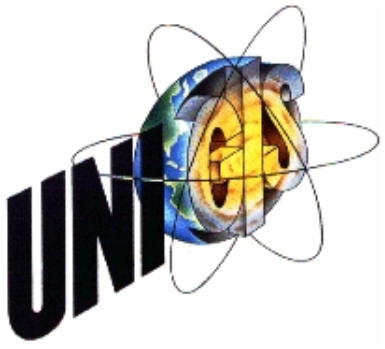


Master Thesis

im Rahmen des
Universitätslehrganges „Geographical Information Science & Systems“
(UNIGIS MSc) am Zentrum für GeoInformatik (Z_GIS)
der Paris Lodron-Universität Salzburg

zum Thema

„Organisation und Verwaltung von Rasterdaten digitaler Geländemodelle“

- Eine Übersicht -

vorgelegt von

Dipl.-Ing. Frank Reihs
U1204, UNIGIS MSc Jahrgang 2005

Zur Erlangung des Grades
„Master of Science (Geographical Information Science & Systems) – MSc(GIS)”

Gutachter:
Ao. Univ. Prof. Dr. Josef Strobl

Clausdorf, 18.04.07

Erklärung der eigenständigen Abfassung der Arbeit

Ich versichere, diese Master Thesis ohne fremde Hilfe und ohne Verwendung anderer als der angeführten Quellen angefertigt zu haben, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat. Alle Ausführungen der Arbeit die wörtlich oder sinngemäß übernommen wurden sind entsprechend gekennzeichnet.

18.04.2007

Datum, eigenhändige Unterschrift

Kurzfassung

Für die Ostseeküste mit ihren zahlreichen überflutungsgefährdeten Zonen sind Untersuchungen der Küstendynamikprozesse sowie die Bewertung der Hochwassergefährdung notwendig. Damit diesbezüglich Analysen möglich sind, wird ein umfangreiches digitales Geländemodell des Küstenbereichs Mecklenburg-Vorpommerns entwickelt. Dieses Projekt 'DGM Küstenbereich' basiert auf einer Rasterdatenverwaltung und dient der effektiven Bereitstellung hochgenauer Geländemodelldaten.

Die vorliegende Thesis bearbeitet strukturiert alle Aspekte, deren Betrachtung zur Konzeption und Erstellung einer rasterbasierten Verwaltung digitaler Geländemodelle notwendig ist. Nach der Definition und Erläuterung des Aufbaus eines Rasterdatensatzes und seiner Eigenschaften werden neben Basisverfahren auch erweiterte Rastermanagementansätze vorgestellt und im Einzelnen charakterisiert. Die zur Geländemodellerstellung notwendigen Messmethoden sowie die Verfahren zu deren Verarbeitung werden erläutert und bzgl. ihrer Verwendbarkeit im Anwendungsgebiet kommentiert. Es werden die Vorteile einer datenbankbasierten Verwaltung der sehr speicherintensiven Rasterdaten herausgearbeitet und der derzeitige Stand der Softwaretechnik durch Beschreibungen grundlegender Workflows und Eigenschaften verschiedener Datenbanksysteme wiedergegeben. Einer Erläuterung der Rasterdatennutzung durch Zugriff und Austausch folgen praxisunterlegte Untersuchungen zur Optimierung einer Verwaltung. Die Reduktion des Speicherplatzes und Transportvolumens durch Datenkompression bildet wie auch die Nutzung von Indexierungsmethoden zur Verbesserung der Zugriffszeiten bei Rasterdatenabfragen einen häufig verwendenden Optimierungsansatz.

Im angewandten Teil der Arbeit wird der Aufbau der Rasterverwaltung hochgenauer Geländemodelle im Küstenbereich konzeptionell und praktisch vorgestellt. Der verwendete Workflow wird beschrieben und die genutzten Verfahren im Zusammenhang näher erläutert. Es werden die derzeitigen Möglichkeiten von Zugriff und Modellnutzung vorgestellt sowie einige Anwendungsbeispiele des Projektes gezeigt. In einem kurzen Ausblick wird auf zukünftige Arbeiten verwiesen.

Abstract

For the Baltic Sea coast with their numerous flooding-endangered zones investigations of the coastal dynamic processes as well as the evaluation of flooding endangerment are necessary. So that in this connection analyses are possible, an extensive digital elevation model of the coastal range of Mecklenburg Vorpommern is developed. This project 'DGM Kuestenbereich' been based on a raster data adminstration and serves the effective supply of highly exact elevation model data.

The available thesis structures all aspects, whose view is necessary worked on for the conception and production of a raster-based administration of digital elevation models. After the definition and explanation of the structure of a raster data record and its characteristics beside basis procedures raster management beginnings also extended are introduced and characterized in detail. The measuring methods necessary for the elevation model production as well as the procedures for their processing are described and commentated concerning their usefulness in the area of application. The advantages of a database-based administration of the very memory-intensive raster data are worked out and the present conditions of the software technology by descriptions of fundamental worksflows and characteristics of different database systems are shown. An explanation of the raster data use by access and exchange practice-underlaid investigations follow for the optimization of an administration. The reduction of memory and transportation volume by data compression forms like also the use from indexation methods to the improvement of the access times with raster data queries a frequently using optimization beginning.

In the applied part of the work the structure of the raster administration of highly exact elevation models within the coastal range is presented conceptionally and practically. The used workflow is described and the used procedures in summary are more near described. The present possibilities of access and model use are presented as well as some sample applications of the project are shown. In a short view to future work one refers.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ADT	abstrakter Datentyp
BIL/P	Band interleaved by line/pixel
BLOB	Binary Large Object
BSQ	Band sequential
CEN	European Comettee on Standards
CLOB	Char Large Object
DBMS	Datenbankmanagementsystem
DCT	diskrete Kosinus-Transformation
DGM	digitales Geländemodell, engl. DTM
(D)GPS	(differential) global positioning system
DHM	digitales Höhenmodell
DOM	digitales Oberflächenmodell, engl. DSM
DWT	discrete wavelet transformation
EPSG	European Petroleum Survey Group
ESRI	Environmental Systems Research Institute
FGDC	Federal Geographic Data Committee
GIST	general search tree
IDW	inverse distance weigthed
ISO	International Standardisation Organisation
JPEG	Joint Photographic Expert Group
LO	Large Object
LP	Linienpaare
LZW	Lempel-Ziff-Welch (Kompression)
MUR	minimal umschließendes Rechteck, engl.: MBR
NN	Nearest Neighbor
NOKIS	Nord-Ostsee-Küsten-Informations-System
OID	Object Identifier
RAM	random access memory
RAID	redundant array of inexpensive disks
RCat	Rasterkatalog
RDS	Raster Data Set
SDO	Spatial Data Option
SQL	structured query language
SRID	spatial referencing system identifier
TIFF	tagged file format
TIN	Triangulated Irregular Network
TK10	topographische Karte M1:10000
WKT	well known text

1 Einleitung

Durch die Entwicklung von Mess-, Rechen- und Speichertechnik ist es möglich geworden, Umweltdaten flächendeckend zu erfassen, umfangreich zu analysieren und langfristig zu speichern. Der Zugriff auf Teile dieser Daten, abgefragt nach thematischen oder räumlichen Kriterien, ist für größere Nutzerschichten zugänglich gemacht worden. Das Rastermodell ist neben dem Vektormodell eine grundlegende Form Messdaten in Objektform strukturiert zur späteren Verwendung abzulegen. Im Gegensatz zu den materialisierten Koordinaten eines Vektoransatzes werden bei einem Raster räumliche oder thematische Objekte in Abhängigkeit von der Auflösung vergröbert und als Zellinformationen abgelegt. Dazu werden sehr große Speicherplatzmengen benötigt. Da Raster durch ihren einfachen strukturellen Aufbau von sehr vielen Verarbeitungsprogrammen verwendet werden können, haben sie eine sehr breite Anwendungsvielfalt und weite Verbreitung erreicht. Rasterdaten besitzen keine internen objektbezogenen Strukturinformationen. Sie können multimediale farbcodierte Daten aus bilderfassenden Aufnahme- und Verarbeitungsmethoden oder direkte Messwerte Umwelt- bzw. statistischer Größen einer Dimension je Rasterschicht enthalten. Mehrere Schichten gleicher Ausdehnung können durch Layer in einem Rasterverbund zusammengefasst werden.

1.1 Digitale Geländemodelle

Eine besondere Anwendung des Rastermodells sind digitale Geländemodelle (DGM). Hier werden, mit Genauigkeiten in Abhängigkeit vom späteren Verwendungsziel, kontinuierliche Höheninformationen in einem 2.5D-Modell abgelegt. Für jede räumliche Position des Modells existiert eine Höheninformation als Zellattributwert, gespeichert durch eine Fließkomma- oder Integerzahl.

Digitale Geländemodelle können als Hintergründe für perspektivische Darstellungen von Geländeausschnitten genutzt werden. Sie dienen dann der Orientierung oder der Darstellung räumlich-thematischer Inhalte. Gleichzeitig können mit Hilfe von Geländemodellen auch vielfältige räumliche Analysen zur Gewinnung umweltrelevanter Informationen durchgeführt werden. Mediale und kartographische Ausgaben zur Unterstützung von Planungen, Bewertungen und Gutachten sind ebenso Anwendungsbereiche von Geländemodellrastern wie die Produktion abgeleiteter Ausgaben wie Flächen- und Volumenberechnungen.

Durch die sehr großen Datenmengen von Modellen größerer Gebiete und sehr hoher Auflösung bietet sich die Verwaltung der Rasterobjekte in einer Datenbankumgebung

an. Dort kann mit Hilfe der Ablage von Metainformationen zu den Modellen eine Ordnungsstruktur geschaffen werden. Schnelle Zugriffe bei Abfragen oder Suchen nach bestimmten Kriterien sind sehr komfortabel durch Benutzeroberflächen oder standardisiert durch Verwendung von SQL-Befehlsfolgen möglich.

Geländemodelle werden als Planungsgrundlagen im Landschafts- und Wegebau, bei der Erschließung von Flächen, in Land- und Forstwirtschaft sowie im Umwelt- und Klimaschutz oder beispielsweise zur Dokumentation von historischen und zur Prognose von zukünftigen Landschaftsveränderungen benötigt. Bauwerks- und Klimafolgenabschätzungen im Katastrophenschutz spielen im Küstenbereich aufgrund von Hochwasserereignissen eine wichtige Rolle. Die Konzeption und Darstellung sowie die Überprüfung der Wirkung von Küstenschutzbauwerken kann durch hochgenaue Geländemodelle maßgeblich unterstützt werden. Auch der Tourismus bietet in Form von Wander- und Radwegeplanung, Geländeansichten oder medialen Werbungsprodukten vielfältige Möglichkeiten zur Verwendung von Geländemodelldaten.

1.2 Ziele der vorliegenden Arbeit

Ziel dieser Arbeit ist die Zusammenstellung aller wesentlichen Aspekte bei der Verwaltung von Geländemodelldaten in Rasterform in einem Datenbankmanagementsystem. Dazu ist das Rastermodell zu definieren und in seiner Struktur zu charakterisieren. Im Vergleich mit dem Vektormodell ist eine Einordnung zu geben. Es sind die wesentlichen Anwendungsklassen des Rastermodells zu unterscheiden. Zur Bestimmung einer optimierten Verwaltung von Rasterdaten sind die Ablageverfahren zu beschreiben und miteinander zu vergleichen. Ziel ist die Darstellung von möglichen datenbankbasierten Datenablagestrukturen und die Vorstellung von allgemeinen Workflows zur Rasterdatenverwaltungserstellung in verschiedenen Anwendungssystemen. Ein Vergleich dieser Systeme soll anhand vorgegebener Datenkriterien und Anforderungen helfen, eine Systemauswahl treffen zu können. Zur Kenzeichnung von Rasterdatensätzen sind deren grundlegende Eigenschaften zu erläutern sowie Methoden und Verfahren für den Umgang mit diesen Daten vorzustellen. Für welche Anwendungsziele wurden Konzepte entwickelt und welche Ansätze sind davon für die Verwaltung digitaler Geländemodelle nutzbar? Zur Erstellung eines Rasterdatensatzes sind Datenerfassungsmethoden digitaler Geländemodelle kurz vorzustellen und auf ihre Eignung zum Einsatz in Küstengebieten, dem Anwendungsbereich des Projektes, zu untersuchen. In der Arbeit

soll ein Überblick über Methoden der Rastergenerierung gegeben sowie die projektrelevanten Verfahren in der Praxis erläutert werden.
Rasterdatenmanagementsysteme werden aufgrund ihrer Vorteile zur Datenstrukturierung und zur Förderung der Interoperabilität von Rasterdaten entwickelt. Der wesentliche Ablauf eines Rasterdatenzugriffs durch eine Abfrage des Datenbestandes sowie die Datenausgabe mit gut nutzbaren Datenaustauschformaten sind hierzu vorzustellen. Diese wesentlichen Komponenten dienen der Verbesserung der Nutzbarkeit von Daten, dem Ziel jeder strukturierten Datenverwaltung. Betrachtet werden sollen auch Rasterdatenoptimierungsansätze zur Kompression und zur Anfragebeschleunigung. Eine Überprüfung der Aussagen wird durch praktische Beispiele ermöglicht.

1.3 Beispiel Geländemodell im Küstenbereich

Eine praktische Anwendung der konzeptuellen Überlegungen findet sich in der Vorstellung einer großräumigen Rasterverwaltung hochgenauer Geländemodelle im Küstenbereich Mecklenburg-Vorpommerns. Ziel war hier die Erforschung der Integration verschiedenster Datenquellen in eine Sammlung blattschnittfreier Geländemodelle mit einheitlichen Eigenschaften. Ein effektiver Datenzugriff nach räumlichen und thematischen Kriterien sollte neben dem unkomplizierten Datenaustausch und leichter Weiterverarbeitung der Raster zu Anwendungsprodukten möglich sein. Die DGM werden durch ständige Aktualisierungsmessungen unterschiedlicher Messmethoden kontinuierlich genauer der Wirklichkeit angepasst. Vorzustellen sind die eingesetzten Verfahren der Datenintegration und Rastererstellung, das Kachelkonzept für optimierte Datenein- und Ausgaben sowie auch die Möglichkeiten des visuellen Datenzugriffs für schnelle optische Überblicke durch dienstbasierte Datenausgaben. Anwendungsbereiche des Projektes sind durch die Begrenzung des Modellgebietes auf den unmittelbaren Küstenstreifen vorrangig Küstenschutzuntersuchungen, Analysen zur Überflutungsgefährdung und von Klimaveränderungen, die Bewertung morphologischer Veränderungen des Gebietes sowie die Bereitstellung von Datengrundlagen für Planungen öffentlicher Einrichtungen.

2 Das Rastermodell

Mit Hinblick auf das im Kapitel 8 zu beschreibende Rasterprojekt war innerhalb der Konzeptionsphase über die grundlegende Verwendung eines Datenmodells zu entscheiden. Ziel ist der einfache, blattschnittfreie und schnelle Zugriff auf die Modelldatenmenge sowie der Datenaustausch mit einem weitverbreiteten Datenformat (Bild). Dabei sollten sowohl Analysen möglich sein (Ausgabe mit maximaler Auflösung) als auch vielfältige visuelle Ausgaben (Bereitstellung von Hintergrundkarten) erstellt werden können. Digitale Geländedaten sind flächenhafte kontinuierliche Informationen, die beispielsweise auch in Vektorform als Triangulated Irregular Network abgelegt werden können. Aufgrund der viel einfacheren Handhabung (Partitionierung, Datenaustausch) sowie bedingt durch das Vorliegen bedeutender Flächenanteile an unstrukturierten bereits rasterbasierten Höhendaten (Laserscanning, Fächerecholot) des Typs Fließkommazahl wurde hier ein rasterbasierter Modellansatz gewählt, der nachfolgend charakterisiert werden soll.

2.1 Datenmodell

Raster besitzen eine Reihen- und Spaltenstruktur. Diese wird geometrisch durch einen Ursprung, die Richtungen von 2 senkrecht aufeinanderstehenden Achsen sowie eine Zellgröße definiert. In einem Zellwert wird an der Reihen- und Spaltenposition relativ zum Ursprung eine Information des Zelltyps abgelegt. Dies kann layerbasiert (je Schicht ein Attribut) oder objektbezogen (je Rasterzelle mehrere Attribute) organisiert werden.

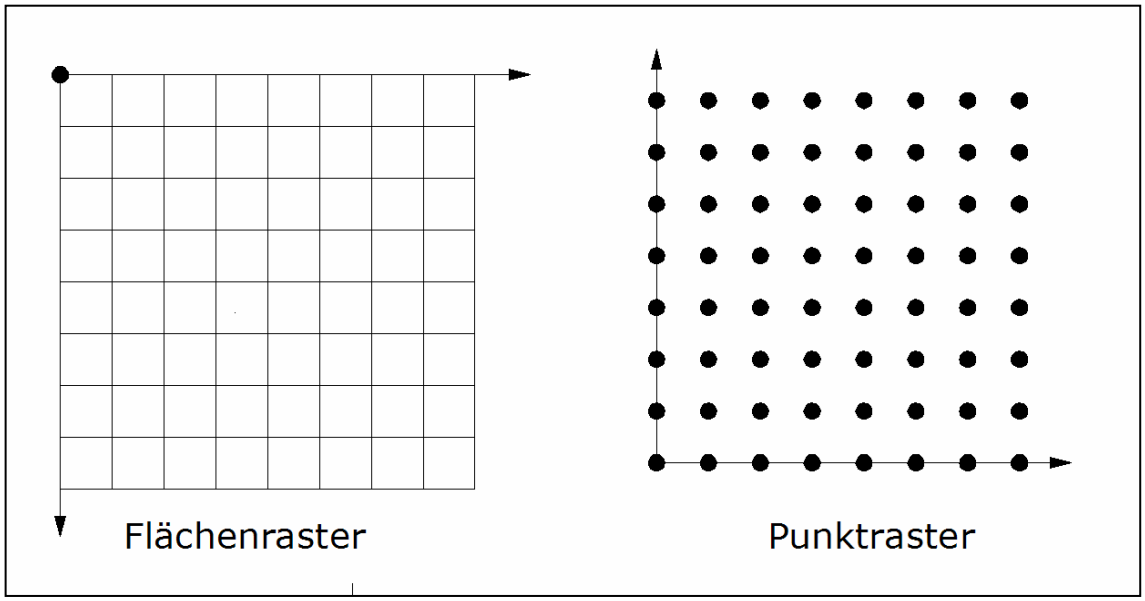


Abb. 1: Flächen- und Punktraster unterschiedlichen Ursprungs und Orientierung

Bei polygonalen Vektormodellen bestehen die Polygone aus Knoten und Kanten, die wie auch die Polygone eigene Attribute besitzen können. Im Rastermodell hingegen bilden einzelne Maschen eine flächenhafte Einheit mit homogenen Eigenschaften. Man verwendet im Allgemeinen regelmäßige, quadratische Maschen (Rasterzellen) als unteilbares Grundelement, da sie eine leichtere Handhabung in Datenmodellen und Operationen bieten. Die Rasterzelle bildet keinen kontinuierlichen Raum (Vektormodell), sondern den diskreten Rasterraum. Zur geometrischen Lagebeschreibung werden statt Koordinaten ganzzahlige Indexangaben, bezogen auf den Rasterursprung, verwendet. Die Form und Größe der Rasterzellen wird einheitlich bei der Rasterdefinition festgelegt. Dies ermöglicht einfach Flächenberechnungen aus Zellanzahlen und hat den Nachteil, der nur näherungsweisen von der Größe einer Zelle abhängigen Darstellung geometrischer Objekte in einem Raster. Punktuelle Objekte können minimal eine Zelle klein sein, linienhafte Objekte bilden eine Zellkette, die bei nichtachsenparalleler Lage Sägezahneffekte aufweisen kann (vergröberte Geometrie).

Raster sind für die flächendeckende Erfassung der Variation eines räumlichen Prozesses gut geeignet. Die Erfassung mit Hilfe scannender Verfahren (Laserscanning, Radar) ist einfach. Logische und algebraische Operationen lassen sich leicht ausführen. Nachteile sind die weder lage- noch formtreue Abbildung von Objekten bzgl. Geometrie und Topologie (Approximation) sowie die großen benötigten Datenmengen gegenüber Vektormodellen. Wenn als Ziel der Rasternutzung die mathematisch exakte Untersuchung von Eigenschaften und Wirkungen von natürlichen und künstlichen Objekten (im DGM beispielsweise Bauwerke) definiert wird, ist eine möglichst geringe Rasterweite und damit gute Approximation der Objekte notwendig.

Es wird zwischen flächenbezogenen und punktbezogenen Rasterdaten unterschieden. Bei den flächenbezogenen Rastern wird eine Rasterzelle durch ihre Umrandung gekennzeichnet. Obwohl häufig der Zellmittelpunkt zur Referenzierung verwendet wird (z.B. WorldFiles), ist kein eindeutiger Bezugspunkt definiert. Die Attributwerte gelten für die gesamte Fläche einer Rasterzelle (Beispiele: Rasterbilder, Rasterkarten). Im Gegensatz dazu ist bei den Punktrastern die Referenzierung eindeutig (Gitterpunkt). Die Attributinformationen beziehen sich dann auf diesen Punkt (Beispiel: DGM) [Neb02].

Der Raumbezug eines Rasters als Grundvorrausetzung zur Verwendung in Geoinformationssystemen und für Transformationen wird durch den Bezug zu einem geodätischen Referenzsystem sowie mit Hilfe der Abbildungsregeln zwischen Raster- und Objektkoordinatensystem hergestellt. Diese Beziehung kann durch innere und äußere Orientierung oder durch gemeinsame Passpunkte im Raster- und

Objektkoordinatensystem bei Originalbildern erreicht werden. Orthoraster werden mit Hilfe von Referenzpunkt (Rasterursprung), Rasterweite und Orientierung des Rasterkoordinatensystems oder durch mehrere Passpunkte georeferenziert [Neb02].
Unterschieden werden 4 Rasterklassen [Neb97]. *Rasterbilder*, beispielsweise Satelliten- und Luftbilder beinhalten Graustufen- oder Farbinformationen mit großer Bandbreite und können durch spezielle Metadaten als räumliche Hintergrundinformationen genutzt werden. *Rasterkarten* besitzen als Ergebnis eines kartographischen Entwurfsprozesses wenige diskrete Farben, die thematischen Ebenen zugeordnet werden. *Thematische Raster* können feldbasierte räumliche Information aus sehr vielfältigen Anwendungsbereichen darstellen. *Digitale Höhenmodelle (DHM)* dienen der Speicherung einer Höhenmatrix und können durch unterschiedliche Mess- oder Rechenprozesse erstellt werden. Sie können in digitale Geländemodelle (DGM) zur Beschreibung der Geländeoberfläche und unter Einbeziehung von Kunstbauten und Vegetation in digitale Oberflächenmodelle (DOM) unterschieden werden [Krau00]. Die vorliegende Arbeit konzentriert sich mit Hinblick auf morphologische Untersuchungen im Modellgebiet auf die Eigenschaften und Besonderheiten von Rastern digitaler Geländemodelle bestehend aus einer Matrix (zweidimensional) von Fließkommazahl-Höhenwerten.

2.2 Verwaltungsmodell

Bisher wurden Rasterdaten sehr häufig in einzelnen Dateien in hierarchischen Verzeichnisbäumen verwaltet. Sie werden dort oft anhand der technischen Erfordernisse der Datenquellen bzw. der Herstellung strukturiert (Anzahl, Format, Struktur), können aber auch speziell an die Erfordernisse einer Applikation angepasst und optimiert werden. Dateibasierte Verwaltungen bieten meist keine Inhaltskontrolle nach festlegbaren Integritätsbedingungen. Sicherungsmechanismen erfordern, sofern vorhanden, langwierige Rücksicherungsprozesse meist älterer Systemzustände. Zur effizienten Verwaltung können spezialisierte Applikationen eingesetzt werden, die viele der zur Rasterverwaltung und Analyse benötigten Funktionalitäten vereinigen und meist bzgl. Dateistruktur und Verarbeitung proprietär sind. Da für die Hersteller solcher Software die Erweiterung, Pflege bzw. Anpassung sehr aufwendig ist, wird eine hohe Vergütung gefordert [Bau00]. Eigene Datenzugänge sind, zumindest bei GIS-Systemen, möglich. Sie erfordern eine hohe Einarbeitungszeit und damit langfristigere Bindung an ein System. GIS-Systeme setzten häufig auf hybride Lösungen aus dateibasierter Verwaltung räumlicher Daten und datenbankbasierter Verwaltung attributiver Daten.

Reine Datenbanksysteme hingegen bieten eine komfortable, nutzergerechte und effektive Verwaltung großer langlebiger Datenmengen, gestützt durch jahrzehntelange Entwicklung und Optimierung von umfangreichen Systemen und Konzepten [Bau00]. Ermöglicht wird die semantische Datenmodellierung, d.h. Datenmodellierung und Abfrage werden unsichtbar vom System und unabhängig von der physischen Organisation dem Nutzer bereitgestellt. Einzelne Objekte sind problemlos auch über Speichermedien hinweg verschiebbar. Datenbanksysteme unterstützen durch die standardisierte Anfragsprache sehr flexibel und schnell Erweiterungen ohne Anpassung von Programmcode oder Schnittstellen. Hinter einer nutzerorientierten Oberfläche wird umfangreiche Funktionalität durch die Struktur 'Sende Anfragestring - Erhalte Ergebnisbild' verborgen. Interne Anfrageoptimierung (Transferkompression, Indexverwendung), gut dokumentierte Systemadministration sowie effiziente Zugriffsverwaltung (z.B. optimierte Suchalgorithmen) sind weitere Vorteile einer datenbankbasierten Verwaltung von Rasterdaten. Eine Datenbank erlaubt ein gemeinsames Datenmanagement von Raster-, Vektor, Meta- und tabellarischen Daten. Sie kann die Speicherung und Verteilung all dieser Daten übernehmen. Eine Datenbank gestattet Multi-User-Zugriffe, d.h. mehrere Client-Applikation können gleichzeitig auf einen Datensatz zugreifen. Lockmechanismen und Versionierung verhindern Probleme bei der gemeinsamen Datensatzbearbeitung. Ein Datenbankmanagementsystem (DBMS) bietet Sicherheit im Umgang mit Daten. Es beinhaltet Hilfsmittel zur Einteilung in verschiedene Sicherheitsstufen sowie zur Zugriffsrechteverwaltung. Durch ein zentralisiertes Archivierungsmanagement (Backup) wird die Ausfallsicherheit beträchtlich erhöht. Eine Datenbank bietet Konzepte wie erweiterten Datenschutz durch ein Sichtenkonzept aus beliebig erweiterbaren und für bestimmte Applikationen bzw. Nutzer einschränkbaren Sichten (Views) oder auch ein automatisiertes Datenbank-Recovery (Sicherung konsistenter Datenbankzustände). Probleme liegen in dem dafür notwendigen Führen einer Log-Datei, die bei Verwendung von Rasterdaten aus Kapazitätsgründen bei bestimmten Operationen abgeschaltet werden sollte (siehe Kapitel 2.3). Ein DBMS besitzt eine einheitliche Umgebung zur Datenabfrage. Mit Hilfe einer Programmiersprache (Application Programming Interface), mit standardisiertem SQL oder auch durch WebServices (z.B. WebMapService WMS) kann auf die Daten von unterschiedlichsten Clientsystemen und Applikationen aus zugegriffen werden. Mehrbenutzersynchronisation wird durch implizite und unsichtbare Serialisierung der Nutzerzugriffe (Schreib- und Lesezugriffe) unterstützt. Transaktionen bieten Datensatzsperrung während einer Aktion sowie Rollbackoperationen.

Datenbanken sind für die Verwaltung großer Mengen simpler Objekte geschaffen worden [Neb97]. Die Unterstützung großer, räumlicher Objekte erfordert meist Datenbank- sowie Datenbankspracherweiterungen (z.B. Oracle Spatial GeoRaster, PostGIS). Die großen über die Netzwerke zu transportierenden Datenvolumina stellen zur schnellen Abarbeitung von Anfragen hohe Ansprüche an das Serversystem (Multiprozessor, Clusterbildung, redundante Sekundärspeicher). Die Implementierung von serverseitigen Prozeduren (stored procedures) kann wie die Entwicklung von direkten Schreibzugriffen eines Clients auf den Serverspeicher die Netzwerkauslastung beträchtlich verringern.

Datenbanksysteme als Rasterserver bieten bereits jetzt höhere Performance, einfachere Verwaltung und mehr Flexibilität als dateibasierte Rasterverwaltungen. "Dateisysteme verwalten Daten - Datenbanksysteme bieten Informationen" [Bau00].

Aufgrund der Datenmengen des Rastermanagementprojektes sowie den beschriebenen Vorteilen wurde zur Konzeption einer Rasterverwaltung ein Datenbanksystem ausgewählt. Neben ArcGIS/ArcSDE (ESRI) mit einem integrativen Ansatz aus Datenbank und GIS-System bietet sich auch Oracle Spatial an, eine Erweiterung der Oracle-Datenbank um raumbezogene Raster- und Vektordatentypen. Nur aus Literaturquellen konnte die Funktionsweise von RasDaMan, einer für viele Datenbanken verfügbaren rasterspezifischen Erweiterung, beschrieben werden. Ausgewählt für das Projekt 'DGM Küstenbereich' wurde die OpenSource-Software PostgreSQL/PostGIS, die kaum rasterunterstützende Funktionen enthält, aber durch ihren objektrelationalen Aufbau zukünftig sehr gut erweitert werden kann.

2.3 Datenstrukturen

Zur Rasterdatenmodellierung innerhalb eines Datenbankmanagementsystems werden derzeit verschiedene Datenstrukturen eingesetzt. Diese Strukturen sollten inhaltsbezogene Abfragen und Operation, den effizienten Zugriff auf Teilbereiche sowie eine räumlich thematische Rasterzellverarbeitung unterstützen. Für statische Daten und kleine Raster können dazu implizite generische Datenstrukturen, deren Zellbeziehungen auf Metadaten beruhen (Dimensionen), verwendet werden [Neb97]. Listen dienen als hochdynamische Datenstruktur beispielsweise der Organisation der Teile von Rasterdatenmengen. Hash-Strukturen (siehe auch Indexierung) ermöglichen durch Adressberechnung den Zugriff auf Teilbereiche beispielsweise bei einem BLOB. Zu den räumlichen Datenstrukturen zählen Baumstrukturen, die den Datenraum hierarchisch dynamisch unter Sicherung der Nachbarschaftsbeziehungen aufteilen

können. Raumordnungen (Reihenordnung, Morton/Z-Ordnung, Hilbert-Ordnung) organisieren durch eine raumfüllende Kurve die Zellen einer Raumaufteilung.

Datenstrukturen bestehen aus einfachen oder komplexen Datentypen, die sich durch ihren Wertebereich sowie durch die Menge der auf ihnen ausführbaren Operationen definieren. Neben den bekannten Basisdatentypen (z.B. Integer, String) gibt es als Schlüsseltechnologie der objektorientierten und objektrelationalen Datenbanktechnologien *ADT's (Abstract Data Type)*, welche durch Kapselung die Veränderung oder Ersetzung einer internen Datenstruktur ohne Schnittstellenveränderung ermöglichen. Damit können sowohl nutzerdefinierte Datentypen als auch Typerweiterungen und zugehörige Funktionen und Operationen definiert und implementiert werden. Möglichst komplex und anwendungsspezifisch ist ein spezieller Rasterdatentyp unter Einbeziehung entsprechender Schnittstellen erzeugbar. Die Definition eines ADT besteht aus der Menge der annehmbaren Werte (Wertebereich), der Menge von Operationen (Funktionen, Methoden, Prozeduren), die für diesen Typ spezifiziert sind sowie aus den Axiomen (Regeln), welche die Semantik des Typs festlegen. Die Menge der definierten Operationen wird als Schnittstelle bezeichnet. Eigenschaften eines ADT sind neben der breiten Anwendbarkeit (Universalität), der Einfachheit der Anwendung mittels Schnittstellenkommunikation, der Geschütztheit (Fehlervermeidung durch Verbergen der inneren Struktur) auch die Modularität, die leichtes Programmieren und Austauschen sowie einfache Fehlersuche ermöglichen [Sar98]. Vorteil der ADT-Verwendung im Rastermanagement ist die Möglichkeit der Schaffung einer auf einen Anwendungsfall bzw. auf bestimmte Rastereigenschaften optimierten Datenstruktur incl. Definition der dabei benötigten Zugriffs- und Verarbeitungsoperationen. Unter anderem zur zukünftigen Verwendung dieser Strukturen wurde im Anwendungsprojekt 'DGM Küstenbereich' als DBMS PostgreSQL/PostGIS gewählt.

Zu den für die Rasterverwaltung verwendbaren Datentypen zählt auch das *Array*. Es besteht aus einer endlichen Reihung von Daten eines einheitlichen Datentyps, die geordnet im Speicher abgelegt werden (z.B. als Spaltentyp). Der Zugriff auf einzelne Elemente (anonyme Variablen) erfolgt mit Hilfe eines ganzzahligen numerischen Index (Standardarray) oder mit Hilfe von Schlüsseln (assoziatives Array, z.B. Hashtabelle). Arrays können ein- bis vieldimensional erstellt werden. Nachteil ist neben fehlenden Schutzfunktionen (Kapselung, Indexüberlauf) auch die statisch bei der Erstellung festzulegende Arraygröße sowie die meist sehr langsam ablaufenden auf Arrays ausführbaren Operationen. Zweidimensionale Arrays (Matrix) des Elementdatentyps

Fließkommazahl sind zur Aufnahme von kleinen in der Ausdehnung festen DGM-Rastern (z.B. Kacheln) durchaus geeignet. Zur Verringerung der Zugriffszeiten sollten die entsprechenden Operationen durch Programmroutinen optimiert werden. Ein Beispiel für die Verwendung von Arrays zum Rastermanagement bildet der RasDaMan-Server (siehe Kapitel 5).

Als einfacher, bereits in SQL3 definierter, und für Rasterdaten nutzbarer Datentyp werden nachfolgend *Large Objects (LO)* kurz vorgestellt. Large Object (LO) ist ein Datentyp, der innerhalb einer Spalte einer Zeile einer Tabelle eine große Datenmenge abzuspeichern erlaubt. DBMS ohne LO-Unterstützung besitzen Datentypen, die Längen bis 32kb umfassen können [Sar98]. Innerhalb eines LO können dagegen, abhängig von der Implementierung, bis 2GB abgelegt werden. Durch LO wird die Speicherung von sehr umfangreichen multimedialen Daten wie Dokumenten, Bildern, Audio- sowie Videodaten innerhalb eines DBMS möglich. Für das DBMS sind die Informationen innerhalb eines LO unstrukturiert, d.h. das DBMS kann den Inhalt nicht typbezogen erfassen oder verarbeiten. Large Objects können als BLOB (binary large Object - sequenzielle Aneinanderreihung von Binärdaten) oder CLOB (char large Objects - sequenzielle Reihung von Textdaten) angelegt werden. Für den Nutzer sind die grundlegenden Zugriffsmöglichkeiten nach SQL wie INSERT, UPDATE und DELETE implementiert worden. Das DBMS legt die LO separat von den anderen Daten einer Tabellenzeile in einer speziellen Speicherseite, teilweise gruppiert über mehrere Objekte, ab. Innerhalb der Datenstruktur der Tabelle des LO wird dagegen nur eine Referenz in Form eines Verweises oder Zeigers (OID) abgelegt [Sar98].

Im DGM-Anwendungsprojekt werden Large Objects zur Rasterkachelablage genutzt. Bei der PostgreSQL-Implementierung sind die Auswirkungen einer Lösch- oder Updateoperation zu beachten. Diese verändern lediglich die Daten der Tabelle, ohne das entfernt abgelegte LO zu beeinflussen. Will man das Anwachsen der Datenbank durch solche Operationen verhindern, so sind regelmäßige Wartungen (z.B. mittels VACUUMLO) zur Entfernung nicht mehr referenzierter LO notwendig.

Large Object als Datentyp unterliegt innerhalb einer Datenbank einigen Einschränkungen. Sie können nicht als Vergleichsobjekte genutzt werden. Eine Indizierung auf ihnen und eine Sortierung nach ihnen sind nicht möglich. Ebenso sind sie nicht als Haupt- oder Fremdschlüssel einer Relation verwendbar. LO ermöglichen ein gemeinsames Management von Standard-DBMS-Datentypen mit spezialisierten unstrukturierten großen Datentypen in einer Umgebung. Die Verknüpfung von Basisdatentypen und unstrukturierten Daten ist damit möglich. Eine Nutzung von LO

innerhalb von DBMS kann sehr zeitintensiv sein. Wenn Client und Server auf unterschiedlichen Systemen aufgesetzt worden sind, wird ein Netzwerktransfer benötigt. Die Rückgabe eines vollständigen LO benötigt sehr große Ressourcen im Bereich Speicherverwaltung, Netzwerkbandbreite und Programmpufferspeicher. Einige Systeme erlauben daher die Datenwiedergabe in Teilen, je nach nachgefragtem Bedarf. Bestimmte Implementierungen bieten den Zugriff auf die LO durch Programmierschnittstellen, die zur spezielleren Verwaltung der Objekte eingesetzt werden können. Solche Erweiterungen ermöglichen die Verarbeitung von LO innerhalb von SQL-Befehlsfolgen. Das Objekt wird mittels serverseitiger in das DBMS eingebetteter Programmierroutinen verarbeitet und nur das Ergebnisobjekt (z.B. Teilraster) an den Client bzw. die Zielrelation gesendet. Bei Standarddatentypen wird im Client kaum Hauptspeicherpuffer benötigt. Bei der Größe der LO ist dies anders. Einige Implementierungen erlauben deshalb den Datentransfer direkt zwischen dem DBMS (Server) und einer oder mehreren Ausgabedateien, anstelle des Umweges über den Hauptspeicher eines Clients.

Mehrbenutzerdatenbanksysteme benutzen Protokollstapel zur Abarbeitung von Schreiboperationen. Bevor und nachdem eine Operation ausgeführt wird, wird auch eine Kopie der Daten innerhalb eines LOG-Systems abgelegt. Diese Ablage ermöglicht den gleichzeitigen Zugriff mehrerer Nutzer auf die Datenbankobjekte und kann eine Datenwiederherstellung im Falle eines Festplatten- bzw. Speicherfehlers erlauben (Recovery). Bei einer Verwendung von Large Objects und häufigen Aktivitäten mit diesen Objekten kann das LOG-System leicht sehr speicherplatzintensiv werden. Bei der Erstellung der Relation ist daher eine Abschaltung der LOG-Aktivitäten von LO zur Leistungssteigerung möglich. Zur Gewährleistung der Transaktionssicherheit wird vom DBMS dann eine Spiegelung des bearbeiteten Datensatzes vorgenommen. Wird nach Abschluss einer Operation das Ergebnis angenommen (COMMIT), so nutzt das DBMS den neuen Datensatz und verwirft die Spiegelung des alten Datensatzes. Alternativ kann auch ein Rückgängigmachen (ROLLBACK) erfolgen. Dann wird lediglich die Spiegelung verwendet. Da durch diesen Mechanismus eine Wiederherstellung von historischen Zuständen der LO innerhalb einer Datenbank ausgeschlossen wird, können im Falle eines Speicherfehlers lediglich die Zustände des letzten vollständigen Backups wiedererlangt werden. Veränderungen seit diesem Zustand, normalerweise gesichert im LOG-System, sind verloren [Sar98].

Vergleicht man die drei vorgestellten Datentypen, so bietet sich aufgrund der einfachen Handhabung durch die innerhalb der Datenbank bereits vorhandenen Zugriffsfunktionen

sowie durch die Schnelligkeit und Universalität des Datentyps die Verwendung von BLOB's zur Rasterdatenhaltung der Kacheln eines DGM-Projektes an. Im Projekt 'DGM Küstenbereich' wird in einen BLOB eine vollständige GeoTIFF-Datei binär abgelegt, die nach ihrer Abfrage außerhalb der Datenbank selbstständig weiterverarbeitet werden kann. Codierungen oder Decodierungen bei Datenaustausch mit dem Dateisystem können so entfallen. Zukünftige Erweiterungen des entworfenen Rasterverwaltungssystems könnten zur Schaffung eines komplexen eigenen Datentyps (ADT) führen, der dann auch den bisher fehlenden Zugriff auf einzelne Zellinformationen ermöglicht. Innerhalb der Datenbank wären eigene Operationen zur Komprimierung/Dekomprimierung, Indexierung und Abfrage zu entwickeln. Möglicherweise kann eine Datentypumwandlung der Ablagestruktur in einen Array und die Nutzung der Array-Zugriffsfunktionen während einer Abfrage hier ebenfalls zielführend und optimierend sein.

3 Rastereigenschaften und Konzepte

Zur optimalen Verwendung von Rastern sind einige Eigenschaften genauer zu untersuchen und für den Anwendungszweck optimiert einzustellen. Die Festlegung der Auflösung eines Rasters bei dessen Erstellung bestimmt die Wiedergabequalität von Kleinstrukturen eines Geländes. Wird eine Verwendung der Raster im Zusammenhang mit anderen räumlich verorteten Daten angestrebt, so ist eine Georeferenzierung zur Lagefestlegung notwendig. Generell sollten zu Geodaten möglichst umfangreiche Metainformationen erfasst werden, die den Datensatz beschreiben und die Weiterverwendung erleichtern. Sollen die Rasterdaten vom DBMS an Clientsysteme mit unterschiedlicher Qualität oder Auflösung zur Optimierung der Transportkapazitäten übermittelt werden, so sind Multi-Resolution-Konzepte anzuwenden. Zur Optimierung des Zugriffs auf Rasterteile von Rastern sehr großer Datenmenge und Ausdehnung werden Konzepte der Rasterteilung benötigt. Ziel ist es hier, für den Nutzer transparent, einen schnellen Zugang zu Teildaten zu ermöglichen. Ebenfalls wichtig für eine Rasterverwaltung auf Datenbankbasis ist die Optimierung der physikalischen Speicherstrukturen. Nur durch an die Anforderungen der Rasterobjekte angepasste Speicherkonzepte lassen sich Performanz und Komfort in einer Rasterverwaltung für den Nutzer realisieren.

