Großvolumige Rasterkarten in Open-Source Datenbanken

Peter Baumann[1, 2], Vikram Unnithan[1], Angela Schäfer[1]

[1]International University Bremen, Campus Ring 1, 28759 Bremen, E-Mail: p.baumann@iu-bremen.de, v.unnithan@iu-bremen.de, a.schaefer@iu-bremen.de

[2]rasdaman GmbH, Adam-Berg-Str. 172a, 81735 München, E-Mail: baumann@rasdaman.com

1 Einleitung

Die zunehmende Hardware-Leistungsfähigkeit, aber auch die Entwicklung offener Web-Standards ermöglichen heute den direkten Zugriff auf großvolumige Rasterdaten, entweder konventionell in Dateien verwaltet oder bereits in Standard-Datenbanken. So betreiben beispielsweise die Bundesländer Brandenburg und Thüringen sowie Vattenfall Europe Mining AG blattschnittfreie Rasterdatenbanken.

In diesem Beitrag berichten wir über unsere Arbeiten an Open-Source-gestützten Rasterdatenbanken für Geo-Anwendungen. Der Rasterserver rasdaman wurde an das quellfrei verfügbare Datenbanksystem PostgreSQL adaptiert, und derzeit laufen mehrere Aktivitäten, diese Kopplung einzusetzen. Ein kommerzieller Einsatz ist das Rasterarchiv des französischen Nationalen Geographischen Instituts, welches Frankreich und seine exterritorialen Gebiete abdeckt; ein Projekt aus der Forschung ist der Aufbau des Geo-Datenzentrums für das International Research Consortium on Continental Margins. Zu einem küstennahen Meeresgebiet ergänzt diese Datenbank die Meta- und Vektordaten um einen Datenpool aus Satellitendaten, ozeanographischen Daten und 3D-geophysikalischen Daten des Meeresbodens. Ziel ist, eine web-gestützte übergreifende Analyse dieser Multi-Terabyte-Objekte zu ermöglichen.

Als Kandidaten für die Open-Source-Datenbanken standen i.w. mySQL und PostgreSQL zur Verfügung. Die Entscheidung für PostgreSQL fiel schließlich aus zwei Gründen: weil für PostgreSQL bereits umfangreiche Geodaten-Unterstützung vorhanden ist, so dass beim Aufbau integrierter Datenbanken z.B. auf Vektorkomponenten zurückgegriffen werden kann, und weil der Auftraggeber IGN France dieses System forderte. Geplant ist, in Zukunft auch mySQL zu testen.

Bereits jetzt lässt sich ersehen, dass PostgreSQL neben der einfachen Installation eine erfreuliche Stabilität und Performance zeigt. Damit werden Open-Source-Datenbanken auch für großvolumige Rasterarchive im Produktiveinsatz interessant.

In Abschnitt 2 wird die Kombination rasdaman /PostgreSQL vorgestellt. Derzeitige und künftige praktische Einsatzfelder für diese Kopplung werden in Abschnitt 3 diskutiert. Die Abschnitte 4 und 5 gehen auf den Stand der Technik und Standardisierung ein. Eine Zusammenfassung wird in Abschnitt 6 gegeben.

2 Der Raster-Server rasdaman

Die rasdaman-Technologie ergänzt Standard-Datenbanksysteme um die Verwaltung sehr großer, multidimensionaler Rasterobjekte. Die Daten werden intern partitioniert und integriert in der Datenbank abgelegt (also nicht in Dateien „neben“ der Datenbank). Der rasdaman-Server, welcher Hardware- und Software-Parallelität nutzt, bietet Navigation und Manipulation derartiger Rasterobjekte über die Anfragesprache rasql an, welche – auf Basis einer formal-algebraischen Semantik – Standard-SQL um multidimensionale Rasterausdrücke erweitert.

Das rasdaman-System entstand im Rahmen mehrerer EU-Projekte und ist inzwischen als kommerzielles Produkt der Ausgründung rasdaman GmbH in 13 Nationen installiert, zumeist als Geo-Server für diverse Kartentypen wie Luft/Satellitenkarten, topographische Karten und Höhendaten. Eine kurze Übersicht zu rasdaman findet sich z. B. in Baumann (2004), eine vertiefende Darstellung in Baumann (2005a, 2005b). Ein knappes Beispiel möge die Funktionsweise der Anfragesprache veranschaulichen.

