transaction is invoked for each tile. The comparison between the two transaction options revealed no significant differences in import performance.

7.4 CURRENT LIMITATIONS

7.4.1 System Limitations

Physical Memory Limitations

A limitation which affected the import and export performance for large raster objects was the amount of main memory available on the test platform (96 MB). In a local database scenario, primary memory has to be shared between the operating system, the DBMS server processes and database buffers as well as the client processes and data segments. The GrIdS prototype system provides a run-time option which allows the amount of client memory to be controlled via the raster strip concept described in Section 6.7.2. However, reducing the maximum strip size automatically increases the number of tiles at the strip border which have to be accessed multiple times. In an ideal case, the entire raster object fits into one strip, which requires only a single access to each affected tile. As part of the tests different strip size limits were tested. Typically strip sizes larger than 20 MB would lead to an excessive amount of memory swapping activity. However, on one occasion, a 48 MB raster object was exported with a 50 MB strip size in a time roughly 30% shorter than the otherwise very consistent average export time (see Figure 7–8). The only plausible explanation for this performance increase is an optimal memory page configuration which required little or no paging activity for this specific case. This example primarily indicates that even on this slow system a certain performance improvement could be achieved by adding more main memory.

Compression Performance

The modest CPU performance of the test system is illustrated in Figure 7–9. Compressed output time is almost three times slower than the time which is required for uncompressed file output. A closer analysis reveals that around 80% of the time is used for the relatively simple PackBits compression using the algorithm of the TIFF library. On a modern system with an integer CPU performance improvement of at least a factor of 10-20, the combined cost of compression and compressed file output should be lower than uncompressed file output alone. The observed average file-write data rate of 2.3 MB/s for large objects on an internal SCSI-2 hard disk is consistent with the results presented in (Ehrig and Maier, 1995).

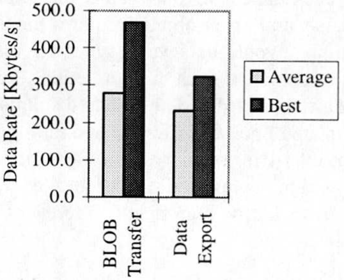


Figure 7–8: Comparison of best export performance observed with average performance.

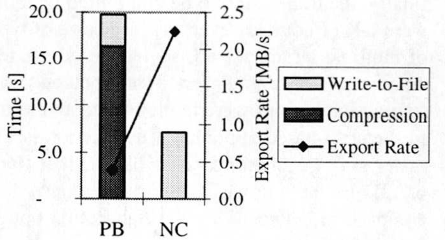


Figure 7–9: Comparison of compressed (left) and uncompressed (right) file output of a 16.8 MB raster map object.

7.4.2 Storage Management

External Storage Support

As part of the tests, only DBMS-internal storage was investigated. DBMS-external, file system-based storage was not implemented in the prototype system at the time of the tests. This was primarily due to the considerable requirements to develop an external storage solution which would be system independent, which would not be affected by file number limitations, which would permit efficient access to very large numbers of tiles and which would also incorporate a suitable data security concept. With the announced integration of file-based storage support in major (object-) relational DBMSs such as in Oracle8 (Oracle, 1997), the decision against developing a proprietary external storage solution proved to be appropriate.

Single- versus Multi-Table Raster Storage

In the tested prototype system, the header information of the over 77'000 tiles was stored in a single table (see also Figure 6–8). The actual tile data, in excess of 3 GB, was also stored in a single database table. This was an ideal scenario for testing the access performance and for simulating large individual future mosaics. The tablespace concept of Oracle which allows database tables to be spread across multiple hard disks proved to be very useful. The multi-Gigabyte size of a single database table caused no problems and no size-related performance degradation was observed.

However, there are some good reasons for assigning the raster data of individual mosaics to separate tables. The 'table-per-mosaic' approach, for example, would allow mosaics to be deleted from the database with a 'drop table' DDL command. The execution of this command is much faster than that of the 'delete' DML command which includes the expensive operation of writing a copy of the entire mosaic to the rollback segment. Deleting a 3 GB mosaic with the 'delete' command from a system with an optimistic rollback write rate of 2 MB/s would still require approx. 30 Minutes. Another argument for the 'table-per-mosaic' approach might be the ease of backing-up and restoring individual mosaics.

7.4.3 Spatial Tile Query Performance and Optimisation

The spatial query optimisation described in Section 5.4.2 had not been implemented at the time of the tests. The average and worst-case performance of the basic spatial query solution was analysed by monitoring the query times and the number of candidate tiles returned. The average results for the different window retrieval tests were[21]:

- average query times: 0.1 - 0.2 seconds
- average query CPU times: 0.0 - 0.1 seconds
- ratio (candidate tiles returned / tiles required): 1.5 - 2.5

The results presented in Table 7–6 represent the worst cases observed during the test series. For two windows, the oversearch factor, i.e. the ratio between the number of candidate tiles returned by the query and the number of tiles actually required, was around 1366. However, it is interesting to notice, that the effect on the elapsed query time and CPU is only significant in case of the RMM 512 PB mosaic but not in case of the RMM 768 PB mosaic. The explanation for this is simple: the query time is primarily affected by the number of candidate tiles which are actually found in the mosaic. For these returned candidate tiles, the relatively expensive spatial code evaluation has to be performed in order to ultimately determine whether it intersects the query window or not. In the case of the relatively small RMM 768 PB mosaic only a few tiles existed within the specific oversearch area, which required the evaluation of only a fraction of the actual candidate tiles.