3.1 Allgemeine Rastereigenschaften

3.1.1 Auflösung eines Rasters

In Abhängigkeit von der Herstellungsart und vom Anwendungsgebiet lassen sich verschiedene Auflösungsarten bestimmen. Bei digitalen Geländemodellen, als Nachbildung der Erdoberfläche, spiegelt die *qualitative Auflösung* die Wiedergabequalität in Abhängigkeit vom Aufnahmesystem wieder. Sie entspricht der kleinsten Objektgröße, die nach Erfassung und Rasterverarbeitung noch dargestellt werden wird. Wesentlich ist die physikalische geometrische Auflösung des Aufnahmesensors, beispielsweise messbar in Linienpaaren/Millimeter (Lp/mm, Photogrammetrie) [Krau04]. Durch eine an die Aufnahme anschließende Rasterverarbeitung wird diese Auflösung aufgrund von Transformationsprozessen und unter Beachtung des Anwendungsziels meist verringert. Bedeutsam sind auch die Veränderungen, die durch die Verarbeitung der Messpunkte zu einem Raster (Interpolation) vorgenommen werden. Im Zusammenhang mit der Georeferenzierung als Zuordnung der räumlichen Referenz- und Positionsinformation zu einem Raster

kann die Größe einer Rasterzelle als Bodenauflösung angegeben werden. Meist werden einheitliche quadratische Rasterzellen verwendet. Für hochgenaue Geländemodelle zur Analyse morphologischer Geländeveränderungen sind, nach Erfahrungen bisheriger Projekte, Zellgrößen von 10, 5 oder 1 Meter zu verwenden, um die Abbildung aller relevanten Objekte zu gewährleisten. Diese Zellgrößen stellen hohe Ansprüche an die für die Rastererstellung verwendeten Eingangsdaten sowie an die Genauigkeit der Rechenprozesse. Bedingt durch die Abbildung der Erdoberfläche auf eine Ebene ist die Auflösung innerhalb eines Rasters nicht konstant. Im Anwendungsbereich digitaler Geländemodelle des betrachteten Projektes sind jedoch durch die geringen Höhenunterschiede (max. 150m) sowie die relative Kleinräumigkeit der Modelle diese Verzerrungen vernachlässigbar klein. Die *radiometrische bzw. spektrale Auflösung* spielt bei DGM keine Rolle. Es werden keine Farbinformationen als primäre Messdaten von einem Aufnahmesystem benötigt. Die *attributive Auflösung* besteht bei DGM ohne zusätzliche an die Zellposition geknüpfte thematische Information lediglich aus der für die Codierung der Höheninformationen benötigten Pixeltiefe. Sie ist eine quantitative Auflösung, kennzeichnet den Wertebereich des Attributs und wird durch die Attributextrema, den Attributtyp oder eine Bit-Auflösung in den Metadaten angegeben.

3.1.2 Georeferenzierung

Die Georeferenzierung beinhaltet die Zuordnung räumlicher Referenzinformationen sowie Positionsinformationen zu einem Raster. Sie ist damit ein fundamentales Konzept zur Abgrenzung räumlicher Rasterdaten, ermöglicht räumliche Abfragen und die Integration des Rasterdatensatzes in eine Umgebung zur Überlagerung mit anderen räumlich festgelegten Daten [Neb97]. Die Referenzinformation setzt sich aus dem geodätischen Referenzsystem, Datum, Ellipsoid incl. Parameter, Projektionstyp, Projektionszone und Zentralmeridian, den Einheiten sowie weiteren Projektionsparametern zusammen (siehe Beispiel). Die Positionsinformationen dienen der Orientierung innerhalb des Referenzsystems und können durch Kontrollpunkte, Passpunkte bzw. Angaben zur Skalierung oder einer Transformationsmatrix festgelegt werden. Geocodierung bezeichnet die geometrische Transformation eines Datensatzes in ein Referenzsystem. Sie kann impliziert, d.h. die Daten werden im Modellsystem belassen und eine Transformation wird beigefügt, oder explizit (Anwendung der Transformation auf die Daten) erfolgen. Wird eine Geocodierung explizit durchgeführt, so kommt es aufgrund der veränderten Lage der Achsen (Drehung, Skalierung) des

Zellrasters zu einer Veränderung der Daten durch Anwendung einer Resamplingmethode.

Beispiel räumlicher Referenzdatensatz aus PostgreSQL/PostGIS für EPSG-SRID=2398

"PROJCS["Pulkovo 1942(83) / Gauss-Krüger Zone 4",
GEOGCS["Pulkovo 1942(83)",DATUM["Pulkovo_1942_83",
SPHEROID["Krassowsky 1940",6378245,298.3,AUTHORITY["EPSG","7024"]],
TOWGS84[24,-123,-94,0.02,-0.25,-0.13,1.1],AUTHORITY["EPSG","6178"]],
PRIMEM["Greenwich",0,AUTHORITY["EPSG","8901"]],
UNIT["degree",0.01745329251994328,AUTHORITY["EPSG","9122"]],AUTHORITY["EPSG","4178"]],
PROJECTION["Transverse_Mercator"],
PARAMETER["latitude_of_origin",0],PARAMETER["central_meridian",12],
PARAMETER["scale_factor",1],PARAMETER["false_easting",4500000],PARAMETER["false_northing",0],
UNIT["metre",1,AUTHORITY["EPSG","9001"]],AUTHORITY["EPSG","2398"]]"

Als Standard im Bereich Georeferenzierung bei Austauschformaten mit räumlichen Parameterdatensätzen haben sich die Arbeiten der 1986 gegründeten European Petroleum Survey Group in Form der EPSG-Datenbank durchgesetzt. Das OGP Surveying und Positioning Comittee (2005) dieser Gruppe beschäftigt sich mit der Sammlung, Sortierung und Pflege dieser Datensätze und integriert die Konzepte als freie Referenz in bekannte Austauschformate [OGP]. Innerhalb der meisten räumlichen DBMS (z.B. Oracle Spatial, PostgreSQL/PostGIS) wird beispielsweise eine Relation mit diesen Referenzinformationen und einem allgemeinen Schlüsselcode (SRID) geführt. Auch innerhalb des GeoTIFF-Formates kann mittels Metadaten ein Referenzsystem (EPSG-Code) angegeben werden. Zum Betrachten und Editieren der Referenzierung ist Spezialsoftware notwendig. Die Bearbeitung mittels eines Standartbildverarbeitungsprogramms führt hingegen zum Verlust dieser Metainformationen.

Im Gegensatz dazu nutzt das von ESRI entwickelte Worldfile-Konzept eine zusätzliche ASCII-Datei gleichen Namens und formatspezifischer Dateiendung, die die 6 Parameter einer Affintransformation enthält. Die Vorteile Einfachheit und Bearbeitbarkeit stehen den Nachteilen der fehlenden Angabe eines Referenzsystems sowie der ungewöhnlichen Referenzierung des Rasterzellmittelpunktes gegenüber [Neb02].

3.1.3 Interleaving

Mit Interleaving wird die Methode der Sortierung von Rasterzellinformationen bei mehrbändigen Rastern bezeichnet. Möglich ist die Speicherung aller Zellen je Band nacheinander als Block gefolgt von einem nächsten Band (BSQ - Band sequential). Folgt einer Rasterzeile der Ebene 1 die entsprechenden Zeilen aller vorhandenen

Bänder, so bildet dies das Format 'Band interleaved by Line' (BIL). Bei BIP (Band interleaved by Pixel) werden die Pixel aller Ebenen zusammengehalten. Die beschriebenen Methoden werden von den Rasterformaten und Systemen in unterschiedlicher Weise unterstützt [Neb97]. Für digitale Geländemodelle sind aufgrund der überwiegenden Speicherung als ein Fließkommaraster mit einem Band diese Ordnungen selten relevant. Ein DGM kann aber aus Kompressionsgründen (Nutzung JPEG) auch als 8-Bit Raster, aufgeteilt auf 3 Bänder (je Höhenwert 24-Bit) und gekoppelt mit entsprechenden Interpretationsmethoden, abgelegt werden.
Da derzeit je Rasterzelle bei den digitalen Geländemodellen nur eine Information (Höhe) abgelegt ist, werden die Interleaving-Methoden im Projekt noch nicht benötigt. Eine mögliche Systemerweiterung des Modelldatenformats hin zur zusätzlichen Ablage beispielsweise von Bodenparametern bzw. Landnutzungsdaten würde dann die Speicherung mehrerer Bänder je Raster erfordern.

3.1.4 Metadaten

Zur Erfassung und Speicherung von Metadaten eines räumlichen Objekts wurden verschiedene Standards entwickelt. Beispiele sind der 1994 vom U.S. Geological Survey herausgegebene FGDC sowie der europäische CEN/TC 287 - Standard. Sehr umfangreich bzgl. Konzept und Inhalt ist der ISO TC/211. Auf die digitalen Geländemodelle der Küstenbereiche Mecklenburg-Vorpommerns könnte der NOKIS-Standard [Traub06] angewandt werden. Basiskategorien der Metadaten sind nach den obigen Standards in nachfolgender Tabelle zusammengefasst.

Informationen zu/zur/zum	**Beispiele**
Identifikation	Beschreibung, Stichwort, räumliche Ausdehnung
Datenqualität	Vollständigkeit, Attributqualität, Positionsgenauigkeit
Datenmodell	Datentyp (Geländemodell, Oberflächenmodell)
räumliches Bezugssystem	Referenzparameter, Koordinatensystem, Einheit
Attribut	Typ, Bereich, Auflösung, Einheit
Nutzung	Format, Transfermechanismus, Nutzungsbedingungen

Tab. 1: Metadatenbasiskategorien, Quelle: [Neb97]

Metadaten für Rasterobjekte können auch nach anderen Kriterien klassifiziert werden. Zu den inhaltsunabhängigen Metadaten werden nach Schmidt [Schm02] die präsentationsbezogenen Metadaten gezählt. Beispiele im Rasterdatenbereich sind hier Auflösung oder das verwendete Koordinatensystem. Ebenso zählen speicherungsbezogene Metadaten (z.B. Rasterdatentyp, Speicherformat) dazu.

Die inhaltsbezogenen Metadaten beschreiben auf einer semantisch niedrigen Stufe die Daten ohne Interpretation und können meist automatisch erfasst werden. Beispiele sind Höhenverteilung oder Textur eines Rasterobjektes. Sind diese Informationen durch das verwendete Managementsystem mit Hilfe von implementierten Funktionen automatisiert ermittelbar, so wird ihre Verwendung in den Metadaten empfohlen.
Zu den inhaltsbeschreibenden Metadaten zählen neben den kontextbezogenen Daten auch kontextbeschreibende und objektbeschreibende Metadaten. Beispiele sind in der folgenden Tabelle aufgeführt.

Klassifikation	**Metadaten**	**Beispiele**
inhaltsbeschreibend (interpretierend)	kontextbeschreibend	Angaben zur Rastererstellungsmethode, Filter- und Verarbeitungsschritte
	kontextbezogen	Identifikation (z.B. Kachelname), Raum- und Zeitdaten (z.B. letztes Revisionsdatum, geometr. Ausdehnung)
	objektbezogen	Land/Seeanteile, Anteile der unterschiedlichen Messmethoden, Genauigkeitsindex
inhaltsbezogen (nicht interpretierend)		Attributverteilung (z.B. mittleres Höhenniveau, min, max. Werte, Präzision) Texturparameter (z.B. Bodenrauhigkeit)
inhaltsunabhängig	präsentationsbezogen	Auflösung (z.B. 5x5m), Koordinaten-system (z.B. epsg=2399)
	speicherungsbezogen	Rasterformat (z.B. TIFF), Rastertyp (z.B. INT4)

Tab. 2: Einteilung von Metadaten, ähnlich [Schm02]

Neben den wertvollen beschreibenden Merkmalen der Rasterdaten für die Nutzer, verwenden Programmsysteme die enthaltenen Informationen beispielsweise auch zum Aufbau eines räumlichen Index oder zur optimierten Ausführung geometrischer Operationen [Brink06].

3.2 Allgemeine Rasterkonzepte und Managementansätze

Für ein modernes räumliches Rasterdatenmanagement sind einige Funktionalitäten grundlegend erforderlich. Nebiker [1997] teilt diese Erfordernisse in Basisfunktionen sichtbar für den Nutzer und erweiterte Funktionalitäten spezialisiert auf die Verwaltung großer Rasterobjekte und transparent für den Nutzer ein.
Die Datenhandhabung und Manipulationen durch einfache Grundoperationen (Erstellen, Löschen, Kopieren) von Einzelrastern und Mosaiken sowie das Erstellen, Einfügen und Extrahieren innerhalb von Rastermosaiken sind für ein Rastermanagementsystem (RMS) unabdingbar. Die Metadatenverwaltung wird als Schlüsseltechnologie in den Bereichen Attributverwaltung, Georeferenzierung und auch in softwarespezifischen

Erweiterungen benötigt. Eine räumliche Abfrage und der Rasterobjektzugriff, basierend auf räumlichen, thematischen und ggf. zeitlichen Metadaten, sollten möglich sein. Innerhalb eines DBMS werden die bisher beschriebenen Funktionalitäten durch die Modellierung eines Rasterobjektes in einer Sprache, beispielsweise SQL, umgesetzt. Die Rastervisualisierung als Grundfunktionalität ist aufgrund von vielen spezialisierten 2D/3D-Einzellösungen bzw. Konzepten meist etwas von der Verwaltung losgelöst und soll hier aufgrund der Fülle der Ansätze nicht explizit betrachtet werden. Kommuniziert wird zwischen den Softwarekomponenten über Schnittstellen bzw. Austauschformate.

Zur erweiterten Rastermanagementfunktionalität zählt Nebiker [1997] die Unterstützung für sehr große Rasterobjekte ohne negativen Einfluss auf die Systemperformance. Das Verwaltungssystem unterteilt dazu, unsichtbar für den Nutzer, die Rasterobjekte in handhabbare Stücke, um hauptspeicheroptimiert und effizient den erforderlichen Funktionsumfang bereitstellen zu können. Es werden spezialisierte Methoden bereitgestellt, die einen räumlichen Zugriff auf Teile bzw. Zellen innerhalb eines Rasters sowie deren Bearbeitung erlauben. Die Rasterdatenausgabe kann optimiert durch das System in verschiedenen Auflösungsstufen erfolgen. Dies verbessert die Anzeige- bzw. Transportgeschwindigkeit und ermöglicht nutzerbezogene Einschränkungen aufgrund von Sicherheits- oder Lizenzvorgaben. Ebenfalls zur erweiterten modernen Rastermanagementfunktionalität zählt die großteils automatisierte Bereitstellung von Speicherstrategien und -technologien, die einfache Datenmigrationen und Hierarchieänderungen erlauben.

Die sehr wesentlichen Konzepte Kompression und Indexierung werden ausführlich und praktisch unterlegt im Kapitel 7 im Zusammenhang der Optimierung von DBMS behandelt.

3.2.1 Multi Resolution Konzepte

Durch die Verwendung eines Multi-Resolution-Konzepts ist die Lieferung eines Rasters an eine Clientapplikation in einer von ihr benötigten oder in Abhängigkeit von Lizenz- bzw. Nutzerrechten erlaubten Auflösung möglich. Durch die geringere übertragene Datenmenge findet eine Optimierung des Transportprozesses statt. In bestimmten Anwendungsbereichen könnte, aufgrund der dortigen Rechengenauigkeit, eine verringerte Auflösung ausreichend sein. Letztlich kann bei visuellem Zugriff, beispielsweise in Übersichtskatalogen von Rastern oder bei optisch-unterstützten Abfragen, ebenfalls eine geringere Auflösung benötigt werden.

Merkmal	Ausprägungen
Zeitpunkt der Berechnung	bei Rastererstellung, bei Abfrage
Art der berechneten Daten	redundant, additiv
Art der Datenablage	diskret, progressiv
Bezugsobjekt	datengestützt, metadatengestützt

Tab. 3: Klassifikation von Multi-Resolution-Konzepten

Als *Basiskonzept*, verwendet vorwiegend in grafischen Rasterformaten (z. B. TIFF, JPEG), kann das im Rasterdatensatz eingebettete und gespeicherte Vorschau- oder Übersichtsbild angesehen werden.

Bei der *Interlaced-Speicherung* (Zeilensprungverfahren) werden Raster in mehreren Schichten unter Auslassung jeweils anderer Rasterzeilen gespeichert und übertragen. Vorteil ist der schelle Aufbau eines Übersichtsbildes bereits nach der Übertragung einer Schicht. Während des Ladevorganges wird mit jeder Schicht die Qualität und Auflösung höher. Verwendet wird diese Technik beispielsweise beim GIF-Format, welches 4 aufeinanderfolgende nichtredundante Teilbilder auch einzeln zugreifbar durch einen Offsetzeiger beinhaltet [Neb97].

Das *Multi-Resolution-Konzept der Bildpyramiden* speichert zusätzlich zur vollen Rasterauflösung hierarchisch geordnet Rasterobjekte gleichen Ausmaßes mit verringerten Auflösungen. Meist werden 3-5 Stufen unter jeweiliger Halbierung der Auflösung beider Rasterausdehnungen je Stufe abgelegt. FlashPix sowie das Kodak-PhotoCD-Format verwenden dabei Pyramidendaten verringerter Auflösung zusätzlich zur Quellauflösung (redundante Pyramiden). Andere Fachgebiete (z.B. Photogrammetrie) speichern stattdessen ein Raster der geringsten Auflösung sowie Differenzraster, mit denen durch Addition eine gewünschte Auflösungsstufe erstellt werden kann (additive Pyramiden) [Neb97].

Beim *Konzept der Rastergeneralisierung* wird der Rasterinhalt zur besseren Kompression und Indizierung durch Verfahren der Generalisierung (z.B. Klassifikation, Auswahl) vereinfacht.

Es gibt ebenso verschiedene Ansätze, die auf der Verwendung von *Metadaten* zu den Rasterdatensätzen beruhen. So können zusätzliche Informationen zu Rastern anderer Auflösung bzw. anderer Herkunft oder Speicherposition bei der Abfrage nach Daten eines Gebietes verwendet werden.

Häufig hat bei einer Auflösungspyramide jedes Niveau die gleiche räumliche Ausdehnung. Die Datenmenge zum Lesen der Rasterdaten eines niedrigen Levels

nimmt hier bzgl. der vollen Auflösung ab, die Übertragungsgeschwindigkeit zum Client zu. Bei großen Rastern bzw. Rastermosaiken wird zur Optimierung des Zugriffs meist eine Partitionierung des Rasters vorgenommen. Jedes Teilraster kann dann mit einer eigenen Auflösungspyramide versehen werden. Die Anzahl der Teilobjekte und damit der benötigten Zugriffe innerhalb eines Rasterdatensatzes steigt bei diesem Verfahren sehr stark an, die Datenmenge je Rasterteilstück nimmt mit abnehmender Auflösung ab. Vergrößert man die räumliche Ausdehnung je Rasterteil mit jeder Stufe der Auflösungsverringerung, so bleibt das zu lesende Datenvolumen je Auflösungsniveau und Rasterteil gleich. Die Anzahl der Rasterteile je Stufe nimmt ab.

Zugriff auf die Rasterteile verringerter Auflösung erhält man durch eine Indizierung der Teile. Jedes Rasterteil erhält eine Referenz auf den zugehörigen Bereich der größten Auflösung. Bei einer Abfrage werden dann die benötigten Teile der höchsten Auflösung ermittelt und durch eine Indexberechnung auf die Gebiete des gewünschten Auflösungslevels umgerechnet [Neb97].

$$Index_{LevelN} = Index_{Level0} - \mathrm{mod}(Index_{Level0}; 2^{2N})$$

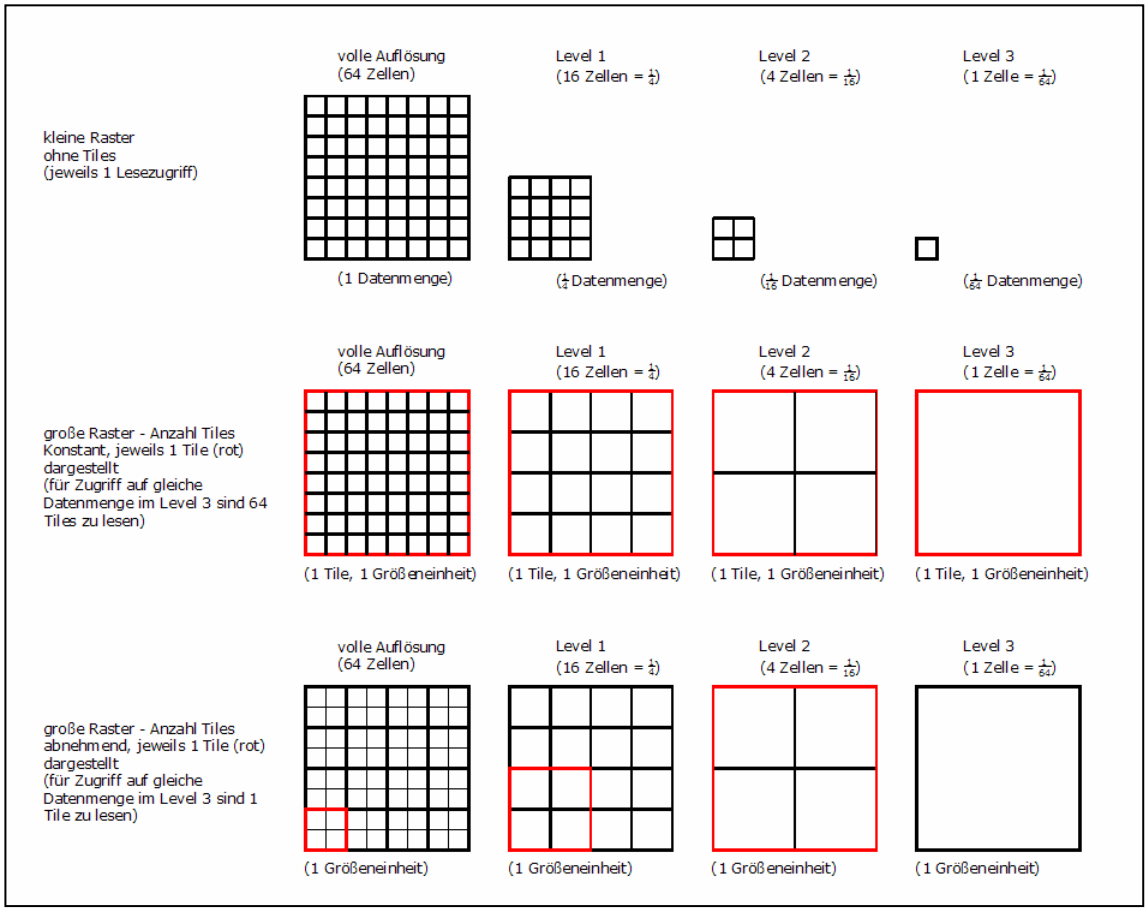


Abb. 2: Möglichkeiten von Auflösungspyramiden

Für Daten digitaler Geländemodelle sind Multi-Resolution-Konzepte mit dauerhafter zusätzlicher Datenablage selten relevant. Wird der Datensatz in verringerter Auflösung zur Analyse oder Berechnung von Geländeparametern benötigt, so werden Resampling- oder Interpolationsmethoden zur Änderung der Zellgröße angewandt. Da Geländemodelle im Allgemeinen nicht mit einer Farbtabelle zur direkten visuellen Ausgabe ausgestattet sind, sondern der Belegung durch ein thematisches Raster (z.B. Luftbild) bedürfen, wird häufig lediglich die beste Auflösung in der Verwaltung abgelegt und erst im Anwendungsfall eine Transformation durchgeführt.

3.2.2 Räumliche Unterteilungen von Rasterobjekten

Das Konzept der für Clientapplikationen und für den Nutzer transparenten Teilung von großen Rastern oder Mosaiken in einzelne Stücke (Partitionierung) zählt zu den erweiterten Funktionalitäten deren ein Rastermanagementsystem bedarf [Neb97]. Eine räumliche Unterteilung wird notwendig, wenn trotz Verwendung von Kompressionsmethoden, die Haltung des gesamten Rasters im Hauptspeicher des Systems nicht mehr möglich ist. Eine räumliche Unterteilung sollte die Unterstützung und Gewährleistung von effektivem räumlichen Zugriff und Abfrage bei geringer Prozessorzusatzbelastung und geringem zusätzlichem Datenaufwand ermöglichen. Räumlich zusammengehörige Rasterelemente werden unter möglichst großer Beibehaltung der Nachbarschaftsbeziehungen im Speicher angeordnet. Eine Kompression der individuell geteilten Rasterstücke sollte unterstützt werden. Raumteilungen können in statische sowie dynamische Strukturen und Transformationen unterschieden werden.

Die Methode der *Scan-Lines* (Streifen) unterteilt ein Raster statisch in Pixelreihen. Sie ist ein fundamentales Konzept für viele Rasterformate (z.B. TIFF), entstanden aus linienhaften Erfassungsmethoden für Rasterobjekte (Scanning) und mit wenig Rechenaufwand geeignet für klein- bis mittelgroße Raster. Es werden externe Metadaten (Dimensionen) zur Zugriffsadressberechnung benötigt. Große Raster bzw. Raster mit sich verändernden Ausdehnungen werden durch diese Methode nicht optimal unterstützt. Erweiterungen des Konzeptes gruppieren Rasterzeilen zu Zugriffsblöcken [Neb97].

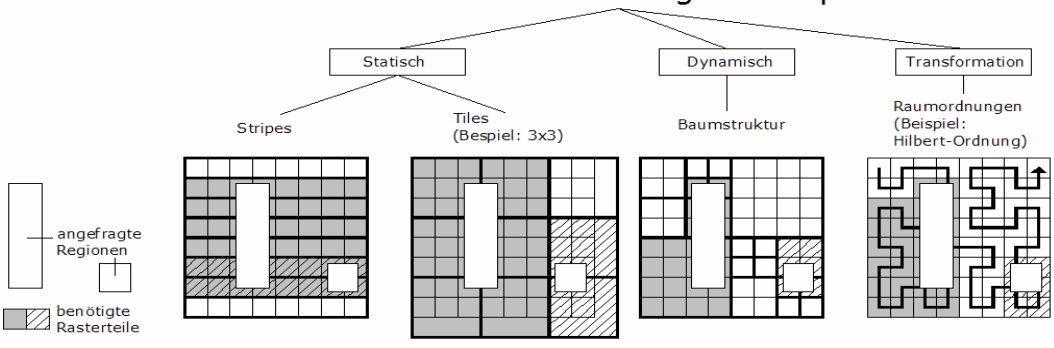
$$AdresseRasterstreifen = Rasterursprungsadresse + (Reihennummer * Rasterbreite * Bytes_{Zelle} * Bands)$$

Das *Tilingkonzept* bildet durch die Unterteilung des Rasterobjekts in gleich große oft quadratische eigenständige Raster eine Aggregation der Zellen auf einem höheren Rasterlevel. Dynamische Änderungen der Größen des Quellrasters werden durch

Erzeugung bzw. Löschung von Raster-Tiles (bei Oracle Spatial: Rasterblocks) unterstützt. Verwendet wird dieses Konzept von einigen Rasterformaten (z.B. TIFF) sowie in räumlich erweiterten RDMS (Oracle SDO). Es benötigt für den Tilezugriff eine separate räumliche Indexstruktur (z.B. Oracle SDO: RasterDataTable).

Baumstrukturen können eine hierarchisch geordnete hochdynamische Unterteilung des Datenraums mit Hilfe von Splitheuristiken vornehmen. Beispiele sind der R-Tree oder der Quadtree.

Raumordnungen transformieren den mehrdimensionalen Rasterdatenraum mittels einer raumfüllenden Kurve in eine lineare Ordnungsstruktur (Beispiel: Z-Ordnung, Hilbert-Ordnung). Die Nachbarschaftsbeziehungen bleiben gut erhalten, Kompressionen der Daten sind möglich [Neb97]. Näheres zu den letztgenannten Strukturen enthält Kapitel 7.2 Indexierung.



Strategie	Rasterzeilen	Rasterteile einheitlicher Größe	hierachische Teilung	Transformation in lineare Struktur
Unterstützung räuml. Erweiterungen	ja (entlang Reihen)	ja (Tiles-Basis)	ja (MBR in Knoten)	ja (entlang Raumordnung)
räumlicher Zugriff	externer Index	externer Index	in Struktur enthalten	in Struktur enthalten
Kompression	ja	ja	ja	ja
Erhaltung Nachbarschaft	wenig	wenig	gut	hoch
Komplexität	gering	mittel	hoch	hoch
Verbreitung	hoch	hoch	gering	gering

Abb. 3: Rasterunterteilungskonzepte, Quelle: [Neb97], verändert

3.2.3 Storage Konzepte

Ein effektives Rasterdatenmanagement benötigt durch die Größe der zu übertragenden Daten eine sinnvoll kombinierte Nutzung verschiedener Speichertechnologien. Nach dem Speicher-Level kann unterschieden werden in in-line (sofort verfügbar, RAM), on-line (zugriffsbereites Laufwerk), near-line (nicht zugriffsbereites Laufwerk, Medienwechsel durch Robotik z.B. Jukebox) oder off-line (externe Archive, Zugriff nur durch Nutzeraktion, z.B. Bandarchive) [Neb97]. Eine Speicherhierarchie sortiert die möglichen Datenablagepositionen nach sinkender Zugriffsgeschwindigkeit. Nach

Prozessorregistern und Prozessorcache folgen dabei Arbeitsspeicher (jeweils RAM), interne und externe Festplatten sowie alle Arten von Wechseldatenträgern. Im Datenbankbereich wird traditionell in Primärspeicher (Arbeitsspeicher), Sekundärspeicher (Festplatten) und Tertiärspeicher (Optische Systeme, Bandlaufwerke) unterschieden. Hauptperformancefaktor bilden durch ihre ständige Verwendung die schnellen RAM-Speicher. Alle datenbezogenen Optimierungsstrategien (z.B. Indexierung) beruhen auf der Verringerung von Zugriffen auf die langsameren Sekundärspeichermedien. Zur Steigerung der Leistungsfähigkeit dieser Medien sind verschiedene Hard- und Softwaretechnologien (z.B. Stripping-RAID-Systeme) entwickelt worden. Die höchsten Datenkapazitäten bieten die Tertiärspeichersysteme, die bei den heute verfügbaren schnelleren Festplattenspeichern nur noch zur Langzeitarchivierung (Backup) verwendet werden.

Speichermanagementkonzepte dienen der optimalen Nutzung vorhandener Speichertechnologien. So können Declusterung (z.B. Verteilung Datenblöcke auf verschiedene physikalische Festplatten beim RAID) oder auch Clusterung (z.B. Sortierung eines Objektes auf ein Medium bei optischen Laufwerken) zu einer Performancesteigerung führen. Spezialisierte Dateisysteme ermöglichen bei den nicht direkt zugreifbaren Tertiärspeichern, beispielsweise durch Directory Caching, die Abfrage auf Basis von Metadaten sowie die daraus folgende gezielte Mediensuche. Hierarchische Speichermanagementstrategien (HSM) integrieren alle Speichertechnologien so, dass beispielsweise durch Datenmigration auf Grund von Nutzungsmustern die jeweiligen Ressourcen zur Steigerung der Gesamtperformance optimal ausgenutzt werden. Hohe Transferraten für ständig benötigte Daten (z.B. Indexdaten) werden mit hohen Kapazitäten für große Datenvolumina (Rasterdaten) verknüpft [Neb97].

Datenbankmanagementsysteme ermöglichen sowohl die Speichermanagementkontrolle durch dieses System selbst als auch die Verwaltung durch nachfolgende Systeme (z.B. Betriebssystem). Hierbei werden lediglich Referenzen auf die Objekte in der Datenbank gespeichert. Nachteile liegen in der fehlenden Transaktionskontrolle durch die Datenbank, dem nicht möglichen direkten Zugriff durch eine universelle Abfragesprache (SQL) sowie die meist propritäre Implementierung der Im- und Exportfunktionen durch die Applikation. Werden die Rasterobjekte durch das DBMS verwaltet, so sind je nach Ausbaustufe des DBMS und verwendetem Datentyp (Array/BLOB/ADT) verschiedenste Speicherstrategien (z.B. Auswahl Medium nach Performance - HSM) und Integrationen in die Abfragesprache SQL möglich.

4 Datenerfassung und Rastererstellung

4.1 Datenerfassung

Zur Herstellung eines digitalen Geländemodells können verschiedenste Erfassungsmethoden genutzt werden. Zu den originären Methoden zählen die direkten Messungen am Objekt sowie die indirekten Erfassungen an dessen Abbild. Eine sekundäre Erfassung basiert auf der Auswertung eines für einen bestimmten Anwendungszweck erstellten Produkts. Bereits die Qualität der Messdaten bestimmt neben der Qualität der Verarbeitungsprozesse die Nutzbarkeit der daraus errechneten Rasterdaten. Ziel sollte daher ein integriertes und dokumentiertes Datenmanagement bis hin zum Endprodukt, dem Rasterobjekt eines Verwaltungssystems, sein.

Tachymetrie als direkte Methode ist die punktweise Erfassung der Situation und Geländeoberfläche durch polare Bestimmung von Objektpunkten von einem günstig gelegenen Standort aus. Aufgenommen werden die charakteristischen Geländelinien (Gerippelinien, Falllinien, Formlinien, Kantenlinien) sowie die in Abhängigkeit vom Genauigkeitsanspruch sachgerecht verteilten Geländepunkte mit ggf. örtlich angepasster Aufnahmedichte [Hak94]. Eine GPS-Messung ermöglicht die effektivere Aufnahme eines größeren Gebietes als bei der Tachymetrie. Die Nutzung eigener oder amtlicher Referenzempfänger beim Differential-GPS führt zu einer für hochgenaue Geländeanalysen ausreichenden Genauigkeit. Die Erfassung der überflutungsgefährdeten Gebiete im Küstenbereich wird durch die Messung von wesentlichen Geländebruchkanten der küstenschutzrelevanten natürlichen und künstlichen Objekte (Küstenschutzbauwerke) sowie flächenhaft durch Geländepunkte bei Begehbarkeit des Geländes (ggf. im Winter) hauptsächlich mittels GPS durchgeführt. Nur die Erfahrung der beteiligten Vermessungsbüros und die örtliche Anpassung der Methodik führen bei dieser Aufgabenstellung zu einem optimierten Messergebnis. Die für die Trennung der Land- und Seebodenmodelle wesentliche Uferlinie (technische Linie der Höhe -0.14m) kann ebenfalls nur manuell gemessen werden, da unterschiedliche Pegelstände und die erforderliche Genauigkeit andere Messmethoden ausschließen.

Laserscanning bezeichnet das zeilen- oder rasterartige Überstreichen einer Oberfläche mit einem Laserstrahl zur Vermessung und/oder zur Bilderzeugung (Intensitätserfassung). Dabei sind die Teilprobleme 'reflektorlose Streckenermittlung' und 'Sensororientierung in einem Koordinatensystem' zu lösen. Es werden derzeit 2 verschiedene Messverfahren angewandt. Beim Impulsverfahren wird nur ein kurzer Laserimpuls ausgesandt. Die Laufzeit bis zum Empfang der reflektierten Strahlung

bildet ein Maß für die doppelte Entfernung vom Messsystem zum Objekt. Beim Phasenmessverfahren hingegen werden kontinuierlich Laserstrahlen, deren Amplitude durch mehrere sinusförmige Wellen unterschiedlicher Wellenlände moduliert wird, zum Messobjekt geschickt. Zur Bestimmung des Abstandes wird die Phasendifferenz zwischen der Phasenlage des gesendeten und empfangenen Signals betrachtet. Das zweite Teilproblem (Georeferenzierung) wird durch synchrone Ermittlung der Position des Lasers mittels DGPS und der Orientierung mit Hilfe eines inertialen Navigationssystems (INS) gelöst. Nach einer Prozessierung werden aus Position, Orientierung und gemessener Strecke dreidimensionale kartesische Punktkoordinaten errechnet (Punktwolke). Eine anschließende Klassifizierung (Last-Pulse-Filterung) dient der Eliminierung der Objektoberflächen aus der Menge der Messpunkte. Laserscanning kann terrestrisch (bodengestützt) oder aus der Luft in einem Flugzeug oder Hubschrauber betrieben werden. In Abhängigkeit von der Morphologie des Geländes wird bei einem gegenüber einer Luftbildbefliegung höheren Flugaufwand (zusätzlicher Kalibrierungsflug mit topographisch bestimmten Passpunkten) in Profilen/Streifen je nach erforderlicher Genauigkeit das Gelände flächig erfasst. Terrestrisch ist eine flächige Geländeaufnahme durch Messung und Orientierung mehrerer in Sichtweite liegender Standorte möglich, wenn die entstehenden Punktwolken durch Verknüpfungspunkte zusammengeführt werden können. Schulz (2004) gibt bei sorgfältiger Planung mit diesem Verfahren Genauigkeiten im Millimeterbereich an. Laserscanning ist ein auch in schwer zugänglichen Gebieten einsetzbares Verfahren mit hoher Flächenleistung bei geringem Personaleinsatz. Es kann im Gegensatz zur Photogrammtrie (wg. Schattenwurf) auch bei durchlässiger Vegetation (z.B. Wald im Winter) und in Gebieten, die mangels Textur keine Stereomessungen zulassen (Sand, Watt), eingesetzt werden. Eigene Erfahrungen mit Messdaten haben aber gezeigt, dass in Bereichen mit sehr feiner dichter Vegetation (z.B. Schilfflächen) die Filterung der Vegetationspunkte aufgrund geringer Bodenpunktdichte nicht möglich ist. Hier sind tachymetrische Kontrollpunkte zur Abschätzung der Überflutungsgefährdung (DGM-Korrektur) zwingend notwendig. In unzugänglichen und abbruchgefährdeten Gebieten (Steilküsten, Spülfelder) wurde im in Kapitel 8 beschriebenen Projekt kleinräumig erfolgreich terrestrisches Laserscanning eingesetzt. Luftgestütztes Laserscanning bietet nach Herstellerangaben Höhengenauigkeiten von 5-15cm bei maximalen Lageabweichungen von 30-50cm und stellt damit für die Erzeugung hochgenauer großflächiger Geländemodelle im kaum bewegten überflutungsgefährdeten Küstenbereich und in Kombination mit konventioneller Kontrollmessung die beste

Messmethodenkombination im Landbereich dar. Aufgrund der Reflektionseigenschaften sowie der veränderlichen Pegelstände des Wassers liefert das Laserscanning im Küstenbereich seeseitig keine verwertbaren Daten. Die Trennung (Land/See) erfolgte durch eine mittels GPS gemessene hochgenaue Uferlinie.

Ohne wetterbedingte Einschränkungen und zusätzlich auch von Satellitenplattformen aus lässt sich mit Hilfe aktiver Mikrowellen eine *Radarmessung* des Geländes durchführen. Aus der Phasendifferenz der ausgesendeten und empfangenen aktiven Mikrowelle kann in Kombination mit Positionsbestimmung (GPS) und Orientierung (INS) die Höhe der Geländeoberfläche bestimmt werden. Die Stärke der rückgestreuten Strahlung ist abhängig von Landbedeckung, Bodenrauhigkeit und Bodenfeuchte [Krau00]. Häufig wird ein Seitensichtradar mit synthetischer Apertur (SAR) verwendet, das flugzeuggestützt Genauigkeiten im Meterbereich liefert. Von einer Satellitenplattform aus und in vegetationsreichen Gebieten werden von Kraus (2000) maximale Höhenfehler von 6 Metern angegeben. Diese Fehler sowie die hohen technischen Anforderungen und Kosten verhindern derzeit den Einsatz von Radar zur Vermessung der überflutungsgefährdeten Küstengebiete. Eine Genauigkeit der Höhenmessung im Laserscanningbereich (max. 25cm) wird angestrebt. Zukünftig sollte die Möglichkeit mittels Radar den Meeresboden erfassen zu können, stärker genutzt werden (z.B. Envisat RA-2).

Im Küstenbereich ist die Vermessung der angrenzenden Gewässerabschnitte von großer Bedeutung. Durch Parallelprofile mit einem *Echographen* oder flächenhaft mit einem *Fächerecholot* kann durch Messung der Ultraschalllaufzeit, abhängig von Wassertemperatur und Salzgehalt, aus einem schwimmenden Fahrzeug mit Positionierung (DGPS) sowie einem Sensorsystem zur Neigungskorrektur und einem Pegelausgleich die Oberfläche des See- bzw. Meeresgrundes ermittelt werden. Die minimal benötigte Wassertiefe liegt dabei bei etwa 1m. Die Daten aus dieser Form der Seebodenvermessung lagen der Modellerstellung des DGM in Teilabschnitten flächig und zum Großteil in Form von gemessenen Einzelprofilen vor. Zur Bildung eines nahtlosen Anschlusses an die Landmessungen sind neben einer vermessenen Uferlinie auch manuell gemessene Profile in See (bis 0.8m Wassertiefe) verwendet worden.

Zu den indirekten Herstellmethoden eines Geländemodells zählt die manuelle oder (teil)automatische Umsetzung einer analogen Vorlage in digitale Form (*Digitalisierung*). Zu beachten sind die maßstabs-, objekt-, und darstellungsbedingten Generalisierungen, die fachspezifische Interpretation bei Herstellung [Hak94] sowie auch die oft geringere Aktualität der Vorlage. Eine großflächige Digitalisierung und

Aufbereitung der Höhenlinien der topographischen Karte 1:10000 des Küstengebietes Mecklenburg-Vorpommerns bildete die erste Grundlage zur Geländemodellerstellung. In diesen Datenpool wurden, soweit vorhanden, die genaueren und aktuelleren Daten anderer Quellen eingearbeitet.

Photogrammetrie ist eine flächenhafte indirekte Datenaufnahmemethode mit hoher Flächenleistung, sehr guten Genauigkeiten und auch hohem Aufwand. Von einem Sensor werden aus einem Flugzeug, Hubschrauber (Aerophotogrammetrie) oder Satteliten mit einem analogen (optisch-photographisch) oder digitalen System (CCD-Sensoren, aktive/passive Abtastsysteme) die von den abzubildenden Objekten unterschiedlich reflektierten Strahlen gesammelt und auf einen Informationsträger gespeichert. Das erfassbare Wellenlängenspektrum erstreckt sich vom sichtbaren Licht, über Infrarot bis in den Mikrowellenbereich. Die Auswertung der Abbildungen (analog oder digital) erfolgt im Postprozessing (z.B. Stereoauswertung) nach innerer und äußerer Orientierung über enthaltene Passpunkte. Zur Ermittlung von Eingangsdaten zur DGM-Erstellung wurden unterschiedliche Verfahren der Auswertung entwickelt. Das personell aufwendige Nachvollziehen einer tachymetrischen Messung ermittelt morphologische Einzelpunkte und wichtige Geländebruchkanten bei geringst möglicher Punktanzahl und größtmöglicher Genauigkeit. Die dynamische Profilierung beinhaltet die automatische Erzeugung von Profilen aufgrund spezifischer Kriterien für eine Punktauslösung mit Genauigkeiten, die abhängig von Abtastgeschwindigkeit und Geländeneigung sind. Möglich sind auch Gittermessungen sowie das Progressive Sampling, bei dem das Gelände entsprechend seiner Bewegtheit mit angepasster Punktdichte vermessen wird [Krau00]. Das Abfahren von Höhenlinien mit automatischer Registrierung von Punkten bietet eine hohe Punktdichte entlang der Linien sowie eine sehr geringe Dichte senkrecht dazu (Flachland) und ist damit für die DGM-Erzeugung nur eingeschränkt geeignet. Neuere PC-Methoden automatisieren den Auswertevorgang durch digitale Musterübereinstimmungsprüfung (Matching-Verfahren) der Stereopartner. Probleme treten hier durch Positionsveränderungen von schnellen Objekten (Fahrzeuge) auf. Wenig geeignet ist die Photogrammetrie in bebauten und vegetationsreichen Gebieten (Schattenwurf, keine Sichtbarkeit der Erdoberfläche) sowie in Gebieten mit für die Aerotriangulation notwendiger schwieriger Blockgeometrie (Steilküsten) oder geringer Textur (Sand, Watt) [Lind02]. Diese Problemgebiete lassen neben dem hohen Aufwand für Geräte und spezialisiertem Personal den Einsatz dieser Technik zur Erstellung von Geländemodellen im

Küstenbereich kaum geeignet erscheinen. Im Projekt wurde die Photogrammetrie daher nicht zur Datengewinnung eingesetzt.
Zur Entscheidung für ein bestimmtes Messverfahren zur DGM-Generierung sind neben Aufwand, Kosten, Flächenleistung, möglicher Aufnahmegeometrie des Verfahrens und benötigter Ausrüstung auch die Eignung für das Untersuchungsgebiet sowie die erreichbaren Genauigkeiten zu beachten.