Gegeben sei eine blattschnittfreie Karte aus 7-Kanal-Landsatdaten (Abbildung 1). Zu bestimmen sei die Anzahl grüner Pixel im Bereich, der durch die Boundingbox (x0,y0) und (x1,y1) gegeben ist; dabei vereinfachen wir für unsere

Zwecke die Definition von „grün“ dahingehend, dass eine einfache Vergleich gegen einen Schwellwert *s* auf dem Grünkanal stattfindet. Die folgende Anfrage leistet dies:

SELECT count_cells(m.green[x0:x1,y0:y1] > s) FROM LandsatMap AS m

In eckigen Klammern wird die Boundingbox angegeben, welche eine Ausschnittsbildung bewirkt. Der Klammerausdruck „*m.green>s*“, eine sog. induzierte Operation, wird für jedes Pixel ausgewertet: zuerst Extraktion des green-Bands, dann der Vergleich mit der Konstanten *s*. Das Ergebnis ist pro Pixel ein boolescher Wert, mithin entsteht also eine boolesche Matrix. Der *count_cells()* Operator liefert schließlich die Anzahl *true*-Pixel.

Diese Anfrageschnittstelle dient, wie SQL allgemein, lediglich als Schnittstelle zwischen Applikation und Datenbank-Server, ist jedoch keinesfalls als Endbenutzer-Schnittstelle gedacht. Für die interaktive Web-Navigation wurde das Geo-Frontend rasgeo realisiert. Es bietet geo-spezifische Funktionalität auf blattschnittfreien Karten für Navigation, Überlagerung, Visualisierung, sowie großvolumigen Export. Eine WMS-Schnittstelle (WMS 2002) erlaubt den Zugriff mit dem rasgeo-Browser-Frontend oder jedem anderen WMS-konformen Client. Intern werden sämtliche WMS- und sonstige Anfragen auf rasdaman-Datenbankanfragen abgebildet, welche unmittelbar die gewünschten Rasterobjekte im angeforderten Datenformat (etwa JPEG für Web-Anwendungen, TIFF für Datenexport) generiert. Demnächst wird eine WCS-Schnittstelle freigegeben. Zusätzlich erlaubt die Anfragesprache über ein graphisch-interaktives Werkzeug bzw. eine Kommandozeilen-Schnittstelle die Suche und Manipulation auf beliebig-dimensionalen Rasterobjekten. Dies ist insbesondere relevant im Kontext der Erdsystemforschung, wo etwa 3D-Seismikdaten und 4D-Klimamodelle auftreten.

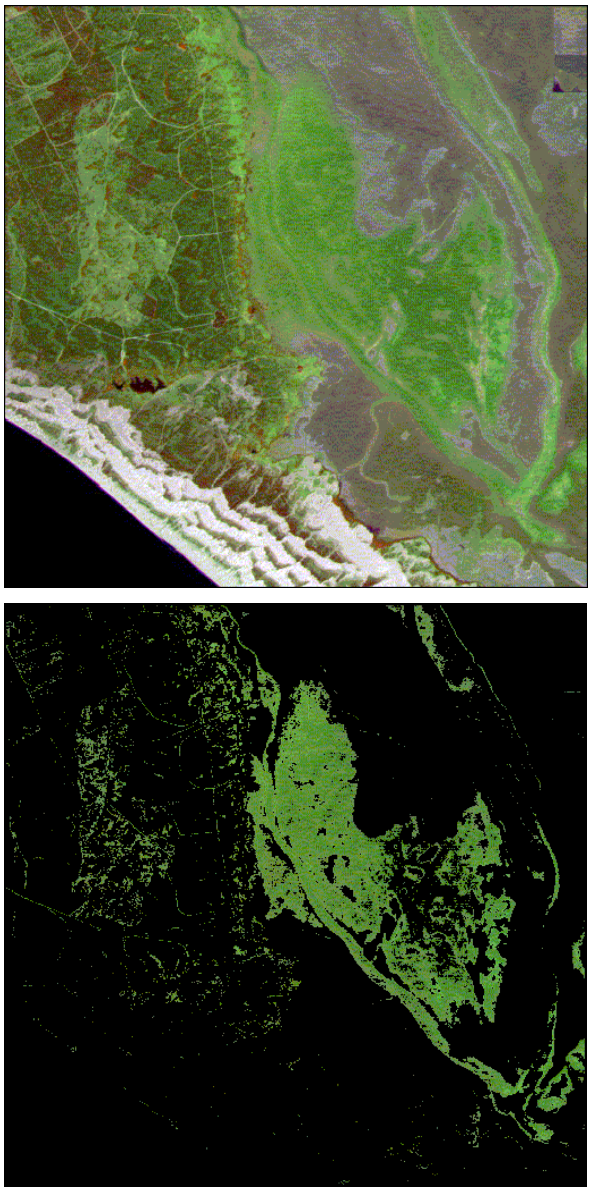

Abb. 1: Landsat-Szene (oben) und naive Vegetationsextraktion (unten).