[21] not including the worst cases discussed below, which clearly require optimisation

Mosaic	RMM 512	RMM 512	RMM 512	RMM 768	RMM 768
Window Dim.	512x512	512x512	512x512	768x768	768x768
Without Query Optimisation:					
SC(START) (row/col)	578282 (200/671)	579818 (200/687)	580330 (200/703)	182971 (133/447)	189115 (133/479)
SC(END) (row/col)	579649 (201/672)	580161 (201/688)	585793 (201/704)	188436 (134/448)	190484 (134/480)
Tiles Queried	1368	344	5464	5466	1370
Time / CPU	3 s / 1.5 s	2 s / 1.0 s	3 s / 1.7 s	0 s / 0.2 s	0 s / 0.2 s
Tiles Required	4	4	4	4	4
Ratio	**342.0**	**86.0**	**1366.0**	**1366.5**	**342.5**
With Query Optimisation (single optimisation step):					
Query 1	578282-578283	579818-579819	580330-580331	182971-182974	189115-189118
Tiles Queried	2	2	2	4	4
Query 2	579648-579649	580160-580161	585792-585793	188433-188436	190481-190484
Tiles Queried	2	2	2	4	4
ΣQueried 1&2	4	4	4	8	8
Tiles Required	4	4	4	4	4
Ratio	**1.0**	**1.0**	**1.0**	**2.0**	**2.0**

Table 7–6: Efficiency of spatial mosaic range queries: without query optimisation (top) and with query optimisation (bottom). Examples shown are worst-case results observed as part of the window retrieval tests.

The effects of a single optimisation step are nicely illustrated in the bottom section of Table 7–6. A single optimisation step which splits the query range into two ranges (see Figure 5–7) dramatically reduces the oversearch ratio for the given examples to values between 1.0 and 2.0. In these cases, no benefits would be gained from a second optimisation step. However, the two-step optimisation solution is required for windows intersecting a horizontal as well as a vertical high-order quadrant boundary.

7.4.4 Database Performance Tuning

In most relational DBMSs such as Oracle, significant performance improvements can be gained from an optimal configuration and tuning of the operating system and database parameters. As part of the performance tests a selection of alternative parameter settings were evaluated with the primary goal to verify the robustness of the chosen standard configuration. Only a limited set of operating system parameters could be evaluated due to the unavailability of the corresponding features (e.g. disk striping) and monitoring tools on the installed operating system version (AIX 3.2.5). The database parameters were evaluated by testing the effects of the following alternative settings. For example, the use of a smaller Shared Pool size of 6-10 MB instead of the normally used 20 MB reduced the import performance by nearly 50%. The use of larger rollback buffers, on the other hand, had no significant effect on the system performance. The main difficulty in tuning the database parameters is caused by the poor to non-existent documentation of the long raw data type in any of the available Oracle documentation and literature.

7.5 SUMMARY

The GrIdS prototype system was successfully implemented and tested on a relatively old, low-performance UNIX platform. The test database consisted of a 3 GB raster database with more than 77'000 mosaic tiles and with a largest individual mosaic of more than 1 GB in size. The database contained (single-layer) orthoimage mosaics and multi-layer raster map mosaics. The results of the tests can be summarised as follows:

- The tests demonstrated the system's capability to import and export raster objects of arbitrary dimension and size. The observed data import performance is relatively low. This is primarily due to the inclusion of the raster data in the transaction and rollback mechanisms of the Oracle DBMS in combination with a modest to low system performance. The export performance is consistently better by a factor of 3-4.
- The prototype system shows a relatively good window retrieval performance from the above-mentioned 3 GB database with retrieval times ranging from 0.6 to 8 seconds – depending on the window dimensions and on the type of mosaic data.
- The spatial tile query concept proved to be very efficient with very low average query times on a fairly large number of tiles. However, the few observed 'worst-case' examples demonstrate the need for and the dramatic effects of the proposed query optimisation.
- The chosen DBMS proved to be very robust and scaleable. For example, all of the mosaic raster data was held in a single database table with no evidence of size-related performance degradation. Several of the standard DBMS features, such as concurrency, were successfully tested.

The tests primarily demonstrated the feasibility of the proposed and implemented solution. Major performance improvements, compared to the results presented, can be expected on a modern platform.

Future System Requirements

The system specifications for a production system largely depend on the type of application and on factors such as number of users, type and volume of data. ***Scalability*** as one of the key features of DBMS-based solutions allows a broad range of requirements to be covered. A minimum system configuration for a spatial raster management system should include the following components:

- high-performance single- or multi-processor system
- min. 256 MB of main memory (for the local database scenario)
- multiple high-performance disks and disk controllers (min. Ultra-Wide SCSI)
- with RAID support (software or hardware)
- optional use of solid state disks (SSDs) for rollback segments

Leer - Vide - Empty

8

CONCLUSIONS AND OUTLOOK

8.1 SUMMARY AND CONCLUSIONS

The presented investigations were motivated by the growing gap between the rapidly increasing production of spatial raster data and the almost complete absence of spatial raster management solutions.

The increasing production and the growing use of raster data in the geomatics community is the result of several factors. One is the well-established, cost-effective use of scanned maps and aerial photographs, of which the latter was an important factor in fully establishing digital photogrammetry. Another reason is that digital photogrammetry now allows spatial raster products such as DTMs and orthoimagery to be generated largely automatically. In the future, high-resolution remote sensors, airborne laser-scanners and digital aerial cameras will further promote the use of spatial raster data and at the same time increase the demands for spatial raster management.