4.2 Konzepte zur Erstellung von Rasterdaten

4.2.1 Interpolationen

Die Weiterverarbeitung der Messpunkte benötigt zur DGM-Generierung eine Vorschrift zur Ermittlung der Zwischenwerte des unregelmäßigen Stützpunktfeldes. Geht die dabei entstehende kontinuierliche Oberfläche durch die Stützpunkte, so wird von einer exakten Interpolation gesprochen. Alternativ wird bei einer Approximation des Geländes eine Glättung bzw. Filterung vorgenommen [Rei01]. Eine sehr einfache Methode der Rasterinterpolation, eingesetzt bei Daten, die bereits die erforderliche räumliche Dichte besitzen, ist die *Nearest-Neighbor-Interpolation*. Sie wird häufig lediglich zur Füllung von Rasterfehlstellen eingesetzt. Im vorliegenden Projekt wurden damit die Laserscanningdaten aufbereitet. Die *Minimum-Curvature-Methode* ermittelt durch die iterative Lösung eines Gleichungssystems die glätteste mögliche Oberfläche durch die Stützpunkte mittels Minimierung der Summe der Krümmungen. Sehr bekannt und weit verbreitet ist die *Methode der inversen Distanzen*. Der gesuchte Zellwert wird mit Hilfe des nach dem umgekehrt proportionalen Abstand gewichteten Mittels der benachbarten Messwerte geschätzt. Als Parameter sind dabei verschiedene Gewichtspotenzen, Suchradien zu den Stützpunkten zur Verrechnung von Anisotropie (Richtungsabhängigkeit) sowie auch ein Glättungswert möglich. Aufgrund der weiträumig aber fast regelmäßig vorliegenden Messdaten des Gewässergrundes wurden für diese Teilbereiche IDW-Interpolationen im Projekt genutzt. Eine *Polynomregression* liefert mittels einer Polynomfunktion eine global angepasste Regressionsoberfläche. Der gewählte Polynomgrad bestimmt die Komplexität des zu lösenden Gleichungssystems und damit die Exaktheit der Anpassung der Oberfläche an die Stützpunkte. Bei der *Flächensummation* wird die Oberfläche aus der Überlagerung der Rotationsflächen einer spezifizierten Kernfunktion um alle Stützpunkte ermittelt. Durch lokal angepasste Funktionen und Glättungen können auch Geländebruchkanten bzw. Gerippelinien berücksichtigt werden. Angenehm 'ästhetische' Oberflächen erzeugen die häufig verwendeten *Spline-Interpolationen*. Lokal angepasste Polynome dritten Grades werden

als Bausteine glatt zusammengesetzt. Es haben sich eine Vielzahl von Varianten dieser Interpolationsgruppe entwickelt, die jeweils spezielle mathematische Vorgaben zur Optimierung der Oberfläche nutzen. Statistische Annahmen über den räumlichen Zusammenhang der Eingangsdaten werden bei der Methodenfamilie der *Kriging-Interpolationen* verwendet. Mittels eines an die Daten angepassten Variogramms kann für alle Stützpunktabstände richtungsabhängig die räumliche Korrelation angegeben werden. Der Zellwert wird mit einem nach dem geostatistischen Modell optimiert gewichteten Mittel der Nachbarwerte geschätzt. Kriging stellt aufgrund der statistischen Modellanpassung eine sehr aufwendige aber auch hochgenaue Interpolationsmethode dar. Die *Dreiecksvermaschung nach Delaunay-Kriterien (TIN)* unter Verwendung von zusätzlichen Geländeinformationen in Form von Bruchkanten kann nach linearer oder kubischer Interpolation von Rasterpunkten auf die entstandene Dreiecksoberfläche auch in ein Raster-DGM überführt werden. Die Vorteile, beispielsweise lokale Anpassungen bei Stützpunktveränderungen sowie Eindeutigkeit und Berechnungsgeschwindigkeit, können dabei nur zum Teil erhalten werden. Eine Dreiecksvermaschung mit anschließender Umwandlung in das Rasterformat wurde zur Aufbereitung der Vermessungsdaten des Projektes angewandt. Da hier neben Geländepunkten auch umfangreich Bruchkanten, beispielsweise von küstenschutzrelevanten Deichen vorlagen, bietet dieses Verfahren die exakteste Modellierung der Geländeoberfläche. Zum Teil wurde auch in Bereichen, in denen Laserscanning als Raster bereits vorlag, die exakte Geländemodellierung aus den mit höherer Genauigkeit gemessenen DGPS-Punkten dem Laserscanning vorgezogen. In Gebieten mit starkem Vegetationsbewuchs, bei gut bestimmbaren Kanten (Ufermauern, Spundwände, Hafenkai) sowie bei Bauten im Wasserbereich (Wellenbrecher, Dämme) bildet die DGM-Berechnung mittels Triangulation und Umwandlung ins Raster die beste und genaueste Modellierungsmethode.

Ziel einer Interpolation ist es, aus einer Stichprobe von Informationen ein gut nachgebildetes Modell einer Geländeoberfläche möglichst schnell und mathematisch einfach zu erzeugen. Die zur DGM-Erstellung verwendeten Arbeitsschritte, Verfahren und Parameter sollten immer der Dokumentation des Rasterdatensatzes als Metadaten mitgegeben werden. Da im Projekt sehr verschiedene Methoden und Verfahren eingesetzt wurden, sollte in einer zukünftigen Projektausbaustufe die kleinräumige kachelweise Erfassung der verwendeten Modellierungsparameter in Form eines Metadatensatzes angestrebt werden.

4.2.2 Mosaiking

Mit Mosaiking wird die Überlagerung mehrerer bereits vorhandener Quellraster zu einem gemeinsamen Zielraster bezeichnet. Dieser Prozess kann vom einfachen Umkopieren von Zellwerten bis zu komplexen Workflows reichen. Werden beispielsweise 2 Luftbildraster zu einem Orthophotomosaik fusioniert, so ist nach einer radiometrischen Transformation beider Eingangsraster (Anpassung des radiometrischen Spektrums) auch die Festlegung der Prioritäten beider Bilder notwendig. Es folgen entweder einfache Transformationen (Verschiebung, Resampling) oder auch komplexe geometrische Verarbeitungsprozesse (z.B. Orthorektifikation) sowie die Verschmelzung beider Zellwerte einer Position mit Hilfe einer festgelegten Operation.

Beim Mosaiking-Prozess von DGM des betrachteten Projektes (keine radiometrischen Informationen) wird von einheitlichen Modellparametern der Eingangsdaten ausgegangen. Sowohl Georeferenzierung (Koordinatensystem) als auch das verwendete Höhensystem sowie die exakte Lage der Rasterzellen und deren Datentyp sollten übereinstimmen. Es ist daher weder eine neue Referenzierung noch ein Resampling oder eine vorausgehende Anpassung der Zellwerte notwendig. Zur Überlagerung der Zellwerte werden lediglich Angaben zur Priorität der Eingangsraster gemacht. Zellen aus Rastern höherer Priorität ersetzen die Zellwerte aus Rastern geringerer Priorität, wobei Rasterfehlstellen (NoData-Werte) mit den vorhandenen Werten der höchsten Priorität aufgefüllt werden. Im Projekt bilden die Grunddaten aus der Digitalisierung (Land) und der Seevermessung die geringste Prioritätsstufe. Es folgen Laserscanning und Vermessungsdaten. Betrachtet man das Aktualisierungsverfahren der DGM-Kacheln, so ersetzen neuere Daten die vorhandenen Altdaten vollständig. In einer zukünftigen Projekterweiterung sollte die Sicherung der Altdaten zur Möglichkeit der Zeitreihenbildung und Analyse vorgesehen werden.

In Abhängigkeit vom verwendeten Softwaresystem sind beim Mosaiking einige Einschränkungen zu beachten. Bei Oracle Spatial dürfen die Eingangsraster lediglich überlappungsfrei sowie unter vollständiger Gebietsabdeckung vorliegen. Das Ergebnis der Mosaikoperation enthält in diesem Fall alle Zellwerte der Quellraster. Zelloperationen (Ersatz, Vereinigung) werden nicht durchgeführt.

Bei der ESRI-Software wird diese Einschränkung nicht gemacht. Unter Angabe einer Funktion werden die Zellwerte der sich überlappenden Bereiche aus den Quellrastern errechnet. Möglich sind die Funktionen First, Last (Prioritätenbildung), Blend, Mean sowie Min und Max.

First	Aus der Liste der Quellraster wird in überlappenden Bereichen der Wert des ersten Rasters mit gültigem Zellinhalt verwendet.
Last	Analog wird hier der Wert des letzten Rasters, sofern vorhanden, verwendet. Möglich ist mit diesen Methoden z.B. ein Einblenden eines Teilgebietes (Update) in ein vorhandenes DGM.
Blend	Hier wird Wichtung der Zellwerte nach ihrem Abstand zum Rand ihres Rasters vorgenommen. Es findet damit ein gleitender Übergang zwischen den Rastern innerhalb der Überlappungszone statt.
Mean	In Überlappungsgebieten wird der Mittelwert der Quellzellen gebildet.
Min/Max	Verwendung des Minimal/Maximalwertes

Tab. 4: Mosaikfunktionen bei ESRI-Software

ESRI bietet bei eventuell vorhandenen Farbtabellen der Quellraster die Übernahme der ersten/letzten Tabelle, die Kombination (MATCH) sowie den Abbruch bei verschiedenen Farbtabellen (REJECT). Für die digitalen Geländemodelle des Projektes (ohne Farbinformationen) ist dieser Parameter nicht relevant.

4.2.3 Rasterabfragen und Map Algebra

Unter Verwendung vorhandener Raster kann mittels mathematischer oder logischer Methoden ein neues Raster erstellt werden. Neben selektiven Abfragen, lokalen und globalen Statistiken, Berechnungen auf Zellbasis sowie Klassifikationen bieten heutige rasterverarbeitende Systeme meist auch komplexere anwendungsspezifische Berechnungsalgorithmen.

Disziplin	**Berechnungsziel**
Geologie	Erosivität
Biologie	Vegetationsindex, Neigung, Ausrichtung
Raumplanung	Sichtbarkeit
Hydrologie	Fließrichtung, Einzugsgebiete

Tab. 5: Beispiele für Rasterabfragen und Berechnungen

Im Projekt wird diese Rastertechnik zu Ausweisung von potenziellen und realen Überflutungsflächen benötigt. Eine potenzielle Überflutungsfläche eines Gebietes wird durch Abfrage aller Zellwerte auf ihre Höhenlage unterhalb eines vorgegebenen Niveaus (Wasserstand) ermittelt. Reale Überflutungsflächen werden daraus durch zusätzliche Überprüfung von schützenden Bauwerken oder natürlichen Höhenzügen berechnet. Sie zeigen nur die überfluteten Flächen, die von der gewählten Sturmflutquelle (seeseitig oder boddenseitig) erreichbar sind. Ergebnis sind Vektorflächen, die mit Landnutzungen und Bevölkerungsdaten zu Schadenspotentialabschätzungen oder mit Biotoptypenkartierungen zu Prognosen der Biotopveränderungen (z.B. Waldsterben) verschnitten werden können. In zukünftigen Modellerweiterungen sind Betrachtungen zu Bodenparametern (Durchfluss von Bodenpassagen) und Zeiträumen (zeitlicher Verlauf des Anstaus) in gesonderten Forschungsprojekten zu betrachten.

4.2.4 Resampling

Bei Interpolationen, Auflösungsänderungen und Drehungen sowie bei einer Georeferenzierung werden die Pixel eines Rasters mittels einer Resampling-Methode in eine neue Anordnung gebracht. Unterschieden wird in direkte und indirekte Umbildung [Krau00]. Bei der direkten Umbildung bzw. Entzerrung wird der Rasterzellenmittelpunkt durch eine entsprechende Operation auf die 'neuen' Koordinaten transformiert. Mit einer Resampling-Methode wird für den Rasterzellenmittelpunkt des neuen Rasters ein geeigneter Wert aus den transformierten Zellwerten ermittelt. Nachteil ist die mögliche Bildung von Lücken bzw. Fehlstellen im neuen Raster. Bei der häufiger verwendeten indirekten Umbildung werden hingegen die Rastermittelpunkte des neuen Rasters auf die Lage des Quellrasters invers transformiert. Dort wird dann ein Wert für die Zelle ermittelt und dem Zielraster übergeben. Nutzt man die *Nearest-Neighbor-Methode*, so wird der Zellwert der dem transformierten Punkt am nächsten liegenden Zelle als neuer Zellwert verwendet. Nachteile dieser Methode liegen im Auftreten von Sägezahneffekten an linearen Strukturen, der Verschiebung der Zellinformationen um bis zu 1/2 Pixel sowie in der Möglichkeit der Veränderung der statistischen Werte des Rasters (min, max...). Bei einer *bilinearen Interpolation* wird aus den 4 Nachbarn ein hyperbolisches Paraboloid gebildet. Mittels des Gleichungssystems der Fläche lässt sich der Zellwert berechnen. Es entstehen keine Versetzungen, durch die Berechnung werden aufgrund einer ausgleichenden Dämpfung die Rasterkontraste vermindert. Die aufwendigste häufig angebotene Methode ist das *bikubische Resampling*. Hier wird aus den 16 Nachbarn mit Hilfe eines Gleichungssystems dritten Grades der benötigte Wert errechnet [Krau00].

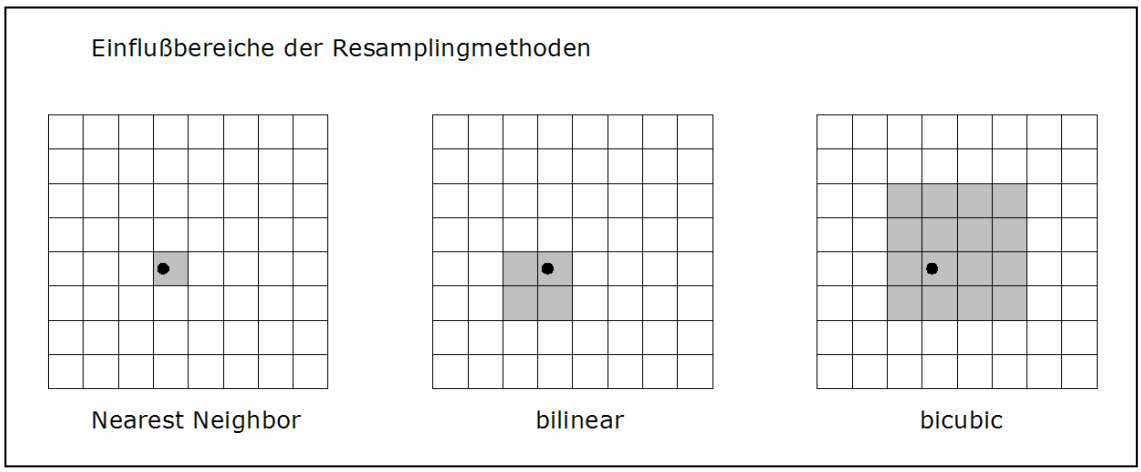


Abb. 4: genutzte Zellen bei Resamplingmethoden, Quelle: [Neb02]

Im betrachteten Projekt wird ausschließlich die Nearest-Neighbor-Methode als Voreinstellung benutzt, um eventuell vorhandene geringe Ungenauigkeiten

auszugleichen. Da bereits bei der Rastererstellung die genaue Lage der Rastermittelpunkte und durch die Rasterauflösung auch die Rasterausdehnung festgelegt wird sowie das Referenzsystem einheitlich vorgegeben ist, kann es bei den Mosaikingoperationen nicht zu einer Umbildung (Resampling) kommen. Die Methoden bilinear und bikubisch eignen sich nur bedingt für die Verarbeitung von DGM, da unerwünschte Glättungseffekte der Höhenwerte auftreten. Bruchkanten (z.B. Steilküste) werden stark verändert.

4.2.5 Formatänderungen und Transformationen

Neue Raster können auch durch Rotation, Translation bzw. Skalierung bestehender Raster errechnet werden. Unter Angabe einer Resampling-Methode werden die Zellwerte des Zielrasters aus denen des Quellrasters bestimmt.

Bei der Änderung von Datenformaten wird durch Parameter oder Formateinschränkungen möglicherweise eine Konvertierung der Zellwerte notwendig sein. Ein Beispiel ist die Umwandlung eines Fließkommarasters durch eine Rundungsregel in ein Integer-Raster (Ganzzahl) mit entsprechenden Veränderungen der Zellwerte. Da häufig Ganzzahlenzellwerte von Programmsystemen umfangreicher verarbeitet werden können (z.B. Komprimierung), bietet sich bei DGM die Umwandlung von gebrochenen Meterwerten in ganzzahlige Werte des Zentimeter- oder Millimeterbereiches an.

Nach erfolgreicher Rastererstellung bzw. bereits bei diesem Prozess kann die Anwendung einer Rasterkompression die Datenmenge beträchtlich verringern. In Abhängigkeit vom Verwendungszweck des Rasters (geforderte Verlustfreiheit der Kompression), vom benötigten Datenaustauschformat und vom verwendeten Rasterzelltyp sind verschiedenste Kompressionsmethoden anwendbar. Die häufigsten verwendeten Verfahren werden im Abschnitt 7.1 mit einem praktischen Beispiel vorgestellt.

5 Rastermanagement

Nachfolgend sollen wesentliche Datenbanksysteme mit und ohne explizite Rasterdatenunterstützung vergleichend charakterisiert werden, um den derzeitigen Stand der Technik wiederzugeben. Dazu wurden Unterscheidungen nach dem Datenablagetyp (BLOB, Array, ADT) und nach der Umgebung bzw. Herkunft des Systems getroffen. Es werden die wesentlichen Projektbearbeitungsschritte zur Rasterdaten-managementerstellung aufgezeigt und kurz beschrieben. Allgemeinere Aussagen, beispielsweise zur Datenbereitstellung, gelten dabei meist für alle Systeme. Der Datenbankeinrichtung und Konfiguration sowie der Überprüfung und Optimierung der Einstellungen und Möglichkeiten von Hardware und Betriebssystem folgt die Erstellung einer Datenrelation, die eine Spalte zur Rasterablage enthält. Nach der Parametrisierung dieser Ablageform und der Bereitstellung der Quelldaten wird der Dateneinleseprozess gestartet sowie die Nachführung der für den optimierten Betrieb der Verwaltung notwendigen Statistiken durchgeführt. Unter datenbankbasiertem Rastermanagement wird hier die Gesamtheit aller Verfahren und Funktionen zur Rasterdatensatzerstellung und Ablage in einer Datenbank, zum Datenzugriff und Datenaustausch sowie zu Wartungs-, Konfigurations- und Optimierungseinstellungen verstanden. Auf die systemeigenen Besonderheiten der Verwaltungen wird im Überblick eingegangen.

5.1 Rasteroptimierte Datenbank-Middleware mit RasDaMan

Der parallelisierte RasDaMan-Server, entwickelt seit 1995 durch die rasdaman GmbH sowie Prof. Baumann Bremen/München, ermöglicht durch eine Erweiterung der Anfragesprache SQL den Zugriff auf multidimensionale Rasterdaten beliebiger Größe [Bau04]. Als Middleware aufgebaut, sind die klassischen Datenbankkomponenten wie Client-Server-Kommunikation, Parser, Optimierer, Datenverzeichnis sowie Transaktions- und Indexmanager enthalten. Über eine Adapterschicht können verschiedene Datenbanksysteme (z.B. Oracle, IBM DB2) als Speicherverwalter und Basis eingebunden werden. Zentrales Entwurfsprinzip war dabei die strikte Trennung von logischer zu physischer Ebene. Datenformate, Reihenfolge der Abarbeitung und Speicherverwaltung erlauben durch die Arbeit im Hintergrund umfangreiche interne Optimierungen. Mit RasDL wurde eine Datendefinitionssprache implementiert, die einen Konstruktor (Template) 'marray' bereitstellt, der mit einer beliebigen Dimensionalität, sehr vielen verschiedenen Zelltypen sowie festen oder variablen räumlichen Grenzen instanziert werden kann. Damit sind sowohl einzelne Rasterobjekte mit Array-Aufbau als auch Kollektionen (Tabellen) aus Verweisen (OID) auf solche

Rasterobjekte möglich. Rasterdaten werden definiert als Menge von Werten, die in einem Raumwürfel auf gleichabständigen Gitterpunkten sitzen [Bau04]. Als Anfragesprache dient das um rasterwertige Primitive und Ausdrücke erweiterte SQL-92. Bereitgestellt werden für die Rasterobjektverarbeitung die bekannten INSERT/UPDATE und DELETE-Kommandos, sowie u.a. auch eine Möglichkeit zur Datenextraktion (Subset). Für jeden Zelltyp wurden induzierte Operationen und Condenser-Operationen entwickelt. Erstere werden simultan auf alle Zellen eines Rasterobjekts angewandt und können unär (z.B. Zugriff auf Farbkanal) oder binär (z.B. Maskierung) sein. Im SELECT-Teil einer Anfrage können sowohl boolesche Operationen als auch Arithmetik und Typumwandlungen verwendet werden. Das Ergebnis einer WHERE-Klausel hingegen muss vom Typ Boolean sein. Condenser-Operationen berechnen summarisch Informationen über ein bestimmtes Gebiet (Rasterobjekt oder Auswahl) und differenzieren sich in Aggregatoren (z.B. Summe, Min, Max) und Quantoren (z.B. Anzahl). Alle Operationen sowie auch die Konzepte für Speicherabbildung und Optimierung lassen sich auf die Basisfunktionen des Array-Konstruktors zur Rastererstellung anhand von vorgegebenen Eigenschaften und die Condenser-Operation (Berechnung eines skalaren Wertes aus Rasterobjekt) zurückführen.

Ein wesentliches Entwicklungsmerkmal von RasDaMan ist die Datenunabhängigkeit. Alle Operationen laufen unabhängig von Datenformat ab und liefern ihre Ergebnisse im Hauptspeicherformat des Clients und damit in Abhängigkeit von der verwendeten Programmierschnittstelle (z.B. C/C++) ab. Alternativ kann ein bestimmtes für die Ausgabe von Daten der gewünschten Dimensionalität geeignetes Format (z.B. Bildformat für 2D) gewählt werden und ist im Server durch bereitgestellte Codierungsfunktionen erzeugbar [Bau04].

Zur Speicherung der Rasterobjekte wird eine Partitionierung mit sequenzunabhängiger Verwaltung der Partitionen und Sequentialisierung innerhalb jeder Partition verwendet. Dieses Verfahren entspricht einer Kachelung, die durch multidimensionale Indexierung kaum ortsabhängige Zugriffskosten hat. Im Massenzugriff sind die Zugriffskosten relativ niedrig. Alternativen zu dieser Strukturierung wären die Verwendung von BLOB (sequentielle, koordinatenfreie Byteketten) je Rasterobjekt mit dem Nachteil der stark orts- und dimensionsabhängigen Zugriffskosten. Wird die Datenablage anders als der spätere Zugriff durchgeführt, so benötigt die BLOB-Variante als nur für einheitlich verteilte dichte Daten ohne Lücke (Speicherplatzausnutzung) geeignete Methode sehr große Zugriffzeiten. Bei der sequenzunabhängigen Ablage von Daten mit

Materialisierung der Koordinaten (Abspeicherung), geeignet nur für dünn besiedelte Räume, werden neben hohen Zugriffszeiten auch größere Speichermengen benötigt.

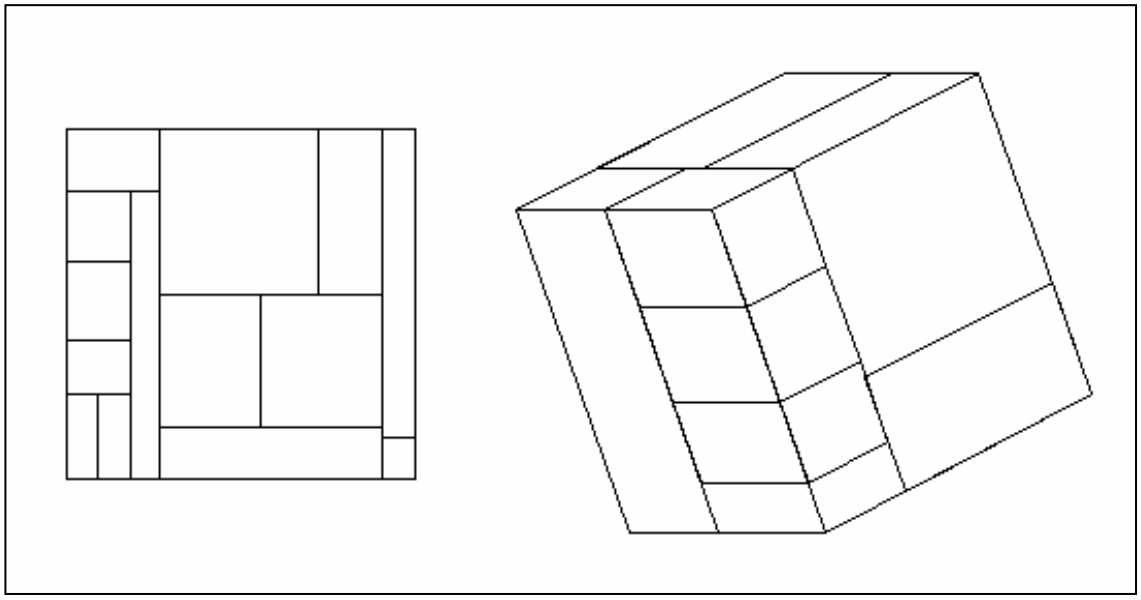


Abb. 5: 3D-Kachelung innerhalb Rasdaman, Quelle: [Bau04]

RasDaMan benutzt die Kachelung, um durch individuelle Auswahl geeigneter Speicherstrukturen eine gute Optimierbarkeit zu erreichen [Bau98]. Die Unterteilung der Rasterobjekte kann dabei, unterstützt durch entsprechende Algorithmen, beliebig sein. Mehrere Strategien sind implementiert worden. Eine Möglichkeit ist die Vorgabe der späteren Anfragemuster an den Zerlegungsalgorithmus, der die Partitionierung durch kleinste Zerlegung und Kachelminimierung optimal an diese Muster anpasst [Bau04]. Die Kacheln bilden die Einheit sowohl für Speicherzugriffe als auch für die Kompression und werden durch einen R-Baum für schnellen Zugriff indiziert. Die Anfragebearbeitung ist ebenfalls kachelbasiert. Optimierungs- und Kanonisierungsregeln sowie Mechanismen wie parallele Anfragebearbeitung und mehrdimensionale Kompression bilden die Erweiterungen durch den RasDaMan-Server. Derzeit werden Konzepte zur Anbindung von Tertiärspeichersystemen und zum durch einen Datenbankcache optimierten Zugriff auf diese Daten sowie spezielle Cachefüllungsoperationen entworfen. Ein Web-Frontend mit WMS/WCS-Schnittstelle zum Datenbankserver komplettiert die Entwicklung um den Rasdaman-Server [Bau04].
Stand der Technik sind derzeit neben den auf die Verarbeitung von Arrays optimierten Sprachen AQL [Lipkin 1996], AML [Salem 1999] und RAM [Ballegooij 2003] auch die nachfolgend beschriebenen kommerziellen Lösungen von ESRI und Oracle [Bau04] sowie die für das Projekt genutzte freie PostgreSQL/PostGIS-Umgebung.

5.2 Rasterfunktionen eines GIS-Softwaresystems (ESRI)

5.2.1 Möglichkeiten der Datenhaltung

Die Firma ESRI bietet die Verarbeitung von Rasterdaten in ihrem derzeitigen Hauptprodukt, der ArcGIS-Reihe (V9.1), sowohl dateibasiert mit Binärcodierung in einigen Formaten (z.B. TIFF) als auch mit Datenbankunterstützung an.

Für die datenbankgestützte Rasterdatenverwaltung hat ESRI die Speicherung als Raster Data Set (RDS) oder als Raster Catalog (RCat) vorgesehen. Ein RCat kann gegenüber einem RDS (einzelnes Raster) mehrere Raster überlappend oder blattschnittfrei, erweitert durch zusätzliche Attributinformationen verwalten. Die Speicherung dieser Datentypen ist innerhalb einer Personal-Geodatabase oder mit Hilfe der ArcSDE-Architektur in einem DBMS möglich. Ein Raster Dataset (RDS) besteht aus einem einzelnen Rasterdatensatz, der durch eine Mosaik-Funktion aus verschiedenen Eingabequellen errechnet werden kann [ESRI1].

Für die Speicherung von kleineren Rasterdatensätzen kann ein Datenbankmanagement durch die *ESRI-Personal-Geodatabase* genutzt werden. Das Größenlimit liegt durch die Nutzung der Microsoft Access Jet Engine derzeit bei 2 GB. Möglich ist eine durch das System gemanagte Verwaltung (nur RCat) oder auch eine ungemanagte Verwaltung (RDS und RCat). Im ersten Fall werden die Rasterdaten in das IMAGINE-Format in einen Ordner relativ zur Datenbank konvertiert sowie Metainformationen über die Datensätze innerhalb der Datenbank abgelegt. Durch entsprechende Operationen können Veränderungen und Löschungen an den Daten vorgenommen werden. Bei der 'ungemanagten' Variante werden unter Beibehaltung der bisherigen Formatierung der Datensätze lediglich Pointer in der Datenbank abgelegt, über die nur lesender Zugriff auf die Daten möglich ist. Bei Rasterkatalogen können neben der Rasterspalte auch eine Geometriespalte zur Ablage eines umschließenden Rechtecks des Datensatzes abgelegt werden. Diese Spalte dient der Optimierung von Anfragen durch Implementierung einer Indizierung [ESRI2]. Dieser Mechanismus wird im Projekt mit Hilfe der PostGIS-Erweiterung der Datenbank ebenfalls genutzt.

Für größere komplexere Projekte kann ein Rasterdatenbankmanagement über die Middleware *ArcSDE* aufgebaut werden. Sie arbeitet mit den DBMS IBM DB2, IBM Informix, Microsoft SQL-Server und mit Oracle zusammen. Auch hier können Raster Data Sets (RDS) und Raster Kataloge (RCat) zur Rasterverwaltung und Speicherung ohne begrenzendes Speicherlimit eingesetzt werden. Durch die Nutzung eines DBMS als Basis sind deren Funktionalitäten z.T. für das Rastermanagement nutzbar.

5.2.2 Workflow

Der Basis-Workflow des Aufbaus einer Rasterdatenbank über ArcSDE sieht wie folgt aus:

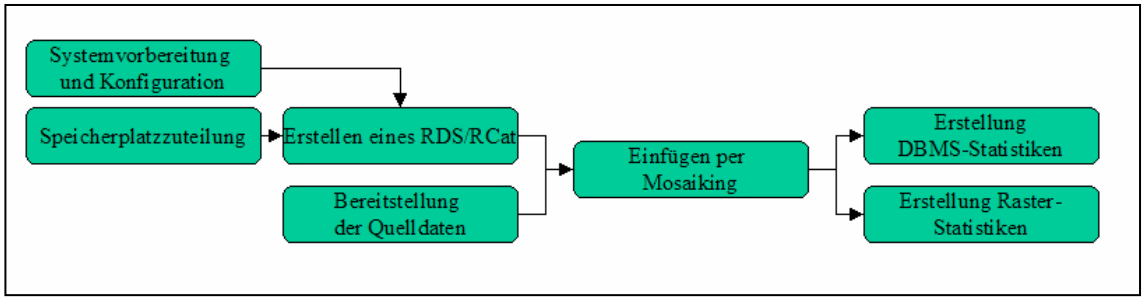


Abb. 6: Workflow ESRI

Als ersten sehr wichtigen Schritt sollte das System auf den Einsatz als Rasterdatenserver vorbereitet werden (*Systemvorbereitung und Konfiguration*). Entscheidend sind dabei die Betriebssysteme von Server und den Clients, die ArcSDE-Server- und Verbindungsarchitektur sowie das zugrundeliegende DBMS. Ebenso muss das der Datenübertragung dienende Netzwerk einen Ausbaugrad erreichen, durch den der Datentransfer bei der Erstellung ohne Beeinflussung von anderen Nutzern durchgeführt werden kann. Alternativ können zur Anlage des Rasterservers auch Zeiten geringerer Netzwerkbelastung genutzt werden. Während der ersten Füllung des DBMS wird die zugrundeliegende Datenbank bezüglich ihrer LOG- und Wiederherstellungsaufzeichnungen umkonfiguriert. Im Allgemeinen werden alle Daten eines Transfers auch in einer LOG-Datei aufgezeichnet. Tritt während der Operation ein Fehler auf oder wird eine Operation rückgängig gemacht, so nutzt das DBMS diese Informationen zur Wiederherstellung eines bestimmten Zustandes der Datenbank. Da bei der Anlage eines umfangreichen Rasterdatensatzes auch in die Log-Datei sehr umfangreiche Daten geschrieben werden würden, die durch das Wiederherstellungssystem der Datenbank auch archiviert werden würden, sollte dieser Mechanismus bei der Rasterdaten-Ladeprozedur angeschaltet werden [ESRI2]. Dies geschieht beispielsweise bei SQL-Server (als DBMS-Basis) durch Veränderung des DB-recovery-models auf 'simple' und bei Oracle durch Veränderung des 'LOG_Checkpoint_Intervalls'. Das Loggen großer Rasterdaten belastet die Performance eines DBMS so stark, das im Allgemeinen auf die Fehlertoleranz in dieser Zeit verzichtet wird und stattdessen bei einer Fehlfunktion der Ladevorgang wiederholt wird. Sicherungs- und Wartungsprozesse sind daher während der Erstellung auszuschalten.

Zur Abfrageabarbeitung eines DBMS sind Informationen über die abzufragenden Daten notwendig. Der kostenbasierte Anfrageoptimierer selektiert den besten

Ausführungsplan durch Nutzung der statistischen Informationen, gewonnen aus den Daten (siehe Abschnitt 6.1). Die Sammlung von statistischen Informationen durch das DBMS sollte daher nicht unterbrochen werden [ESRI2]. Zur optimalen Nutzung der Netzwerkkapazität durch das DBMS wird die Datenblockpuffergröße durch entsprechende Konfiguration maximiert. Auch die Datenblockgröße (kleinste Speichereinheit eines DBMS) sollte möglichst den Eigenschaften der Rasterdaten angepasst werden (Blockgröße). Bei Oracle ist diese Angabe bereits bei der Erstellung des Tablespace (Speicherort von Relationen einer Datenbank) zu maximieren. ArcSDE übernimmt die Steuerung des Datenflusses von einem Client zur Datenbank. Dazu werden Transportpuffer benötigt. Der Client schreibt seine Daten in diesen Puffer und überträgt diesen an den Server. Die Puffergröße, einstellbar durch 'RASTERBUFSIZE', sollte vom Standardwert 100KB auf 10MB während des Ladevorgangs erhöht werden [ESRI2]. Da für die Einrichtung dieser Transportpuffer sowohl auf dem Client als auch auf dem Server Hauptspeicher der Systeme verwendet wird, ist ein entsprechender erhöhter Ausbaugrad dieser Hauptspeicher sicherzustellen. Die zur Darstellungsoptimierung verwendeten auflösungsreduzierten Pyramid-Layer (siehe Kapitel 3.2.1) werden beim Einlesen der Raster in die Datenbank bereits angelegt. Da dieser Prozess sehr viel Prozessorzeit benötigt, sollte er von dem System mit dem schnellsten Hauptprozessor (CPU) durchgeführt werden. Je nach Verteilung der CPU-Fähigkeiten können daher für die Verbindung von Client und Server unterschiedliche Möglichkeiten gewählt werden. Wird eine direkte Verbindung aufgebaut, so rechnet der Client. Beim Laden vom Server aus sowie beim Ausführen einer Mosaikprozedur wird hingegen viel Rechenzeit beim Server benötigt [ESRI2].

Zum Erstellen eines Rasterdatensatzes in einer Datenbank ist neben dem Speicherplatz für den Datensatz selbst auch Datenbankspeicher für die Anpassung der Datenverzeichnisse sowie Temporärspeicher für Sortierungen notwendig (*Speicherplatzbereitstellung*). Die Bereitstellung von ausreichendem Speicherplatz vor dem Einlesen der Rasterdaten in die Datenbank verhindert nachfolgende Probleme, die bei Betriebssystem, Hardware oder Datenbank auftreten können. Ebenso wird die zeit- und ressourcenaufwendige dynamische Speicherplatzvergrößerung, sofern überhaupt ohne Verlust der logischen Struktur möglich, vermieden. Zur Bereitstellung ist eine Abschätzung oder Berechnung des benötigten Platzes notwendig. Dazu können zum Einen einige Beispieldatensätze in die Datenbank gelesen werden, um dann durch Extrapolation den Gesamtspeicherbedarfbedarf zu ermitteln. Diese Beispieldatensätze werden mit den gleichen Parametern (Kompressionstyp, Kompressionsqualität,

Pixeltiefe, Bandanzahl, Partitions-Größe, DBTUNE-Parameter) eingelesen. Die Effizienz der Ablage eines Rasterdatenblockes wird durch den bei Oracle 11 Parameter umfassenden DBTUNE-Konfigurationsabschnitt festgelegt. Bei SQL-Server sind 22 Parameter einzustellen, die u.a. den Indexfüllfaktor der beteiligten Tabellen beeinflussen. Sind keine Änderungen nach Anlage des Rasterdatensatzes mehr geplant, so kann der Füllgrad 100% betragen, alternativ sind niedrigere Werte zu wählen [ESRI2]. Zur Ermittlung des genutzten Speicherplatzes der eingelesenen Beispieldaten wird die Rasterspaltenidentifikationsnummer mit einem SQL-Befehl ermittelt. Es folgt das Update der DBMS-Statistik für die Raster-Block-Tabelle, sowie die Abfrage der Blockanzahl. Die Multiplikation dieser Anzahl mit der bekannten Datenblockgröße (Tablespace-Parameter) sowie die verhältnisgerechte Hochrechnung auf den vollständigen einzulesenden Rasterdatensatz ergibt den Gesamtspeicherplatzbedarf innerhalb der Datenbank. Liegen noch keine Beispieldatensätze vor, so kann auch über eine Berechnungsformel der Speicherbedarf etwas ungenauer abgeschätzt werden [ESR2]:

$$Speicherbedarf = \sum_{Raster1}^{RasterN} (Anzahl\_Pixel * Byte\_je\_Pixel * Bandanzahl * PyrLay * Kompression * DBbed)$$

Anzahl_Pixel:	z.B. Fläche * Auflösung (Pixel je Fläche)
Byte_je_Pixel:	Umrechnung der Bittiefe eines Zellwertes in Byte, 8 Bit = 1 Byte
Bandanzahl:	bei DGM häufig 32-Bit-Fließkommazahlen (4 Byte) in einem Band = 1
PyrLay:	Speicherplatzbedarfsfaktor für Pyramidenlayer = 1.33
Kompression:	Faktor abhängig von Kompressionsart (z.B. LZ77 = 0.5)
DBbed:	Speicherplatzbedarf durch Datenbankkonfiguration (Tilesize) und Attributtabellen, vereinfacht 10% = 0.1

Beispiel:
60 DGM-Raster der Größe 500x500 Pixel (32Bit-Float), LZ77-Kompression benötigen:
60*250000*4*1*1.33*0.5*0.1 = 3.8MB

Die zur Erstellung der Datenbank benötigten Quelldaten (*Quelldatenbereitstellung*) sollten während des Einleseprozesses auf einem sehr schnellen Datenträger für die Datenbank verfügbar sein. Bedeutsam ist dabei die für die Schreib- und Lesezugriffe benötigte Zeit des Quellmediums bzw. die erzeugte Netzwerklast [ESRI2].
Zur *Erstellung des Rasterdatensatzes* innerhalb der Datenbank sind einige Parameter erforderlich, die durch Planung und Vorüberlegungen ausreichend an die speziellen Gegebenheiten angepasst werden sollten. Eine nachträgliche Änderung ist zumeist nicht

möglich. Unter Angabe von Datensatzname, Zellgröße, Pixeltype, Koordinatensystem, Bandanzahl, Anzahl der zu erstellenden Pyramidenlayer sowie deren Resampling-Methode und eines Referenzpunktes, der Größe der Datenblöcke ('TILESIZE'), Kompressionsmethode und Qualität sowie der Angabe eines Datenbankoptimierungsprofils ('DBTUNE') wird dann das RDS mittels Script oder Benutzeroberfläche erzeugt. Die Kompressionsmethode wird bestimmt durch die Art der späteren Verwendung der Rasterdatenbank sowie deren Pixeltiefe (Bittiefe). Sollen lediglich Hintergrundinformationen beispielsweise zur Kartenherstellung abgelegt werden, so kann eine verlustbehaftete Kompression (JPEG2000) mit geringerer Qualität gewählt werden. Sind hingegen zukünftige Analysen und Berechnungen vorgesehen, so wird die verlustfreie LZ77-Methode zu wählen sein. Die Erzeugung von Pyramidenlayern dient der Optimierung der Performance bei der graphischen Darstellung des Rasterdatensatzes. In Abhängigkeit von der vom Client angeforderten Auflösung werden reduzierte Datenteile zurückgeliefert. Die Resampling-Methode bei der Erstellung der Pyramidenlayer sollte durch eine vergleichende Probe ermittelt werden [ESRI2]. Die bei ESRI als Standardwert verwendete 'TILESIZE' (128x128) ist für die Verwendung mit RGB-Rastern optimiert. Bei Nutzung von 1-Band-Graustufen-Rastern (DGM) sollte diese Angabe so verändert werden, dass eine möglichst gute Ausnutzung der Datenbankblöcke durch die Rasterdatenblöcke ('TILESIZE') erreicht wird. Je größer dieser Wert gewählt wird, umso effizienter können Kompressionsalgorithmen arbeiten.

Das *Einlesen der Quellraster* mittels Programmoberfläche oder Script kann für Raster Data Sets lediglich sequentiell durchgeführt werden, da alle Operationen sich auf ein zukünftiges Rastermosaik beziehen. Bei der Verwendung von Rasterkatalogen, die die Quellraster lediglich strukturiert in der Datenbank ablegen ohne sie zu mosaikieren, sind auch parallele Zugriffe, bestenfalls von unterschiedlichen physikalischen Clientsystemen mit dortiger Verarbeitung, möglich. Werden Pyramidenlayer erzeugt, so kann die Angabe eines optimierten Pyramidenreferenzpunktes deutlich zur Performancesteigerung beitragen [ESRI2]. Werden Daten in das RDS eingelesen, die links oder oberhalb des Pyramidenreferenzpunktes liegen, so ist das System zur vollständigen Neuberechnung aller Pyramidenlayer gezwungen. Dieser zeitaufwendige 'Shifting'-Prozess kann durch Wahl eines entsprechenden Pyramidenreferenzpunktes vermieden werden. Bei Einfügung von Daten rechts unterhalb des Punktes werden lediglich die betroffenen Teile der Pyramidenlayer neu errechnet [ESRI2].