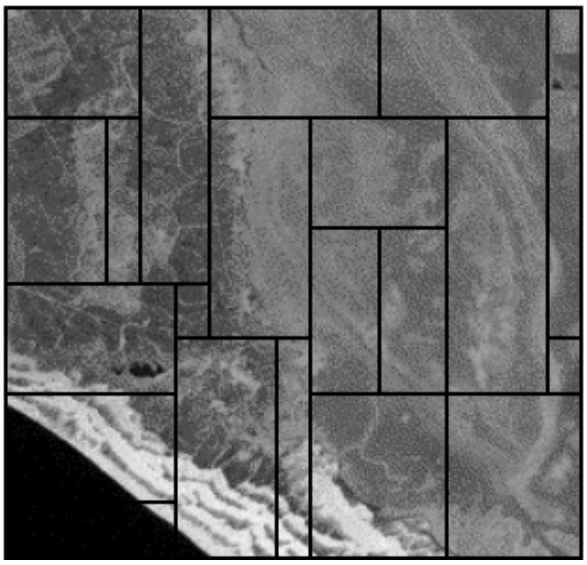

Abb. 2: Partitionierung eines zweidimensionalen Rasterobjekts.

3 Rasterdaten in PostgreSQL

Der Rasterserver rasdaman ist als Middleware konzipiert, welche n-dimensionale Rasterobjekte in Standard-Datenbanken ablegt. Dazu werden die Objekte in sogenannte Kacheln (*tiles*) partitioniert (Abbildung 2), pro Kachel wird ein Blob-Objekt (Blob = *binary large object*; ein Datenbank-Datentyp für variabel lange Byte-Container ohne weitere Semantik; LORIE 1982) in der Datenbank angelegt. Die Art der Partitionierung und damit die Blob-Größe ist als Tuning-Parameter in rasdaman einstellbar. Typischerweise

liegt bei PC-Servern die Blob-Größe bei etwa 1 MB.

Zur Anpassung von rasdaman an PostgreSQL war die Implementierung der Treiberschicht in rasdaman erforderlich, welche die Low-Level-Datenbankzugriffe erledigt. Dies beinhaltet einerseits klassische Tabellenstrukturen mit alphanumerischen Datentypen für das Data Dictionnary von rasdaman, andererseits die Zugriffsmechanismen für die Blobs. Wie aus der Erfahrung mit den anderen vorhandenen Adaptern (Oracle, DB2, Informix) bereits bekannt, war die Anpassung der regulären Tabellenzugriffe unproblematisch und fast ohne Änderungen zu übernehmen; die Abfrage von detaillierten Fehlersituationen bildete eine Ausnahme. Neu implementiert werden musste der Blob-Zugriff – auch dies ein bekanntes Phänomen, da Blobs in SQL nicht wirklich standardisiert sind.

Interessanterweise erlaubt PostgreSQL, anders als alle bisher von rasdaman unterstützten Datenbanksysteme, die direkte Definition von multidimensionalen Arrays über allen vorhandenen Basistypen als Tabellenattribute. Allerdings ist dies für rasdaman nicht nutzbar, da keine Speicheroptimierung (räumliche Clusterung etc.) vorhanden ist. Es blieb daher bei den Blobs wie auch bei anderen Systemen.

PostgreSQL bietet zwei Blob-Typen an, *bytea* und *LO*. Im ersten Verfahren, *bytea*, werden die Blob-Daten innerhalb der Tabelle abgelegt. Die Handhabung von Binärdaten erfordert eine Ersatzdarstellung für manche reservierte Bytewerte, etwa Anführungszeichen und binäre Null. Diese Ersatzdarstellung belegt drei Bytes. Nun ist eine Grundregel der Implementierung von Datenbankdiensten, Kopieren von Puffern zu vermeiden, und genau dies wird durch das jedesmalige Umkopieren erforderlich. Daher ist *bytea* für die Zwecke von rasdaman, wo permanent beliebige Bytemuster auftreten, wenig geeignet.