Existing raster-based systems such as raster GISs, digital photogrammetric stations, and satellite image processing products are almost exclusively focused on the aspect of data processing. The data management support provided by these systems is typically application-specific, highly integrated and specialised with little to no DBMS-integration. The main shortcomings of these solutions include limitations in the number and size of raster objects which can be handled, a missing or limited query support and the unavailability of traditional DBMS services, such as concurrent multi-user support.

The basic spatial raster management requirements, irrespective of application domain or raster class, can be summarised as:

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.

For most applications, these requirements can only economically be fulfilled by the utilisation of modern DBMS technologies. The primary objectives of these investigations were to research and identify the technological key issues and to develop concepts for the implementation of spatial raster management solutions on the basis of standard, state-of-the-art DBMS technology.

Developed Concepts

For the management of individual spatial raster objects, a number of basic concepts from existing raster storage solutions could be adapted, e.g. the compression support of the TIFF format. The management of spatial raster mosaics and the integration within a DBMS-framework, however, required a number of concepts to be developed or modified. These concepts can be summarised as follows:

- regular tile-based raster partitioning
- tile-indexing using variable-length Morton coding scheme
- spatial mosaic query concept with query optimisation
- tile-based multi-resolution support for raster mosaics
- multi-level raster object georeferencing

These concepts are largely database model independent and they support DBMS-internal and -external raster data storage. For DBMS-internal storage under transaction control, the BLOB (Binary Large Object) concept was chosen. The BLOB concept ensures compatibility with current DBMS technologies and facilitates the migration to future, SQL-3 compliant technologies. For DBMS-external storage, an application-controlled file-based mechanism was chosen, which, in its initial form, makes use of the TIFF format.

Implementation and Tests

An important part of the investigations was the design and implementation of the prototype system GrIdS which should allow the practicability and performance of some of the concepts listed above to be tested. GrIdS is based on a middleware architecture, with an extensible client-side compression and decompression support. GrIdS was implemented on top of a standard relational DBMS on a low-performance UNIX platform.

The tests were performed on a series of raster mosaics, which were generated from raster maps and orthoimages. The maximum mosaic size was in the excess of 1 GB. The tests demonstrated:

- the system's capability to manage raster objects of large dimensions and size
- a good system scalability with no signs of size-related degradation
- a good spatial mosaic query performance (especially with query optimisation)

- a good mosaic window retrieval performance (in consideration of test platform performance)

The tests also highlighted the advantages and drawbacks of the two main storage scenarios.

- DBMS-internal raster data storage, which is typically subjected to transaction control, involves a significant amount of overhead. This particularly affects the performance of data manipulation operations, e.g. insert, update and delete. On modern hardware platforms, DBMS-internal storage is well suited for frequently and concurrently accessed raster mosaics, e.g. seamless map and orthoimage mosaics. For this type of spatial raster data the additional manipulation overhead is outweighed by the availability of the full DBMS services.
- DBMS-external raster storage offers a better performance if the transaction control mechanism and the DBMS server memory are bypassed, i.e. if the data can directly be transferred from the storage system to the client memory and vice versa. DBMS-external storage allows advanced mass storage systems to be supported by the application in cases where direct support through the DBMS is not available. DBMS-external storage is well suited for managing individual raster data sets and 'raw' raster data in particular.

In summary, it can be stated that robust and efficient spatial raster data management solutions can be implemented on state-of-the-art relational database technology. The developed concepts proved to be well adapted to the DBMS environment and they allowed an operational basic prototype system to be implemented within a relatively short time frame. The prototype system fulfils the functionality and scalability requirements stated earlier in this section. There is also every reason to believe that running the prototype system on a modern hardware platform will dramatically improve the moderate I/O performance which was observed on the available out-of-date test platform.

It should be noted that the presented concepts primarily provide some of the essential building blocks for the design and implementation of spatial raster management solutions and that most of them are not limited to a specific database technology. The current management functionality implemented in the relational context could be significantly enhanced by integrating the same concepts with some of the emerging database and mass storage technologies discussed below.

8.2 OUTLOOK

The following summary documents the status and trends with respect to some of the main issues related to the DBMS-based management of spatial raster data.

From BLOBs to Abstract Data Types (ADTs)

The BLOB data type, currently used for storing raster data, is a base type with a limited set of supported operations, such as writing and reading sub-strings. Despite some ongoing debates over its benefits and deficiencies, the BLOB concept is very useful for managing arbitrary-length byte strings in DBMSs. With the inclusion of BLOB as a standard SQL-3 base type, it will become one of the primary components for creating user-defined raster ADTs, for example, in object-relational DBMSs. This will allow the current BLOB-based raster storage concept to be migrated without strategic changes.

New Database Technologies

Probably the most important motivation factor for migrating the current solution to object-relational or object-oriented database technologies are type-specific, user-defined operations. They, for example, enable content-based querying and allow certain raster operations to be performed on the DBMS server. Of the two technologies, it is expected that object-relational DBMSs will become the main next-generation database technology for spatial raster data and for other spatial data types. Object-relational technology, for example, allows current (spatial raster) management functionality to be extended and migrated in a SQL context. It also promises to maintain or even improve the 'relational' performance on large unstructured objects. Object-oriented DBMSs, the other relatively new DBMS technology, primarily address the management of complex data. It was shown that complexity is not the primary issue in spatial raster data management. The commercial argument which supports this trend towards object-relational technology is the enormous market share of the major (object-) relational DBMS vendors, which includes the GIS market.