Zum Finden des besten Ausführungsplanes bei Abfragen nutzen DBMS im Algemeinen eine kostenbasierte Optimierungsmethode, die auf einer Statistik der Datenbankobjekte beruht und nach Änderungen überarbeitet werden muss. (*Überarbeiten der Datenbankstatistiken*). Ist keine Statistik verfügbar, so wird eine regelbasierte Abarbeitung ohne vorausschätzbare Ergebnisse durchgeführt. Bei der Anlage von RDS werden Schreib- und Leseoperationen auf eine Tabelle beispielsweise zur Pyramidlayererstellung benötigt. Zum optimierten Lesen und Schreiben von Datenblöcken ist daher ein zusammengesetzter Index auf diese Tabelle zu empfehlen, der jeweils nach etwa 10 Einfügungen nachzuführen ist [ESRI2]. Ist so ein Index vorhanden, so kann statt eines aufwendigen Full-Scans der Tabelle (alle Datensätze einlesen und durchsuchen) ein partieller Index-Scan durchgeführt werden. Die Aktualisierung der Statistiken kann durch native SQL-Kommandos, spezialisierte ArcSDE- bzw. Geodatabase-Befehle oder allgemein bei DBMS durch eine 'Analyze'-Wartungsoperation erreicht werden. Da diese Reindexierung teilweise sehr zeit- und ressourcenaufwendig ist, kann beispielsweise bei Oracle eine Statistik mit einer auf 1% der Anzahl aller Datensätze verringerten Stichprobe mit Einschränkungen der Statistikgenauigkeit durchgeführt werden [ESRI2].

Im Gegensatz zur Datenbankstatistik beruht die *Rasterstatistik* auf einer Ermittlung der Eigenschaften zu einem einzelnen Objekt, abgelegt bei ESRI in einer Zusatztabelle bzw. Datei (AUX, unmanaged-Variante). Dort werden Minimum, Maximum, Mittelwert und Standardabweichung der Zellwerte gespeichert. Da diese Statistiken zur Optimierung der Datendarstellung (z.B. Kontrastverbesserung) genutzt werden, sollte bei farbkorrigierten Rastern die Statistikverwendung im Anzeigeclient (z.B. ArcGIS) abgeschaltet werden [ESRI2]. Die mögliche Berechnung der Statistiken auf einem höheren Pyramidenlevel kann bei interpolierenden Resampling-Methoden (bilinear, bikubisch) zu einigen Zeitgewinnen bei der Erstellung führen.

5.3 DBMS mit Rasterdatenfunktionalität am Beispiel Oracle Spatial

5.3.1 Datenbankmodell

Seit der Version 10 bietet Oracle in seinem gleichnamigen Datenbank-Managementsystem als Teil der als Erweiterung verfügbaren 'Spatial Option' die Verarbeitung von Rastern unter dem Oberbegriff GeoRaster an. GeoRaster erlaubt neben Im- und Export von Rasterdaten durch verschiedenste Schnittstellen auch deren Indexierung sowie Anfrage und Analyse. Dazu werden neben Werkzeugen und Programmen auch Erweiterungen der SQL-Syntax bereitgestellt [Brink06].

Zur Ablage eines Rasters ist ein Objekt der Oracle-Klasse SDO_GEORASTER erforderlich. Die Struktur dieser Klasse, bestehend aus Rastertyp, SpatialExtent, Rasterdatatable, RasterID und Metadata sei nachfolgend wiedergegeben:

Der Rastertyp wird durch eine Nummer des Formates '[d][b][t]01' mit der Anzahl der Dimensionen d (derzeit nur d=2 unterstützt), einem Attribut zur benötigten Ebenenmenge (b=0 eine Ebene, b=1 mehrere Ebenen), sowie dem derzeit noch nicht verwendeten Parameter t (zeitliche Dimension) und dem Wert 01 für Erweiterungszwecke festgelegt. In Form eines SDO_Geometry-Polygon-Objektes wird die räumliche Ausdehnung (SpatialExtent) des Rasterobjektes hier geometrisch in Weltkoordinaten eines definierten Systems abgelegt. Als Text wird im Attribut Rasterdatatable der Name der Rasterdatentabelle abgelegt. RasterID dient als Fremdschlüssel zur eindeutigen Identifikation eines GeoRaster-Objektes. Die Metadata-XML-Spalte enthält ergänzende Informationen zur Beschreibung der Ablage des Rasters. Beispiele sind Interleaving- oder Blocking-Parameter [Kot04].

Die Daten eines oder mehrer GeoRaster werden innerhalb einer Rasterdatentabelle in Form von Binary Large Objects (BLOB) abgelegt. Zur Speicherung eines Rasters in der Rasterdatentabelle (Typ SDO_RASTER) können ein oder mehrere BLOB (Blocks) in Abhängigkeit von Objektgröße, Bildpyramiden und Blocking-Parametern genutzt werden.

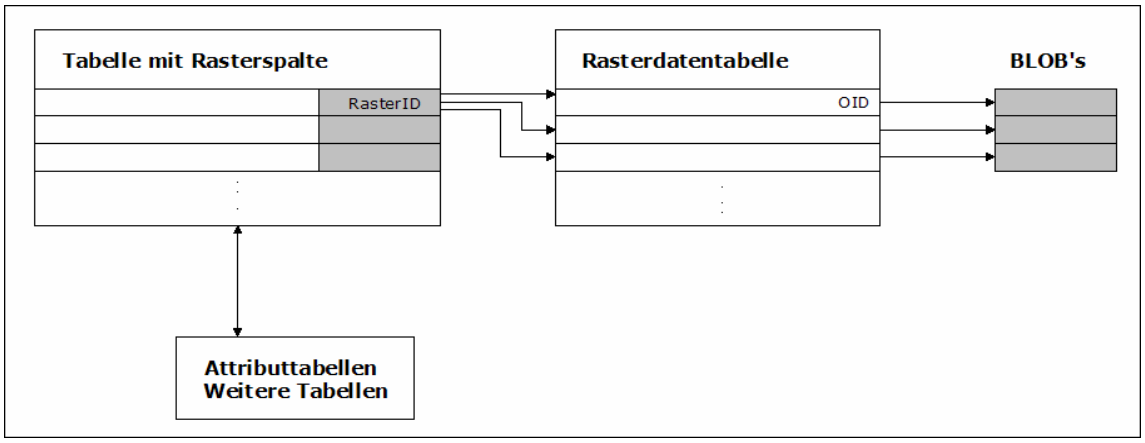


Abb. 7: Struktur der Rasterspeicherung in Oracle, Quelle: [Kot06]

Objekttabellen in Oracle besitzen neben beschreibenden Attributen eine Spalte mit vom System vergebenen und verwalteten eindeutigen Object Identifier (OID), hier zur Aufnahme der Referenz auf ein BLOB, über die der Zugriff auf die Rasterdaten erfolgt. Die Klasse SDO_RASTER besitzt nachfolgenden Aufbau:

Der RasterID-Schlüssel dient der Verbindung der Blockteile zu einem GeoRaster-Objekt. Die PyramidLevel-Nummer regelt die Zuordnung des Blocks zu einer

Pyramidenebene (Index 0 steht für die Originaldaten). Die in BandBlockNumber abgelegte physische Bandnummer entspricht der um 1 verminderten Ebenennummer. RowBlockNumber und ColumnBlockNumber indizieren die Blockkacheln beginnend vom Rasterursprung rechts oben (0,0). Als Punkt-Vektorobjekt (SDO_Geometry) wird der Datenraum des Blocks in Pixelkoordinaten im Attribut BlockMBR beschrieben. Die OID-Spalte (RasterBlock) referenziert zum BLOB des Rasterblocks [Kot04].

5.3.2 Workflow in Oracle Spatial

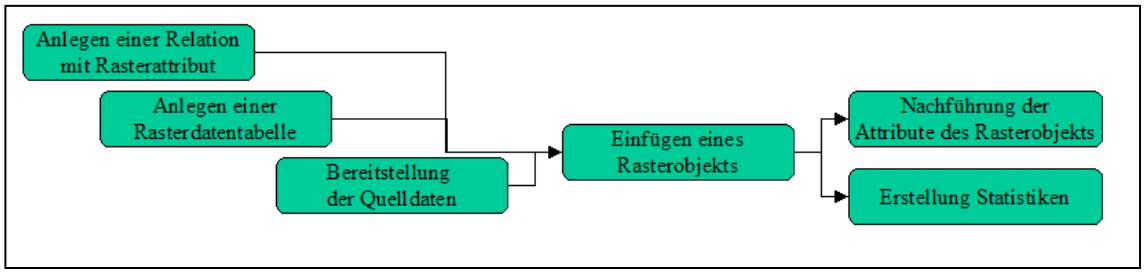


Abb. 8: Workflow Oracle

Erster Arbeitsschritt ist das *Anlegen der Rasterdatentabelle.*

```
CREATE TABLE DGMRaster (
        id              DECIMAL(8)              -- ID
        rastername      VARCHAR(32)             -- ggf. mehrere beschreibende Attribute
        raster          SDO_GEORASTER,
        CONSTRAINT ps PRIMARY KEY (id) );       -- Primärschlüssel der Tabelle
```

Zur Sicherung der referentiellen Integrität der GeoRaster-Struktur wird von Oracle die Anlage eines Triggers für jede Georasterspalte einer Tabelle empfohlen [Ora1]:

```
EXECUTE SDO_GEOR_UTL.createDMLTrigger('DGMRaster', 'raster');
```

Danach kann die Rasterdatentabelle eindeutigen Namens als Objekttabelle mit einem zusammengesetzten Primärschlüssel sowie Parametern zur BLOB-Ablage angelegt werden:

```
CREATE TABLE DGMBlock OF SDO_RASTER (
        CONSTRAINT ps_blck PRIMARY KEY (rasterId, pyramidLevel, bandBlockNumber,
        rowBlockNumber, columnBlockNumber))
        LOB (rasterblock) STORE AS (NOCACHE, NOLOGGING);
```

Die LOB-Parameter verhindern das ressourcenaufwendige Zwischenspeichern und Mitschreiben (Logging). Weitere Parameter wie Tabellenpartitionierung und Tablespace sowie ein großer CHUNK-Wert (Datenvolumen je Schreib/Lesezugriff) sind Ansätze für die Datenbankoptimierung [Ora3].
Dem *Laden eines GeoRasters* wird die Initialisierung durch SDO_GEOR.init vorangestellt:

```
INSERT INTO DGMRaster VALUES (1, 'DGM01', SDO_GEOR.init('DGMBlock', 1));
```

Danach bietet Oracle verschiedene Möglichkeiten des Imports von Rasterdaten in das angelegte Objekt an. Die SDO_GEOR.importFrom-Prozedur setzt Leserechte für die zu ladende Datei voraus. Vor dem Laden sollten die nachfolgenden Parameter des 'JavaPoolSize' angepasst werden. Ist die automatische Speicherverwaltung angeschaltet, so enthalten die Parameter Minimalwerte, alternativ Maximalwerte:

```
ALTER SYSTEM SET SGA_TARGET=248M;       --automatische Verwaltung 248MB ein
ALTER SYSTEM SET DB_CACHE_SIZE=8M;
ALTER SYSTEM SET SHARED_POOL_SIZE=50M;
ALTERSYSTEM SET LARGE_POOL_SIZE=0;
ALTER SYSTEM SET JAVA_POOL_SIZE=80M;
```

Für die Beschreibung der Parameter sei auf die Oracle-Dokumentation [Ora3] verwiesen. Nach Restart der Datenbank sind die Einstellungen aktiv.
Beispielhaftes *Einlesen eines Rasterdatensatzes:*

```
BEGIN
 SELECT  raster INTO r1 FROM DGMRaster WHERE id=1 FOR UPDATE;
 SDO_GEOR.importFrom(r1,NULL, 'TIFF, 'file', 'dgm1.tif');
 UPDATE DGMRaster SET raster = r1 WHERE id=1;
 COMMIT;
END;
```

Einfügeparameter sind neben dem GeoRaster-Objekt auch Speicherparameter (hier NULL), das Rasterquellformat (hier 'TIFF'), der Typ ('file') und Name sowie, wenn vorhanden, Format (z.B. 'worldfile'), Typ (z.B. 'file') und Name einer Header-Datei zur Beschreibung der räumlichen Ausdehnung des Rasters. Die Speicherparameter werden als String, bestehend aus Key-Value-Paaren, angegeben.

Parameter (Key)	gültige Werte (Value)	Beschreibung
blocking	TRUE/FALSE	ggf. Übernahme der Parameter der Eingabedaten
blockSize	(row, column, [band])	definiert die Blockgröße in 3 Dimensionen
cellDepth	[1-64]BIT, _u,_s,_real	Bitlänge und Information über Datentyp (z.B. real)
compression	JPEG-B/F,DEFLATE, NONE	siehe Abschnitt Kompression
interleaving	BSQ, BIP, BIL	siehe Abschnitt Interleaving
pyramid	TRUE/FALSE	Erstellung mit SDO_GEOR.generatePyramid
quality	0-100	100 für beste JPEG-Qualität

Tab. 6: Oracle Speicherparameter

Das Zellformat (cellDepth) kann Formate von 1BIT bis 64BIT_REAL annehmen. Die verlustbehaftete JPEG-Kompression wird für 8Bit-unsigned-Raster sowie 1, 3 oder 4 Bänder und eine maximale Blockgröße von 50MB unterstützt. Bei JPEG-B (abbreviated baseline Format) sind im Gegensatz zu JPEG-F (full-format baseline JPEG) die Tabellen zur Quantifizierung und zur Huffmann-Codierung nicht in den komprimierten Daten enthalten. Die verlustlose DEFLATE-Kompression verarbeitet alle Zelltypen sowie eine maximale Blockgröße von 1GB und ist für digitale Geländemodelle zu empfehlen (siehe auch Kapitel 7.1 Kompression). Sind mehrere Rasterebenen vorhanden, so kann mit dem Interleaving-Parameter die Sortierung der Informationen der Ebenen gesteuert werden. Je nach Dateninhalt und späterer Verwendung können durch geschickte Parameterwahl Optimierungen in der Zugriffszeit bei Darstellungen und Anfragen erzielt werden.

Andere Möglichkeiten der Rasterdateneingabe sind die Nutzung des Java-Werkzeugs 'GeoRasterTool' sowie die Java-Anwendung 'GeoRasterLoader', die durch spezifische Programmschnittstellen TIFF-, JPEG-, BMP-, GIF- und PNG-Dateien, auch mit Angaben zur Geocodierung in einem Worldfile, einlesen kann [Ora3].

Zur Vervollständigung des Einlesevorganges werden die GeoRaster-Attribute SpatialExtent sowie das genutzte Koordinatensystem als EPSG-Code nachgeführt (*Nachbearbeitung*):

```
BEGIN
 SELECT  raster INTO r1 FROM DGMRaster WHERE id=1 FOR UPDATE;
 r1.spatialextent:=SDO_GEOR.generateSpatialExtent(r1);
SDO_GEOR.setModelSRID(r1,2398);
 UPDATE DGMRaster SET raster = r1 WHERE id=1;
 COMMIT;
END;
```

5.4 PostgreSQL/PostGIS ohne spezifische Rasterunterstützung

Zurückgehend auf eine Entwicklung der University of California in Berkeley ist das objektrelationale DBMS PostgreSQL bereits ein sehr fortgeschrittenes OpenSource-Projekt, welches die SQL-Standards SQL92-2003 unterstützt und geringe Systembelastung bei hoher Geschwindigkeit verspricht [PG06]. Neben den Grundfunktionalitäten wie Unterstützung komplexer Abfragen, Mengenoperationen, Views oder gespeicherten Prozeduren in verschiedenen Programmiersprachen werden auch Joins sowie ein Regelmanagement angeboten. Regeln können relations- und ereignisbezogen Abfragen modifizieren oder ergänzen und helfen, vorgegebene Abläufe und wiederkehrende Operationen zu automatisieren. Ereignisbezogene Prozeduren (Trigger) können auf Relations- oder Systemzustände reagieren. Neben Transaktionen, der Zusammenfassung von SQL-Kommandos in einer blockbasierten Abarbeitung, werden Konzepte wie das der referentiellen Integrität (Abhängigkeitskontrolle) oder des Multi Version Concurrency Control (Bearbeitung einer eigenen Datenkopie für jede Transaktion) bereitgestellt. PostgreSQL ist eine objektrelationale Datenbank, die objektorientierte Merkmale (Vererbung, Definition eigener Datentypen/ Operationen/ Funktionen) sowie sehr große Datentypen (Large Objects / Toast) unterstützt. Eine Erweiterung des PostgreSQL-Systems um die Unterstützung von räumlichen 'Simple Features' nach OGC, räumlicher Indexierung sowie gängige Vektor-GIS-Analysefunktionen bietet die ebenfalls unter OpenSource entwickelte Software PostGIS.

Diese Funktionalitäten schaffen die Grundlagen zur Verwaltung auch großer Rasterdatensätze mit PostgreSQL. Noch nicht vollständig implementierte Ansätze sind der mit Hilfe der GDAL-Bibliothek (www.gdal.org) zu füllende Raster-ADT PGCHIP. Bereits mit einfachen Datenbankmitteln kann aber unter Verwendung des Datentyps Binary Large Object (BLOB) eine ausbaubare Rasterverwaltung aufgebaut werden. Dazu wird nach dem Anlegen von Tablespaces (Speicherort) und einer Datenbank sowie deren Konfiguration nachfolgender allgemeiner Workflow umgesetzt, der auch für die Rasterdatenverwaltungserstellung des Projektes genutzt wurde:

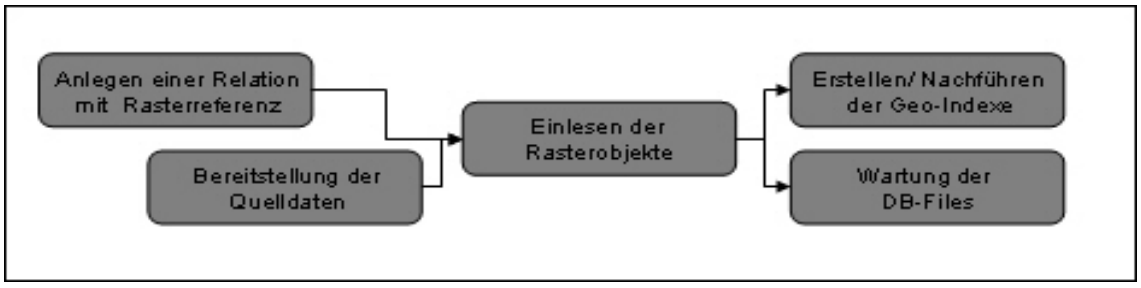


Abb. 9: Workflow PostgreSQL

Beispielhaft sollen nun die Arbeitsschritte aufgezeigt werden:

1. Erstellung der Relation

```
CREATE TABLE dgm (gid serial not null,  raster oid,  name text,  gebiet varchar(20));
ALTER TABLE dgm ADD PRIMARY KEY (gid);
SELECT AddGeometryColumn('','dgm','geom','2398','POLYGON',3);
```

Nach dem Anlegen der Datenbanktabelle und dem Einrichten eines Primärschlüssels wird eine Geometriespalte zur Aufnahme des Rasterumrumrings (Polygon) im Koordinatensystem 42/83 3° 3.Streifen (EPSG=2398) angefügt.

2. Einlesen der Quelldaten (1 Beispiel aus 25850 Kacheln)

```
INSERT INTO dgm (name, gebiet, raster, geom) VALUES ('45950_59975','Gebiet 1',
lo_import('d:/rastereingabe/45950_59975.tif'),GeomFromText('POLYGON((4595000
5997500,4595000 5998000,4595500 5998000,4595500 5997500,4595000 5997500))',2398));
```

Zum Einlesen der Quelldateien werden die Daten auf einem vom Server aus zugreifbaren Verzeichnis erwartet. Die geometrischen Umringe der Raster werden durch eine PostGIS-Funktion konstruiert.

3. Erstellen der Indexe

```
CREATE INDEX dgm_gist ON dgm USING gist (geom GIST_GEOMETRY_OPS);
CREATE INDEX dgm_btree ON dgm USING btree (name);
```

Es wird hier ein Index zur Abfrage nach den Metadaten (name) und ein Index zur räumlichen Abfrage (geom) erstellt. Eine sich nun anschließende Wartungsoperationen könnte beispielsweise zur Bereinigung verwaister Raster-OID's aus der Datenbank db_dgm dienen:

```
vacuumlo -v -U postgres -W db_dgm
```

Auf einzelne Besonderheiten des PostgreSQL/PostGIS-Systems wird durch die Verwendung im Projekt in den einzelnen Kapiteln sowie in der Projektvorstellung ebenfalls eingegangen.

5.5 Vergleich der Rastermanagementlösungen

Die spezialisiertesten Möglichkeiten für die Verwaltung sehr hochdimensionaler Raster bietet RasDaMan. Durch umfangreiche Spracherweiterungen der Datendefinitions- und Abfragesprache SQL gelingt es hier, den Array-Ansatz effizient zu nutzen. Geeignet ist die Middleware für große Multimediasammlungen sowie für mehrdimensionale Datenarchive. Beispieldaten sind medizinische bildgebende Messergebnisse (Tomographie) oder auch Klimamodelldaten (3D) erweitert um eine Zeitkomponente (4D). Vorgestellt und beschrieben wird von den Herstellern die sehr aufwendige Datenabfrage senkrecht zur Datenablagerichtung (Scanrichtung). Ergebnisse könnten dann beispielsweise frei positionierte Schnitte durch den menschlichen Körper oder orts- und zeitbezogene Analysen von Umweltdaten sein [Bau04]. Diese Lösung sollte gewählt werden, wenn auf größte Datenmengen hoher Dimensionalität in besonderer Art und Weise schnell zugegriffen werden soll.

Die Firma ESRI bietet Rasterdatenmanagementfunktionalität eingebettet in einer weitverbreiteten GIS-Umgebung und unterstützt durch eine Fülle von rasterverarbeitenden Werkzeugen mit Hilfe von GIS-Erweiterungen (z.B. Spatial Analyst). Geboten werden die Gestaltung von Anwendungsszenarien und Workflows unterstützt durch Scripting und ein hoch komplexes Objektklassenmodell, welches objektrelational erweiterbar ist (Integration VBA) und auf das aber kaum durch Standardschnittstellen zugegriffen werden kann. Spezialisierte Werkzeuge für bestimmte Anwendungsbereiche (z.B. hydrologische Modellierung) stehen einer sehr propritären Politik der Daten- und Austauschformate gegenüber. In der eigenen Arbeit mit ArcGIS 9 fielen häufige schlecht dokumentierte Programmfehler sowie Probleme bei sehr großen Datensätzen auf, die in den Folgeversionen noch zu bereinigen sind. Das Rastermanagement mit den Werkzeugen von ESRI wird für mittlere Datengrößen und wenig anspruchsvolle Analysen empfohlen. Sehr umfangreich sind die Möglichkeiten in den GIS-Spezialgebieten Visualisierung und Kartengestaltung.

Für die Verwendung von Oracle Spatials GeoRaster sprechen die klare Objektstruktur und die vielfältigen Verarbeitungsmöglichkeiten in einer mächtigen ausgereiften Datenbankumgebung. Die effektive Unterstützung sehr großer Rasterdatenmengen sowie die Anbindung von thematischen Informationen an diese Raster, der standardisierte Zugriff durch Verwendung von SQL und OGC-konformen Erweiterungen, umfangreiche Schnittstellen sowie die sehr gute Dokumentation punkten für Oracle. Zur Nutzung von Standardrasterdaten (2D) und zur Anbindung an Unternehmenssoftware kann Oracle Spatial empfohlen werden.

PostgreSQL ist die beste Wahl für Anwendungsbereiche und Nutzergruppen, die spezialisierte anpassbare Funktionalitäten und gute Erweiterbarkeit benötigen. Die kostengünstige OpenSource-Lizensierung ermöglichte vielfältige freie Erweiterungs- und Anwendungsprogramme verschiedenster Hersteller und Anwender mit kurzen Entwicklungszyklen. Basis bilden die konsequente Einhaltung von Standards, beispielsweise der OGC-konforme Funktionsumfang im Geometriebereich der Erweiterung PostGIS. Sehr gute Dokumentationen und die weite Verbreitung von Schnittstellen von und zu einer PostgreSQL-Datenbank sprechen für diese Lösung.

5.6 Standardisierung der Rasterverwaltung

In der OGC-Implementations-Spezification Grid Coverage (2001) werden grundlegende Anforderungen an eine Softwareimplementierung zur Rasterverwaltung dargelegt. Ziel dieses Dokuments ist die Förderung der Interoperabilität zwischen Datenerstellern und Softwareproduzenten. Anforderungen wie die Unterstützung sehr großer Datensätze, vieler Dimensionen sowie vielfältiger Formate von RAW-Daten bis zu thematisch klassifizierten Rastern werden neben der Unterstützung für verschiedene Farbmodelle und der Umsetzung von erweiterten Möglichkeiten der Bildverarbeitung und Analyse an eine Software definiert. Drei grundlegende Pakete werden dazu vorgeschlagen. Das Basispaket General Coverage Specification (CV) umfasst dazu die Definition eines Interfaces zum Zugriff auf einen Rasterdatensatz. Neben Informationen wie der Datenquelle, des Koordinatensystems und des Rasterrechteckumrings (MUR - minimal umschließendes Rechteck) werden auch Rasterstatistiken wie Minimal- und Maximalzellwerte, NoData-Werte und Zelleinheiten definiert sowie Funktionen zur Abfrage eines Rasterzellwertes und zum Zugriff auf die Informationen zu Farbtabelle und Pixeltiefe bereitgestellt. Das Paket Grid Coverage Specification (GC) beinhaltet die Schnittstellen zum Rasterdatenaustausch, zum Rasterformat sowie Funktionen zum Zugriff auf Rasterteile (Partitionierung in Blocks). Weiterhin wird die Ablage der Rasterteile in Geometrie und Datencodierung (GC_GridGeometry, GC_GridPacking) definiert. Funktionen, beispielsweise zur Rastererstellung (createFromName), zur Rasterausgabe in eine Datei (exportTo) oder zur Koordinatentransformation (move) sind im Paket enthalten. Das optionale Paket Grid Coverage Processing (GP) enthält Funktionen und Operationen zur Rasterverarbeitung. Beispiele sind die Ermittlung von Extremwerten im Interface GP_GridAnalysis (min/maxValue) oder zur Metadatenverarbeitung im Interface GP_GridCoverageProcessor (getMetadataValue). Weiterhin werden mögliche Rasterverarbeitungsoperationen wie Interpolationen (Resampling) oder Filterungen (z.B. Median Filter) vorgestellt. Zentrales immer zu

unterstützendes Datenaustauschformat soll das GeoTIFF-Format sein. Es wird repräsentiert durch eine WKBGeoTIFF-Definition (Well known binary GeoTIFF) und und wurde aufgrund seiner Universalität und breiten Unterstützung von Rasterkonzepten ausgewählt. Derzeit sind lediglich 4 Produkte der Firma Cadcorp Inc. bei OGC registriert, die die Grid Coverage Implementation Specification (GC 1.0) umgesetzt haben und die daraufhin erfolgreich von OGC auf Konformität geprüft wurden. [OGC7]

Setzt sich dieser Vorschlag in Zukunft durch, so wird in Kombination mit der Standardabfrage- und Datenmanipulationssprache SQL sowie den relevanten OGC-Datenaustauschmöglichkeiten (z.B. WMS) die Interoperabilität der Systeme und ihre größere Verbreitung sowie die Nutzung großvolumiger Daten sehr stark gefördert werden. Eine Unterstützung und die Implementierung der Vorschläge incl. Prüfung auf Konformität durch das OGC ist den Datenbank- und Anwendungserstellern vor diesem Hintergrund zu empfehlen.

6 Konzepte zur Rasterdatennutzung

Die Konzepte zum Zugriff und Austausch der Rasterdaten eines Verwaltungssystems bilden häufig neben den datenbanktechnischen Möglichkeiten den Grund zur Implementation eines solchen Systems. Je nach Anforderungen, Datenumfang und Nutzeranzahlen können die Ausgaben nach erfolgreichem Zugriff dienst- oder dateibasiert gestaltet werden. Auf die Möglichkeiten der Nutzung eines Webinterface für die Datenabfrage und -anzeige wird im Projektkapitel 8.4 als Praxisbezug im Zusammenhang kurz eingegangen.

6.1 Rasterdatenzugriff

Die Zugriffe auf Objekte innerhalb einer Datenbank erfolgen meist durch die standardisierte Abfragesprache SQL. Durch die einheitliche Syntax sowie die vielfältigen Möglichkeiten der Sprache ist dieser Weg jeder Rasterdatenverwaltung auf Dateibasis überlegen (vgl. Abschnitt 2.1). Wird ein Rasterzugriff in einem DBMS durchgeführt, so entscheidet das System selbst über die Reihenfolge der Verarbeitung. Möglicherweise ist die Reihenfolge durch die benötigten Operationsschritte zwingend nacheinander erforderlich oder es kann mehrere Wege zur Abarbeitung der Anfrage gegeben. Der erste Schritt des Anfrageweges ist der Parser. Hier wird eine syntaktische Prüfung des Anfragestrings auf Korrektheit durchgeführt. Es folgen die Erzeugung des Anfragebaumes sowie eine Transformation des Anfragestrings in für die Datenbank verständliche Identifikatoren sowie die Trennung nach Anfragetypen. Das anschließende Rewrite schreibt, wenn vorhanden, die Anfrage von Sichten (Views) in physische Relationen (Tabellen) um. Die nächsten Arbeitsschritte erfolgen im Planer/ Optimierer der Datenbank. Über die möglichen Verarbeitungsreihenfolgen werden alle möglichen Ausführungspläne erstellt. Das System wählt anhand interner Kostenansätze, vorgegebener Regeln und statistischer Schätzungen dann den Plan aus, der die niedrigsten Gesamtkosten (Aufwand) benötigt. Das Ergebnis ist ein Plan-Baum, der die Reihenfolge der Abarbeitung umschreibt. Meist werden die Antwortmenge stark einschränkende Operationen zuerst ausgeführt. Der Ausführungsplan wird dem Executor zur Abarbeitung übergeben. Die Bildung der Ergebnismenge und die Rückgabe an den aufrufenden Client folgen [Stif04].

Bei der Planauswahl prüft das System durch Vergleich der Kosten auch die Verwendung von Indexen bei der Anfrageausführung. Optimiert wird dabei die Anzahl der Lesezugriffe auf den Sekundärspeicher. Ein vollständiger bzw. sequentieller Scan (Einlesen aller Tupel einer Relation zur Auswahl von Elementen) wird durchgeführt,

wenn kein Index angelegt worden ist oder das vollständige Einlesen der Relation nur sehr wenige Lesezugriffe benötigt. Alternativ wird ein Index verwendet oder auch vom System temporär angelegt. Das Anfrageergebnis wird davon nicht beeinflusst, lediglich die Anfragedauer wird durch das gezielte Lesen von Informationen verringert. Alle selektiven Anfragen profitieren von dem sortierten Tupelzugriff. Auch die sortierte Ausgabe einer Ergebnismenge sowie die Realisation von Eindeutigkeiten werden ermöglicht und beschleunigt. Nachteile sind die notwendige Indexpflege bei Datenveränderungen, die wachsende Größe der Datenbank durch das Halten redundanter Informationen sowie der erhöhte Verwaltungsaufwand des Datenbanksystems [Stif04]. Im angewandten Teil der Arbeit ist eine Untersuchung zur Indexoptimierung durchgeführt worden.

6.2 Rasterdatenaustausch

Zum Rasteraustausch zwischen DBMS, zwischen DBMS und Dateisystem oder zwischen den Anwendern können Mechanismen verschiedener konzeptueller Level genutzt werden. Der einfachste, immer noch häufig genutzte, Weg ist dabei die Verwendung *allgemeiner Bildformate*, die von vielen Herstellern mit teilweise geringer Verbreitung entwickelt wurden. Auch für andere Anwendungsbereiche erstellte Massenformate (z.B. GIF) ohne Integration raumbezogener Attribute fallen in diese Gruppe. Attribute können lediglich als externe Metadaten mitgeliefert werden.

Austauschkonzept	**interne Metadaten**	**externe Metadaten**	**Beispiele**
einfache Bildformate	Größe Kompression radiometrische Auflösung	Format räumliches Bezugssystem Lageinformation	PBM/PGM/PPM BMP GIF TIFF
räumliche Rasterformate	zusätzl. räumliches Bezugssystem Lageinformationen	Format	GEOTIFF NITF HDF
räumliche Datenaustauschmechanismen	zusätzl. Informationen über nicht-räumliche Attribute Qualitätsinformationen	Transfermechanismus	EDBS SAIF SDTS WTS,WCS,WFS

Tab. 7: Rasteraustauschkonzepte, Quelle: [Neb97]

Räumliche Rasterdatenformate beinhalten bereits Metadaten zur räumlichen Referenzierung der Daten. Das GeoTIFF-Format hat sich dabei als universelles und am weitesten verbreitetes Format als Standard durchgesetzt. Es beherrscht viele der bereits beschriebenen Rastermanagementkonzepte und ist durch die Vergabe von reservierten Metatags auch erweiterbar. Es ist ein gut dokumentiertes Format, dessen Patentschutzablauf auch zur weiteren Verbreitung beiträgt. Das hierarchisch aufgebaute

Datenformat (HDF) wird in der Fernerkundung durch seine große und spezialisierte Metadatenbandbreite für die Übertragung von Raster- und Vektordaten verwendet.

Räumliche Datenaustauschmechanismen werden durch Standardisierungsbemühungen internationaler Organisationen (ISO, CEN) weiterentwickelt und in verschiedenen Formaten zusammengefasst. (EDBS, SAIF) [Neb97]. Zukünftig soll der dateibasierte Datenaustausch durch einen dienstbasierten Austausch erweitert oder ersetzt werden. Als Beispiel soll hier die Dienstestruktur des Open Geospatial Consortium (OGC) näher betrachtet werden. Das OGC hat sich als Ziel die Förderung eines hohen Grades an software- und firmenübergreifender Interoperabilität gesetzt. Es ist ein weltweites Industriekonsortium, dessen Mitglieder die erarbeiteten Empfehlungen und Standards zeitnah umsetzen und so in die Praxis einführen möchten. Tritt ein Problem räumlichen Bezugs auf, so wird nach Analyse und Diskussion der Mitglieder eine zu testende Lösungsmöglichkeit entworfen, die als Standard verabschiedet werden kann. Es werden umfangreiche Konformitätstests von Softwareprodukten bzgl. der Standards durchgeführt.

Derzeit sind bereits einige Standards erfolgreich eingeführt worden (z.B. GML, WMS, WCS). Für die Rasterverwaltung kann, entsprechende Implementierung und Verbreitung vorausgesetzt, beispielsweise ein Web Terrain Service eingesetzt werden. Er produziert Ansichten georeferenzierter Daten und damit eine als Bild gerenderte visuelle Repräsentation eines Geländeausschnitts zur Gewinnung eines 'räumlichen' Eindrucks beim Nutzer [OGC1]. Die Geländemodelldaten können dazu beispielsweise durch einen Web Feature Service (WFS), welcher auf ein DBMS zurückgreift, als Punktgeometrie geliefert werden. Ein Web Coverage Service (WCS) stellt georeferenzierte Bilder (Raster) der Erdoberfläche bereit, die auf das Geländemodell projiziert werden [OGC3]. Zur Steuerung des WTS liegen 3 grundlegende Anfragen, die durch HTTP-Get oder Post-Request an einen Server gestellt werden, vor. Die GetCapabilities liefert, wie auch in den anderen von OGC standardisierten Services, Metainformationen über den Dienst sowie seinen Inhalt und akzeptierte Parameter. Der Request GetView liefert ein perspektivisches dreidimensionales Bild eines Ausschnitts der Erdoberfläche anhand verschiedener Visualisierungsparameter [OGC1].

Parameter	**Inhalt**
POI (x y z)	Point of interest, Sichtpunkt
Distance	Abstand zwischen POI und Standpunkt
Pitch	Winkel zwischen Standpunkt und POI
Yaw	Azimutwinkel zur Nordrichtung
AOV	Angle of view, Sichtfeld

Tab. 8: WTS-Visualisierungsparameter, Quelle: [OGC1]

Weitere Möglichkeiten wie die Definition des an den Aufrufenden zurückzuliefernden Ausschnittes mit einer rechteckiger Box und deren Umrechung in die obigen Parameter sowie Verwendung von nutzerbezogener Symbolisierung (SLD), Transparenz und auch die Unterstützung von Koordinatensystemen sind in den Standard einbezogen worden.

Ein Web Coverage Service dient der Unterstützung des elektronischen Austauschs räumlicher Rasterdaten, die im derzeitigen Ausbaustand (WCS 1.1) auf reguläre Grids beschränkt sind. In 1-3 Raumachsen eines Koordinatensystems sowie mit Hilfe einer Zeitangabe können räumlich-zeitliche regelmäßig verteilte Phänomene als Menge von Einzelwerten je Position und Zeit oder auch als Wertlisten (z.B. Spektralwerte) je Position an einen aufrufenden Client geliefert werden. Der GetCapabilities-Aufruf liefert Dienstbeschreibung sowie die verfügbaren Datenzusammenstellungen. DescribeCoverage beschreibt ein oder mehrere Datensätze (Coverages) durch ihre zeitliche und räumliche Domäne, durch die Art und Variation der Coveragewerte (RangeSet), durch das Koordinatensystem (z.B. als EPSG-Code), durch mögliche lieferbare Formate (GeoTIFF, HDF-EOS, DTED, NITF, GML) sowie mögliche bei Resampling oder Generalisierung verwendbare Interpolationsmethoden (z.B. Nearest Neighbor, bilinear, bikubisch). GetCoverage übermittelt aufgrund der an den WCS übergebenen Parameter (z.B. Name, Auflösung, Gebietsgrenze, Format, einschränkender Wertebereich, Zeitpunkt) den gewünschten Datensatz an den Client [OGC3].

Ein Web Feature Service (WFS) besitzt den gleichen Dienstaufbau wie bisher beschrieben, liefert mit GetFeature an den Client aber Geometriedatensätze, die auch aus einer räumlichen Datenbank stammen können (z.B. Oracle, PostgreSQL/PostGIS). Durch diesen Dienst können die Daten ggf. erzeugt, geändert oder gelöscht werden. Es ist mit Hilfe eines WFS möglich, eine universelle Schnittstelle zu einem datenbankbasierten Rasterdatenmanagementsystem zu realisieren.

Derzeit in der Entwicklung bei OGC sind Verfahren und Workflows zur Behandlung von Bildkatalogen und Bildarchiven. Mit nachfolgenden Arbeitschritten kann durch einen Image-Client eine Abfrage auch nach Geländemodelldatensätzen codiert als Bild gestaltet werden [OGC4].

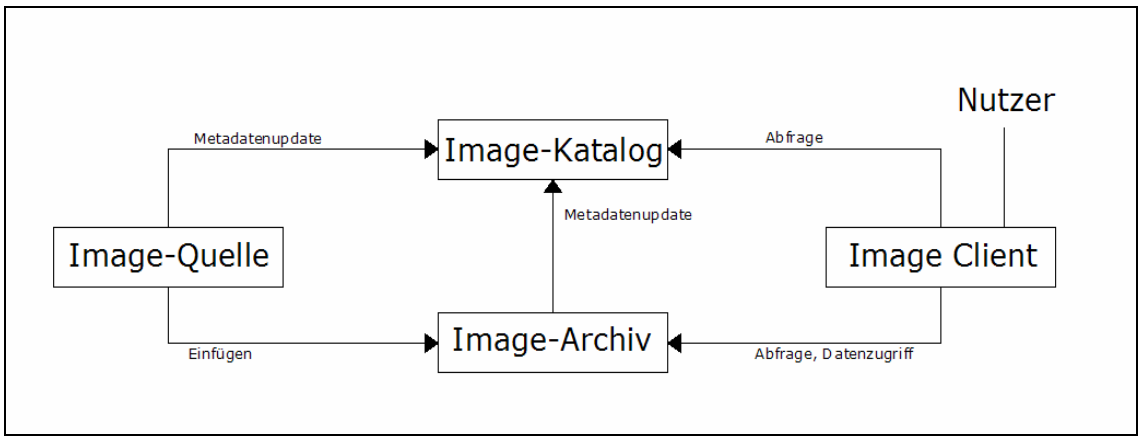


Abb. 10: Aufbau Imageverwaltung, Quelle: [OGC4]

Als Imagearchive dient dabei ein Server, der Einfügen und Löschen sowohl von Coverages (GetCoverage) als auch Karten (GetMap) und Objekten (GetObject) unterstützt. Ein Imagekatalog-Server dient der Metadatenverwaltung dieser Archivdaten und ermöglicht deren Erstellung und Abfrage. Umfangreiche Metadatenschemata beispielsweise bzgl. Imagetype, Kategorie, verwendetem Sensor, Komprimierung, Größe und Formate sowie auch 'USE CASES' verschiedener Anwendungsfälle (z.B. Suche eines Bildes eines Gebietes) auch verschiedener Anwendungsgebiete (z.B. Landwirtschaft) werden in Dokumenten des OGC vorgestellt und zur Diskussion gegeben [OGC5], [OGC6]. Ein Standard in diesem Bereich ist noch nicht verabschiedet worden. Zukünftig kann auch über diese Schnittstelle eine Rasterdatenbereitstellung an eine Nutzergruppe durchgeführt werden.

7 Optimierungsansätze einer Rasterverwaltung

Wesentlich zur Erstellung einer anwendungsorientierten Rasterverwaltung ist die Nutzung von Optimierungspotentialen. Algemeingültig für viele Managementsysteme werden in diesem Kapitel die Bereiche Indexierung und Komprimierung von großräumigen Rasterdaten vorgestellt, ausgehend von den Verfahren und Methodiken bis hin zu je einem praktischen Performancevergleich.

Die Verbesserung der Performance eines DBMS vor Beginn des laufenden Betriebes wird durch die Möglichkeiten des physischen Datenbankentwurfs erreicht. Unter Beachtung der Erkenntnisse aus den Datenbankentwurfsphasen sowie Überlegungen zu den zukünftigen typischen Arbeitslasten (Lastprofil) werden Speicherungsstrukturen und Zugriffsmechanismen, die logische Speicherorganisation sowie das interne Datenbankschema und die zugehörigen Systemparameter festgelegt. Es werden Entscheidungen zum Dateiformat, zur Redundanz der Datenspeicherung sowie Datei-Zugriffs-Performance und zur Abwicklung von Datenbankänderungen (Update) getätigt. Eine Form der normalerweise zu vermeidenden Datenredundanz ist die Indexverwendung, die aus Sicherheits- und Performancegründen meist genutzt wird. Indexe werden angelegt, wenn Selektionen oder Anfragen vorgesehen sind, die davon stark profitieren werden. Kandidaten für Indexfelder sind die Attribute einer Relation, die für Vergleiche benötigt werden. Höchstens ein Index ist zur Clusterung einer Relation einsetzbar, bei der gemeinsam von einer Abfrage benötigte Werte auch physikalisch dicht beieinander abgelegt werden (z.B. für Bereichsanfragen). Bei der Indexanlage sind neben dem Performancegewinn auch der Speicherplatzaufwand sowie der Aufwand für die Indexwartung und Erstellung zu beachten. Eine weitere Form der Datenredundanz wird durch Datenreplikation erstellt. Die Erhöhung der Zuverlässigkeit und Verfügbarkeit kann hierbei durch redundante Datenverteilung (RAID-Architektur, Relationsverteilung) erreicht werden. Durch Denormalisierung (Attributredundanz) werden Attribute, auf die einzeln zugegriffen werden soll, repliziert. Bei Datenveränderungen sind hier entsprechende Maßnahmen zur Sicherung der Datenintegrität zu ergreifen. Materialisierte Sichten (Views) bieten als abgeleitete redundante Informationen schnellen Zugriff auf vorher berechnete Ergebnisse, die ebenfalls bei Datenupdates nachgeführt werden müssen.