Der zweite Blob-Typ, *Large Object API* (kurz: *LO*), ist nicht in SQL eingebettet, sondern verwendet wie bereits der Name sagt eine eigene Schnittstelle. Diese ist, wie bei den meisten Systemen, nach Art der Dateibearbeitungs-Systemaufrufe gestaltet: es gibt Aufrufe *lo_creat()*, *lo_open()*, *lo_close()*, *lo_read()*, *lo_write()* etc. Die Größenbeschränkung von 2 GB ist für rasdaman irrelevant, da bei den derzeitigen Hardware-Architekturen von PCs die optimale Blobgröße weit darunter liegt. Abgelegt werden die Blobs in separaten Bereichen außerhalb der Tabellen.

Aus diesen Gründen schied *bytea* aus, so dass die Implementierung nun *LO* benutzt.

Die Installation von PostgreSQL benötigt gcc und ggf. einige weitere Pakete (etwa Java, falls

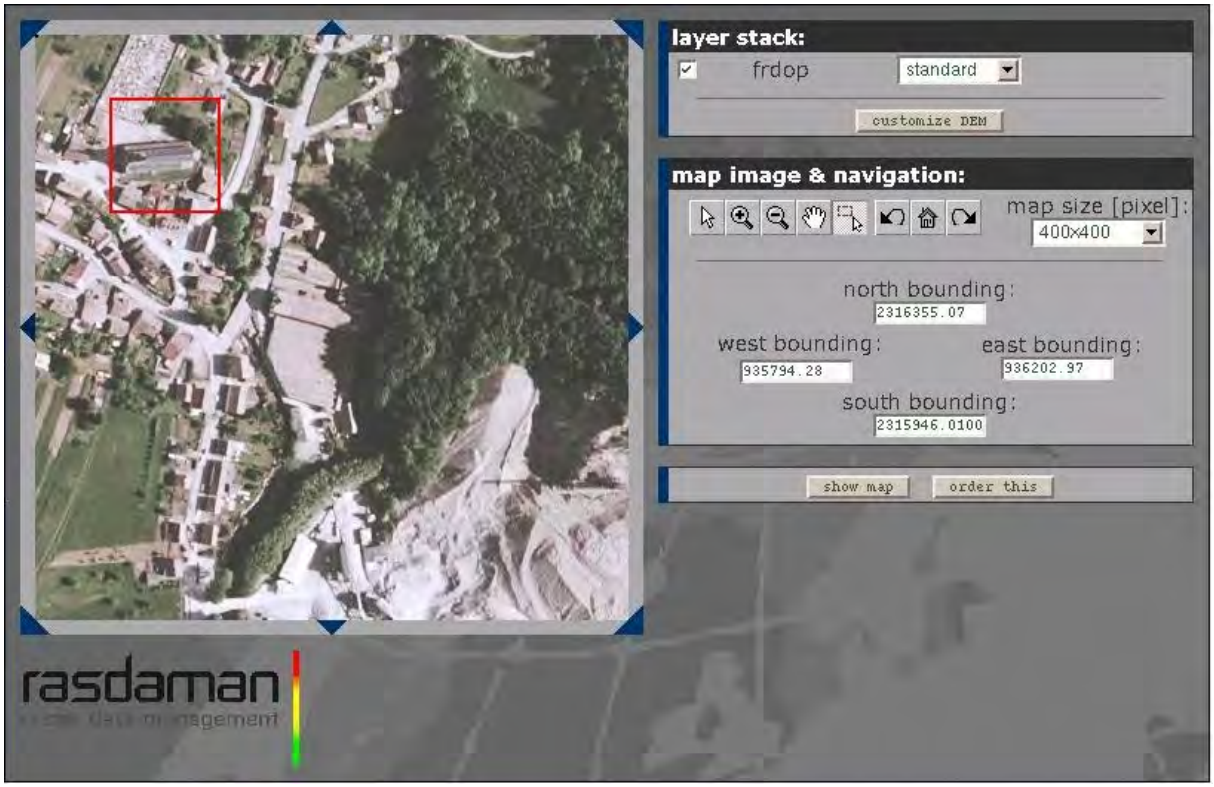

Abb. 3: Web-Client-Zugriff auf Rasterdaten des IGN über die WMS-Schnittstelle.