DBMS Extensions and Spatial Raster Data Management

Extensibility is one of the key concepts of object-relational and object-oriented DBMS technologies. Major object-relational DBMS vendors already offer a number of DBMS extensions which they refer to as DataBlades, Data Cartridges or similar. 'Spatial' and 'image' extensions are examples of DBMS extensions with a particular relevance to the management of spatial raster data. However, these extensions will not automatically solve the problems listed earlier. Image extensions, for example, are typically aimed at managing collections of relatively small images. They do not address very large raster objects, composite objects such as raster mosaics and spatial aspects such as georeferencing or spatial querying. Nevertheless, the basic spatial and image extensions will provide useful tools for the economical development of future spatial raster management solutions.

Integration with Other Spatial Data Types

One of the major benefits of object-relational DBMS technologies is the possibility of improving the integration of spatial raster data with other spatial data types, such as complex 3-D objects. An interesting perspective is the gradual integration of spatial raster management support within existing spatial DBMS extensions which are being developed by a number of GIS manufacturers. This would allow all types of spatial data, including raster, to be managed with a single DBMS architecture and through a uniform programming interface. This would also rapidly improve the current situation with the limited spatial raster management support in GISs.

Storage Management and Mass Storage Support

The future storage capacity and performance requirements of spatial raster data will be enormous, probably only surpassed by digital video data in multi-user video server applications. Current solutions are often based on a dual storage architecture with a separate DBMS-controlled on-line storage and application-controlled mass storage. Major improvements over this solution can be expected from the emerging integration of mass storage support within DBMSs. The support of Hierarchical Storage Management (HSM), in particular, will permit the seamless integration of high-performance RAID technology with high-capacity optical jukebox systems. Both storage technologies will be required to fulfil the performance and capacity requirements of operational spatial raster management solutions.

Leer - Vide - Empty

REFERENCES

Abel, D.J., 1983. A data structure and algorithm based on a linear key for a rectangle retrieval problem. Computer Vision, Graphics, and Image Processing, 24(1): 1-13.

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.

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.

Aldus, 1992. TIFF Specification, Revision 6.0, Aldus Corporation, Seattle, WA.

Anderson, J.T. and Stonebraker, M., 1994. SEQUOIA 2000 Metadata Schema for Satellite Images. ACM SIGMOD Record, Volume 23(Number 4).

Angleraud, C., 1996. GIS-Geospot Format/Toolbox. Spot Image, (http://www.spot.com).

Atkinson, M. et al., 1992. The Object-Oriented Database System Manifesto. In: Bancilhon, Delobel and Kanellakis (Editors), Building an Object Oriented Database System. Morgan Kaufmann.

Baltsavias, E.P., 1991. Multiphoto Geometrically Constrained Matching, Institut für Geodäsie und Photogrammetrie, ETH Zürich, Zürich.

Bartelme, N., 1995. Geoinformatik. Springer Verlag, Berlin Heidelberg.

Bashkaran, V. and Konstantinides, K., 1995. Image and Video Compression Standards - Algorithms and Architectures. Kluwer Academic Publishers Group, Dortrecht, NL.

Batini, C., Ceri, S. and Navathe, S.B., 1992. Conceptual Database Design - An Entity Relationship Approach. The Benjamin / Cummings Publishing Company, Inc., Redwood City, California.

Baumann, P., 1995. Liberating Images from Data Exchange Formats: Data Independence for Raster Data!, Photonics East, Philadelphia (http://www.forwiss.tu-muenchen.de/~rasdaman).

Baumann, P., Furtado, P., Ritsch, R. and Widmann, N., 1997a. Geo/Environmental and Medical Data Management in the RasDaMan System, VLDB '97, Athens (http://www.forwiss.tu-muenchen.de/~rasdaman).

Baumann, P., Furtado, P., Ritsch, R. and Widmann, N., 1997b. The RasDaMan Approach to Multidimensional Database Management, SAC '97, San Jose, CA (http://www.forwiss.tu-muenchen.de/~rasdaman).

Baxes, G.M., 1994. Digital Image Processing. John Wiley & Sons Ltd., Chichester, West Sussex, UK.

Bhargava, B.K., 1996. Digital Libraries. Ecole Polytéchnique Fédérale de Lausanne, Cours Postgrade en Informatique, Lausanne.

Bill, R., 1996a. Aspects of Data Base Implementation for Spatial Objects, ISPRS Tutorial on Coneptual Aspects of GIS Technology, Vienna, Vienna.

Bill, R., 1996b. Grundlagen der Geo-Informationssysteme, Bd. 2. Analysen, Anwendungen und neue Entwicklungen. Wichmann, Karlsruhe.

Bill, R. and Fritsch, D., 1991. Grundlagen der Geo-Informationssysteme, Band 1. Wichmann, Karlsruhe.

Blott, S., Relly, L. and Schek, H.-J., 1996. An Open Abstract-Object Storage System, Proceedings of the ACM/SIGMOD Conference, Montreal Canada, June 1996. ACM/SIGMOD, Montreal, Canada.

Blott, S. and Vckovski, A., 1995. Accessing Geographical Metafiles through a Database Storage System, Fourth International Symposium on Large Spatial Databases. Institute for Information Systems, ETH Zürich.