Zur Optimierung der Datei-Zugriffs-Performance kann neben der Erhöhung der I/O-Bandbreite durch Declusterung von Zugriffsseiten über mehrere Sekundärspeicherplatten auch die Datensatzclusterung eingesetzt werden. Diese dient

der Maximierung des Lesens relevanter Informationen in einer Zugriffsoperation bei der Minimierung der Zugriffe zur Beantwortung einer Anfrage [VOS00]. Benötigt werden dazu Aussagen zu zukünftigen Zugriffsmustern auf die Datenbank. Sind Datenredundanzen vorhanden, so kann der Datenbankoptimierer wählen, welcher Zugriffsweg bzw. Datenablageort genutzt wird.

Unter Datenbank-Tuning werden sämtliche Aktivitäten verstanden, welche zur Laufzeit die Zeiteffizienz oder die Durchsatzrate einer Datenbankapplikation verbessern [VOS00]. Hier bieten sich neben Datenstrukturmodifikationen und Anpassungen der Parameter des Datenbanksystems auch Veränderungen der Betriebssystemkonfiguration (Puffergrößen, Plattenlayout) sowie Modifikationen der Hardware (Ausbau Primärspeicher, Prozessoren) an. Die angemessene Aufgabenverteilung zwischen den Systemkomponenten (Client, Server) in Abhängigkeit von den benötigten Ressourcen sowie der gewünschten Lokalisation der Informationen wird ebenso untersucht und optimiert wie auch das Verhältnis von inertialem zu laufendem Aufwand von Operationen. Bei ständig wiederkehrenden Anfragen kann, beispielsweise durch Kompilierung, der Aufwand für Parsing, Anfragenoptimierung und Codeerzeugung einspart werden (siehe Kapitel 6.1). Da selten alle Systemkomponenten gleich stark ausgelastet sind, sollte durch räumliche und zeitliche Aufteilung der Bildung von Datendurchsatzengstellen (z.B. lange Anfragen behindern kurze Anfragen) vorgebeugt werden. Nach Identifikation eines Performanceproblems wird durch lokale Optimierung mittels Systemeingriff ein möglichst großer globaler Effekt zu erreichen versucht.

7.1 Kompression von Rasterdaten

7.1.1 Einteilung und Allgemeines

Verfahren zur Reduktion des Speicherbedarfes von Daten werden als Datenkompression bezeichnet. Sie dienen der Verringerung des Datenaufkommens während einer Übertragung oder bei einer Lagerung. Kompressionsalgorithmen spielen beim Rasterdatenmanagement eine sehr große Rolle, da durch heutige Datenerfassungsmethoden sehr umfangreiche Datensätze entstehen können. Die Optimierung von Speicherung und Zugriff dieser Datensätze wird wesentlich durch eine adäquate Kompression gestützt.

Zuständig für die Komprimierung und Dekomprimierung als reverser Funktionalität ist ein Codec (Coder/Decoder), meist als ein eingebetteter Algorithmus oder eine spezialisierte Bibliothek. Datenbanken nutzen diese Codecs als an die möglichen Eingangsdaten angepasste Erweiterungen zur optimierten Ablage, Ausgabe und

Übertragung bzw. zur Umwandlung in andere oder eigene Formate oder zur Kompression. Das wesentliche Merkmal eines Komprimierungsalgorithmus ist seine Packrate: Packrate = Länge der Originaldaten / Länge der komprimierten Daten.

Die Packrate ist stark abhängig vom Rastertyp und von Informationsgehalt des Datensatzes. Ebenso ist die Unterscheidung in verlustfreie Kompressionsmethoden und verlustbehaftete Methoden wesentlich. Verlustfreie Methoden erreichen Packraten bei DGM von 1.1-1.5. Bei ihnen ist die Rekonstruktion der Originaldaten aus den gepackten Daten ohne Verluste möglich. Der Kompressionsprozess ist also reversibel. Es werden lediglich redundante Informationen aus den Daten entfernt [Neb97].

Im Gegensatz dazu können bei der verlustbehafteten Kompression aus den gepackten Daten nicht wieder die Originaldaten erzeugt werden. Dieser Nachteil ermöglicht aber deutlich höhere Kompressionsraten. Durch ein Modell wird hier der entbehrliche Teil der Daten ermittelt und damit eine Irrelevanzreduktion durchgeführt.

Es sind sehr unterschiedliche Kompressionsalgorithmen entwickelt worden, die sich durch ihre Effizienz (Packrate), Implementierungskomplexität, Codierungslaufzeit sowie bei Erweiterung auch auf die verlustbehafteten Methoden durch ihre Signalqualität unterscheiden.

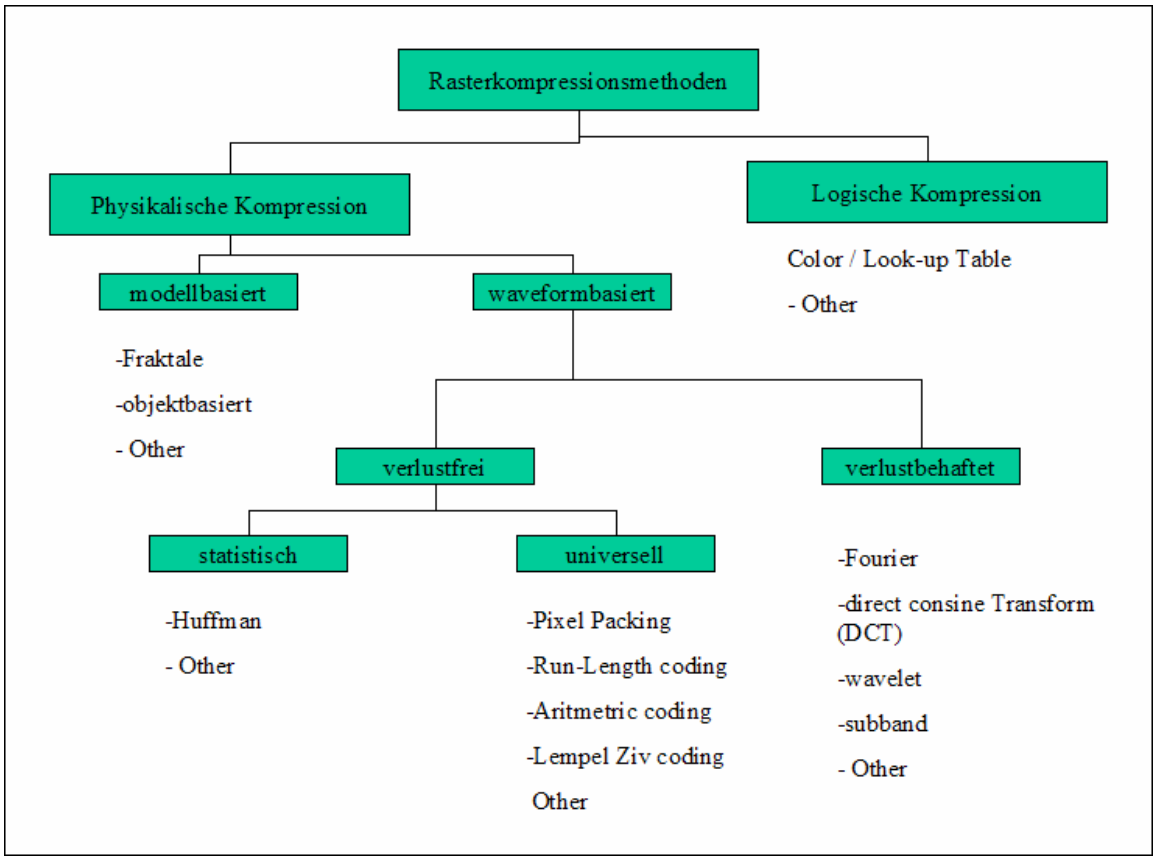


Abb. 11: Kompressionsmethodeneinteilung, Quelle: [Neb97]

Logische Kompressionsmethoden, basierend auf externem Wissen über die Daten, eliminieren durch Ablage von weniger speicherintensiven Indexwerten stark datenabhängig mit guten Packraten redundante Informationsteile.
Bei der physischen Kompression wird die statistische Datenredundanz, zurückzuführen auf räumliche (Flächen gleicher Attributwerte), zeitliche (z.B. Bewegungsdaten) oder spektrale (z.B. Fernerkundungsdaten) Korrelation der Zellwerte, ausgenutzt, um die nicht benötigten Informationsanteile zu entfernen.
Statistische Kompressionsverfahren, auch Präcodierung genannt, nutzen die Häufigkeit des Auftretens eines Wertes oder einer Gruppe von Werten. Anstelle der Werte werden meist kürzere Bitfolgen gespeichert. Statistische Algorithmen bestehen aus einer Modellierungskomponente, einem Wahrscheinlichkeitsmodell, das aus den Eingangsdaten oder aus apriorischen Annahmen über die Daten abgeleitet wird und aus einer Abbildungskomponente, die die Symbol-zu-Codewort-Umwandlung bewältigt. Das Wahrscheinlichkeitsmodell kann statisch als Statistik für alle Daten vor der Verarbeitung erzeugt werden. Dieser einmalige Prozess ist sehr schnell, leider aber lokal nicht optimal an die Daten angepasst. Außerdem muss diese Statistik (Beispiel: Huffman) im komprimierten Datensatz enthalten sein. Dynamische Modelle ermitteln vorwärts- oder rückwärtsdynamisch nach jedem bearbeitetem Zeichen die Statistik zeitaufwendig neu. Vorteile sind hier die lokale Anpassung der Statistik und die nicht benötigte Übertragung an den Decodierer [Neb97].
Universale Kompressionsalgorithmen, die kein Wissen über die Daten vor der Kompression benötigen, können relativ einfache mathematische Methoden oder auch komplexe hochentwickelte Abläufe enthalten.

7.1.2 Ausgewählte Verfahren kurz vorgestellt

Bei der *Huffman-Codierung* wird aus den Daten eine Zeichenliste (Alphabet) ermittelt und die Zeichen mit einer unterschiedlichen Bitanzahl codiert (Enthropiecodierer). Dadurch werden nicht mehr beispielsweise 4 Byte je Doublewert benötigt, sondern eine geringere Bitanzahl. Zur Vermeidung von Mehrdeutigkeiten müssen die verwendeten Bitfolgen präfixfrei sein, d.h. keine Bitfolge, die für einen bestimmten zu codierenden Wert steht, darf am Anfang einer Bitfolge eines anderen Zeichens stehen. Zur Erzeugung dieser Präfixfreiheit wird ein Binärbaum, dessen Teilbäume mit 0 oder 1 belegt werden, generiert. Der Pfad von der Baumwurzel zum codierten Wert, aufgereiht in einer Bitfolge, ergibt die gewünschte Codierung.

Es wurden mehrere Ansätze zur Ermittlung der Binärbaume angefertigt. Zum Decodieren müssen die Bäume innerhalb der codierten Datei möglichst kompakt mitgeliefert werden. Die Nutzung eines allgemeinen nicht mitzuliefernden Baumes würde zu nicht optimalen Kompressionsergebnissen führen. Zur kompakten Speicherung wird der Baum durch Knotenaustausch von rechts nach links in der Tiefe ansteigend sortiert. Die Angabe der Bitanzahl des verschlüsselten Zeichens ermöglicht die Rekonstruktion des Baumes und damit die Decodierung. Sind die Häufigkeiten der Zeichen innerhalb der Zeichenliste sehr unterschiedlich, so kann dies durch Codierung von Zeichenketten mit Häufigkeiten in vorher unbelegten Bereichen optimiert werden. Das Bedürfnis einer ganzzahligen Bitfolge je Zeichen wird durch die Zeichenkettenverschlüsselung verhindert. Eine Huffman-Codierung ist nur für Datensätze mit einem diskreten Wertebereich geeignet. Die Codierung von digitalen Geländemodellen mit kontinuierlichem Spektrum bedarf daher einer Vorverarbeitung.

Beim *Lauflängenverschlüsselungsalgorithmus* werden die Werte des Rasters durch Paare aus Wert und Anzahl ihres Auftretens ersetzt. Sehr effizient ist diese Methode bei Rastern mit hoher räumlicher Korrelation, d.h. großen Flächen einheitlicher Zellwerte.

Im *Packbits-Verfahren* wechseln sich Wortverschlüsselung (jeder Datensatz wird einzeln codiert und durch Symbolwert ersetzt) und Lauflängencodierung ab. Die Wortverschlüsselung bildet bei der Kompression von Rastern wegen ihrer einfachen Handhabung und der Abwesenheit von hochkomplexen Algorithmen die Basis aller Methoden [Neb98].

Die kleinste adressierbare Speichereinheit eines Datenverarbeitungssystems ist ein Byte. Zur Speicherung von DGM werden als grundlegender Zelltyp Werte des Typs Double (4Byte) benötigt. Bei Rastern geringerer Auflösung oder einfachen Zellwerten sind kürzere Bitfolgen (ein Byte = 8 Bit) zur Codierung ausreichend. Die Aneinanderreihung von Datenbits ohne Auffüllung zur Byteebene ermöglicht beim *Pixel/Bitpacking* z.T. die Speicherung mehrerer Zellwerte in einem Byte bzw. mehreren Bytes. Der TIFF-Kompressionsalgorithmus nutzt beispielsweise das Bitpacking in Kombination mit anderen Techniken.

Die Speicherung von DGM und auch Luft- und Satellitenbildern erfolgt in Rastern kontinuierlicher Zellwerte, bei denen eine starke Abhängigkeit von räumlich aufeinanderfolgenden Werten besteht. Statt des originalen Zellwertes wird bei der Methode der *Differentialcodierung* lediglich der Differenzbetrag zum vorherigen Zellwert gespeichert, da diese Differenz weniger Speicherplatz als der vollständige

Wert einnehmen kann. Differentialcodierung wird oft als Vorbereitung eines Rasters zur Nutzung anderer hochentwickelter Kompressionsmethoden genutzt [Neb97].

Die Grundlage der *Wavelet-Transformation* ist die Fourier-Transformation, die eine gegebene zeit- oder ortsabhängige Funktion in die in ihr vorkommenden Frequenzanteile zerlegt. Diese Anteile werden durch neue frequenzabhängige Funktionen beschrieben (Fourier-Transformierte). Zur gleichzeitigen Information über die vorkommenden Frequenzen sowie deren ortsabhängiges Auftreten wird durch Multiplikation des Eingangssignals mit einer Fensterfunktion ein Stück aus den Eingangsdaten herausgeschnitten (Short-Time-Fourier-Transformation). Aufgrund der Heisenbergschen Unschärferelation kann nicht gleichzeitig eine gute Zeit- bzw. Ortsauflösung (schmale Fenster) und eine gute Frequenzauflösung (breite Fenster) erfolgen. Dieses Problem vermeidet die Wavelet-Transformation, die mit Fenstern variabler Größe und fester Schwingungsanzahl je Fenster operiert. Durch Skalierung und Translation eines Basiswavelets zur Anpassung an das Ausgangssignal (Waveletkoeffizient) werden sehr rechenintensiv Wavelets unterschiedlicher Auflösungen erzeugt. Die Überlagerung aller Wavelets ergibt wiederum das Ausgangssignal. Für verschiedene Kompressionsverfahren wurden unterschiedliche Basiswavelets entwickelt. Beispiele sind 'Daubechies' für verlustbehaftete sowie 'LaGall' für verlustfreie Kompressionen im JP2000-Format. Um den Rechenaufwand zu verringern, werden die Parameter (Zeit/Skalen) diskretisiert (diskrete Wavelet-Transformation). Dies führt auch zu einer Verringerung der Redundanz der Informationen der Wavelets ohne die Vollständigkeit der Signalbeschreibung zu beeinflussen. Ergebnis einer Wavelet-Transformation sind je Datenzelle ein Waveletkoeffizient. Diese Koeffizienten können anschließend leichter komprimiert werden. Es folgt eine Quantisierung sowie eine Entropiecodierung (z.B. Huffman) und/oder eine Lauflängencodierung. Vorteil der Waveletkomprimierung ist die Einstellung der Verfahrensparameter von der verlustfreien bis zu einer beliebig hoch verlustbehafteten Datenverarbeitung. Sie findet im MrSid-Format (Multi resolution Seamless image database) sowie im JPEG2000-Format Anwendung. [Dan04]

Der bekannte *LZW-Algorithmus* (Lempel-Ziv-Welch) gehört zu einer aufgrund zahlreicher Ergänzungen und Abwandlungen entstandenen Gruppe von Kompressionsalgorithmen (LZ-Gruppe). Da hier die Informationen implizit im komprimierten Datensatz enthalten sind, muss kein gesondertes Lexikon permanent geschrieben werden. Die Wörterbucheinträge werden zur Laufzeit der Kompression und Dekompression erzeugt und dienen mit Hilfe eines Index zur Daten-Code- sowie Code-

Daten-Wandlung. Die Unterscheidung zwischen einem Zeichen und einem Code erfolgt durch ein Flag. Die LZ-Algorithmen arbeiten verlustfrei und bilden Teile von GIF (LZ78), TIFF (LZ77) oder JPEG (LZ77).Abweichend zu den Entropiecodierern (z.B. Huffman) werden Zeichenfolgen durch Zeichenfolgen eines anderen Alphabets ersetzt (Stringersatzverfahren). Der LZ77-Algorithmus ist im Gegensatz zur LZ78-Methode nicht mit Patenten belegt und wird in Kombination mit Huffman beispielsweise im Deflate-Algorithmus (TIFF) oder beim Datenformat PNG verwendet.

Zur Codierung des LZ77 wandert ein aus Textfeld und Vorschaupuffer bestehendes Fenster über den Datensatz. Komprimierte Daten werden in 3-Tupel-Form abgelegt. Der erste Wert entspricht der Position des Lexikoneintrags, der nächste Wert der Länge der zu kopierenden Daten. Abschließend wird das nachfolgende Zeichen abgelegt. Für neue Zeichen wird die Folge "0,0,neues Zeichen" in das Wörterbuch eingefügt. Die Dekompression erzeugt verlustfrei aus den Lexikoneinträgen wieder die Originaldaten. Die Kompression ist von der Lexikongüte anhängig. Für gute Raten ist eine Mindestfenstergröße notwendig, die durch die umfangreichen Suchen im Wörterbuch aber zu steigender Kompressionszeit führt.

Für den LZ78-Algorithmus wird ebenso sequentiell der zu codierende Datensatz verarbeitet. Ist das aktuelle Zeichen im Lexikon bereits enthalten, so wird die Suche unter Erweiterung um das nächstfolgende Zeichen erneut gestartet. Ist das Muster noch nicht vorhanden, so wird das aktuelle Muster (bzw. Zeichen) und dessen Nachfolger mit dem aktuellen Index versehen im komprimierten Datensatz abgelegt. Zur Dekompression wird beim zweiten Zeichen beginnend, jeweils das ausgegebene Zeichen und dessen Vorgänger incl. Index als Lexikon wiederhergestellt. Findet der Decoder nun einen Index, so wird dieser in der Ausgabe durch seinen Wert ersetzt [Jäg].

7.1.3 Dateiformate und Standards

Sehr eng verknüpft sind Kompressionsaspekte mit den Überlegungen zu Datenaustausch und Datenübertragung. Wesentliche Standards im Bereich der in Austauschformate integrierten Kompressionsalgorithmen sind das Tagged Image File Format (TIFF) sowie JPEG (Joint Photographic Expert Group) und dessen Weiterentwicklung JPEG2000.

Die Firmen Aldus und Microsoft entwickelten das *TIFF*-Format (aktuell Version 6, 1992), dass durch zahlreiche Erweiterungen zu einer sehr großen Komplexität aber auch zu einer weiten Verbreitung gelangte. Wesentliche Eigenschaften sind die Multi-Page-Möglichkeit (mehrere Bilder je Datei), wählbare Farbräume sowie in Form von

Informationseinheiten (Tags) abgelegte Metadaten. Je Bildpixel können beliebig viele Einzelwerte (zusammengefasst zu Bändern) der Datentypen Integer oder Float sowie auch Transparenzinformationen gespeichert werden. Das Pixelbild wird zur Verbesserung der Zugriffsmöglichkeiten und Ladezeiten in Stripes (Pixelzeilen) oder rechteckige Tiles (Kacheln) unterteilt, die mit einem 32-Bit-Offset angesprochen werden (Partitionierung). Als Kompressionsalgorithmen stehen verlustfrei LZW und Lauflängencodierung sowie verlustbehaftet eine Form der JPEG-Komprimierung zur Verfügung. Erweiterungen des Formats ermöglichen die Deflate-Kompression, eine Kombination von LZ77 und Huffman mit parametrisierbarer Fenstergröße und Suchintensität sowie unterschiedlichen Arten von Huffmantabellen (siehe oben). Die Nutzung des TIFF-Formates ist lizenzfrei beispielsweise durch die GDAL/libtiff-Bibliothek möglich. Eine Erweiterung um Koordinatenverarbeitung stellt das GeoTIFF-Format ebenso lizenzfrei nutzbar, beispielsweise mit der GDAL/libgeotiff-Bibliothek, dar [TIFF]. Aufgrund des Float-Datenzelltyps, der weiten lizenzfreien standardkonformen Verbreitung sowie den Tag- und Kompressionsmöglichkeiten eignet sich GeoTIFF sehr gut als Schnittstelle zum Austausch von DGM.

JPEG ist das am weitesten verbreitete Grafikformat. Es wurde von der Joint Photographic Experts Group, einer Zusammenarbeit der Internationalen Fernmeldeunion, International Elecktrotechnical Commission sowie der ISO, 1992 in der Norm ISO 10918-1 festgelegt. Bei seiner Kompression wird zuerst eine Farbraumumrechnung vorgenommen. Dann folgen eine Tiefpassfilterung und die verlustbehaftete Unterabtastung der Farbraumabweichungssignale. Nach einer Einteilung in 8x8 Blöcke wird eine diskrete Kosinustransformation (DCT) der Blöcke, implementiert durch eine Fast-Fourier-Transformation durchgeführt. Es folgen Quantisierung, Zick-Zack-Umsortierung anhand der Frequenz sowie abschließend eine Entrophiecodierung (z.B. Huffmann). Bei der ebenfalls verlustbehafteten Quantisierung werden die DCT-Koeffizienten durch eine im Dateiheader gespeicherte Matrix geteilt und auf Ganzzahlen gerundet. Diese Matrix beeinflusst wesentlich Kompressionsrate und Qualität. Zusatzinformationen als Metadaten können an die Dateien angehängt werden. Eine verlustfreie Variante (JPEG Lossless Mode) nutzt eine Kombination aus Präcodierung und Entrophiecodierung. Neu beim JPEG2000-Format ist die Verwendung einer diskreten Wavelettransformation (DWT) anstelle der DCT. Ein gleitender Übergang zwischen verlustloser und verlustbehafteter Kompression ist nach Aufteilung des Eingangsbildes in Teilbilder und als Folge der obigen Schritte hier bei sehr guten Kompressionsraten möglich [Cop06].

7.1.4 Kompressionsvergleich am Beispiel

Für einen Vergleich einiger der beschriebenen Kompressionsverfahren wurden 3 DGM-Raster unterschiedlicher Eigenschaften (Auflösung 1x1m) ausgewählt. Das kleinste Raster 'Ribnitz' enthält einen 4500x2500m großen Innenstadtbereich einer Kleinstadt mit Daten aus einer Laserscanningbefliegung. Das 'ländliche' Raster 'Michaelsdorf' der Größe 5000x5000m enthält Daten sehr unterschiedlicher Herkunft. Etwa 10% stammen aus einer Laserscanningbefliegung, weitere 25% sind durch terrestrische GPS-Vermessung entstanden. Die restlichen Bereiche beruhen auf der Digitalisierung von Höhenlinien im Wasser- und Landbereich. Das Raster 'Insel Ummanz' (8000x9500m) wurde zu etwa 80% aus digitalisierten Höhenlinien einer topographischen Karte gewonnen. Neben etwa 5% Vermessungsdaten (Deiche) sind auch 15% Seevermessungsdaten (Multibeam) enthalten. Die Erstellung der Raster erfolgte in einem Mosaikingprozess.

Die Tabelle 9 zeigt die Anwendung der verlustfreien Verfahren LZW, PackBits und Deflate im TIFF-Format und des verlustbehafteten Verfahrens JPEG2000 mit den Qualitätsstufen 99%, 5% und 1%. Im oberen Teil der Tabelle wird als Zelldatentyp ein 32-Bit-Float (Fließkommazahl) verwendet. Im unteren Teil wurden die Zellwerte durch Multiplikation in Zentimeter-Höhen und unter Abschneidung der Nachkommastellen in 32-Bit-Integerwerte umgewandelt. Da die Messgenauigkeit der Datengrundlagen der DGM oberhalb dieser Qualitätsschwelle liegt, werden durch diesen Schritt lediglich irrelevante Informationsanteile abgetrennt. Das 32-Bit-Format (4 Byte je Zellwert) wurde zum Vergleich mit den Fließkommazahlenrastern beibehalten. Die Vergleiche wurden mit der GDAL-Bibliothek der FW-Tools 1.0.9 als Batch-Script durchgeführt (www.gdal.org).

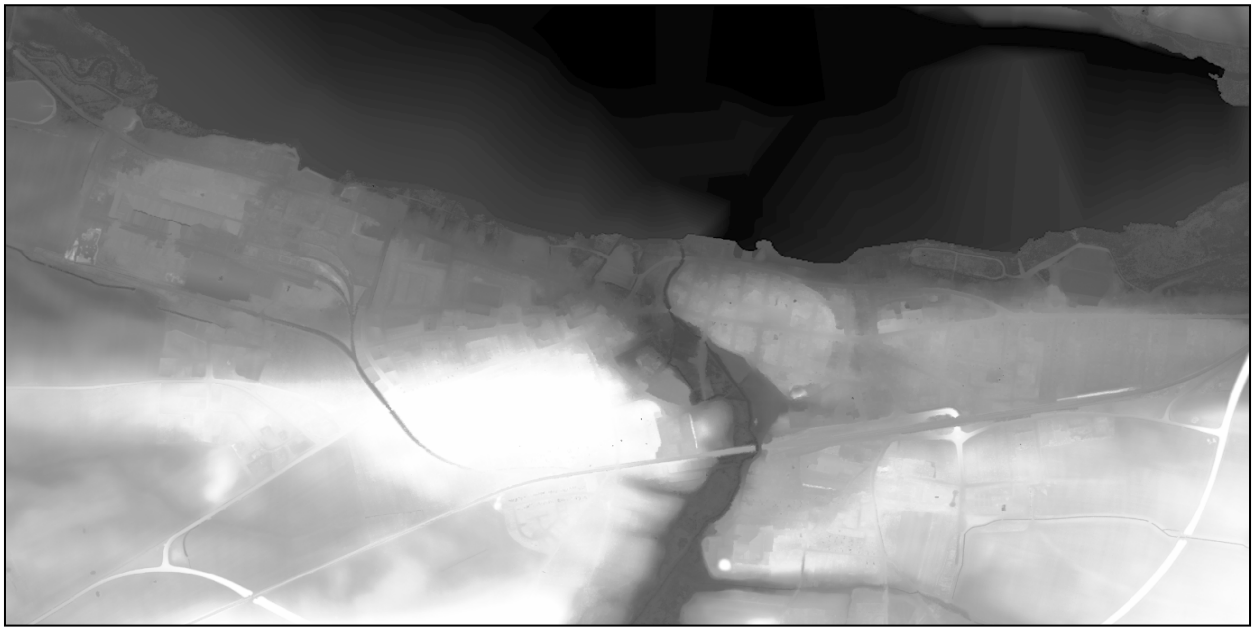
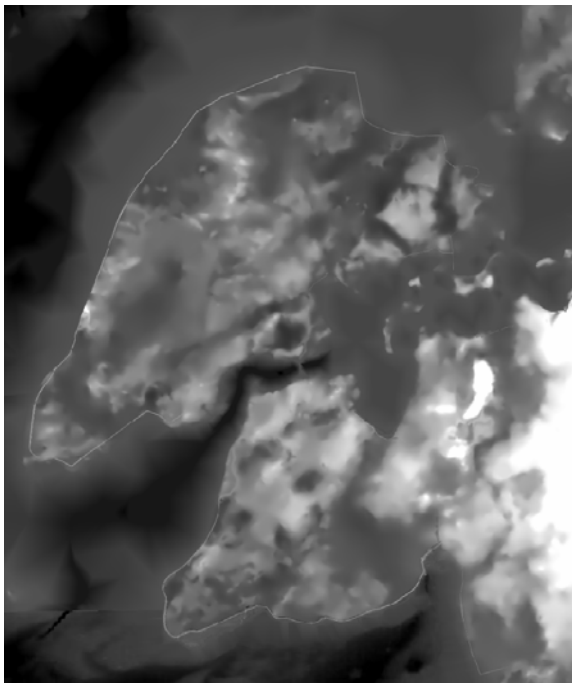


Abb. 12: Graustufen-DGM Ribnitz (verkleinert)



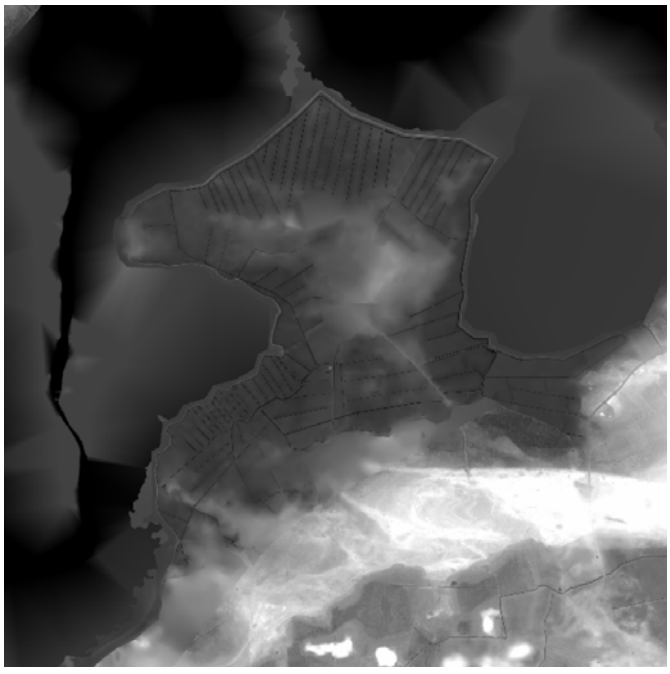


Abb. 13: links: Graustufen-DGM Ummanz (verkleinert)
Abb. 14: rechts: Graustufen-DGM Michaelsdorf (verkleinert)

FLOAT32													
		DGM-Raster Auflösung 1x1m											
Format	Kompression	Ribnitz	4500x2500			Michaelsdorf	5000x5000			Ummanz	8000x9500		
		Bytes		PR	Zeit [s]	Bytes		PR	Zeit [s]	Bytes		PR	Zeit [s]
GTIFF													
	None	45.020.242	100%	1.00		100.080.912	100%	1.00		304.152.912	100%	1.00	
	LZW	46.264.688	103%	0.97	4.81	103.114.162	103%	0.97	11.25	325.743.576	107%	0.93	38.96
	PackBits	45.362.538	101%	0.99	6.60	98.052.112	98%	1.02	10.87	303.685.446	100%	1.00	31.97
	Deflate	37.168.134	83%	1.21	8.34	80.885.022	81%	1.24	12.95	257.943.244	85%	1.18	42.02
JPEG-2000(Kakadu)													
	99%	6.129.128	14%	7.35	8.17	5.546.571	6%	18.04	7.38	1.753.003	1%	173.50	13.97
	5%	2.252.492	5%	19.99	6.95	5.004.678	5%	20.00	8.23	1.766.758	1%	172.15	14.08
	1%	452.170	1%	99.56	6.91	1.004.218	1%	99.66	7.01	1.767.605	1%	172.07	13.00
INT32													
		Ribnitz				Michaelsdorf				Ummanz			
GTIFF													
	None	45.020.242	100%	1.00		100.080.912	100%	1.00		304.152.912	100%	1.00	
	LZW	19.800.348	44%	2.27	5.19	35.021.936	35%	2.86	6.81	112.143.552	37%	2.71	20.37
	PackBits	45.209.894	100%	1.00	6.92	95.680.598	96%	1.05	10.03	293.970.804	97%	1.03	31.07
	Deflate	11.996.062	27%	3.75	9.40	17.452.596	17%	5.73	11.00	44.125.486	15%	6.89	28.16
JPEG-2000(Kakadu)													
		n.a.				n.a.				n.a.			

PR = Packrate, Zeit = Kompressionszeit

Tab. 9: Kompressionsvergleich

Basis: Float32-Ausgangsbild Ribnitz

Basis	NONE	NONE	NONE	NONE	NONE	NONE	JP2 99%
Vergleich	LZW	PackBits	Deflate	JP2 99%	JP2 5%	JP2 1%	JP2 1%
min	0.00	0.00	0.00	-2.82	-2.82	-2.82	-0.57
max	0.00	0.00	0.00	23.94	23.94	23.94	0.55
mean	0.00	0.00	0.00	5.65	5.65	5.65	-4.05E-06
Std.	0.00	0.00	0.00	5.64	5.64	5.64	0.05

Basis: Float32-Ausgangsbild Michaelsdorf

Basis	NONE	NONE	NONE	NONE	NONE	NONE	JP2 99%
Vergleich	LZW	PackBits	Deflate	JP2 99%	JP2 5%	JP2 1%	JP2 1%
min	0.00	0.00	0.00	-11.00	-11.00	-11.05	-0.44
max	0.00	0.00	0.00	11.09	11.09	11.22	0.41
mean	0.00	0.00	0.00	0.66	0.66	0.66	-8.01E-06
Std.	0.00	0.00	0.00	1.87	1.87	1.87	0.02

Basis: Float32-Ausgangsbild Ummanz

Basis	NONE	NONE	NONE	NONE	NONE	NONE	JP2 99%
Vergleich	LZW	PackBits	Deflate	JP2 99%	JP2 5%	JP2 1%	JP2 1%
min	0.00	0.00	0.00	-4.48	-4.48	-4.48	0.00
max	0.00	0.00	0.00	14.71	14.71	14.71	0.00
mean	0.00	0.00	0.00	0.49	0.49	0.49	0.00
Std.	0.00	0.00	0.00	1.82	1.82	1.82	0.00

Tab. 10: Vergleich der Höhenabweichungen

Erkennbar ist deutlich, dass bei den Float-Rastern die Kompressionsverfahren LZW und PackBits keine wesentlichen Kompressionserfolge erzielen. Teilweise sind sogar Dateivergrößerungen zu beobachten. Beide Verfahren beinhalten Wort- oder Zeichenverschlüsselungen jeweils eines Zellwertes. Da bei 32-Bit-Fließkommazahlen kaum identische Werte in einem DGM zu finden sind, verursacht die Verschlüsselungstabelle der Komprimierung eine Dateivergrößerung. Nach der Wandlung in Ganzzahlen (Integer) im unteren Tabellenteil verbessert das LZW-Verfahren durch die häufiger vorkommenden identischen Werte (Rundung) die Kompressionsergebnisse sehr deutlich. PackBits hingegen kann von der Zelltypveränderung nicht profitieren. Die nachfolgende Lauflängencodierung egalisiert hierbei die Gewinne der Wortverschlüsselung. Die Deflate-Kompression hingegen zeigt auch mit Fließkommazahlen verlustfrei einige Speicherplatzgewinne durch die Anwendung des Huffman-Algorithmus. Durch die Zelltypwandlung im unteren Tabellenteil werden die Kompressionsraten noch einmal deutlich verbessert.

Das untersuchte verlustbehaftete Verfahren JPEG2000 nach der KAKADU-Implementation führt zu deutlich besseren Kompressionsraten. Für die 32-Bit-Integer-Zellwerte stand das JPEG2000-Verfahren nicht zur Verfügung. Ermittelt man die Abweichungen durch Subtraktion der komprimierten Daten von den Originaldaten, so zeigen sich nicht akzeptable Veränderungen der Zellwerte von bis zu 23m bei dem verlustbehafteten Verfahren (Tabelle 10). Dies ist unabhängig von den eingesetzten Qualitätsparametern. Bei den verlustfreien Verfahren wurden keine Abweichungen festgestellt. Das JPEG-2000 Verfahren ohne 'Lossless Mode' ist für die Komprimierung von Daten digitaler Geländemodelle zu Analysezwecken nicht geeignet. Empfohlen werden kann aufgrund der Untersuchungsergebnisse der TIFF-Deflate-Algorithmus.

In der Tabelle 9 sind die Kompressionszeiten der Verfahren angegeben. Sie sind erkennbar deutlich von der Datengröße (Schreib-/Lese-Zeiten) abhängig und zeigen auch die Komplexität des jeweiligen Algorithmus (Rechenaufwand).

7.2 Indexierung

Ein Index innerhalb einer Datenbank ist eine separate Struktur, die die Suche nach bestimmten Schlüsselfeldern stark beschleunigen kann. Sie dient als dynamisches Inhaltsverzeichnis für ein einzelnes Attribut, eine Gruppe von Attributen oder wird auf Basis eines Attributausdrucks berechnet. Ein Index besteht aus einer Menge von Zeigern bzw. Verweisen, die eine Ordnung auf die Indexgrundlage definieren. Bei Anfragen kann das DBMS statt des ohne Index notwendigen sequentiellen Durchsuchens der gesamten Tabelle (Full-Scan) eine eingeschränkte Suche anhand der Indexinformationen durch Ermittlung bzw. Berechnung von Datensatzpositionen mit einigen Laufzeitvorteilen vornehmen. Bei einem Primärindex, basierend auf einem Primärschlüssel, sowie einem Clusterindex, basierend auf einem nicht eindeutigen Attribut, wird nicht für jeden Datensatz ein Indexeintrag angelegt (nicht dicht, je Relation einmalig). Die Indexeinträge sind wie die Datensätze sortiert. Im Gegensatz dazu wird bei einem beliebig oft anlegbaren Sekundärindex, basierend auf Schlüssel- oder Nichtschlüsselattributen, für jeden Datensatz ein Indexeintrag angelegt (dichter Index) und die Indexordnung nicht der Datensatzordnung angepasst. Ein funktionaler Index nutzt nicht die Attributwerte direkt, sondern behandelt mittels einer Funktion transformierte Werte. Räumliche Indexe basieren auf geometrischen Attributen und beschleunigen als räumliche Zugriffsmethode neben exakten Suchen auch räumliche Operationen und Bereichsanfragen. Die Speicherung von häufig gemeinsam angefragten Daten, auch physisch nah in einem Datenblock, wird mit Clusterbildung bezeichnet [Brink06].

Der Vorteil eines Index liegt in der Verringerung der zur Bereitstellung eines Datensatzes notwendigen Zugriffe auf die Datenablageblöcke eines Hintergrundspeichermediums. Die Zugriffszeit (access time) auf einen wahlfreien Block einer Festplatte, als kleinste Speichereinheit bestehend aus einem oder mehreren Sektoren, setzt sich aus der Bewegung des Schreib-/Lesekopfes zu einem Zylinder (seek time), der Rotationszeit des Sektors zum Kopf (latency time) sowie der vernachlässigbar kleinen Lesezeit zusammen und beträgt etwa 9ms [Brink06]. Sie ist damit sehr aufwendig gegenüber einem Hauptspeicherdatenzugriff, sodass zur Leistungssteigerung die Zahl der Plattenzugriffe minimiert werden sollte. Durch Ablage einer Indexstruktur ganz oder teilweise im Hauptspeicher eines Systems können Datenbankblöcke ohne sequentielle Suche laufzeitoptimierend vom Hintergrundspeicher gelesen werden. Nachfolgend werden einige Indexierungsverfahren, die bei der Verwaltung großräumiger Rasterdaten genutzt werden können, vorgestellt. Neben Verfahren, die explizit für räumliche Daten (z.B. Ausdehnungspolygon eines Rasters) geeignet sind (z.B. GIST/R-

Baum), werden auch kurz Verfahren zur Indexierung von Text bzw. numerischen Attributen (z.B. Hash/B-Tree), verwendet bei den wichtigen Metadatenabfragen einer Rasterverwaltung, beschrieben.

Sehr weit verbreitet sind Indexe, die unter Zurhilfenahme einer Funktion einen Schlüsselwertebereich auf einen Adressraum abbilden können (*Hash-Verfahren*). Aus dem Indexschlüssel kann direkt die Speicheradresse eines Datenblockes berechnet werden. Hashing eignet sich für nur wenig veränderliche Daten, da bei einem Datenblocküberlauf eine Umspeicherung in andere Blöcke erfolgt sowie ggf. eine Bearbeitung durch verkettete Überlaufblöcke angeschlossen werden muss. Dies erhöht je Überlaufblock die Plattenzugriffe bei der Suche von Datensätzen dieser Blöcke. Möchte man die Blocküberläufe verringern, so führt die Speicherung von weniger Datensätzen je Block zu einer schlechteren Speicherplatzausnutzung. Dynamische Hash-Verfahren können einen variabel großen Adressraum ohne Überlaufblöcke in Abhängigkeit von Anzahl und Größe der gespeicherten Datensätze verwalten. Ein Beispiel dafür ist das Lineare Hashing, welches zeitgleich 2 Hashfunktionen nutzt [Brink06]. Für die Indexierung von Geometriewerten ist Hashing nicht geeignet. Die Suche nach Gebietsbezeichnungen von DGM-Kacheln hingegen kann mit einem Hash-Verfahren stark beschleunigt werden. Besonders gut wird die exakte Suche nach einzelnen Datensätzen, basierend auf einem eindeutigen Attribut, unterstützt (Abfrage nach Gleichheit) [PG06].

Jedes DBMS ermöglicht heute die Erstellung des universellen *B-Baum-Index*. B-Bäume minimieren die Anzahl der Zugriffe auf ein Hintergrundspeichermedium durch Anpassung der Indexstruktur an die Blockgröße des Mediums mit Hilfe des wählbaren minimalen Verzeigungsgrades m. Ein Knoten benötigt zur Speicherung gerade eine Blockgröße des Speichermediums. Je Knoten kann eine variable Anzahl Schlüssel (m-1 bis 2*m-1) sowie eine variable Anzahl von Verweisen auf Kindknoten (m bis 2*m) gespeichert werten. Durch einen großen Verzweigungsgrad reduziert sich die Baumhöhe h, die den maximalen Suchweg zu einem Schlüssel bestimmt. Der Baum besteht aus inneren Knoten mit s Schlüsseln und s+1 Verweisen auf Nachfahren sowie aus Blattknoten, die alle auf der Tiefe h des Baumes liegen. Die Baumwurzel eines nicht leeren Baumes speichert 1 bis 2*m-1 Schlüssel sowie ggf. 2 bis 2*m Verweise auf Kindknoten. Alle Schlüssel sind aufsteigend sortiert. B-Bäume sind balanciert und besitzen bei insgesamt n Schlüsseln die Höhe $h \leq \log_m(\frac{n+1}{2})$. Die Haltung der oberen Baumebenen im Hauptspeicher reduziert die zeitintensiven Speicherzugriffe nochmals erheblich. Die Suche nach einem Schlüssel innerhalb des Baumes liefert den Knoten sowie die Position innerhalb des Knotens, sofern der Schlüssel vorhanden ist. Beim Einfügen von

Schlüsseln ist die maximale Schlüsselanzahl je Knoten (2*m-1) zu beachten und der Knoten ggf. zu teilen. Eine Schlüssellöschung kann beim Unterschreiten der minimalen Knotenfüllung (m-1) zur Verschiebung eines Schlüssels des Vorgängers oder Nachfolgers oder zur Verschmelzung zweier Knoten führen [Brink06]. Varianten des B-Baumes sind der B*-Baum mit einer größeren Mindestfüllung je Knoten und damit einer besseren Speicherausnutzung aber auch häufigeren Verschmelzungs- und Teilungsoperationen sowie der B+-Baum, der Datenelemente nur in den Blattknoten enthält, beispielsweise um Bereichssuchen zu optimieren [Pag96].