JDBC verwendet werden soll) und erfolgt nach dem üblichen Schema bei Open-Source-Systemen: Entpacken des heruntergeladenen Pakets; Konfigurieren (./configure); Generieren (gmake); Installieren, wahlweise als Superuser (gmake install); aus Sicherheitsgründen empfohlen: Anlegen einer eigenen Kennung „postgres“ für die Datenbankdateien.

PostgreSQL verwaltet Datenbanken in sogenannten *database clusters*; jeder Cluster entspricht einem Verzeichnis auf der Festplatte, er kann jeweils mehrere Datenbanken enthalten. Nach Initialisieren eines Clusters mit dem Kommando initdb und Starten des Servers (Kommando postmaster) kann mit *createdb* eine Datenbank angelegt werden.

Diese Schritte sind bei uns problemlos verlaufen; im Gegensatz zu kommerziellen Systemen, welche des öfteren Probleme z.B. mit der (immer benötigten) Java-Installation haben, ist hier Übersetzen und Binden des gesamten PostgreSQL-Codes auf Anhieb fehlerfrei durchgelaufen. Selbstverständlich ist bei dieser Betrachtung zu berücksichtigen, dass besagte kommerzielle Systeme einen erheblich größeren Leistungsumfang besitzen, von dem allerdings üblicherweise in Applikationen nur ein Bruchteil tatsächlich eingesetzt wird.

Die Zugriffsgeschwindigkeit konnte noch nicht systematisch gemessen werden, ist jedoch gefühlsmäßig ungefähr bei den Antwortzeiten der kommerziellen Systeme angesiedelt.

4 Praktischer Einsatz

In diesem Abschnitt stellen wir aktuelle Arbeiten vor, die auf dem praktischen Einsatz der Kombination rasdaman/PostgreSQL aufbauen. Seit Anfang 2005 ist ein derartiger Rasterserver beim französischen Nationalen Geographischen Institut (IGN) in Paris installiert. Im Rahmen des Aufbaus eines Datenzentrums für marine Forschung wird derzeit eine Datenbank für 2D/3D-Rasterdaten aufgebaut.

4.1 IGN France

Die erste praktische Installation der rasdaman/PostgreSQL-Kopplung erfolgte Anfang 2005 beim französischen IGN in Paris. Das IGN setzt derzeit einen Web-Service auf, der strikt auf offenen Standards und weitestgehend auf Open-Source-Software basiert. Die Metadatenhaltung erfolgt im XML-Datenbanksystem eXist, Vektordatenhaltung wird von dem kanadischen Hersteller Global Geomatics geliefert, die Rasterdatenhaltung erfolgt in rasdaman / PostgreSQL. Zur Integration des WMS-basierten Vektor/Raster-Zugriffs wird ein kaskadierender WMS-Dienst eingesetzt.

Abb. 4: SeepFinder, ein an der IUB entwickelter Crawler (Unterwasser-Roboter).

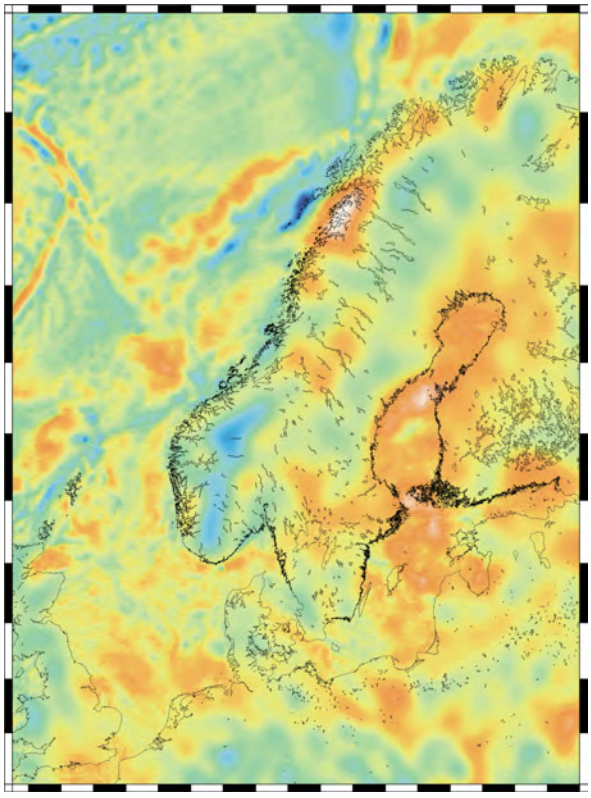

Abb. 5: Kartierung der Gravitationsanomalien an der norwegischen Küste. Helle Farbe zeigt hohe, dunkle niedrige Anomalien an.