Bobrowski, S., 1997. Divide and Conquer, Oracle Magazine, Volume XI, Issue 3, pp. 66-68.

Bonham-Carter, G.F., 1994. Geographic Information Systems for Geoscientists: Modelling with GIS. Pergamon / Elsevier Science Ltd., Kidlington, UK.

Breutmann, B. and Burkhardt, R., 1992. Objektorientierte Systeme. Carl Hanser, München.

Campell, J.B., 1996. Introduction to Remote Sensing. The Guilford Press, New York.

Camps, R., 1996. Domains, Relations and Religious Wars. ACM SIGMOD Record, Volume 25(Number 3, September 1996).

Carosio, A., 1996. Vorlesungsskript in Raumbezogene Informationssysteme. Institut für Geodäsie und Photogrammetrie, Zürich.

Carosio, A., Grün, A. and Schek, H.-J., 1996. Raster-GIS - Integration of Raster Images and 3D-Objects into Geodatabases (Proposal - not published), ETH Zürich, Zürich.

Cattell, R.G.G. et al., 1996. The Object Database Standard: ODMG-93, Release 1.2. Morgan Kaufmann Publishers, Inc., San Francisco, California.

CEN/TC 287, 1996. Geographic Information - Data Description - Metadata (Draft). preN 12657:1996 E, European Committee for Standardisation (CEN), Brussels.

Cheiney, J.-P. and Kerhervé, B., 1990. Image Data Storage and Manipulation for Multimedia Database Systems. In: K. Brassel and H. Kishimoto (Editors), Proceedings of the 4th International Symposium on Spatial Data Handling 1990. Department of Geography, University Zurich, Zurich, Switzerland, pp. 611-620.

Cole, J.L. and Jones, M.E., 1995. The IEEE Storage System Standards Working Group Overview and Status, Fourteenth IEEE Symposium on Mass Storage Systems. IEEE Computer Society Press, Los Alamitos, California, Monterey, California, USA, pp. 306-311.

Darwen, H. and Date, C.J., 1995. The Third Manifesto. ACM SIGMOD Record, Volume 24(Number 1, March 1995).

Date, C.J. and Darwen, H., 1997. A Guide to the SQL Standard - Fourth Edition. Addison-Wesley Longman, Inc.

Demuth, B. and Bruns, K., 1994. Wege zur Struktur - Relationale Datenbanken und ihre multimedialen Grenzen, iX Multiuser-Multitasking-Magazin. Verlag Heinz Heise, Hannover, pp. 164-169.

DeWitt, D.J., Kabra, N., Luo, J., Patel, J.M. and Yu, J.-B., 1994. Client-Server Paradise, 20th International Conference on Very Large Data Bases, Santiago, Chile (http://www.cs.wisc.edu/paradise).

Dröge, G., 1995. Eine anfrage-adaptive Partitionierungsstrategie für Raumzugriffsmethoden in Geo-Datenbanken. Dissertation Thesis, Eidgenössische Technische Hochschule, Zürich.

Eastman, R., 1997. The IDRISI Project Web Site, Clark Labs for Cartographic Technology and Geographic Analysis, Worcester, Massachusetts, USA (http://www.idrisi.clarku.edu).

Ehrig, M. and Maier, B., 1995. Fast + Wide SCSI für HP-Workstations, Offene Systeme - Zeitschrift deutschsprachiger UNIX-Benutzer-Vereinigungen. Springer Verlag, Berlin Heidelberg, pp. 209-216.

Ekstrom, M.P. (Editor), 1984. Digital Image Processing Techniques. Computational Techniques, 2. Academic Press Inc., Orlando, Florida.

Elmasri, R. and Navathe, S.B., 1994. Fundamentals of Database Systems. The Benjamin / Cummings Publishing Company, Inc., Redwood City, California.

ESRI, 1996. Spatial Database Engine (SDE) - White Paper, Environmental Research Institute, Inc., Redlands, CA.

Evans, J.D., Joseph Ferreira, J. and Thompson, P.R., 1992. A Visual Interface to Heterogeneous Spatial Databases based on Spatial Metadata. In: P. Breshanan, E. Corwin and D. Cowen (Editors), 5th International Symposium on Spatial Data Handling 1992. International Geographical Union, Commission on GIS, Charleston, South Carolina, USA, pp. 283-293.

Farine, A., 1995. Galore - API Description and Application Notes. Alain Farine - Computer Science Engineering, Lausanne.

FGDC, 1994. Content Standards for Digital Geospatial Metadata, Federal Geographic Data Committee (FGDC) c/o U.S. Geological Survey, Reston, Virginia.

FGDC, 1995. Content Standards for Digital Geospatial Metadata Workbook, Federal Geographic Data Committee (FGDC) c/o U.S. Geological Suvey, Reston, Virginia.

Foley, J.D., Dam, A.V., Feiner, S.K. and Hughes, J.F., 1992. Computer Graphics: Principles and Practice. Addison Wesley, Reading, Massachusetts.

Frank, A., 1983. Probleme der Realisierung von Landinformationssystemen. Teil 2 - Storage Methods for Space Related Data: The FIELD TREE. Grey Series No. 71, Institut für Geodäsie und Photogrammetrie, ETH Zürich, Zürich.

Frei, U., 1984. Geometrische Korrekturen von NOAA - AVHRR - Daten. Vol. 8, Department of Geography, University of Zurich, Zurich.