Räumliche Indexstrukturen dienen der Indexierung von Geodaten. Sie sollen Basisanfragen mit wenigen Plattenzugriffen ermöglichen sowie räumlich benachbarte Daten in einem gemeinsamen Datenblock organisieren (Bildung räumlicher Cluster). Neben einer guten Speicherplatzausnutzung, einer Robustheit gegen Ungleichverteilungen der Geobjekte wird auch dynamisches Einfügen, Löschen und Verändern der Geoobjekte ohne erheblichen Effizienzverlust der Indexstruktur gefordert. Geometrische Datentypen sind im Vergleich zu Zahlen oder Zeichenketten aufgrund ihrer höheren Dimensionalität weit schwieriger zu indexieren. So benötigen B-Bäume und deren Varianten eine lineare Ordnung, in die Geodaten nur schwer einzusortieren sind. Hash-Verfahren könnten nach einer Zerlegung des Raumes in kleine Bereiche mittels einer Hashfunktion die Raumbereiche Datenblöcken zuordnen. Problematisch ist die effiziente Verwaltung ungleichverteilter Daten sowie die Zuordnung der Geoobjekte zu den gebildeten Bereichen [Brink06]. Handelt es sich bei den zu indexierenden Geoobjekten um Punkte, so können herkömmliche Indexstrukturen genutzt werden. Werden Linien, Polygone und komplexere Daten (z.B. Raster) indexiert, so wird häufig das minimal umschließende Rechteck (MUR) des Objekts zur Indexgenerierung verwendet. Es wurden bereits verschiedene Verfahren zur Abbildung von komplexen Geoobjekten in Indexstrukturen entwickelt, von denen einige nachfolgend beschrieben werden.

Beim *Clipping* wird das MUR eines Geoobjekts jeder Blockregion zugeordnet, die es schneidet. Bei einer großen Partitionierung des Datenraumes durch die Speicherung vieler Objekte oder bei großer Variabilität in Lage und Ausdehnung der Objekte steigt die Anzahl der Indexeinträge überproportional an und verursacht damit eine schlechte Speicherplatzausnutzung. Neben dem aufwendigen Einfügen und Löschen von Indexeinträgen wegen der Bearbeitung oft mehrerer Blöcke, ist auch die Duplikatentfernung als ein zusätzlicher Arbeitsschritt einer Anfragebearbeitung notwendig [Brink06].

Mittels *Transformation* können MUR's von 2D-Objekten in 4D-Punkte umgewandelt und dann in jeder dafür geeigneten Indexstruktur abgelegt werden. Die Eckentransformation speichert dazu die Minimal- und Maximalwerte der beiden Dimensionen (xmin, ymin, xmax, ymax). Die Mittentransformation speichert den Mittelpunkt sowie die halbe Ausdehnung des MUR des Geoobjekts (xmp, ymp, xd, yd). Nachteilig ist der Verlust der geometrischen Verhältnisse sowie der räumlichen Nachbarschaft der Objekte, was die Effizienz von räumlichen Abfragen mindert. Da die Objekte meist sehr viel kleiner als der Datenraum sind, führt die Transformation der MUR in höherdimensionale Punkte zu einer extremen Ungleichverteilung dieser Punkte entlang der Ausdehnungsachse (Mittentransformation) oder entlang der Diagonale der X-Achsen (Eckentransformation) mit der Folge einer schlechten Verteilung der Schlüsselpunkte im Index.

Eine weitere Möglichkeit der Indexierung von Geoobjekten beruht auf der *Einbettung in den eindimensionalen Raum.* Ein Datenraum wird durch wechselseitiges Halbieren der Dimensionen unter Angabe der Auflösung (Anzahl der Dimensionswechsel) soweit wie nötig in Zellen zerlegt. Zu diesen Zellen zugeordnete Objekte werden dann durch eine fraktale (raumfüllende) Kurve in eine lineare indexierbare Ordnung gebracht. Beispiele sind die Z-Ordnung (auch Morton-Ordnung) oder die Hilbert-Ordnung.

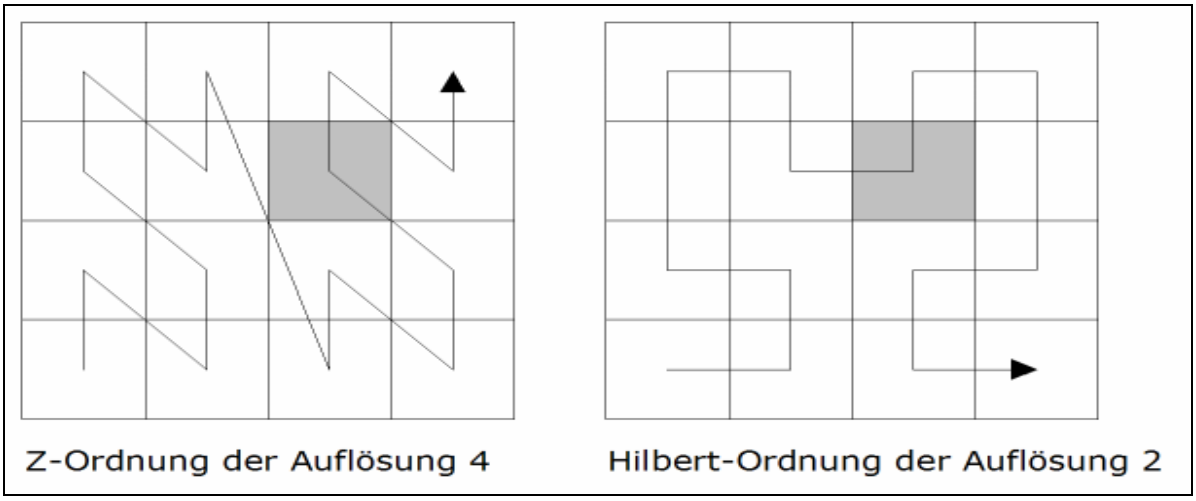


Abb. 15: Z-Ordnung, Z-Wert 12,4, Hilbert-Wert 9,2

Mit Hilfe der Kurve kann eine Ordnungsnummer, beispielsweise der Z-Wert, aus der Binärfolge der durchlaufenden Kurve ermittelt werden und mittels einer geeigneten Indexstruktur (z.B. B+-Baum) indexiert werden. Z-Wert und Auflösung bilden ein Paar, das bei konstanter Auflösung sortiert werden kann und fast vollständig die räumliche Nähe erhält. Die Geoobjekte werden je nach Lage und Ausdehnung einer Einzelzelle oder durch Approximation mehreren Zellen zugeordnet. Problematisch ist dann das Vorhandensein mehrerer Schlüssel je Geoobjekt. Die Hilbert-Ordung besitzt gegenüber der Morton-Ordnung

weniger starke Positionssprünge, ein leicht besseres Leistungsverhalten aber auch eine schwierigere Handhabung bedingt durch den komplexeren Ordnungsnummerberechnungsalgorithmus. Die räumliche Nähe der Geoobjekte wird etwas besser erhalten [Brink06].

Durch die *Verwendung von überlappenden Blockregionen* kann eine Zerschneidung von Geoobjekten vermieden werden. Selektive Anfragen werden erschwert, da mehrere potenzielle Blockregionen durchsucht werden müssen. Dies verringert die Effizienz der Indexstruktur. Die Überlappungen sollten daher möglichst klein gehalten werden.

Mit der *Mehrschichtentechnik* werden durch paralleles Führen unterschiedlicher Raumpartitionierungen die Geoobjekte immer in die Partitionierung eingefügt, die dies ohne Objektzerschneidung erlaubt. Jede Partition für sich ist dabei disjunkt. Da während einer Anfrage alle Partitionen durchsucht werden müssen, sollte ihre Anzahl minimiert werden.

Technik	**keine Redundanz**	**räumliche Ordnungserhaltung**	**keine Überlappungen**
Clipping	-	+	+
Transformation	+	-	+
Einbettung	-	o	+
Überlappung	+	+	-
Mehrschichtentechnik	+	+	-

Tab. 11: Vergleich Geoindexierungstechniken, Quelle: [Brink06] (+ Vorteil, - Nachteil)

Ein *R-Baum* ist eine räumliche Indexstruktur, die zwei- oder mehrdimensionale Rechtecke mit überlappenden Blockregionen indexiert. Er bildet einen balancierten Baum bestehend aus Directoryknoten und zur Wurzel gleichabständigen Datenknoten (Blattknoten). Die Datenknoten enthalten neben dem Datensatz auch dessen MUR. Ein Directoryknoten enthält einen Verweis auf die direkten Nachfahren sowie das MUR aller Rechtecke der nachfolgenden Sohn-Knoten als geometrischen Schlüssel für Such- und Einfüge- bzw. Löschoperationen. Ein R-Baum ist eine vollständig dynamische Struktur mit garantierter Speicherplatzausnutzung durch Regeln der Minimalfüllung und Kapazität einen Knotens und optimierter Höhe. Operationen auf dem R-Baum beginnen mit der Suche des benötigten Datenknotens anhand der MUR-Direktory-Schlüssel. Führt eine Datensatzlöschung zu einer Datenknotenunterfüllung, so werden alle Datensätze des Knotens entfernt und neu eingefügt. Kommt es zu einem Knotenüberlauf, so wird ein Split der Datensätze so durchgeführt, das jeweils die Minimalanzahl an Datensätzen enthalten bleibt. Möglich ist dabei der lineare Splitalgorithmus, bei dem als Kristallisationspunkte die MUR der Geoobjekte mit maximalem normalisiertem Abstand ausgewählt werden. Die restlichen Datensätze werden zu den Gruppen so zugeordnet, dass jeweils der geringste Flächenzuwachs des Gesamt-MUR zu

verzeichnen ist. Beim quadratischen Splitalgorithmus werden die MUR der beteiligten Objekte für jede Paarung von Geoobjekten vom Paar-MUR abgezogen. Die Objekte mit der so ermittelten größten Fläche bilden hier den Anfangspunkt für die Aufteilung. Weitere Optimierungen des als Splitheuristik bezeichneten Verfahrens finden sich in der Arbeit von Pagels [Pag96]. Eine Variante des R-Baums mit veränderter Splitheuristik auf Basis der Berechnung von Umfangssummen sowie verbesserten Organisationskriterien bildet der R*-Baum. Die Minimierung der Überlappung der Blockregionen verringert die Plattenzugriffe. Die Minimierung des Umfanges der Regionen unterstützt quadratische Regionen und die Maximierung der Speicherplatzausnutzung führt zu weniger benötigten Blöcken bei Anfragen. Der R*-Baum beinhaltet die Reorganisation des Baumes durch Löschen und Wiedereinfügen von Objekten zur Optimierung der obigen Kriterien und zur größeren Robustheit gegenüber dem Entarten der Baumstruktur. Nachteilig ist der sehr hohe Rechenzeitaufwand [Brink06]. Der R+-Baum verzichtet durch Clipping auf überlappende Blockregionen in den Directoryknoten. Die Splitheuristik beruht auf einer minimalen Kostenfunktion bei der Wahl der Partitionierungslinie. Nachteilig sind die Entstehung von möglicherweise nur gering gefüllten Knoten sowie die Veränderung von großen Teilen des Index bei Knotenanzahlveränderungen aufgrund von Operationen auf dem Index (Knotenverschmelzung oder Knotensplit). Der Hilbert-R-Baum verwendet als Seperator eine Hilbert-Kurve. Es ist eine Mischung aus R- und B+-Baum.

Bei einer *Quadtree-Struktur* wird der Datenraum rekursiv in disjunkte gleichgroße Rasterzellen zerlegt, bis die gewünschte Feinheit (Auflösung) erreicht ist. Jede neue Zerlegung einer Zelle bringt bis zu 4 neue Tochterzellen hervor. Die Organisation erfolgt in einer Baumstruktur, deren Wurzel dem quadratischen Datenraum entspricht. Die Rekursion endet jeweils in einem Blattknoten, der entsprechend seiner Kapazität mehrere Geoobjekte speichern kann (Bucket Quadtree). Die inneren Knoten beinhalten, sofern benötigt, die maximal 4 Verweise auf die nachfolgenden Knoten bzw. auf Blattknoten. Bei der datenraumbezogenen Indexierung mit einem Quadtree wird der gesamte Datenraum durch Quadtreezellen überdeckt. Je nach Eigenschaften und Verteilung der zu indexierenden Geoobjekte können dabei die Blattknoten sehr unterschiedlich stark gefüllt sein und auch die Zellauflösung (Baumtiefe) sehr stark variieren. Der datenbezogene Quadtreeansatz benutzt eine Approximation der Geoobjekte durch Quadtreezellen. Möglich sind dabei neben konservativen einelementigen Approximationsformen (Beispiel: MUR) auch Approximationen mit mehreren Elementen einheitlicher Zellgröße (z.B. Oracle - Fixed Indexing) oder unterschiedlicher Zellgrößen (z.B. Oracle - Hybrid Indexing). Verwaltet wird

auch hier mit Hilfe einer Baumstruktur. Die Schlüssel zu den Geoobjekten werden mit einer geeigneten fraktalen Kurve (Z-Ordnung, Hilbert-Ordnung) erzeugt [Brink06].

Das *Gridfile* ist eine aus Gitterzellen aufgebaute Indexstruktur zur Speicherung von mehrdimensionalen Punkten, bestehend aus einem Directory zur Verwaltung der Verweise auf Datenblöcke. Ein Datenblock (Gridbag) kann mehrere benachbarte Geoobjekte enthalten. Durch Zerteilung des Datenraumes in ungleichgroße disjunkte Rechtecke (Gitterzelle) kann nach Ermittlung des Directoryblockes anhand der Zellskalen, dem Lesen des Directoryblockes sowie dem Lesen des dort referenzierten Datenblockes die Anzahl der Plattenzugriffe zum Zugriff auf ein Geoobjekt minimiert werden. Zur Verbesserung der Speicherplatzausnutzung können mehrere Gitterzellen auf einen Datenblock verweisen. Diese Zellen werden in einer Blockregion zusammengefasst. Für die weitere Datenraumaufteilung bei einem Datenblocküberlauf wird eine Zerteilung einer Gitterzelle durch die Aktivierung einer Partitionierungslinie einer Blockregion durchgeführt. Wird eine neue Partitionierungslinie benötigt, so existieren Optimierungskriterien für die Wahl von Splitachse (z.B. Anpassung der Form der Blockregion an Anfrageform) und Splitposition (z.B. Split der längsten Seite, Mitten-, Mediansplit). Beim Löschen von Objekten können Blockregionen mit Nachbarregionen verschmolzen werden. Auch hierfür wurden unterschiedliche Strategien entwickelt (Nachbarsystem, Buddysystem) [Brink06]. Zur Zuordnung der Geoobjekte zu den Gitterzellen können die oben beschriebenen Möglichkeiten genutzt werden (Transformation, Clipping). Das Griddirectory kann als 2-Level-Gridfile mit einem im Hauptspeicher gehaltenen Wurzelgridfile sowie disjunkten Subgridfiles mit eigener Datenraumzerlegung, die jeweils einer Blockregion des Wurzelgrids entsprechen, organisiert werden. Eine Anpassung an lokale Häufungspunkte und Ungleichverteilungen ist damit möglich. Verwendet werden kann auch ein mehrstufiges Directory in Form einer Baustruktur, ein mit BANG-File bezeichnetes System nichtrecheckiger Blockregionen oder auch das Buddy-Tree-Verfahren. Hier liegen die Kanten der Gitterzellen auf einem vorgegebenen Raster, eine vollständige Datenraumabdeckung ist nicht notwendig. Der Einsatz eines mehrdimensionalen Hash-Verfahrens (z.B. PLOP-Hashing) ermöglicht auch den Verzicht auf die Directorystruktur [Brink06].

Die hauptsächlich verwendeten Indexierungsverfahren wurden hier etwas ausführlicher beschrieben, um ihre Wichtigkeit bei der Verwaltung von Objekten in einem DBMS zu unterlegen. Für die Rasterverwaltung können Indexe für räumliche Abfragen von Rasterdatensätzen, zur Auswahl von Teilen eines Rasters (Block-Zugriff), für den Zellzugriff (Array, BLOB) und für die Beschleunigung von Metadatenanfragen verwendet werden.

Besonderes Merkmal von Rasterdaten ist ihr sehr umfangreiches Datenvolumen, auf das mit guter Performance nur mit Hilfe von angepassten Indexen zugriffen werden kann.

7.2.1 Räumliche Indexstrukturen in Oracle und PostgreSQL/PostGIS

In *Oracle Spatial* können Geoobjekte, die nicht in einem geographischen Koordinatensystem vorliegen, mittels eines Quadtree-Index verwaltet werden. Als Zellapproximation kann zwischen einheitlicher Zellgröße (Fixed Indexing) und unterschiedlicher Größe (Hybrid Indexing) gewählt werden. Der ebenfalls erzeugbare R-Baum kann für kartesische (projizierte) und für geographische Koordinatensysteme genutzt werden. Daten- und Directoryknoten können hier zur weiteren Optimierung der Plattenzugriffe in getrennten Indextabellen gespeichert werden. Bei großen Indexen kann dann die Directorytabelle im Hauptspeicher gehalten werden [Ora2], [Kot04].

Relativ wenige Indexstrukturen fanden aufgrund der hoch komplexen und kostenintensiven Entwicklung der Struktur und ihrer Integration in die Datenbank eine große industrielle Akzeptanz und Verbreitung [Korn00]. Da die Integration eines Index einen starken Eingriff ins System benötigt und dies die Weiterentwicklung behinderte, wurde durch die Datenbankhersteller basierend auf der Arbeit von J. Hellerstein (1995) [HNP95] mit dem GIST (generalized search tree) eine Struktur bereitgestellt, die die Implementierung von Indexen für spezielle Einsatzzwecke einfacher ermöglicht. Der GIST besteht aus einem balancierten Baum, der allgemeine Algorithmen zur Navigation innerhalb des Baumes und zur Änderung seiner Struktur in Form von Vorlagen bereitstellt. Die Anbindung des GIST an die Speicherverwaltung auf der einen Seite und die Anfragebearbeitung auf der anderen Seite übernimmt der Datenbankhersteller. Da die GIST-Struktur weniger Einschränkungen bzgl. Schlüsselordnung, Dimensionalität der Daten sowie Disjunktheit bzw. Abdeckung des Datenraumes macht, sind verschiedene Indexmethoden als GIST-Erweiterung mit weniger Programmieraufwand implementierbar. Dies ermöglicht die Optimierung von Indexstrukturen beispielsweise für räumliche Datentypen. Der Entwickler hinterlegt lediglich die Semantik des Datentyps sowie ggf. spezielle Operationen auf die Indexstruktur, die sich von den bereitgestellten Methoden des GIST unterscheiden.

Auch in *PostgreSQL/PostGIS* wird die GIST-Struktur vom DBMS bereitgestellt. Eine implementierte Zugriffsmethode innerhalb der GIST-Vorlage ist ein auf räumliche Datenformate optimierter R-Tree. Aufgrund der Objektrelationalität der Datenbank können auch eigene Indexmethoden basierend auf eigenen Datentypen (ADT) unter Nutzung der GIST-Hülle entwickelt werden. Weiterhin bietet PostgreSQL neben einem B-Tree auch Hashing und einen für räumliche Daten geeigneten R-Tree an, der aber Einschränkungen

gegenüber R-Tree-over-GIST bezüglich der Größe der räumlichen Objekte (8k) sowie der Verwendung von Nullwerten macht [PG06]. Für nichträumliche Daten bieten Hash und B-Tree etwa gleich gute Ergebnisse, sodass wegen der schnelleren und flexibleren Indexerstellung hier ein B-Tree zu empfehlen ist [Eis03].

7.2.2 Indexauswahl und Verwendung

Schlecht entworfene oder fehlende Indexe sind eine Hauptquelle für Engpässe im Datenzugriff bei einer Datenbankanwendung [Micro]. Da das Löschen, Verändern oder Einfügen eines Index keinen Einfluss auf das Datenbankschema und den Anwendungsentwurf hat, sondern lediglich die Ausführungszeit von Abfragen verringert, sollte zur Auswahl der benötigten Indexe unter Beachtung von Datenbankauslastung, gewünschter Abfragegeschwindigkeit sowie den benötigten Kosten für die Indexerstellung und Wartung mit allen in der Implementierung verfügbaren Indexen experimentiert werden. Je mehr Indexe dem Abfrageoptimierer bereitgestellt werden, umso mehr Auswahlmöglichkeiten hat dieser zur Wahl der effektivsten Strategie. Zum Entwurf der nötigen Indexe sollte die Häufigkeit der Veränderung von Datensätzen in der Datenbank beachtet werden und sehr veränderliche kleine Relationen nicht indexiert werden (hoher Indexwartungsaufwand, meist schneller Full-Scan). Viele Indexe hingegen können für Tabellen mit geringen Aktualisierungsanforderungen und großen Datenmengen sowie für Sichten bestehend aus aggregierenden oder verknüpfenden Operatoren angelegt werden. Bezogen auf das DGM-Projekt wäre hier die Indizierung von ggf. zu erstellenden Sichten für Küstenschutzvorranggebiete (Kachelsammlungen) zu empfehlen.

Indiziert werden sollten die Spalten der am häufigsten verwendeten Abfragen. Dazu sind Abschätzungen der späteren Nutzung bzw. Indexnachführungen nach einer gewissen Nutzungsdauer durchzuführen. Jeweils für Selektions-, Verknüpfungs-, Sortierungs- und Aggregationsspalten sind entsprechende Indexe vorzusehen. Nicht benötigte Spalten sind zur Verringerung des Wartungsaufwandes nicht zu indexieren. In der derzeitigen Ausbaustufe des Projektes 'DGM Küstenbereich' sind dazu neben der Geometriespalte (Umring), der eindeutigen Namensspalte ggf. auch eine Spalte mit dem Aktualisierungsdatum o.ä. in jeweils einem Index zu verarbeiten.

In Abhängigkeit vom verwendeten Datentyp einer Spalte ist ein dazu optimierter Index auszuwählen. Generell sind mögliche Eindeutigkeit der Schlüssel, die Datenverteilung sowie die Benötigung einer Sortierung zu prüfen und bei der Indexanlage zu spezifizieren. Für die Datums- und Namensspalte kann daher ein B-Baum-Index, für die Geometriespalte ein R-Baum-over-GIST-Index (siehe PostgeSQL-GIST-Indexe) empfohlen werden.

Sofern Indexoptionen bei der Erstellung wählbar sind, können diese zu einer weiteren Leistungssteigerung beitragen. Letztlich ist auch der Speicherort des Index sowie etwaige Relations- und Indexpartitionierungen zur Optimierung des Sekundärspeicher-Zugriffs (siehe RAID-System) zu beachten.

7.2.3 Indexverwendung am Beispiel

Am Beispiel des PostgreSQL/PostGIS-Systems (Version 8.2) soll die Indexverwendung eines Datenbankmanagementsystems an Beispielen untersucht werden. Andere DBMS bieten ähnliche Möglichkeiten mit etwas veränderter Befehlsstruktur bzw. Nutzeroberfläche. Generell wird bei der Durchführung von Performancetests die Nutzung von Beispieldaten sehr kritisch gesehen [Behl02]. Bei diesem Test zur Indexoptimierung wurden durch eine Skriptsprache (SAX-Basic) 500000 Datensätze disjunkter quadratischer Kacheln (500x500m) mit realistischen Koordinaten als Well-Known-Text in SQL erzeugt und in die Datenbank eingelesen (siehe Anlage A.1). Auf die Bereitstellung von realen Geländemodelldaten für diese Kacheln als BLOB wurde verzichtet, da lediglich die Anfrageoptimierung auf Basis der MUR-Geometrie untersucht werden sollte. Die Vorteile einer Indexverwendung für die Metadatensuche werden durch Abfragen nach der Kachelnamensspalte aufgezeigt. Die Datenbank und auch der abfragende Client sind auf einem Notebook (1.73GHz Prozessor, 512Mb RAM) installiert worden. Zur Untersuchung der Indexverwendung wurden verschiedene Anfragen auf diesen Datenbestand durchgeführt.

Durch Ausschalten bestimmter Plantypen (z.B. sequentieller Scan) kann der Optimierer zur Verendung eines Index (z.B. Index-Scan) gezwungen werden. Auch die Veränderung der Aufwandswerte des PostgreSQL-Systems für die Kalkulation der Gesamtkosten kann eine Änderung des ausgewählten Planes bewirken. Die Gesamtkosten werden aus den Kosten je Zeile eines Planknotens (cpu_tuple_cost) multipliziert mit der Anzahl der zu durchlaufenden Zeilen je Knoten gebildet. Hinzu kommt der Aufwand zum Lesen einer Datenbankseite (seq_page_cost) multipliziert mit der Anzahl verwendeter Seiten. Gegebenenfalls sind Kalkulationen über die Verwendung von Indexen (cpu_index_tuple_cost) oder durch den Aufwand von Operationen (cpu_operator_cost) durchzuführen. Der SQL-Befehl ANALYZE erstellt Statistiken über die Werteverteilung innerhalb einer Relation, mit deren Hilfe realistische Kostenschätzungen vom System durchgeführt werden können. Mittels des SQL-Befehls EXPLAIN lassen sich der Ausführungsplan sowie die geschätzten Kosten sichtbar machen. Wiedergegeben werden die geschätzten Startkosten, welche die Zeit bis zum Beginn der Anfragenausführung, beispielsweise zum Sortieren von Datensätzen, bewerten. Die geschätzten Gesamtkosten zur Rückgabe aller Ergebniszeilen, sowie die geschätzte Anzahl

der Zeilen eines Planknotens bei vollständiger Ausführung und die Breite (Bytes) einer Datenzeile des Knotens werden ebenfalls ausgegeben. Ein übergeordneter Knoten enthält alle Kosten der Unterknoten (siehe auch Anlage A.4).

Beispiel: Dieser Index-Scan setzt sich aus 2 Unterabfragen zusammen.

```
Bitmap Heap Scan on indextest (cost=1507.95..13443.45 rows=54895 width=136) (actual time=35.363..236.128 rows=99799 loops=1)
  Recheck Cond: ((name < '4400500_6054000'::text) OR (name > '4800500_6154000'::text))
  -> BitmapOr (cost=1507.95..1507.95 rows=54900 width=0) (actual time=34.729..34.729 rows=0 loops=1)
     -> Bitmap Index Scan on indextest_btree (cost=0.00..4.47 rows=50 width=0) (actual time=0.058..0.058 rows=107 loops=1)
           Index Cond: (name < '4400500_6054000'::text)
     -> Bitmap Index Scan on indextest_btree (cost=0.00..1503.47 rows=54850 width=0) (actual time=34.661..34.661 rows=99692 loops=1)
           Index Cond: (name > '4800500_6154000'::text)
Total runtime: 419.212 ms
```

Die Einheit ist dabei Fetch (Aufwand zum Lesen einer Diskseite), Prozessorzeiten werden auf diese Einheit umgerechnet. Angegeben werden nur die durch die Planung beeinflussbaren Kosten. Die bei der Betrachtung von großvolumigen Rasterdaten anfallenden sehr langen Transferzeiten vom Server zum Client abhängig von der Netzwerkkapazität, werden nicht beachtet. Die Schätzungen beruhen auf der Berechnung einer Datensatzauswahl (Sample). Beim Durchführen einer erneuten Analyse (ANALYZE) zeigen sich durch die veränderte Auswahl auch geringfügig geänderte Aufwandsschätzungen.
EXPLAIN ANALYZE zeigt die geschätzten Kosten in Fetch sowie die nicht direkt vergleichbare gemessene Laufzeit in Millisekunden. Optimierungen können nur durch qualitative Vergleiche der Angaben überprüft werden. Leider sind die Ergebnisse kaum repräsentativ und übertragbar, da neben hardwareseitigen Möglichkeiten (Prozessor, Primärspeicher) auch die Tabellengröße selbst durch die nichtlinearen Kostenschätzungen die Planauswahl und damit die Anfragezeiten stark beeinflusst. [EIS03]

Für die Untersuchung wurden beispielhaft vorstellbare Anfrageszenarien nachgestellt. Die Möglichkeit des Mehrbenutzerbetriebes wird dabei vernachlässigt, da durch die geringe erwartete Frequentierung der Nutzung des Projektes diese Parallelität kaum benötigt werden dürfte. Die Transportzeiten zum Client wurden trotz ihres großen Einflusses auf die Gesamtlaufzeit aufgrund der kaum vorhersehbaren Konstellationen von Client-, Server- und

Netzwerkkommunikation ebenfalls vernachlässigt. Nachfolgende Anfragetypen wurden untersucht (Indexerstellung und Abfragen siehe Anlagen A.2/A.3):

1. Anfragen nach Metadaten - Szenarien:
Für die Zusammenstellung eines Geländemodells wird eine Kachel eines bestimmten Namens benötigt (1A). Für die Darstellung einer Karte eines Gebietes wird ein Kachelstreifen mit vorgegebenen Namensgrenzen gesucht (1B). Für eine Untersuchung werden alle Kacheln eines Hochwertbereiches benötigt (1C). Sind vielfältige Metadaten erfasst worden, so können beispielsweise auch Abfragen nach einem Zeitraum oder Zeitpunkt der Kachelaktualisierung, nach abgelegten Geländeparametern (Extremwerte, Textur) sowie Gebietszugehörigkeit (z.B. Amtsbereich) o.ä. durchgeführt werden.

2. Anfragen aufgrund geometrischer Verschneidungsoperationen - Szenarien:
Für die Standortanalyse einer Windkraftanlage (Punkt) wird die diesen Punkt enthaltende Kachel gesucht (2A) oder es werden alle Kacheln im Abstand bis 5km um diese Anlage benötigt (2B1/2 - 2 Versionen der Abfrage). Ein vorliegendes Untersuchungsgebiet soll durch die von ihm überlappenden Kacheln abgedeckt werden (2C). Ein Visualisierungs-Client fordert eine Rasterkarte (Kachelzusammenstellung) basierend auf einem Anzeigefenster (BOX) an (2D).

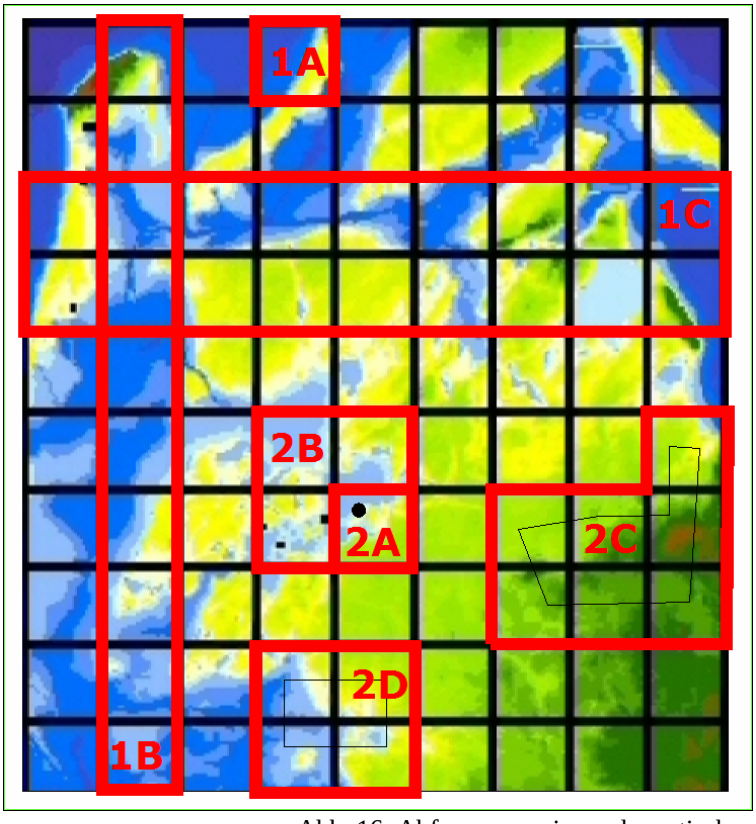


Abb. 16: Abfrageszenarien, schematisch

Ergebnisse (Abfragen in Anlage A.3):

Abfrage	Anzahl Ergebniszeilen	ohne Index (seq Scan) fetch	ohne Index (seq Scan) Laufzeit ms	mit Index B-Tree fetch	mit Index B-Tree ms	mit Index Hash fetch	mit Index Hash Laufzeit ms	mit Index GIST fetch	mit Index GIST Laufzeit ms	Index-Gewinn max(Faktor)
1A	1	17362	1068.1	8.12	0.2	8.2	0.1			5340-10681
1B	19999	18612	1319.6	12326.8	206.9	18612	531.3			6.3-2.5
1C	10000	17362	665.8	11787.9	45.3	17362	389.9			14.7-1.7
2A	1	18612	14130.6					18612	1094.5	12.9
2B1	9	18612	12243.9					18612	1196.7	10.2
2B2	9	17362	387.1					8.16	0.2	1935.5
2C	3087	17362	401.3					6829	34.1	11.7
2D	3116	17362	412.1					6998.3	32.9	12.5

Tab. 12: Ergebnisse Indexvergleich

Anhand der Messergebnisse ist sehr gut die vorteilhafte Verwendung von Indexen zu erkennen. Leistungssteigerungen (Indexgewinn) vom Faktor 2 bis in den kaum messbaren Bereich (Anfrage 1A) sind zu verzeichnen. Da die in PostgreSQL implementierten Indexe sehr spezialisiert für bestimmte Datentypen vorliegen und auch die Anzahl der zur Verfügung stehenden Indexe derzeit noch gering ist, kann eine Reihenfolge bzgl. der Performance-Unterschiede von Indexen nicht getroffen werden. Auch aufgrund der ungenau bekannten zukünftigen Anfragenszenarien und der problematischen Verwendung von Beispieldaten wird deshalb hier nur eine qualitative Aussage gemacht. Der interne Datenbankoptimierer wählt, sofern für die Anfragezeit positiv, einen Index aus oder führt alternativ einen sequentiellen Scan durch. Unpassende Indexe, die ggf. die Laufzeit verschlechtern würden, werden nicht benutzt. Es wird ebenfalls kein Index benutzt, wenn die Kostenkalkulation basierend auf cpu_tuple_cost, cpu_index_tuple_cost und cpu_operator_cost sowie seq_page_cost beispielsweise bei sehr wenigen Ergebniszeilen die Indexverwendung als nicht lohnend kennzeichnet.

Die Ablage des Index in verschiedenen Tablespaces (Speicherort) brachte im Einzelnutzerbetrieb keine nachweisbaren Laufzeitgewinne. Bei der Erstellung des Geometrie-Index GIST (R-Tree) wird eine deutlich höhere Laufzeit zur Indexierung als bei den Indextypen B-Baum und Hash benötigt. Eine Indexgenerierung wird deshalb erst nach Abschluss von Datenänderungsarbeiten empfohlen.

Größere Auswirkungen einer Indexnutzung sind bei stärker partitionierten Rasterdaten zu erwarten. Werden im nachfolgend ausführlicher beschriebenen Projekt die Rasterkacheln verkleinert und damit bei gleicher DGM-Modellgebietsgröße ihre Anzahl erhöht, so ist eine feinere Gebietsauswahl und -zusammenstellung möglich (Optimierung des Datenaustauschvolumens). Aufgrund des sich stark vergrößernden Speicherbedarfes (Metadaten, GeoTIFF-Header-Daten) je Kachel, einer Mindestkachelgröße zur erfolgreichen Kompression und der zu erwartenden Projekterweiterung (Erhöhung der Kachelanzahl aufgrund einer Kachelversionierung) wurde mit der Kachelgröße 500x500m (derzeit 25850 Stck.) ein Kompromiss gefunden.

8 Projekt 'DGM Küstenbereich'

8.1 Projektbeschreibung

Die nachfolgende Skizierung einer Rasterdatenverwaltung auf einem PostgreSQL/PostGIS-System ohne spezifische Rasterdatenfunktionalität basiert auf den Erfahrungen und Ergebnissen bei der Erstellung des 'DGM Küstenbereich', eines Verwaltungssystems zur Bereitstellung von hochgenauen Geländemodelldaten für Untersuchungen zur Bewertung von Überflutungsgefährdungen. Ziel ist die Integration unterschiedlicher Datenquellen sowie von laufenden Aktualisierungen in ein rasterbasiertes schnelles Datenhaltungssystem.

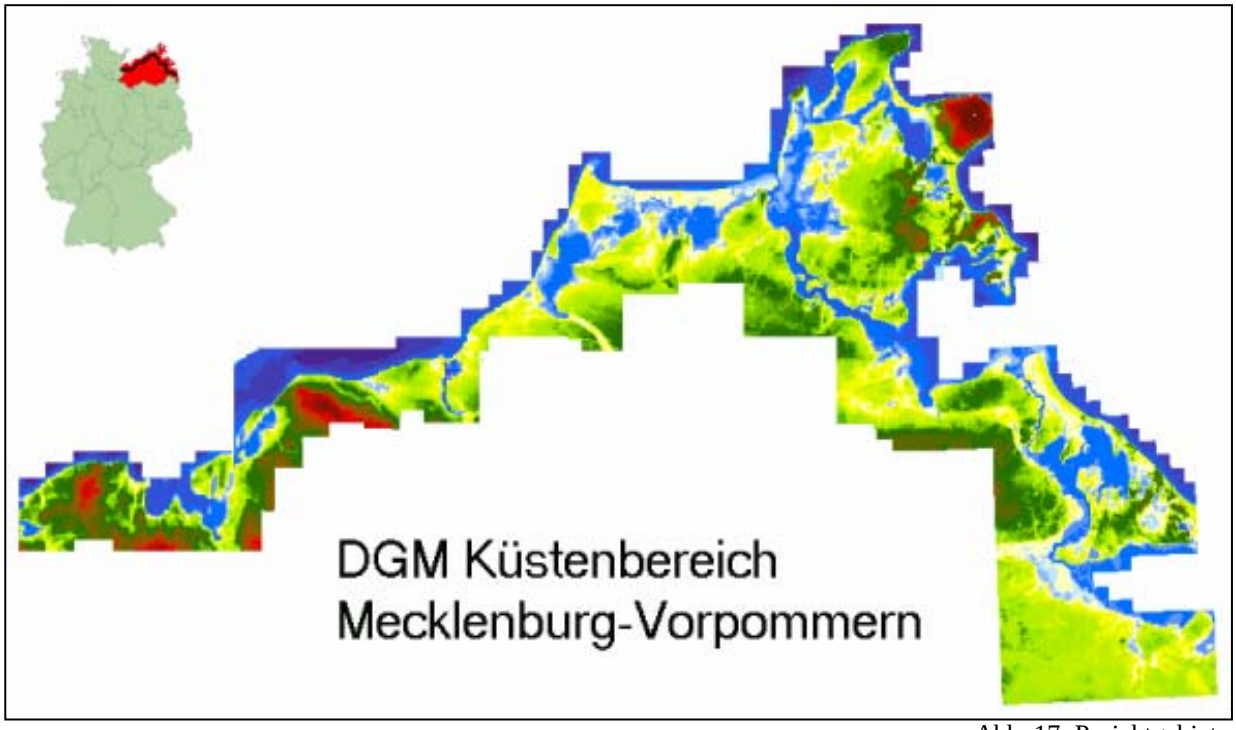


Abb. 17: Projektgebiet

Das Betrachtungsgebiet des Projektes, dass sich entlang der Ostseeküste Mecklenburg-Vorpommerns / Deutschland in den überflutungsgefährdeten Zonen dessen Bodden- und Ausgleichsküste erstreckt, umfasst ca. 6500km². Mecklenburg-Vorpommern besitzt etwa 381km Außenküste bestehend aus 136km Steilküste und 245km Flachküste. Hinzu kommen 1562km Innenküste (Boddenküste). 70% der Außenküste unterliegt aufgrund natürlicher Prozesse einem mittleren Küstenrückgang [Traub06]. Die Beobachtung und Untersuchung dieser Küstendynamikprozesse sowie die Bewertung von Hochwassergefährdung und der Schutz des Küstenbereichs sind behördliche Aufgaben, zu dessen Hilfe die Modelldaten dienen sollen. Eingangsdaten waren zum Zeitpunkt der ersten Modellerstellung aus sehr unterschiedlichen Quellen mit sehr unterschiedlicher Auflösung und Aktualität vorhanden.

Datenquelle	Flächenanteil 2006 (geschätzt)	Datenart	Struktur-informationen	Erstellungszeitraum
topographische Karte M. 1:10000	40%	Vektor - Höhenlinien	Bruchkanten, Steilufer	1998-2002
Küstengewässervermessung	35%	Vektor - Einzelpunkte	nein	1998-2003, laufend
Uferlinie - Vermessung	0.01%	Vektor - Linie	nein	1998-2006, laufend
Bauwerks/Gebiets-Vermessung	10%	Vektor - Linien, Punkte	Bruchkanten	1998-2006, laufend
Laserscanning	15%	Raster	nein	2003-2006, laufend

Tab. 13: Datenquellen des Projektes

Die Anforderungen an das System ergaben sich aus der Verwaltung der bisher verwendeten kleinräumigen, unzusammenhängenden und unterschiedlich modellierten Geländedaten auf Dateibasis in einer raumbezogenen Ordnungsstruktur. Zukünftig sollte ein flexibles und skalierbares Datenbanksystem die Verwaltung übernehmen, um bestehende Datenredundanzen zur verringern und eine gemeinsame Verwaltung von Quelldaten, zugehörigen Metadaten und Modelldaten sowie einer dienstorientierten Infrastruktur zur Verwendung der Daten innerhalb von Informationssystemen zu ermöglichen. Bestehende Daten wurden konvertiert, Metadaten werden erfasst und z.T. vervollständigt. Eine schnelle, blattschnittfreie und nutzungsgerechte Extraktion der rasterbasierten Geländemodelldaten für spezielle Anwendungen sollte neben der schrittweisen Integration von zukunftsorientierten Schnittstellen auf Basis von anerkannten Normen und Standards für Raster- und Vektordaten unterstützt und gefördert werden.