Rasterseitig sind derzeit XX GB importiert. Hardwareseitig laufen rasdaman und PostgreSQL auf unterschiedlichen Servern, da die Datenbank auch für andere Zwecke benutzt wird. Der Zugriff über das Lokalnetz hat sich als unkritisch hinsichtlich der Zugriffsgeschwindigkeit erwiesen.

4.2 IRCCM Datenzentrum

Das *International Consortium on Continental Margins* (IRCCM, www.irccm.de) ist eine transatlantische Partnerschaft im Bereich mariner Forschung und Ausbildung. Das Konsortium wurde formal im Juni 2002 gegründet, um die Prozesse an den kontinentalen Rändern zu studieren, insbesondere wie Flüssigkeitstransport, den Methan/Karbon-Zyklus sowie Bildgewinnung und –auswertung an den Oberflächen und darunter. IRCCM baut auf den drei Säulen Forschung, Infrastruktur und Ausbildung. IRCCM stellt eine Plattform für den Zugang zu Grossgeräten bereit (z.B. insgesamt 22 Schiffe), arbeitet an der Entwicklung innovativer mariner Forschungstechnologien (z.B. Unterwasser-Roboter, Abbildung 4) und bietet eine Vielfalt von Ausbildungsmöglichkeiten und –richtungen für direkte und intensive Kooperation zwischen Industrie und Forschung. Zu den insgesamt ca. 10 Partnern zählen u.a. Statoil, Maersk, das Alfred-Wegener-Institut Bremerhaven, International University Bremen (IUB) sowie Rice University und mehrere weitere US-Universitäten. Mitte 2004 wurde zwischen IUB und IRCCM ein Konzept beschlossen, wonach IUB das Datenzentrum aufbauen soll. Basierend auf einem umfassenden Modell soll freier Zugang zu kombinierten Erdbeobachtungs-, ozeanographischen und seismischen Daten ermöglicht werden (Abbildung 5). Das Modell muss Daten unterschiedlicher Herkunft, Dimensionalität und Charakteristika in einheitlicher Weise abbilden, so dass übergreifende Analyseanfragen beantwortbar werden. Das online zugreifbare Datenvolumen des operativen Systems dürfte mindestens mehrere Terabyte betragen, betrachtet man die bisher bereits von den Partnern gesammelten Daten.

Die Realisierung des Datenzentrums erfolgt auf einem PC-Cluster mit Windows- und Linux-Maschinen. Vektordaten werden mit ESRI-Produkten verwaltet, Rasterdaten mit rasdaman. Unter anderem aus Kostengründen wurde entschieden, für die Rasterdatenhaltung ein Open-Source-Datenbanksystem zu verwenden. Die Wahl fiel auf PostgreSQL, da rasdaman bereits einen Adapter hierfür bietet.

Der Zugang zu den Daten wird u.a. über OGC-konforme Dienste erfolgen, insbesondere WMS und WCS. Weiterhin sollen Raster-Analyse-Methoden via Web zur Verfügung gestellt, werden, um komplexere ad-hoc-Analysen auch auf umfangreicheren Datenvolumina „vor Ort“ und ohne Download zu ermöglichen.

Für IUB und IRCCM bilden diese Arbeiten, in denen die üblichen Meta- und Vektordaten um grossvolumige multidimensionale Rasterdaten zu einem offenen, standard-basierten Webdienst ergänzt werden, einen Meilenstein auf dem Weg zu einem umfassenden, integrierten Langzeit-Datenmanagement. Dazu soll Zusammenarbeit mit verwandten existierenden Diensten aufgebaut werden, etwa mit dem World Data Center MARE/PANGAEA (www.pangaea.de) und dem EU-Projekt Hermes. Weiterhin wurde IUB kürzlich Mitglied in GiN (Geoinformatik Niedersachsen, www.gin-online.de). Die Vision ist, aufbauend auf dieser kritischen Masse ein World Data Center für integrierte Erdsystemdaten zu etablieren, welches alle relevanten Disziplinen und Informationskategorien umfasst.

```
declare
        g sdo_georaster;
        b blob;
begin
        select raster into g
        from uk_rasters
        where id = 4;
        dbms_lob.createTemporary(b,true);
        sdo_geor.getRasterSubset(
                georaster => g,
                pyramidlevel => 0,
                window => sdo_number_array(0,0,699,899),
                bandnumbers => '0',
                rasterBlob => b);
end;
```

```
select g.red[0:699,0:899]
from   uk_rasters as g
where  oid(g) = 4
```