Fritsch, D. and Anders, K.-H., 1996. Objektorientierte Konzepte in Geo-Informationssystemen, Geo-Informationssysteme (GIS). Wichmann, Karlsruhe, pp. 2-14.

Fritz, L.W., 1996. Commercial Earth Observation Satellites, ISPRS Conference. International Archives of Photogrammetry and Remote Sensing, Vienna, pp. 273-282.

Gennart, B.A. et al., 1996. The Gigaview Multiprocessor Multidisk Image Server. Scientific Programming, 5: 3-13.

Geographic Data BC, 1995. Spatial Archive and Interchange Format (SAIF): Formal Definition Release 3.2, Province of British Columbia, Ministry of Environment, Lands and Parks.

Gnägi, H.-R. and Nebiker, S., 1997. INTERLIS Switzerland. In: H. Moellering and R. Hogan (Editors), Spatial Data Transfer Standards 2. Elsevier Science Ltd., Oxford, U.K., pp. 247-258.

Göpfert, W., 1991. Raumbezogene Informationssysteme. Wichmann, Karlsruhe.

Graf, G., 1995. Entwicklung einer EDV-gestützten Rasterdatenverwaltung auf der Basis von C-Programmen und von UNIX-Shell-Skripts, Landesvermessungsamt Baden-Württemberg, Stuttgart.

Grün, A., 1996. Digital Photogrammetric Stations Revisited, International Archives of Photogrammetry and Remote Sensing, Volume XXXI, Part B2, pp. 127-134.

Hainaut, J.-L., 1996. Rétro-ingénierie de bases de données. Cours Postgrade en Informatique: Bases de données, École Polytechnique Fédéral de Lausanne, Lausanne.

Hartmann, J., 1996. HuS Digitales Datenarchiv - Dokumentenmanagement für das Landesvermessungsamt Sachsen. Hartmann & Sältzer GmbH, Giessen, Germany.

Held, G., 1996. Data and Image Compression. John Wiley & Sons Ltd., Chichester, West Sussex, UK.

Hick, J.-M., Englebert, V., Henrard, J., Roland, D. and Hainaut, J.-L., 1995. The DB-MAIN Database Engineering CASE Tool. Computer Science Institute of the University of Namur, Namur, Belgium.

Hochstöger, F., 1996. Software for Managing Country-Wide Digital Elevation Data, International Archives of Photogrammetry and Remote Sensing, Volume XXXI, Part B2, pp. 160-163.

ICG, 1997. PCRaster - Web Site, The Netherlands Centre for Geo-ecological Research (ICG), Department of Physical Geography, Faculty of Geographical Sciences, Utrecht University, Utrecht, NL (http://www.frw.ruu.nl/pcraster).

Informix, 1997. Informix Spatial DataBlade Module - Data Sheet, Informix Software Inc., Menlo Park, California (http://www.informix.com).

ISO/TC 211, 1997. WD 15046-15 Geographic Information - Metadata - Version 2.0.

Jaedike, M., 1997. Datenbank-Innovationen: SQL3 - Standardisierung. Institut für Informatik, Forschungs- und Lehreinheit Informatik III, TU München, München.

Jensen, J.R., 1996. Introductory Digital Image Processing - A Remote Sensing Perspective. Prentice Hall Series in Geographic Information Systems. Prentice Hall Inc., Upper Saddle River, NJ.

Kern, P.F. and Carswell, J.D., 1994. An Investigation into the Use of JPEG Image Compression for Digital Photogrammetry: Does the Compression of Images Affect Measurement Accuracy ?, EGIS/MARI '94, Paris, France.

Kersten, T. and O'Sullivan, W., 1996. Project SWISSPHOTO - Digital Orthophotos for the entire Area of Switzerland, International Archives of Photogrammetry and Remote Sensing, Volume XXXI, Part B2, pp. 186-191.

Kodak, 1997. Photo CD Products and Features. Eastman Kodak Company, (http://www.kodak.com).

Koshafian, S. and Baker, A.B., 1995. MultiMedia and Imaging Databases. Morgan Kaufmann Publishers, Inc., San Francisco, California.

Lamb, P., 1994. Tiling Very Large Rasters, Advances in GIS Researches, Proceedings of the Sixth International Symposium on Spatial Data Handling. Taylor and Francis, London, pp. 449-461.

Lang, S.M. and Lockemann, P.C., 1995. Datenbankeinsatz. Springer Verlag, Berlin Heidelberg.

Leffler, S., 1995. libtiff - TIFF Software Library. Silicon Graphics, Inc. (http://www.sgi.com).

Liebig, V., 1996. Earth Observation in the Information Age - Trends in Utilisation of Networks for Earth Observation Applications, International Archives of Photogrammetry and Remote Sensing, Volume XXXI, Part B4, pp. 514-519.

Lotz-Iwen, H.-J., Goebel, R. and Markwitz, W., 1995. ISIS-Fernerkundung für jedermann. H. Wichmann Verlag, Heidelberg, 152 pp.

Lüscher, A. and Straubinger, A., 1996. Objektorientierte Technologien - Eine Einführung. vdf Hochschulverlag, Zürich.

MacEachren, A.M. and Taylor, D.R.F. (Editors), 1994. Visualization in Modern Cartography. Elsevier Science Ltd., Kidlington, Oxford, U.K.