8.2 Konzeption des Projekts 'DGM Küstenbereich'

Ziel des Verwaltungssystems ist die Bereitstellung von hochgenauen blattschnittfreien Geländemodelldaten für ein sehr großes Untersuchungsgebiet. Zur Datenhaltung wurde das *Rasterformat* GeoTIFF gewählt, da eine Analyse der zukünftigen Nutzungen und Nutzer die universelle Einsetzbarkeit dieses Formates bewiesen hatte. Es unterstützt sehr vielfältig die beschriebenen Rasterkonzepte (Georeferenzierung, Partitionierung, Kompression) und hat als einziges 'freies' Format eine sehr große Verbreitung erlangt. Sowohl die 'einfache' Nutzung in Bildverarbeitungsprogrammen als auch die Verwendung als Modellgrundlage für komplexe Analysen in verschiedensten GIS-Systemen ist möglich. Ebenso können durch die im Format enthaltenen Metainformationen auch standardisiert Angaben zu Quellen, Verarbeitungsschritten u.ä. dem Nutzer übermittelt werden. Derzeit wird hier lediglich die Unterstützung von Koordinatensystemen als Metatag (EPSG-Code) verwendet. Ein weiterer Vorteil dieses Formates ist die Unterstützung von 32-Bit-Float-Attributwerten. Eine direkte

Ablage von Fließkommazahl-Höhenwerten je Zelle ist damit gegeben. Durch die Verwendung der Open-Source-Bibliothek GDAL (www.gdal.org) können die Rasterdaten außerhalb der Datenbank einfachen automatisierbaren Operationen unterworfen werden. Neben Koordinatentransformation sind auch Merge (Fusion), Mosaiking oder Subset-Operationen sowie die Umwandlung in diverse Formate möglich. Letztlich bietet die Obergrenze von 4GB je TIFF-Datei genügend Spielraum zur strukturierten Datenablage.

Bereits bei den ersten Berechnungen der Geländemodelle wurden die dimensionalen Beschränkungen der Datenhaltung in einem bzw. wenigen Modellen sichtbar. Abhilfe schuf ein *Kachelkonzept*, d.h. das Untersuchungsgebiet wurde in derzeit 25850 je 500x500 Meter große Geländemodellstücke zerlegt. Diese Zerlegung geschieht jeweils erst nach den Modellberechnungen, sodass keine Randeffekte an Kachelgrenzen zu erwarten sind. Bei der gewählten Modellauflösung von 1x1m je Zelle enthält eine Kachel damit 250000 Höheninformationen. Die exakte Lagefestlegung der Zellen führte zu 'geraden' Eckkoordinaten der Kacheln im 500m-Abstand. Anhand der Kachelbezeichnung, zusammengesetzt aus den jeweils ersten 5 Stellen der Rechts- und Hochwertkoordinaten im Landessystem S42/83 3°, lässt sich damit auch die Position einer Kachel im Referenzsystem bestimmen. Zur Modellerzeugung eines Gebietes werden lediglich die benötigten Kacheln ermittelt und zu einem Modell zusammengesetzt. Aufgrund der Menge an Kacheln wurde die Verwaltung der Daten an eine Datenbank übergeben. Wie bÿÿeits in den Konzepten beschrieben, gÿÿt esÿÿur Rasterdatenfürwkomple in einer Datenbank verschiedene Datentypen, von denen, aufgrund der einfachen Im- und Exportfunktionalitäten ohne zusätzlichen Codierungs- bzw. Decodierungsaufwand, das BLOB-Konzept ausgewählt wurde. Nachteil ist hier die fehlende Möglichkeit der Veränderung oder Abfrage von Zellwerten innerhalb der kleinsten Dateneinheit (Kachel) und innerhalb der Datenbank. In einem BLOB wird jeweils ein Kachel-GeoTIFF abgelegt und von der Datenbank verwaltet. Die redundanten Kachel-Metadaten zur Geocodierung und Positionierung sowie Auflösung, abgelegt bereits in der zugrunde liegenden GeoTIFF-Datenquelle, wurden zugunsten einfacherer Schnittstellenoperationen im BLOB belassen. Da derzeit noch keinerlei Funktionalität zum Zugriff auf diese Metadaten innerhalb Datenbank implementiert werden konnte, sind beim Datenimport einzelne Metadaten neu in der Zielrelation angelegt worden.

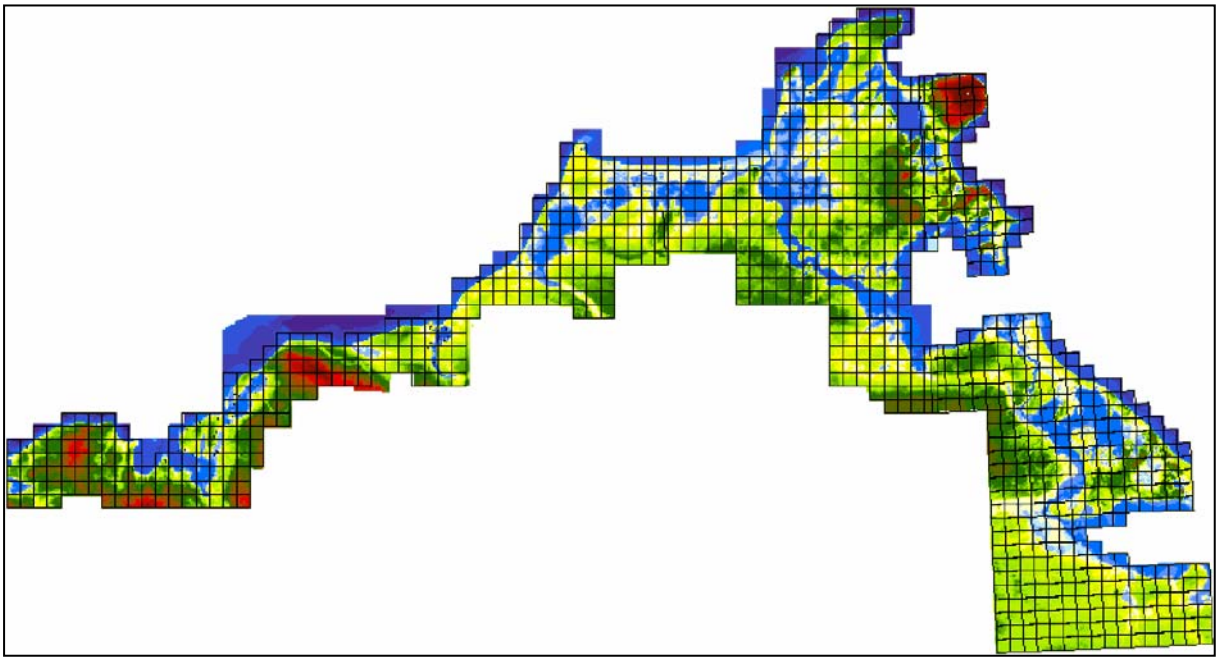


Abb. 18: Kacheleinteilung des Projektgebietes (Zusammenfassung je 25 Kacheln)

Als Datenbanksystem wurde mit PostgreSQL/PostGIS ein objektrelationales modernes Open-Source-System ausgewählt. Neben den umfangreichen Möglichkeiten von zukünftigen Objektklassenerweiterungen (ADT), der relativ einfachen Installation und Administration, den umfangreichen Schnittstellen zu Formaten und Programmiersprachen sowie der Möglichkeit serverseitige Prozeduren ablegen zu können, bildeten auch die gute Dokumentation sowie kurze Entwicklungszyklen entscheidungsrelevante Parameter. Die ebenfalls unter Open-Source-Lizenz stehende Erweiterung PostGIS ermöglicht dem DBMS die Anbindung an die Welt der räumlichen Datenverarbeitung durch Unterstützung einiger OpenGIS-Standards des OGC. Die Verarbeitung räumlicher Datentypen, die Durchführung raumbezogener Operationen, die Möglichkeit räumliche Indexe zu bilden sowie zukünftige Erweiterungen bzgl. Topologie und Modellberechnungen waren Grundlage der Systemauswahl. Rasterdaten werden hier bisher noch nicht durch eine spezielle Datenstruktur direkt unterstützt. Der 'Umweg' über die Ablage von BLOB gekoppelt mit räumlichen und textlichen Attributen in einer Relation ist jedoch ein häufig beschrittener Weg bei der Verwaltung von großvolumigen Rasterdaten. Zur räumlichen Einordnung und Abfrage dient die Anlage eines Polygon-Geometrieattributes mit dem quadratischen Umring einer Kachel.

Da vorerst nur eine Relation benötigt wurde, waren Überlegungen und Optimierungen zu Datentypen und Normalisierung noch nicht notwendig. Bei der späteren Erweiterung des Systems in Richtung Vektordaten (siehe Ausblick) wird dieser konzeptionelle Aspekt noch einmal überarbeitet werden.

Die Befüllung der Datenbank erfolgte mit den im Abschnitt 8.5 beschriebenen SQL-Scripten. Zur Performanceoptimierung wurden für die Suche nach einer bestimmten Kachel aufgrund ihres Namens ein unique-B-Baum sowie für die raumbezogene Suche über das Umring-Geometrie-Feld ein GIST-Index (R-Tree) nach dem Datenimport eingefügt.

8.3 Datenerfassung und -aufbereitung des Projektes

Eine wesentliche Anforderung bei der Konzeption des Systems war die Integration verschiedener vorliegender Datenquellen in ein einheitliches Geländemodell. Neben Vektordaten aus landseitigen und seeseitigen Vermessungen in für die späteren Nutzer wesentlichen Gebieten (überflutungsgefährdeter Küstenbereich, ca. 2Mio. Punkte) aus tachymetrischen und GPS-Vermessungen wurden auch Digitalisierungen der Haupthöhenlinien der topographischen Karten 1:10000 vorgenommen. Bereits in einem Rasterformat vorliegende Lasers-canningdaten (ca. 900km²) wurden nachbearbeitet. Im Seebereich sind etwa 2200km² Seebodendaten unterschiedlichster Herkunft und Auflösung zu verarbeiten gewesen. Laufende Neuerhebungen von Daten führen zur periodischen Änderung der Rasterdaten der Datenbank. Die Einarbeitung dieser Neudaten erfolgt durch Abfrage der betroffenen Kacheln, externer Aktualisierung (ArcGIS) und Wiedereingliederung in den Datenbestand.

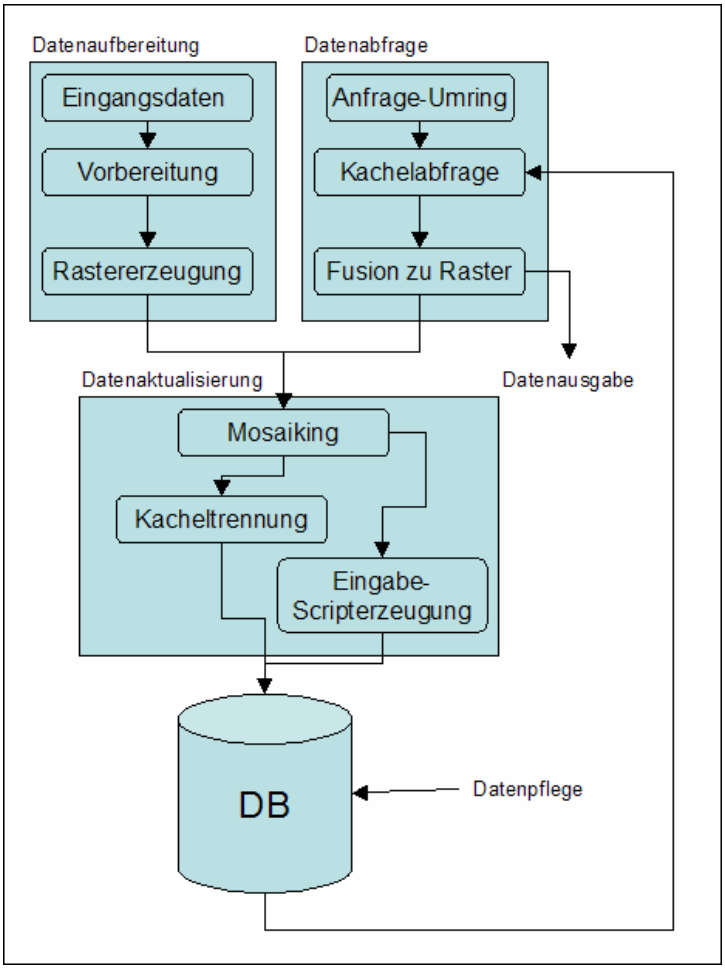


Abb. 19: Workflow Projekt

Zur ersten Datenbankbefüllung wurden für den Gesamtbereich ca. 20 disjunkte Geländemodelle innerhalb von ArcGIS/Spatial Analyst erzeugt, da die Berechnung eines Modells des gesamten Untersuchungsgebietes aufgrund der Ausdehnung und Datenmenge sowie der vorhandenen Soft- und Hardwareausstattung nicht möglich war. Diese

'Startmodelle' setzten sich aus der vorliegenden Digitalisierung der Höhenlinien landseitig sowie Vermessungen des küstennahen Gewässergrundes seeseitig, getrennt durch eine exakt bestimmte Uferlinie zusammen (Abb. 20-1). Die Erstellung dieser Uferlinie an der sich z.T. stark verändernden Küste stellte die Vorraussetzung für die Zusammensetzung der Teilbereiche dar. In den Steiluferabschnitten des Küstenbereichs wurden Unter- und Oberkante bestimmt und als Geländebruchkante genutzt. Näheres zur anfänglichen Vorgehensweise zeigt die Arbeit 'Erstellung, Anwendung und Qualitätsuntersuchung von Digitalen Geländemodellen zur Überflutungssimulation' [Rei01]. Das anschließende Verfahren zur Kacheltrennung wird im Abschnitt 8.5 beschrieben.

Bei der periodischen Einarbeitung von Neudaten wurde je nach Datenquelle angepasst an die spezifischen Eigenschaften der Messdaten verfahren. Die bereits in einem Rasterformat vorliegenden Laserscanningdaten sind nach einer Koordinatentransformation einem Resampling-Prozess unterworfen worden, um die exakte Lage der Rasterzellen zu gewährleisten. Da die Auflösung dieser Datenquelle (2.5 Punkte je Quadratmeter) die des späteren Geländemodells noch übertraf und Datenveränderungen durch Mittelwertbildung nicht erwünscht waren, wurde als Resampling-Methode Nearest-Neighbor gewählt. Landseitige flächige Vermessungsdaten zur Geländemodellverbesserung werden nach baulichen Veränderungen des Geländes (Küstenschutzbauwerke) oder nach Einstufung eines Gebietes als 'überflutungsgefährdet' ausgelöst und bereitgestellt. In einem ersten Verarbeitungsschritt werden aus den tachymetrisch oder mittels GPS gemessenen dreidimensionalen Höhendaten anhand von Feldrissen vorhandene Geländebruchkanten erzeugt. Zusammen mit der bisher noch nicht automatisierbaren Erstellung eines Umringes um die Vermessung zur Begrenzung des räumlichen Einflusses einer Messkampagne auf das Modell bilden Bruchkanten und Geländepunkte die Eingangsdaten für die Berechnung eines Triangulated Irregular Network (TIN) und dessen Umwandlung in ein Grid (Raster) unter Beachtung der exakten Positionierung der Zellen (ArcGIS Spatial Analyst/3D-Analyst).

Ergebnis der Datenaufbereitung ist ein regelmäßiges Raster eines Gebietes im benötigten Koordinatensystem vorliegend als ArcGIS-Floating-Point-Grid, welches im Verarbeitungsschritt Datenaktualisierung mittels Mosaiking in den Bestand der Rasterverwaltung überführt wird.

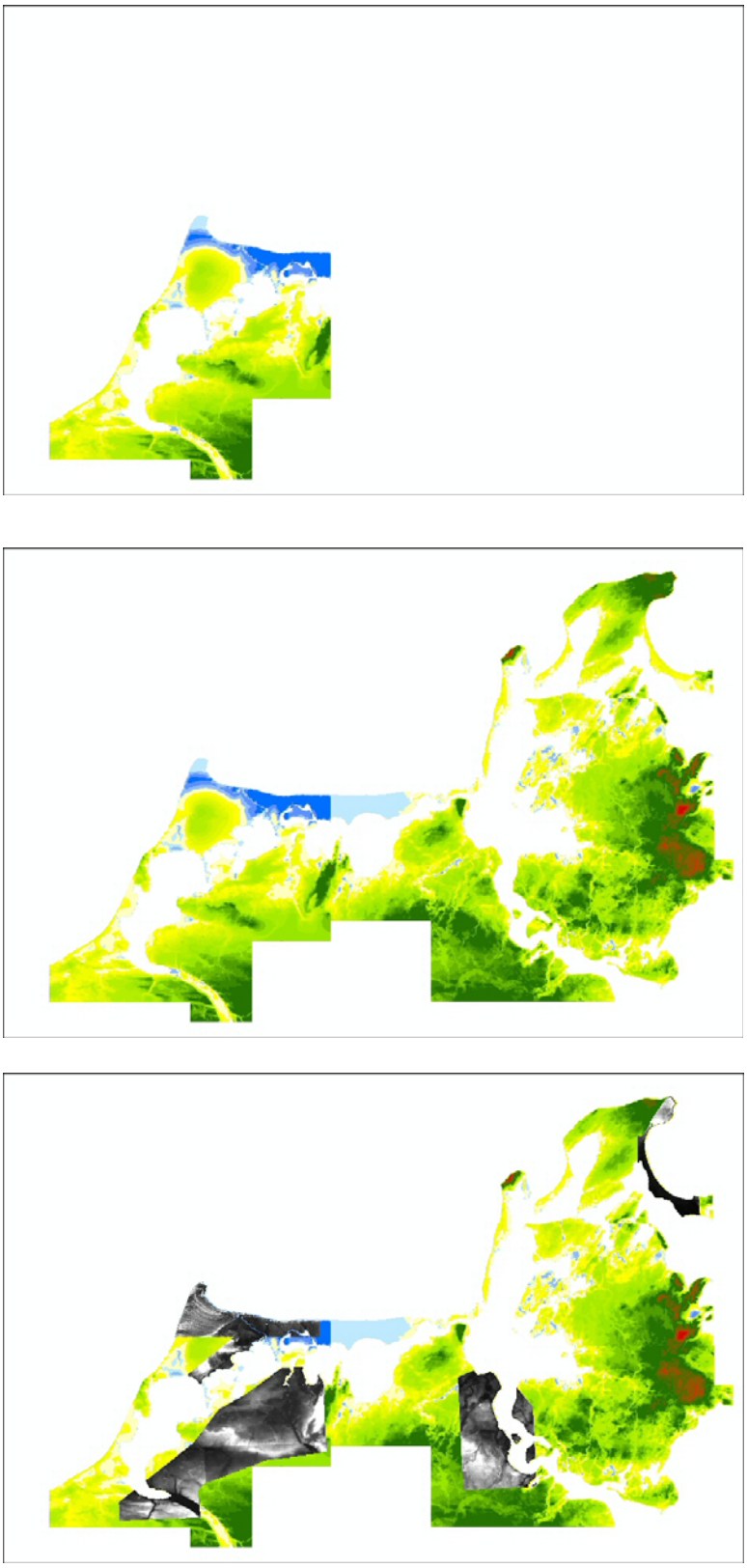


Abb. 20-1: Erstellung der Kacheln

Bild 1/2: Zusammensetzen der Modelle aus Digitalisierung
Bild 3: Zusammenstellung von Laserscanning-Daten (grau)

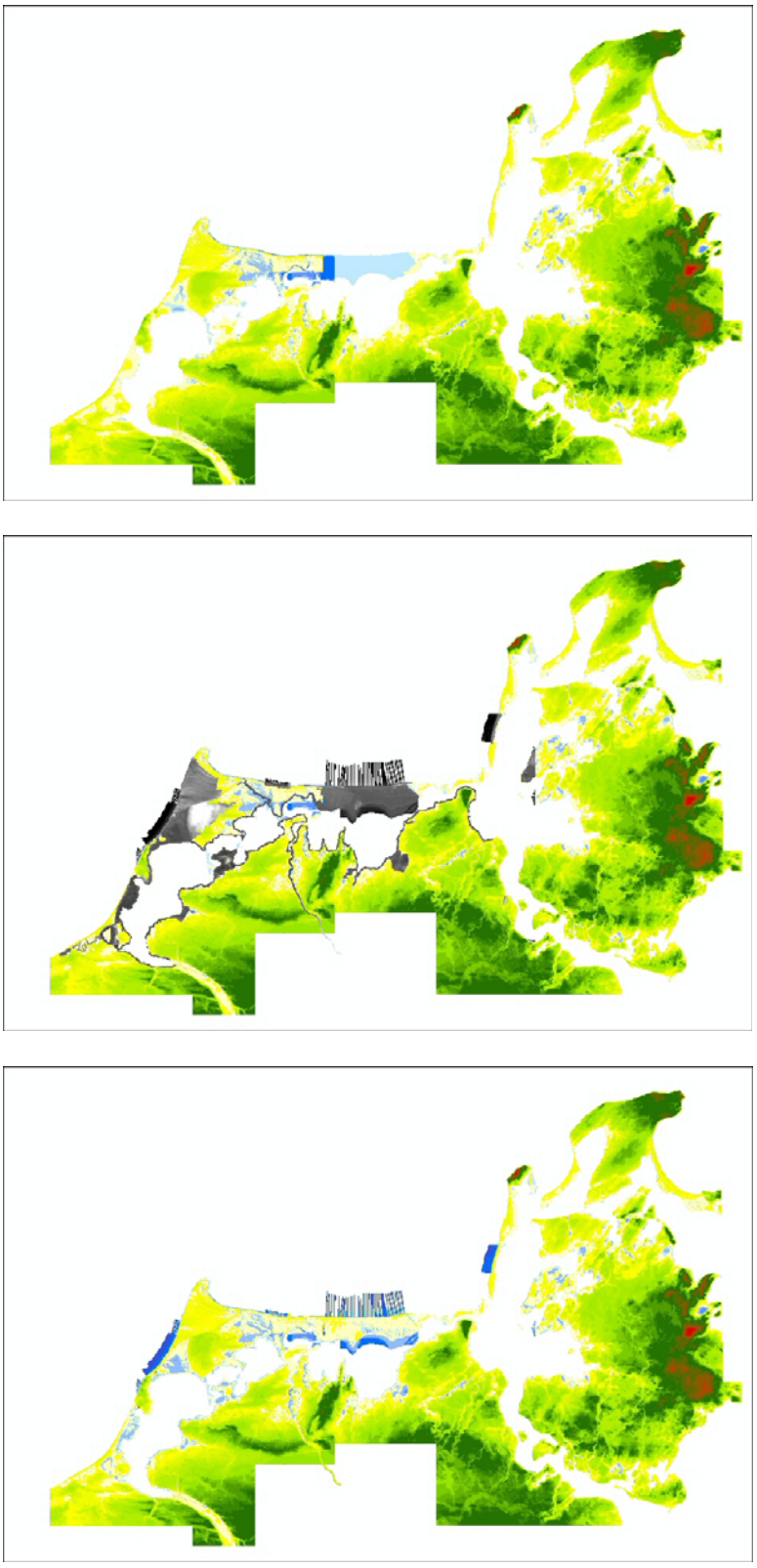


Abb. 20-2: Erstellung der Kacheln

Bild 4: Integration von Laserscanning-Daten in das Modell
Bild 5/6: Zusammenstellung (grau) und Integration von GPS/Tachymetrie-Messdaten

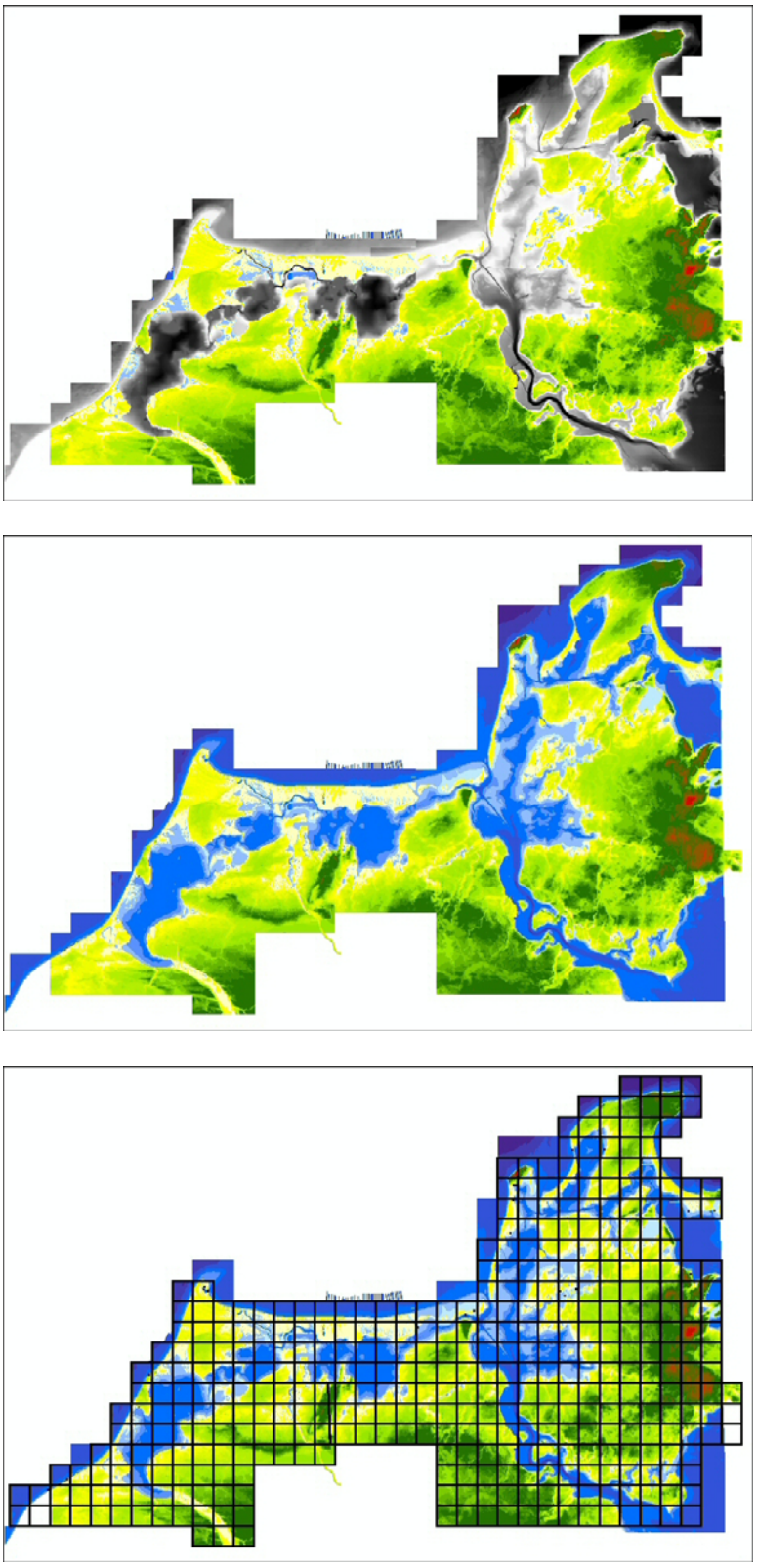


Abb. 20-3: Erstellung der Kacheln

Bild 7/8: Hinzufügen der Daten des Seegrundes (Bodden und Küstengewässer)
Bild 9: Überlagerung und Teilung mit Kachelsystem

8.4 Rasterdatenzugriff und -austausch

Die Datenabfrage aus der Rasterverwaltung ist anhand textbasierter Metainformationen durch Formulierung eines entsprechenden Anfragestrings in einer Clientapplikation möglich. Überwiegend werden jedoch Abfragen des Datenbestandes aufgrund räumlicher Kriterien durchgeführt. Eine Möglichkeit des Datenaustausches besteht in der Verwendung eines WMS-Dienstes in einem Webinterface. Hier werden die DGM-Rasterdaten mit verringerter Auflösung belegt durch Luftbilder oder einer topographischen Karte zum Erzeugen eines halbtransparenten 'Überflutungsbereiches' verwendet, wobei das Überflutungsniveau verändert werden kann. Die optische Information über Gebietszustände ist damit möglich.

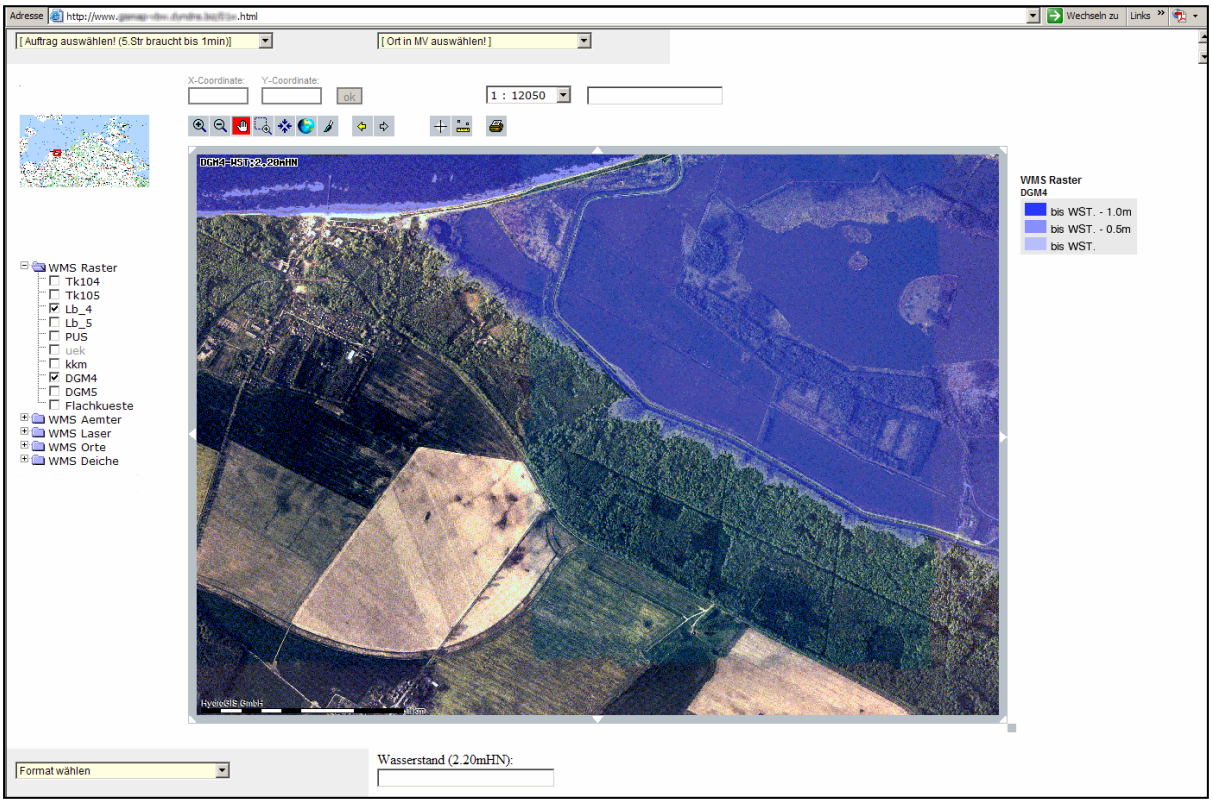


Abb.: 21: WMS-Interface zum Datenaustausch -Wasserstandssimulation 2.20müHN

Eine Auswahl von Analysegebieten (Kachelauswahl) kann durch die Erstellung eines Abfragepolygons (Speicherung im DBMS) vorbereitet werden. Dies wird durch clientseitige Erfassung der benötigten Koordinaten vor dem Hintergrund einer Orientierungskarte und Umwandlung in eine SQL-Einfügeoperation (INSERT) incl. Darstellung des Polygons als WKT-String sowie zugehöriger Metadaten und Übertragung an die Datenbank durch die ODBC-Schnittstelle ermöglicht. Eine anschließende räumliche Abfrage auf den Rasterdatenbestand liefert die Kacheln zurück, die von Anfragepolygon geschnitten werden:

```
select raster,name from dgm
   where intersects(geom,(select geom from gebietsabfragen4 Where gid = 112)) order by name;
```

Sie liegen dann als Einzeldateien im GeoTIFF-Format im Dateisystem vor. Gleichzeitig wird ein Script zur Fusion der disjunkten Kacheln als Batch-Datei erzeugt und anschließend ausgeführt. Dazu wird ein Teilprogramm der Rasterbibliothek GDAL (gdal_merge) aufgerufen. Ergebnis dieses Prozesses ist ein mosaikiertes GeoTIFF des Abfragegebietes, welches durch seine eingebetteten Metadaten universell in diversen Anwendungssystemen (GIS, Bildverarbeitung) genutzt werden kann.

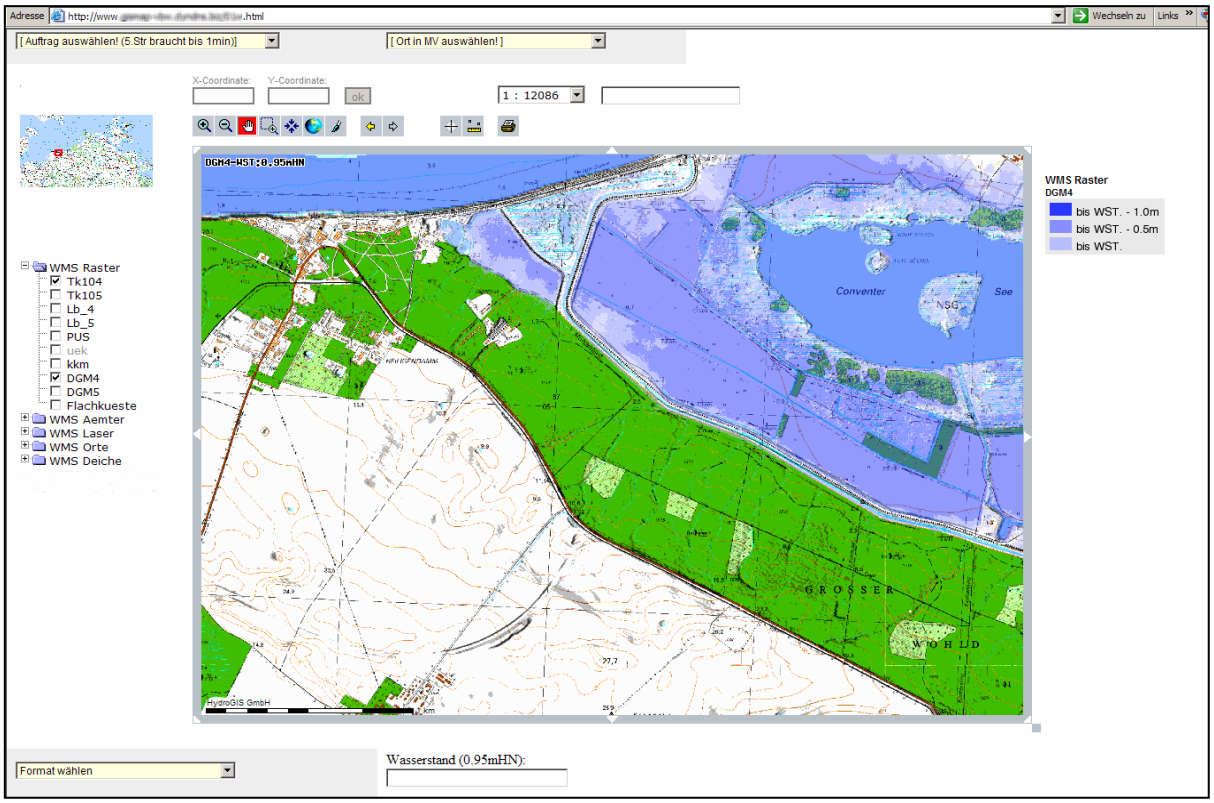


Abb. 22: DGM mit hinterlegter TK (Wst: 0.95müHN)

8.5 Datenaktualisierung und Pflege

Sollen die Kacheln bzw. Geländemodelle eines Gebietes aufgrund von neuen Messdaten aktualisiert werden, so ist in einem ersten Schritt ein Raster der Neudaten zu erzeugen. Da nur flächig zusammenhängende Vermessungen einer Mindestgröße verwendet werden, kann mit dem um 2-5m gepufferten Umring der Messdaten das Raster der Neudaten beschnitten werden. Eine Möglichkeit zur Abgrenzung ist die auf die Umringgenerierung folgende Projektion dieser Linie auf den Altdatenbestand. Man erhält daraus eine dreidimensionale Umringlinie (ESRI: PolyLineZ), die zur 'glatten' Einpassung mit in die Modellberechnung der Neudaten einbezogen wird (TIN-Clipping). Zu beachten ist die exakte Lage der Rasterzellkoordinaten.

Mit Hilfe des Umringes wird weiterhin der 'Alt'-Rasterdatenbestand abgefragt. Man erhält, wie oben beschrieben, ein Geländeraster zusammengesetzt aus den ganz oder teilweise zu

aktualisierenden Kacheln im GeoTIFF-Format. Zur Nutzung in ArcGIS ist nun die Erstellung der Rasterstatistiken mit Hilfe der Toolbox notwendig. Nach Beschneidung des Neudaten-Rasters erfolgt in ArcGIS eine Mosaikierung, wobei aufgrund der exakten Zellpositionen von Alt- und Neubestand kein Resampling notwendig wird. An den Zellpositionen, an denen ein Neudatenbestand vorliegt, ersetzt dieser den Altbestand vollständig. Ergebnis ist ein Geländeraster eines Gebietes bestehend aus Alt- und Neubestand mit angeglichenen Übergangsbereichen im ArcGIS-Floating-Point-Grid-Format.
Zur Aktualisierung des Datenbankbestandes ist die Bereitstellung dieser aktualisierten Rasterdaten als Menge lagerichtiger Kacheln einheitlicher Größe und Auflösung sowie den zugehörigen Metainformationen im GeoTIFF-Format notwendig. Die Trennung des mosaikierten Rasters in die benötigten Kacheln erfolgt mit Hilfe der Informationen aus der Altbestandsabfrage. Mit Hilfe eines Script wird die Zerlegung und Formatänderung in die einzelnen, namentlich enthalten in einer während der Abfrage erstellten Textdatei, und komprimierten GeoTIFF-Kacheln vorgenommen. Gleichzeitig wird ein SQL-Transaktionsblock erzeugt, der die Aktualisierung des Datenbestandes mit einem SQL-Update durchführt.

```
Bsp.: UPDATE dgm SET raster=lo_import('d:/ra/54470_59845.tif') where name ='54470_59845';
```

Im Mittel werden bei jeder Datenaktualisierung 50-100 Kacheln neu erzeugt und in der Datenbank verändert. Große Laserscanning-Messungen können aber auch den Austausch von bis zu 1000 Kacheln hervorrufen. Nach dem Transfer der aktualisierten Kacheln auf den Datenbankserver wird mit einer SQL-Befehlsfolge der Updatevorgang innerhalb der Datenbank durchgeführt. Die in das DBMS importierten Kacheln können im Dateisystem nun gelöscht werden, um die Haltung redundanter Daten zu vermeiden. In diesem Sinne dient die Datenbank als Medium zur strukturierten Datenhaltung der Geländeraster.
Als Eigenart des genutzten Datenbanksystems werden nur die Metadatenrelation verändert. Dies bedeutet, das lediglich die Kachelmetaattribute wie auch die Referenz auf das BLOB (OID) der betroffenen Kacheln ersetzt werden. Innerhalb der Datenbank bleibt aber das 'alte' Kachelrasterobjekt unter seiner Referenz innerhalb einer Datenbankdatei erhalten. Dieser Mechanismus könnte zum Monitoring von Kachelveränderungen verwendet werden (z.B. Küstenrückgang), erzeugt aber auch sehr große sehr selten benötigte redundante Datenmengen. Aufgrund der derzeit noch sehr begrenzten Sekundärspeichergröße wird daher nach jeder Datenbankaktualisierung eine Wartung derselben vorgenommen. Die geschieht durch ein serverseitiges Datenbankzusatzprogramm (vacuumlo), welches die nicht mehr aktiv

durch ihre OID referenzierten BLOB aus dem Datenbankbestand und dem Dateisystem (verwaltet durch das DBMS) endgültig entfernt (siehe Kapitel 5.4)
Neuere Überlegungen sehen eine Veränderung hin zur Einführung einer temporalen Gültigkeitsprüfung der Kacheln und daraus folgender Möglichkeit der Kachelversionierung vor (z.B. durch Trigger). Eine Indexaktualisierung des Kachelbereiches (MUR-Geometrieattribut) ist aufgrund des ausschließlichen Ersatzes des BLOB ohne Veränderung der Kachellage (Kachelumringgeometrie bleibt erhalten) nicht notwendig. Eine periodische Datenbankwartungsoperation (wöchentlich, Sonntagnacht) aktualisiert die Datenbankstatistiken und entfernt mögliche nicht verwendete Datenartefakte.

8.6 Zusammenstellung verwendeter Konzepte und Verfahren

Aufgrund der sehr großen Ausdehnung des Projektes (6500km², 6.5Mrd. Rasterpunkte) konnte die Rasterverwaltung zur Lösung der Anforderungen nur innerhalb einer Datenbank realisiert werden. Sie bildet ein Ordnungssystem für die Rasterdaten, die nur noch an dieser zentralen Stelle als aktualisiert und bereinigt gelten. Zur Verhinderung vor unbefugtem, direktem Zugriff wird für alle Nutzer nur ein eingeschränkter, lesender Zugriff von einem Webinterface (WMS) aus erlaubt, welches die abfragenden Geometrien sowie eine verringerte Rasterauflösung anzeigen kann und mittels Scripterzeugung mit einem eindeutigen Abfrageweg den Abfragevorgang startet. Die räumlichen und thematischen Abfragen (Umringe und Metadaten) werden zu Dokumentationszwecken in der Datenbank gespeichert. Ein administrativer Zugang zur Kachelaktualisierung und zur Ausführung von Erweiterungs- und als SQL-Befehlsfolge hinterlegten Wartungsoperationen ist eingerichtet worden.
Als *Datenstruktur* konnten im DBMS PostgreSQL BLOB's zur Aufnahme einer vollständigen binärcodierten GeoTIFF-Datei verwendet werden, die direkt in anderen Anwendungsprogrammen verarbeitbar ist. Nachteile sind der bisher fehlende Zugriff auf einzelne Rasterzellen einer Kachel, sodass das kleinste austauschbare Rasterteil (1 Kachel) 500x500m groß ist.
Es wurden vielfältig die beschriebenen Rasterkonzepte genutzt sowie einige der Möglichkeiten spezieller Rastereigenschaften optimiert. Aufgrund der verfügbaren hochauflösenden Laserscanningdaten und der Verwendung eines DBMS konnte die *Rasterauflösung* im Projektverlauf auf 1x1m verbessert werden, um die Wiedergabe von Kleinstrukturen (Deiche, Schutzmauern) noch zu ermöglichen. Als Zellauflösung wurde eine 32-Bit-Fließkommzahl gewählt, um die DGM-Höhenwerte mit der erforderlichen Genauigkeit abzulegen und eine Vielzahl diesen Zelltyp nutzenden Operationen verwenden zu können. Zur *Georeferenzierung* der Rasterkacheln wird der im GeoTIFF-Format

implementierte Mechanismus aus Positionierung und einer EPSG-Codierung zur Erzeugung exakt positionierter blattschnittfreier Kacheln eingesetzt. Derzeit werden nur *Basismetadaten* (Name, Umring, Gebiet) erfasst. Es soll ein Schema entwickelt werden, welches die notwendigen Informationen ermittelt und geordnet bereitstellt (z.B. Aktualisierungsdatum, Datenquellen). Zur Nutzung eines *Multi-Resolution-Konzeptes* werden derzeit keine reduzierten Daten innerhalb der Datenbank sondern durch periodische Generierung als Eingangsrasterdaten des Webzugriffssystems (10x10m-Auflösung) und zusammengefasst zu je 25 Kacheln als GeoTIFF-Dateien außerhalb des DBMS abgelegt. Sehr wichtig für Zugriff und Aktualisierung ist das verwendete Kachelkonzept zur *Partitionierung* der Rasterdaten. Dabei wurde aufgrund des Fehlens eines automatischen Verfahrens innerhalb des DBMS eine scriptbasierte Zerlegung in die 25850 Kacheln mit eindeutigen geraden Kachelaußengrenzen zur Kachelerstellung oder Aktualisierung bereitgestellt. Diese generiert neben den binären Kacheldaten auch die *Metadaten* (Name aus der Lage) sowie die Geometriedaten (Umring aus der Lage). Zur Rasternutzung werden die benötigten Kacheln durch einen Mosaikingprozess zu einem DGM-Gebiet zusammengefügt. Durch Nutzung eines DBMS wird auch die *Speichermanagementkontrolle* der Datenbank, die die BLOB-Kacheldaten zur sehr großen indizierten Datenbankdateien (1GB) zusammenfasst, verwendet. Getestet werden derzeit die Möglichkeiten der Relationsclusterung und -partitionierung.