Abb. 6: Der gleiche Rasterzugriff formuliert in Oracle GeoRaster (links, aus Oracle-Vertriebspräsentation) und rasql (rechts).

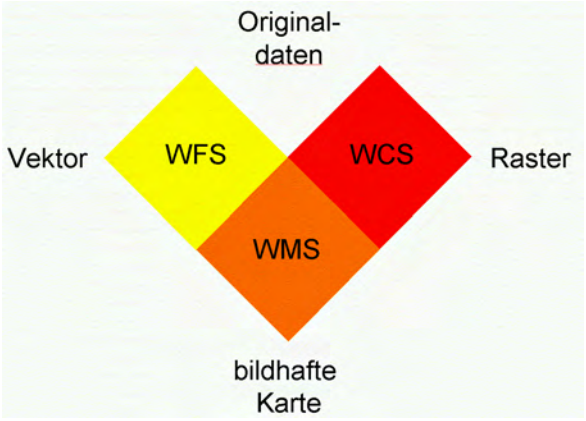

Abb. 7: WMS als Überlappung von Vektor- und Rasterdiensten, allerdings mit Schwerpunkt Bildgenerierung.

5 Stand der Technik

In der Datenbank-Forschung sind bisher bemerkenswert wenige Aktivitäten bei Rasterdatenbanken zu verzeichnen. Einige konzeptuelle Modelle sind vorgestellt worden (BAUMANN 1994; LIBKIN et al. 1996; MARATHE & SALEM 1999). Die optimierte Speicherung großvolumiger Geo-Daten in Bandarchiven wird in SARAWAGI & STONEBRAKER (1994) und REINER et al. (2002) untersucht, Parallelisierung von Geo-Anfragen in HAHN & REINER (2002). Von dem Versuch, Rasterdaten mit objektrelationalen Mitteln (STONEBRAKER et al. 1998) zu implementieren, ist man i.w. wieder abgekommen – da Raster kein Datentyp sind, sondern ein Datentyp-*Konstruktor*, welcher parametrisiert ist mit Dimension, Ausdehnung und Zelltyp, bieten die objektrelationalen Konzepte keinen wirkliche Hilfe, insbesondere nicht für Speicherverwaltung und Optimierung.

Lediglich Oracle scheint intern die Erweiterungsmechanismen zu nutzen. In Oracle 10g wird unter der Bezeichnung *GeoRaster* datenbankseitige Unterstützung für Rasterdaten angeboten. Dabei handelt es sich um einfache Zugriffsoperationen, beschränkt auf 2D-Rasterbilder und die gängigsten Rastertypen. Soweit bekannt bzw. dem Autor von Oracle-Mitarbeitern mitgeteilt findet keine interne Raster-Optimierung statt. Abbildung 6 zeigt eine einfache Zugriffsoperation, formuliert auf Basis von GeoRaster und zum Vergleich in rasql.

ESRI bietet in seiner aktuellen ArcSDE-Version ebenfalls Datenbank-Unterstützung für 2D-Rasterdaten mit Geo-Semantik. Dem rasdaman-Ansatz folgend werden blattschnittfreie Karten partitioniert und in Blobs abgelegt. Der Speicherzugriff geschieht über einfache Lese- und Schreiboperationen, ohne spezifische Raster-Anfragesprache; höherwertige Dienste werden in den oberen Schichten der ESRI-Architektur realisiert. Ein Nachteil dabei ist, dass der Speicherebene kaum Wissen für Optimierung mitgegeben werden kann. Die Aktualisierung einer Karte erfordert den völligen Neuaufbau der intern mitgeführten Bildpyramide und ist damit bei großen Objekten sehr zeitaufwändig.