Moellering, H., 1996. Minutes of Meeting, Summer Meeting, Den Haag, NL, ICA Commission on Standards for the Transfer of Spatial Data.

Moellering, H. and Hogan, R. (Editors), 1997. Spatial Database Transfer Standards 2: Characteristics for Assessing Standards and Full Description of the National and International Standards in the World. Elsevier Science Ltd., Oxford, U.K.

Molenaar, M., 1996. Spatial Data Modelling for Geographic Information Systems, ISPRS Tutorial on Conceptual Aspects of GIS Technology, Vienna.

Morain, S. and Baros, S.L., 1996. Raster Imagery in Geographic Information Systems. OnWord Press.

Murray, J.D. and VanRyper, W., 1994. Encyclopedia of Graphics File Formats. O'Reilly & Associates, Inc., Sebastopol, CA.

Nebiker, S., 1996a. Kartographisches Rasterdatenverwaltungssystem, Kartographie im Umbruch - neue Herausforderungen neue Technologien, Tagungsband zum Kartographiekongress Interlaken 96. SGK Publikationen, Bern, pp. 169-179.

Nebiker, S., 1996b. Verwaltung hybrider Daten, Kombinierte Anwendung von Vektor- und Rasterdaten in Geoinformationssystemen. EPFL, Département de Génie Rural., Lausanne.

Negroponte, N., 1995. being digital. Vintage Books, New York.

Nievergelt, J. and Hinrichs, K., 1993. Algorithms and Data Structures - With Applications to Graphics and Geometry. Prentice-Hall, Eglewood Cliffs, New Jersey.

Nulty, W.G. and Bartholdi, J.J., 1994. Robust multidimensional searching with spacefilling curves. In: T.C. Waugh and R.G. Healey (Editors), Sixth International Symposium on Spatial Data Handling. Taylor & Francis, Edinburgh, pp. 805-818.

OGC Earth Imaging Committee, 1996. Earth Imaging Committee Report, Open GIS Consortium (OGC), (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.

Oracle, 1995a. Oracle7 MultiDimension User's Guide. Oracle Corporation, Redwood City, CA.

Oracle, 1995b. Oracle 7 Spatial Data Option Application Developer's Guide, Version 1.3.5. Oracle Corporation, Redwood City, CA.

Oracle, 1997. Oracle8 Image Data Cartridges - Integrating Media Assets with Mission-Critical Business Data through Oracle8 (White Paper), (http://www. oracle.com).

Ousterhout, J.K., 1994. Tcl and Tk Toolkit. Addison Wesley, Reading, Massachusetts.

Ousterhout, J.K., 1997. Scripting: Higher Level Programming for the 21st Century. Sun Microsystems Laboratories, Mountain View, CA (http://www.sunlabs.com).

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.

Patel, J. et al., 1997. Building a Scalable Geo-Spatial DBMS: Technology, Implementation , and Evaluation, Paper to appear in proceedings of SIGMOD 97 conference. (http://www.cs.wisc.edu/paradise).

Persson, J. and Jungert, E., 1992. Generation of multi-resolution maps from run-length encoded data. International Journal of Geographical Information Systems, 6(6): 497-510.

Pistor, P., 1994. Objektorientierung in SQL3: Stand und Entwicklungstendenzen. IBM Wissenschaftliches Zentrum, Heidelberg.

Poindexter, T., 1997. Oratcl - Tcl interface to Oracle database server, Denver, Colorado (ftp://ftp.neosoft.com).

POSC, 1996. Additions to POSC literature pertaining to Geographical and Projected Coordinate System Transformations, Petrotechnical Open Software Corporation.

Ranade, S., 1991. Mass Storage Technologies. Meckler Ltd., London, UK.

Ranade, S. (Editor), 1992. Jukebox and Robotic Libraries for Computer Mass Storage. Meckler Ltd., London, UK.

Relly, L., Schek, H.-J., Henricsson, O. and Nebiker, S., 1997. Physical Database Design for Raster Images in CONCERT, to appear in: Proceedings of the 5th International Symposium on Spatial Databases (SSD'97). Springer, Berlin.

Rink, J., 1997. Bildergeschichten - Digitale Wasserzeichen unterstützen den Urheberrechtsschutz im Internet, c't magazin für computer technik, pp. 162-175.

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.

Samet, H., 1990. Application of Spatial Data Structures: Computer Graphics, Image Processing and GIS. Addison-Wesley Publishing Company.

Sedona, 1996. Sedona GeoCATALOG - Softcopy Catalog Development/Distribution Software, Sedona GeoServices Inc. (http://www.sedonageo.com).

Seidel, K. and Therre, J.-P., 1994. Remote Sensing Image Archive for National User Requirements, EEOS Workshop on European Data Networks and Earth Obervation User Information Services, Marino (Rome).

Seidel, K. and Therre, J.-P., 1997. The Swiss National Remote Sensing Image Archive - On-line Project Information, ETHZ, Zürich (http://www.vision.ee.ethz).

Shahin, F., 1994. A Hierarchical Approach to Digital Image Compression, ASPRS/ACSM. (http://wwwsgi.ursus.maine.edu/gisweb/spatdb/acsm).

Singer, C., 1993. Relationale Datenbanksysteme als Basis Geographischer Datenbanken., Geo-Informationssysteme (GIS), pp. 9-16.

Smith, T.R., 1997. Alexandria Digital Library - Annual Progress Report, Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA (http://www.alexandria.sdc.ucsb.edu).