Die *Datenerfassung* wurde rasterbasiert mit Laserscanning und Fächerecholot, punktuell mit Tachymetrie und GPS sowie linienhaft durch Digitalisierung von Höhenlinien durchgeführt. Als *Interpolationsmethoden* sind bisher die Nearest Neighbor Methode zur Bereinigung von Fehlstellen der rasterbasierten Eingangsdaten, ein 'TINtoRaster'-Verfahren für Bruchkanten und Geländepunkte sowie eine Spline-Interpolation zur Verarbeitung der Digitalisierung und die Methode der inversen Distanzen (Seeboden) verwendet worden. Durch eine Prioritätenbildung konnte *Mosaiking* zur Zusammenstellung und Integration der Rasterquelldaten unter Beachtung von Messzeitraum und Messziel (z.B. Baumassnahme) eingesetzt werden. Land- und Seebereich wurden durch eine hochgenaue Uferlinie getrennt. Das Konzept der *Rasterabfragen* eines DGM-Datenbestandes kann im Projekt zur Rasteranalyse speziell zur Erstellung potenzieller Überflutungsflächen (Höhenniveau) eingesetzt werden. Aufgrund der großen Rasterauflösung wird lediglich die Nearest Neighbor Resamplingmethode bei *Rastertransformationen* verwendet. Die Veränderung der Höhenwerte (z.B. bei bikubisch) wird bei einem maximalen Lagefehler von 0.5m (1/2 Zellgröße) ausgeschlossen, um Bruchkanten (Steilküste) nicht zu verfälschen.

Der *Rasterdatenzugriff* durch automatisiert erzeugte oder direkt erstellte SQL-Befehlsfolgen liefert zum Datenaustausch Dateien des GeoTIFF-Formates, die direkt in vielen Anwendungsprogrammen weiterverarbeitet werden können. Häufig wird auch eine Umwandlung in die sehr speicherplatzintensiven XYZ-ASCII-Daten vorgenommen. Zur dienstbasierten Ausgabe steht ein Webinterface (WMS) bereit, welches die schnelle optische Abfrage und Information ermöglicht. Eine Erweiterung durch Konzepte des WTS/WCS ist geplant.

Die im Abschnitt *Optimierung* vorgestellten Ansätze sind im Projekt bereits erprobt und verwendet worden. Das GeoTIFF-Format beinhaltet die verlustlose Deflate-Datenkompression, die Speicherplatzgewinne von etwa 15% ermöglichte. Ebenso wurde die Indexierung durch einen B-Tree auf die Namensspalte und einen R-Tree-over-GIST auf die Umringgeometriespalte bereits eingesetzt.

8.7 Datenausgaben und Anwendungsbeispiele

Aus dem Datenbankmanagementsystem der digitalen Geländemodelle können neben den dienstbasierten Ausgaben zur Datenvisualisierung in einer angepassten Auflösung einer Webanwendung auch hochwertige Modellausgaben, übertragen durch das GeoTIFF-Format, herausgelöst werden. Diese hochgenauen Geländemodelle der maximalen Auflösung von 1x1m sind vielfältig für Planung, Analyse und Monitoringaufgaben verschiedenster Anwendungsbereiche einsetzbar. Neben Untersuchungen zur Überflutungsgefährdung von Flächen, der Analyse und Bewertung von morphologischen und klimatologischen Küstenveränderungen sind auch Anwendungen im Katastrophen- und Umweltschutz, im Planungssektor mit Flächenerschließung, Wegebau sowie dem Bau, Erhaltung und Überprüfung der Wirksamkeit von Küstenschutzanlagen und auch im Tourismus oder in Land- und Forstwirtschaft möglich. Kartographische und mediale Produkte, entwickelt aus den Modellen, können aus abgeleiteten Höhenlinien und Geländeschnitten, Volumen- und Flächenberechnungen, thematischen Karten und auch dreidimensionalen perspektivischen Bildern oder Animationen bestehen. Die Weitergabe von Geländemodellteilen des für einen großen Bereich Mecklenburg-Vorpommerns nach einheitlichen Verfahren und Kriterien sowie unter ständiger Aktualisierung entwickelten Modells mit zentraler Datenhaltung führt zu einer Vereinheitlichung des Produktes 'Hochgenaues DGM' bzgl. Qualität und Verfügbarkeit.

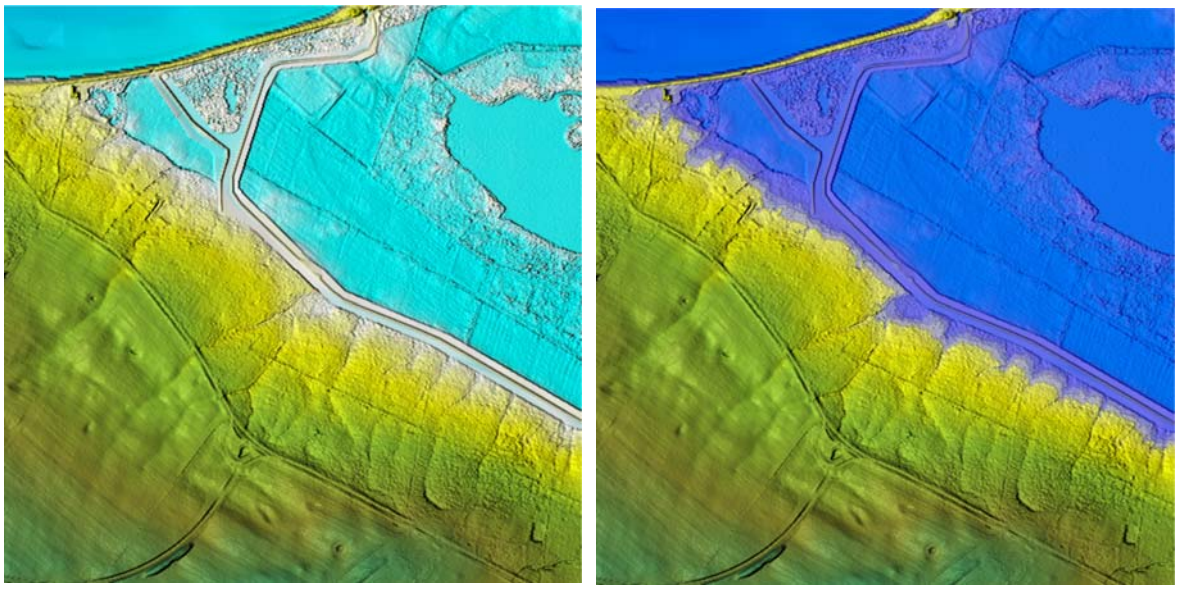


Abb. 23: Visualisierung ohne/mit potenzieller Überflutung bei Bemessungswasserstand

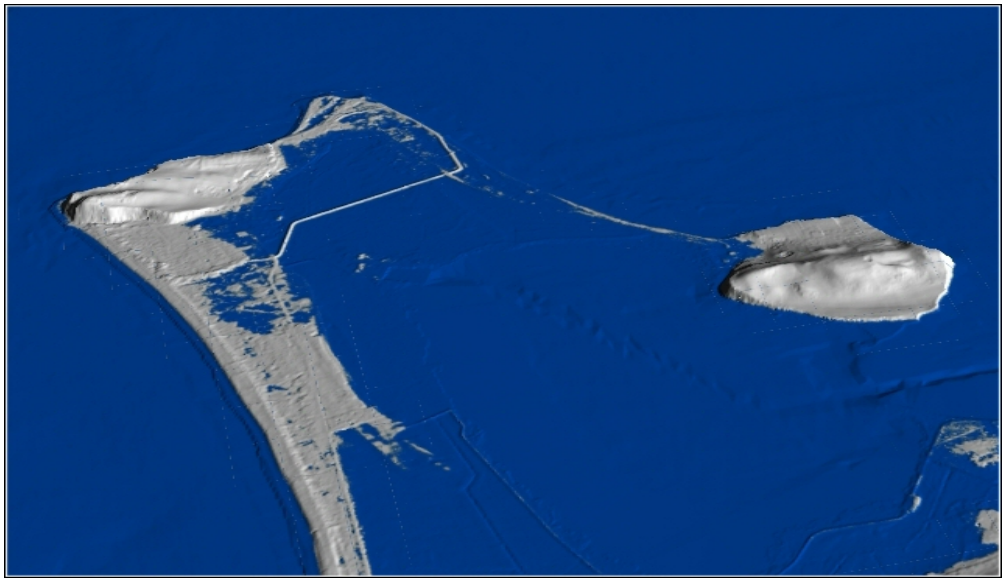


Abb. 24: Überflutungs-simulation perspektivisch in Adobe-3D

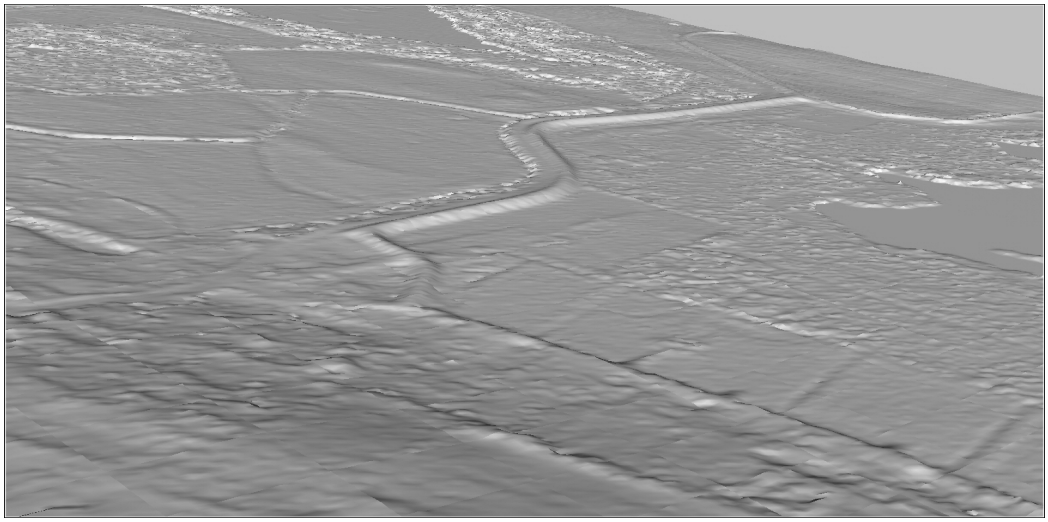


Abb. 25: DGM-Gebiet mit Laserscanning-Daten und Küstenschutzbauwerk (Deich)

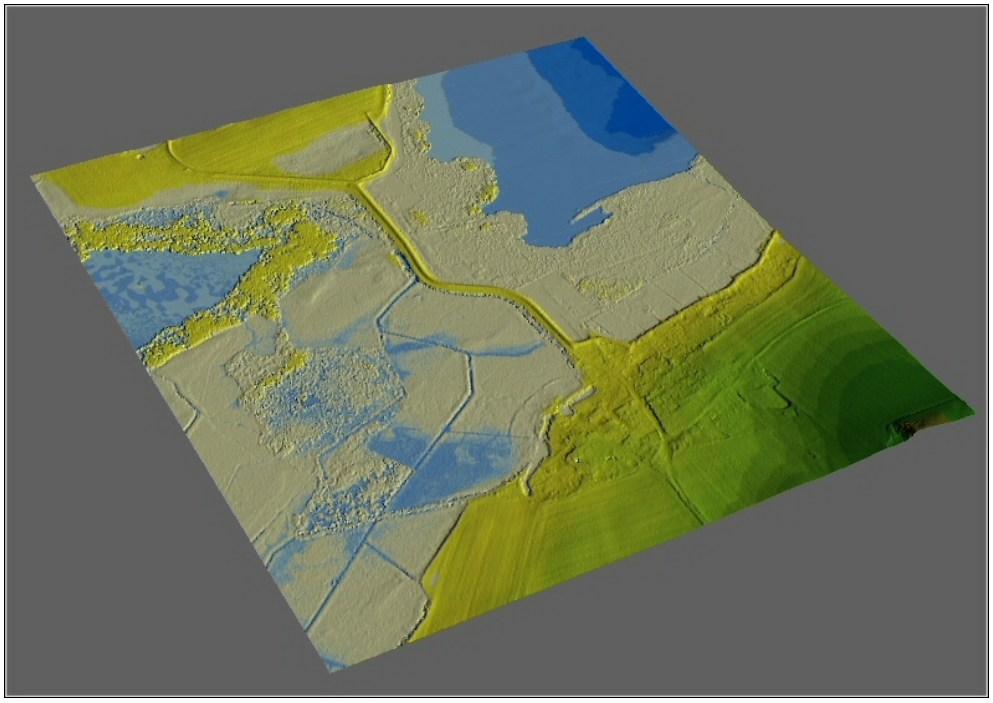


Abb. 26: perspektivische DGM-Ausgabe in Adobe-3D

8.8 Ausblick

Derzeit laufen am Projekt noch umfangreiche Erweiterungen und Veränderungen. Die Integration der Quelldaten in die Datenbank in Vektorform soll die Abfrage nach individuellen Messkriterien (Zeitpunkt, Messauftrag) ermöglichen. Eine verbesserte Metadatenerfassung könnte die Suche nach Objekten (Küstenschutzbauwerk X) und nachfolgende Ausgabe des zugehörigen Geländemodells unterstützen. Ebenso wird versucht, durch Möglichkeiten der Haltung von Datensätzen verschiedener Auflösungen (Level of Detail) bzw. der Erzeugung einer bestimmten Auflösung nach einer Datenabfrage, die auszugebende Datenmenge an den Verwendungszweck anzupassen. Auch die generelle Datenhaltung in Vektorform als TIN mit Rastererzeugung erst bei Datenabfrage wird diskutiert und geprüft.

Soll die Abfrage nach kachelinternen Eigenschaften möglich sein, so ist eine programmtechnische Codierung und Decodierung der BLOB-Daten beispielsweise in ein ADT-Format zu erstellen. Datenabfragen nach Höhen- oder Texturkriterienkriterien sowie Verschneidungen mit Vektordaten (Beispiel: 'Zeige Wiesen unter Meeresspiegel') sind dann auf dem Datenbestand möglich.

Zur einfacheren Administration des Workflows wird eine verbesserte webbasierte Scriptsteuerung der beteiligten Softwarekomponenten incl. Nutzerverwaltung entwickelt werden. Die gemeinsame Datenausgabe der Raster- und Vektordaten über OGC-konforme Schnittstellen (WCS/WTS) ins Intranet mit webbasierten Tools zu einfachen Analysen, beispielsweise der Modelleinfärbung nach Höhenschichten oder der punktuellen Höhenwertabfrage, sind weitere mögliche Entwicklungsschritte.

9 Zusammenfassung

Zentraler Ansatz der Arbeit war die Zusammenstellung der für den Aufbau einer Rasterverwaltung digitaler Geländemodelle wesentlichen Verfahren und Konzepte. Die Nutzung des Rastermodells gegenüber einem vektorbasierten Ansatz (z.B. TIN) kann aufgrund der viel einfacheren Handhabung durch simplere Zugriffsmethoden und vielfältige Rechenoperationen, häufig bereits in einem Rasterformat vorliegenden Eingangsdaten (Laserscanning) sowie der weiten Verbreitung und universellen Konvertierbarkeit der Rasteraustauschformate und -mechanismen für die Erstellung großräumiger Geländemodelle empfohlen werden. Da dabei sehr große Datenmengen anfallen, sind die Möglichkeiten einer dateibasierten Verwaltung der Datensätze gegenüber einer Ablage in einem Datenbankmanagementsystem sehr beschränkt. Vorteile einer Datenbank sind neben der Nutzung einer standardisierten Datendefinitions- und Anfragesprache (SQL) für den effektiven Umgang, datenbankinternen Sicherheitsmechanismen wie Nutzer- und Zugriffsrechteverwaltung sowie Speicher- und Archivierungsmanagement auch die Möglichkeit von Multi-User-Zugriffen. Es kann ein integriertes Management aus Raster-, Vektor- und tabellarischen Daten aufgebaut werden. Zur Nutzung erweiterter rasterspezifischer Methoden und Managementansätze wie Multi-Resolution-Konzepten oder Rasterpartitionierungen sind Erweiterungen der Standarddatenbankfunktionalitäten notwendig. Die Software RasDaMan besitzt hier sehr spezialisierte Methoden für die Rasterverarbeitung hochdimensionaler großer Rasterdatensätze. Die Einbettung in eine GIS-Umgebung zur Nutzung deren Möglichkeiten der Ausgabegestaltung ist der Vorteil einer Datenbankmanagementlösung mit ESRI-Software. Oracle bietet mit Oracle Spatial GeoRaster eine umfangreiche, gut dokumentierte und für große Rasterdatensätze geeignete Verwaltung an. Werden gute Erweiterbarkeit und umfangreiche Schnittstellen bei Konzeption, Erstellung und Betrieb einer Rasterverwaltung benötigt, so kann der u.a. bereits im Projekt 'DGM Küstenbereich' praxiserprobte Einsatz des objektrelationalen PostgreSQL/PostGIS-Systems empfohlen werden. Die Datenablage innerhalb eines Datenbanksystems kann mit Hilfe einer speziell an die Anforderungen angepassten ADT-Struktur, mit einer den Zugriff auf einzelne Rasterzellen ermöglichenden Array-Struktur oder sehr universell und einfach handhabbar mit Hilfe von Binary Large Objects gestaltet werden. Im Laufe der praktischen Projektbearbeitung hat sich die erfolgreiche Verwendung von BLOB's im gewählten Datenbanksystem unter Verwendung einiger der in der Arbeit vorgestellten Konzepte gezeigt. Die verlustfreie Kompression der Daten mit Hilfe des TIFF-Deflate-Algorithmus sowie die

Verwendung von indexierten Metadaten für effektiven Zugriff bilden neben dem noch auszubauenden Datenaustausch durch mehrere Rasterformate und Dienste einige der Bausteine der Projektbearbeitung des 'DGM Küstenbereich'. Es haben sich die Vor- und Nachteile von Datenerfassungs- und Verarbeitungsmethoden beim Einsatz im Küstenbereich gezeigt, sodass die vorgestellte allgemeine Datenintegrationsstrategie zur DGM-Erstellung erfolgreich umgesetzt werden konnte.

Zur Konzeption einer Rasterverwaltung digitaler Geländemodelle sind Vorüberlegungen zur Gestaltung der Datenbank mit einem Rasterdatentyp und den aufzunehmenden Metadaten, zu einem Kachelkonzept und benötigten erweiterten Rastermanagementkonzepten sowie zu Datenaustauschformen und Datenzugriffsmustern und -szenarien zu treffen. Aktualisierungsarten und Zeiträume der Daten sind ebenso zu betrachten, wie mögliche Erweiterungen der Datenstruktur bzw. des Datenaustausches beispielsweise durch dienstbasierte Datenvisualisierung. Optimierungsansätze zur Speicherplatz- und Anfragelaufzeitverringerung sind anforderungsbezogen zu planen.
Notwendige Teile zum Aufbau einer Rasterdatenbankinfrastruktur sind damit neben den bereitzustellenden Quelldaten mit einheitlichen Eigenschaften, einer Datenbank mit Unterstützung räumlicher Strukturen und Datentypen zur Ablage der Rasterdaten auch die für den schnellen Datenzugriff notwendigen Hardwareausstattungen.
Als Ergebnis konnte gezeigt werden, das mit den verfügbaren Rasterverfahren und Softwaresystemen der Aufbau einer datenbankbasierten Verwaltung digitaler Geländemodelle für effektivere Nutzung und verbesserten Zugriff dieser Daten möglich ist. Zukünftige Entwicklungen sind durch die erweiterte Nutzung und Standardisierung des Rastermanagements und Rasteraustauschs durch das OGC und deren Umsetzung durch die Softwarehersteller sowie durch verbesserte Visualisierungs- und Speichermanagementtechniken zu erwarten. Eine verbesserte Dokumentation sowie eine Vereinheitlichung von Rasterdatenerfassungs- und Erstellungsmethoden incl. der erweiterten Untersuchung ihrer Eignung für bestimmte Anwendungsgebiete (z.B. Küstenbereich) und Anwendungszwecke (z.B. Visualisierung) kann die Qualität der DGM-Quelldaten weiter vereinheitlichen und verbessern. Ebenso könnten speziell für großräumige digitale Geländemodelle geeignete Ausgabeprodukte spezifiziert, dokumentiert und standardisiert werden.

Literaturverzeichnis

[Bau97] *Baumann, P.*, et al: Geo/Environmental and Medical Data Management in the RasDaMan System, 1997,http://www.informatik.uni-trier.de/~ley /db/conf/ vldb/BaumannFRW97.html (Proceedings of the 23rd VLDB Conference, Athen) [25.10.06]

[Bau98] *Baumann, P.*,et al: Spatio-temporal Retrieval with RasDaMan, 1998 http://www.informatik.uni-trier.de/~ley/db/conf/vldb/BaumannDFRW99.html (Proceedings of the 25th VLDB Conference, Edingburg) [25.10.06]

[Bau00] *Baumann, P.*: Warum Datenbanken auch für Raster-Geodaten ?, 2000 http://www.rasdaman.com/Product/WhitePapers/warumGeoDatenbanken.pdf [23.03.06]

[Bau03] *Baumann, P.*, et al: Lage-scale, standard-based earth observation imagery and web mapping services, 2003, http://www.vldb.org/conf/2003/papers/S37P08.pdf (Proceedings of the 29th VLDB Conference, Berlin) [27.10.06]

[Bau04] *Baumann, P.*: Rasterdatenbanken am Beispiel rasdaman, 2004 http://www.datenbank-spektrum.de/pdf/dbs-10-30.pdf (Datenbankspektrum, Vol.10 Nr.09, S.30-37) [27.10.06]

[Behl02] *Behl, J.*: Performance-Optimierung einer Datenbankanwendung auf IBM's DB2 UDB, Universität Stuttgart FB Informatik, ftp://ftp.informatik.uni-stuttgart.de/pub/library/medoc.ustuttgart_fi/DIP-2037/DIP-2037.pdf, Diplomarbeit, 2002 [10.02.07]

[Bla03] *Blazewicz, J.*, et al (Hrsg.): Handbook on data management in information systems, 1. Auflage, Berlin/Heidelberg: Springer, 2003

[Brink05] *Brinkhoff, T.*: Geodatenbanksysteme in Theorie und Praxis, 1. Auflage, Heidelberg: H. Wichmann Verlag, 2005

[Cat00] *Cattell, R.G.G.*, et al: The object data standard ODMG 3.0, 1. Auflage, San Francisco: Morgan Kaufmann Publishers, 2000

[Chu96] *Chung, S.M.*: Multimedia information storage and management, 1. Auflage , Boston/London/Dordrecht: Kluwer Academic Publishers, 1996

[Cop06] *Cope, S.*: Image Compression - Past, Present and Future, GeoInformatics Nr. 8 Vol. 9, Emmeloord: cmedia productions bv, S.32/33, 2006

[Cor93] *Corrigan, P. and Gurry, M.*,:Oracle performance tuning, Sec. Edition, Sebastopol: O'Reilly & Acc., 1993

[Dan04] *Dannenberg, K.*: Wavelets, http://www.fh-wedel.de/cis/archiv/seminare/ws0304/sz/grundlagen/wavelet0.htm, 2004, [11.02.07]

[Eis03] *Eisentraut, P.*: PostgreSQL: Das offizielle Handbuch, 1. Auflage, Bonn: mitp Verlag, 2003

[ESRI1] Raster Data in ArcSDE 9.1, White Paper, http://www.esrisa.com.my/download/RasterData_in_ArcSDE91.pdf, 2005, [10.02.07]

[ESRI2] Managing a Raster Database, Technical Paper, http://support.esri.com/index.cfm?fa=knowledgebase.whitepapers.viewPaper&PID=17&MetaID=1018, 2005 [10.02.07]

[Gut84] *Gutmann, A.*:R-Trees, a dynamic index structure for spatial searching, Berkeley: ACM, http://www.sai.msu.su/~megera/postgres/gist/papers/gutman-rtree.pdf, 1984 [10.02.07]

[Hak94] *Hake, G., Grünreich, D.*: Kartographie, 7. neubearb. Aufl., Berlin: de Gruyter, 1994

[HNP95] *J. Hellerstein, J. Naughton, and A. Pfeffer:* Generalized Search Trees for Database Systems. In: Proc. 21st Int'l Conference on Very Large Databases (VLDB), Zürich, S. 562–573, 1995

[Jäg] *Jäger, M.-A.*: Ziv-Lempel-Kompression, http://www.lempel-ziv.de/ [11.02.07]

[Neb97] *Nebiker, S.*: Spatial raster data management for geo-information systems : a database perspective, Ing.-wiss. Fakultät ETH Zürich, Dissertation Nr. 12374, 181 Seiten, 1997, http://e-collection.ethbib.ethz.ch/show?type=diss&nr=12374 [10.10.06]

[Neb02] *Nebiker, S., et al*: Transformation von Rasterdaten mit finiten Elementen, Studienauftrag CC RD/LV95, FH beider Basel, http://www.fhbb.ch/tools/publikationen/ pdf/0203_11_ Studie_Rastertransformation_d.pdf, 2002 [03.03.07]

[Ney00] *Ney, H.*: Algorithmen und Datenstrukturen, RWTH Achen, http://www-i9.informatik.rwth-aachen.de/dateien/lehre/2/NeySkript.pdf, 2001 [03.03.07]

[Korn00] *Kornacker, M.*: Access Methods for next-generation database systems, University of California/Berkeley, http://citeseer.ist.psu.edu/cache/papers/cs/22615/http:zSzzSzs2k-

ftp.cs.berkeley.edu:8000zSz~marcelzSzdisszSzdiss.pdf/access-methods-for-next.pdf, Dissertation, 2000 [10.02.07]

[Kot04] *Kothuri, R.*, et al: Pro Oracle Spatial.1. Auflage, Berkeley: apress, 2004

[Krau00] *Kraus, K.*: Photogrammetrie, Bd. 3, 6.Auflage, Köln: Dümmler, 2000

[Krau04] *Kraus, K.*: Photogrammetrie, Bd. 1, 7. Auflage, Berlin: de Gruyter, 2004

[Lin93] *Lindenberger, J.*: Laser-Profilmessungen zur topographischen Geländeaufnahme, Universität, FB f. Bauingenieur- und Vermessungswesen, Dissertation, 1993

[Lind05] *Linder, W.*: Digitale Photogrammetrie am PC, Universität Düsseldorf, http://docserv.uni-duesseldorf.de/servlets/DerivateServlet/Derivate-3042/1042.pdf, Habilitationsschrift, 2005 [03.03.07]

[McD04] *McDonald, S.*,et al (Hrsg), Advances in Information Retrieval: proceedings of the 26th european conference on ir research, (LNCS Nr. 2997), 1. Auflage, Berlin: Springer, 2004

[Micro] Hinweise zur Indexverwendung im SQL-Server, http://msdn2.microsoft.com/de-de/library/ms189605.aspx [07.03.07]

[O'Neil01] *O'Neil, E. und P.*: Database: Principles, programming, performance, Sec. Edition, San Francisco: Morgan Kaufmann Publishers, 2001

[OGC1] *Singh, R.R.* (Edt.): OGC Web Terrain Server (WTS), Disskussion Paper, http://portal.opengeospatial.org/files/?artifact_id=1072, 2001 [10.02.07]

[OGC2] *Baumann, P.* (Edt.): Web Coverage Processing Service (WCPS), Best Practices, http://portal.opengeospatial.org/files/?artifact_id=14022, 2006 [10.02.07]

[OGC3] *Whiteside, A., Evans, J.D.* (Edts.): Web Coverage Service (WCS) , Implementation Specification, https://portal.opengeospatial.org/files/?artifact_id=18153, 2006 [10.02.07]

[OGC4] *Percivall, G.*(Edt.).: OWS1.2, Image Handling Design, Disskussion Paper, http://portal.opengeospatial.org/files/?artifact_id=6621, 2004 [10.02.07]

[OGC5] *Whiteside, A.* (Edt.): OWS1.2, Image Handling Requirements, Disskussion Paper, http://portal.opengeospatial.org/files/?artifact_id=6660, 2004 [10.02.07]

[OGC6] *Coene, Y.* (Edt.): OWS-3 Imagery Workflow Experiments, Disskussion Paper, http://portal.opengeospatial.org/files/?artifact_id=13916, 2006 [10.02.07]

[OGC7] *Burry, L.* (Edt.): Grid Coverage , Implementation Specification, http://portal.opengeospatial.org/files/?artifact_id=6628, 2001 [10.02.07]

[OGP] OGP Surveying & Positioning Committee, http://www.epsg.org/, 2007 [10.02.07]

[Ora1] Oracle Database 10gR2: Performance Tuning Guide, http://download-uk.oracle.com/docs/cd/B19306_01/server.102/b14211.pdf, 2005 [10.02.07]

[Ora2] Oracle Database 10gR2: Concepts, http://dba-services.berkeley.edu/docs/oracle/manual-10gR2/server.102/b14220.pdf, 2005 [10.02.07]

[Ora3] Oracle Spatial 10gR2: GeoRaster,http://dba-services.berkeley.edu/docs/oracle/manual-10gR2/appdev.102/b14254.pdf, 2005 [10.02.07]

[Pag96] *Pagel, B.-U.*: Analyse und Optimierung von Indexstrukturen in Geodatenbanksystemen, Sankt Augustin: infix, 1996, (Dissertationen zu Datenbanken und Informationssystemen, Bd. 14)

[PG06] *The PostgreSQL Global Development Group*: PostgreSQL 8.2 Documentation , http://www.postgresql.org/files/documentation/pdf/8.2/postgresql-8.2-A4.pdf, 2006 [10.02.07]

[Rap00] *Raper, J.*: Multidimensional Geographic Information Science, 1. Aufl., London: Taylor & Francis, 2000

[Rei01] *Reihs, F.*: Erstellung, Anwendung und Qualitätsuntersuchung von Digitalen Geländemodellen zur Überflutungssimulation, Universität Rostock FB LKU, Diplomarbeit, 2001

[Rig02] *Rigaux, P.*, et al: Spatial databases with application to gis, 1. Auflage, San Francisco: Morgan Kaufmann Publishers, 2002

[Sar98] *Sarocco, C.M.*: Universal database management: a guide to object/relational technology, 3. Auflage, San Francisco: Morgan Kaufmann Publishers, 1998

[Schm02] *Schmidt, I.:* Retrieval in Multimedia-Datenbanksystemen, http://www.datenbank-spektrum.de/pdf/dbs-04-28.pdf (Datenbankspektrum, Vol.8 Nr.04, S.28-35) , 2002 [27.10.06]

[Schu04] *Schulz, T.*: Terrestrisches Laserscanning und seine Anwendung in der Ingenieurvermessung, http://www.geometh.ethz.ch/downloads/SchulzT_IV2004.pdf, 2004
Laserscanning - Genauigkeitsbetrachtungen und Anwendungen, http://www.geometh.ethz.ch/downloads/TSchulz_Oldenburg2004.pdf, 2004 [03.03.07]

[Stiff04] *Stiffel, M.*: Anfragengetriebenes Index-Tuning in PostreSQL, Technische Universität Ilmenau FB Informatik und Automatisierung, http://mordor.prakinf.tu-ilmenau.de/papers/dbis/Diplomarbeiten/Stiffel04.pdf, Diplomarbeit, 2004 [10.02.07]

[Tauch03] *Tauch, R., et al*: Integration geodätischer und photogrammetrischer Informationen zur Erstellung von Orthophotos mit Archimedes3D, Berlin: FPK Ing.-Gesellschaft, http://www.al-wie.de/lit/D3D_TauchWie.pdf, 2003 [03.03.07]

[Tho96] *Thomasian, A.*: Database concurrency control: Methods, Performance an Analysis, 1. Aufl., Boston/London/Dordrecht: Kluwer Academic Publishers, 1996

[TIFF] TIFF/GeoTIFF-Spezifikation, http://www.remotesensing.org/libtiff/, http://www.remotesensing.org/geotiff/geotiff.html

[Traub06] *Traub, K.., Kohlus, J.* (Hrsg.): GIS im Küstenzonenmanagement, 1. Auflage, Heidelberg: H. Wichmann Verlag, 2006

[Tri06] *Triglav, J.*: Bid Performance Challenges Can't be Solved with Just Hardware, GeoInformatics Nr. 8 Vol. 9, Emmeloord: cmedia productions bv, S.36-39, 2006

[Vorn05] *Vornberger, O.*: Datenbanksysteme, 2005, http://www.inf.uos.de/~dbs/2005/PDF/folie-04.pdf [08.02.07]

[Vos00] *Vossen, G.*: Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme, 4. korrigierte und erg. Auflage, München: Oldenburg Verlag, 2000

[Wev06] *Wevers, R.*: Data Compression Techniques and Formats, GeoInformatics Nr. 8 Vol. 9, Emmeloord: cmedia productions bv, S.28-31,2006

[Wil91] *Wilke, H.D.*: Oracle Datenbank-Management professionell: Design, Realisierung und Optimierung, 2. überarbeitete Auflage, Bonn/München/Reading: Addison-Wesley, 1991

[Wong98] *Wong, S.T.C (Edt.)*: Medical Image Databases, 1. Auflage, Boston/Dordrecht/London: Kluwer Academic Publishers, 1998

Anlage A Materialien zur Indexuntersuchung

A.1 Programm (SAX-BASIC) zur Beispieldatenerstellung (500000 Datensätze)

```
Sub Main
Open "c:\indextest.sql" For Output As #2
a=1
startx = 4400000
starty = 6000000
kachelgroesse=500
For i=1 To 1000
For j=1 To 500
k= LTrim(Str(kachelgroesse*i))+startx
l= LTrim(Str(kachelgroesse*j))+starty
m= LTrim(Str((i+1)*kachelgroesse))+startx
n = LTrim(Str((j+1)*kachelgroesse))+starty
Print #2, "INSERT INTO INDEXTEST VALUES (" & Str(a) & ",'" & k & "_" & l & "',GeomFromText('POLYGON((" & k & " " & l & "," & m & " " & l & "," & m & " " & n & "," & k & " " & n & "," & k & " " & l & "))',2398));"
a=a+1
Next j
Next i
Close #2
End Sub
```

A.2 SQL-Befehlsfolgen zur Indexerstellung

```
CREATE INDEX indextest_btree ON indextest USING btree (name);
Laufzeit: 22094ms
CREATE INDEX indextest_hash ON indextest USING hash (name);
Laufzeit: 20188ms
CREATE INDEX indextest_gist ON indextest USING gist (geom GIST_GEOMETRY_OPS);
Laufzeit: 63235 ms
```

A.3 Abfragen der beispielhaften Szenarien

```
1A -    SELECT * FROM indextest WHERE name='4600500_6054000';

1B -    SELECT * FROM indextest WHERE name>'4600500_6054000' and name<'4620500_6054000';
1C -    SELECT * FROM indextest WHERE name like '461%';
2A -    SELECT * FROM indextest
        WHERE geom @> GeomFromText('POINT(4510008 6110105)',2398);
2B1 -   SELECT * FROM indextest
        WHERE distance( geom, GeomFromText( 'POINT(4510250 6110250)', 2398 ) ) < 500;
2B2 -   SELECT * FROM indextest
        WHERE geom && buffer (GeomFromText( 'POINT(4510250 6110250)', 2398 ),500);
2C -    SELECT * FROM indextest
        WHERE geom && geomfromtext('POLYGON((4530000 6190000,4538660.25
        6185000,4553660.25 6210980.76,4545000 6215980.76,4530000 6190000))',2398);
2D -    SELECT * FROM indextest
        WHERE geom && SETSRID('BOX3D(4470000 6020000, 4488000 6060010)'::box3d,2398);
```

A.4 Planausgabe des PostgreSQL-Systems durch EXPLAIN ANALYZE

Ohne Indexverwendung

```
1A  -
"Seq Scan on indextest  (cost=0.00..17362.00 rows=1 width=136) (actual time=308.592..1067.934
rows=1 loops=1)"
"  Filter: (name = '4600500_6054000'::text)"
"Total runtime: 1068.084 ms"

1B
"Seq Scan on indextest  (cost=0.00..18612.00 rows=19367 width=136) (actual
time=76.312..1235.905 rows=19999 loops=1)"
"  Filter: ((name > '4600500_6054000'::text) AND (name < '4620500_6054000'::text))"
"Total runtime: 1319.612 ms"

1C
"Seq Scan on indextest  (cost=0.00..17362.00 rows=9959 width=136) (actual time=68.429..629.091
rows=10000 loops=1)"
"  Filter: (name ~~ '461%'::text)"
"Total runtime: 665.830 ms"

2A
"Seq Scan on indextest  (cost=0.00..18612.00 rows=500 width=136) (actual
time=7896.043..14130.552 rows=1 loops=1)"
"  Filter: ((geom)::box @> '(4510008.5,6110105.5),(4510007.5,6110104.5)'::box)"
"Total runtime: 14130.630 ms"

2B1
```

"Seq Scan on indextest (cost=0.00..18612.00 rows=166667 width=136) (actual time=7497.333..12243.821 rows=9 loops=1)"
" Filter: (distance(geom, '010100.....00A4F5741'::geometry) < 500::double precision)"
"Total runtime: 12243.896 ms"

2B2
"Seq Scan on indextest (cost=0.00..17362.00 rows=2 width=136) (actual time=175.545..387.067 rows=9 loops=1)"
" Filter: (geom && '01030000205E090.....000800A4F5741'::geometry)"
"Total runtime: 387.129 ms"

2C
"Seq Scan on indextest (cost=0.00..17362.00 rows=2138 width=136) (actual time=260.955..395.409 rows=3087 loops=1)"
" Filter: (geom && '01030000205E.....EC9C5741'::geometry)"
"Total runtime: 401.306 ms"

2D
"Seq Scan on indextest (cost=0.00..17362.00 rows=3226 width=136) (actual time=10.023..406.347 rows=3116 loops=1)"
" Filter: (geom && '0103000020.....8F65641'::geometry)"
"Total runtime: 412.117 ms"

Verwendung eines B-Tree-Index

1A
"Index Scan using indextest_btree on indextest (cost=0.00..8.12 rows=1 width=136) (actual time=0.070..0.073 rows=1 loops=1)"
" Index Cond: (name = '4600500_6054000'::text)"
"Total runtime: 0.165 ms"

1B
"Bitmap Heap Scan on indextest (cost=565.08..12326.77 rows=18898 width=136) (actual time=23.916..128.352 rows=19999 loops=1)"
" Recheck Cond: ((name > '4600500_6054000'::text) AND (name < '4620500_6054000'::text))"
" -> Bitmap Index Scan on indextest_btree (cost=0.00..565.08 rows=18898 width=0) (actual time=23.656..23.656 rows=19999 loops=1)"
"   Index Cond: ((name > '4600500_6054000'::text) AND (name < '4620500_6054000'::text))"
"Total runtime: 206.927 ms"

1C
"Bitmap Heap Scan on indextest (cost=311.06..11787.87 rows=10296 width=136) (actual time=4.864..26.532 rows=10000 loops=1)"
" Filter: (name ~~ '461%'::text)"
" -> Bitmap Index Scan on indextest_btree (cost=0.00..311.06 rows=10296 width=0) (actual time=4.793..4.793 rows=10000 loops=1)"
"   Index Cond: ((name >= '461'::text) AND (name < '462'::text))"
"Total runtime: 45.301 ms"

Verwendung eines Hash-Index

1A
"Index Scan using indextest_hash on indextest (cost=0.00..8.20 rows=1 width=136) (actual time=0.025..0.038 rows=1 loops=1)"
" Index Cond: (name = '4600500_6054000'::text)"

"Total runtime: 0.078 ms"

1B
"Seq Scan on indextest (cost=0.00..18612.00 rows=19420 width=136) (actual time=30.282..493.508 rows=19999 loops=1)"
" Filter: ((name > '4600500_6054000'::text) AND (name < '4620500_6054000'::text))"
"Total runtime: 531.331 ms"

1C
"Seq Scan on indextest (cost=0.00..17362.00 rows=10169 width=136) (actual time=28.571..371.620 rows=10000 loops=1)"
" Filter: (name ~~ '461%'::text)"
"Total runtime: 389.904 ms"

Verwendung eines GIST-Index

2A
"Seq Scan on indextest (cost=0.00..18612.00 rows=500 width=136) (actual time=487.184..1094.374 rows=1 loops=1)"
" Filter: ((geom)::box @> '(4510008.5,6110105.5),(4510007.5,6110104.5)'::box)"
"Total runtime: 1094.468 ms"

2B1
"Seq Scan on indextest (cost=0.00..18612.00 rows=166667 width=136) (actual time=534.444..1196.616 rows=9 loops=1)"
" Filter: (distance(geom, '01010000205E090000000000808A345141000000800A4F5741'::geometry) < 500::double precision)"
"Total runtime: 1196.675 ms"

2B2
"Index Scan using indextest_gist on indextest (cost=0.00..8.16 rows=1 width=136) (actual time=0.050..0.146 rows=9 loops=1)"
" Index Cond: (geom && '01030000205E0.....0800A4F5741'::geometry)"
" Filter: (geom && '01030000.....00000800A4F5741'::geometry)"
"Total runtime: 0.203 ms"

2C
"Bitmap Heap Scan on indextest (cost=106.53..6828.98 rows=2985 width=136) (actual time=4.178..20.842 rows=3087 loops=1)"
" Filter: (geom && '01030.....00EC9C5741'::geometry)"
" -> Bitmap Index Scan on indextest_gist (cost=0.00..106.53 rows=2985 width=0) (actual time=4.049..4.049 rows=3087 loops=1)"
"   Index Cond: (geom && '01030000.....C5741'::geometry)"
"Total runtime: 34.082 ms"

2D
"Bitmap Heap Scan on indextest (cost=111.40..6998.31 rows=3102 width=136) (actual time=3.898..20.297 rows=3116 loops=1)"
" Filter: (geom && '0103000.....F65641'::geometry)"
" -> Bitmap Index Scan on indextest_gist (cost=0.00..111.40 rows=3102 width=0) (actual time=3.704..3.704 rows=3116 loops=1)"
"   Index Cond: (geom && '01030.....65641'::geometry)"
"Total runtime: 32.895 ms"