Space Imaging, 1997. The New Ear in the Information Age Begins - Space Imaging EOSAT Corporate Profile, Space Imaging EOSAT, Thornton, CO (http://www.spaceimage.com).

Spot Image, 1997. DALI and DIVA Product Descriptions, Spot Image, Tolouse (http://www.spotimage.fr).

Stengele, R., 1995. Kartographische Mustererkennung - Rasterorientierte Verfahren zur Erfassung von Geodaten, Institut für Geodäsie und Photogrammetrie, ETH Zürich, Zürich.

Stonebraker, M., 1997. Limitations of Spatial Simulators for Relational DBMSs, Informix Software Inc. (http://www.informix.com), Menlo Park, California.

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

Thurgood, J., 1997. Conversation on features of the Space Imaging remote sensing system.

USGS, 1997. Digital Orthophoto Quadrangles, National Spatial Data Infrastructure. National Geospatial Data Clearinghouse - U.S. Geological Survey Node (http://www.usgs.gov/nsdi/product/doq.html).

Walcher, W., 1996. Visual Interaction with Very Large Spatial Data Sets, International Archives of Photogrammetry and Remote Sensing, Volume XXXI, Part B1, pp. 197-202.

Walker, A.S. and Petrie, G., 1996. Digital Photogrammetric Workstations 1992-96, International Archives of Photogrammetry and Remote Sensing, Volume XXXI, Part B2, pp. 384-395.

Welch, B.B., 1995. Pratical Programming in Tcl and Tk. Prentice Hall PTR, Upper Saddle River, New Jersey.

Williams, J., 1995. Geographic Information from Space - Processing and Applications of Geocoded Satellite Images. John Wiley & Sons Ltd.., Chichester, West Sussex, UK.

Worboys, M.F., 1995. GIS - A Computing Perspective. Taylor and Francis., London.

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.

Zanini, M., 1996. 3D-Modellierung und -Visualisierung kartographischer Raster- und Vektordaten, Kartographie Kongress 96. Kartographische Publikationsreihe Nr. 14. Schweizerische Gesellschaft für Kartographie, Interlaken, Switzerland, pp. 120-129.

Zanini, M. and Carosio, A., 1995. 3D-Landscape-Modelling and -Visualisation based on Digital Topographic Maps, Proceedings of the 17th International Cartographic Conference 1995, Barcelona, Spain. Institut Cartografic de Catalunya, Barcelona, pp. 1471-1475.

Zimmermann, H. and Sandoz, V.B., 1996. Geo-Informationssysteme des Bundes. Geo-Informationssysteme im Dienste der Raumplanung. Schweizerische Vereinigung für Landesplanung, Bern, 32-58 pp.

Leer - Vide - Empty

ACKNOWLEDGEMENTS

First, I would like to thank my examiner, Prof. Alessandro Carosio, for giving me the opportunity and freedom to perform this highly interesting research, for providing me with an excellent working environment and for his useful comments and suggestions. I am greatly indebted to Prof. Ralf Bill for serving as my co-examiner, for the stimulating discussions, for carefully reading the thesis and for his critical and encouraging suggestions and comments.

Many thanks go to my friend and colleague Matthew Amor for meticulously reading and checking the manuscript, both from a native-speaker's and a professional perspective, and for his constructive advice. I would also like to thank my colleague Steffen Frischknecht for reading the entire manuscript and for providing many useful comments.

These investigations helped to initiate the interdisciplinary ETH-project Raster-GIS. I would like to thank the partners of this project, namely Lukas Relly and Stephen Blott, for the highly interesting discussions which helped me to get a better insight into some DBMS kernel issues, which otherwise would have remained hidden to me.

I am indebted to the Swiss Federal Office of Topography, Wabern-Bern and SwissPhoto Surveys Ltd., Regensdorf and their representatives Martin Roggli and Thomas Kersten. They provided me with valuable real-world data and gave me the opportunity to demonstrate the prototype system to their respective teams which gave me many useful hints.

I would also like to thank my colleagues of the GIS group at the Institute of Geodesy and Photogrammetry for the many lively discussions, for reading parts of the manuscript and for their many useful comments. I would especially like to thank Marc Zanini for his support and cooperation in our mutual effort to tackle the numerous Tcl/Tk and C programming problems. I am also grateful to Michael Sonderegger for his excellent job in keeping 'my' system up and running and for his encouragement.

Leer - Vide - Empty

CURRICULUM VITAE

Name:	Stephan Nebiker
Date of Birth:	08 October, 1962
Place of Origin:	Itingen BL and Häfelfingen BL, Switzerland

Education

1996	Postgraduate course in database technologies, Ecole Polytéchnique Fédérale, Lausanne (EPFL), Switzerland
1982 - 1987	Diploma course in geodesy and surveying, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland:
1978 - 1981	Gymnasium Liestal, Switzerland

Degrees

1992	Swiss National Surveyor's License (Eidg. Geometerpatent)
1988	Diploma in Geodesy and Surveying, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.
1981	Matura Type C (Gymnasium leaving certificate)

Professional Experience

1993 - present	Research assistant, Institute of Geodesy and Photogrammetry, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland
1990 - 1993	Development team leader and GPS coordinator, Geodetic and Construction Survey Ltd., Switzerland
1989	Project team leader and course instructor, Geodetic International Inc., Houston, TX, USA
1988	Project surveyor, Geodetic and Construction Survey Ltd., Switzerland