Insgesamt sind industriell punktuell Systeme mit datenbankgestützter Verwaltung von 2D-Rasterkarten zu finden, jedoch insgesamt derzeit kein Ansatz erkennbar, der multidimensionale Verwaltung und Datenanalyse in der Art von rasdaman bietet. Insbesondere ist bisher die Verwaltung von Rasterdaten in Open-Source-Datenbanken nicht untersucht worden.

6 Standardisierung

Treibende Kraft im Bereich der Standards für interoperable, web-gestützte Geo-Dienste ist das OpenGeospatial Consortium (OGC, www.opengis.org). Historisch der erste Ansatz zur Einbindung von Rasterdaten war der *Web Map Service Implementation Specification* (WMS), aktuell in der Version 1.1.1 vorliegend (WMS 2002. WMS erlaubt die blattschnittfreie Darstellung von Vektor- und Rasterkarten, einzeln oder in Kombination, als Rasterbild, welches vom Server dynamisch aus dem Kartenmaterial generiert wird. Dieses Kartenbild kann z.B. unmittelbar in einem Web-Browser angezeigt werden. Da dieser Dienst sehr einfache Web-Clients ermöglicht, wurde WMS von der Industrie bisher mit Abstand am besten aufgenommen.

Neben diesem Dienst zur Generierung von Kartenbildern existieren weitere zur Abfrage der Basisdaten (Abbildung 7). Zur Vektorabfrage steht die *Web Feature Service Implementation Specification* (WFS) zur Verfügung; das Pendant auf Rasterseite bildet die *Web Coverage Service Implementation Specification* (WCS; WCS 2003). Der WCS erlaubt die Ausschnittsbildung in Raum, Zeit und Kanälen (*domain and range subsetting*) sowie Reprojektion auf Coverage-Daten, d.h. derzeit: auf zwei- und mehrdimensionalen Rasterobjekten. Derzeit arbeiten wir an einer Erweiterung des WCS um benutzerdefinierte komplexe Ausdrücke, was in mehreren Vorteilen resultieren soll:

- neuartige Anforderungen erfordern lediglich die Formulierung neuer Requests, ohne client- oder serverseitige Programmierung;

- auch komplexere Aufgaben lassen sich ohne clientseitige Programmierung lösen, im Klartext: komplexe Verarbeitungsleistung lässt sich über Web-Browser abrufen und steuern;
- da nur das jeweilige Endprodukt zum Client geliefert wird, reduziert sich für viele Anfragen der Netztransfer erheblich;
- für die Bearbeitung der üblicherweise sehr datenintensiven Algorithmen kann im Server die jeweils angemessene Rechenleistung bereitgestellt werden.

Die hier vorgestellten Arbeiten liefern wichtige Erfahrungen und Demonstrationsimplementierungen zur Unterstützung des Standardisierungsvorschlags.

7 Zusammenfassung und Ausblick

In unseren Arbeiten versuchen wir zwei aktuelle Trends in Geo-Diensten zusammenzuführen: einerseits werden zunehmend Datenbanksysteme eingesetzt, da die klassischen Vorteile – Informationsintegration, Skalierbarkeit, Offenheit, Flexibilität etc. – inzwischen erkannt worden sind. Andererseits ergänzen in wachsendem Umfang Rasterdaten das Online-Geodaten-Angebot, da auch Multi-Terabyte-Volumina heute kein ernsthaftes Hindernis mehr darstellen. In der Kombination, also datenbankgestützten Raster-Diensten, sehen wir einen wichtigen Schritt, um die gleiche Dienstequalitäten, welche bereits für Meta- und Vektordaten existieren, auch für Rasterdaten zu ermöglichen.

Da die Kosten kommerzieller Datenbanksysteme oft beträchtlich sind, bietet sich an, Open-Source-Datenbanksysteme in die Betrachtung einzubeziehen. Wir wollen untersuchen, inwieweit solche Systeme, die unseres Wissens noch nie mit solchen Multi-Terabyte-Lasten konfrontiert worden sind, stabil und performant bleiben.

Erste Ergebnisse mit PostgreSQL sind vielversprechend. Die Installation ist erheblich komplikationsfreier als bei vielen kommerziellen Systemen, bei vergleichbarer Stabilität (im Rahmen der geschilderten Bedingungen). Verlässliche Leistungsvergleichsdaten liegen derzeit noch nicht vor. Offensichtlich bestimmt die Leistungsfähigkeit der Blob-Implementierungen der verschiedenen Datenbanksysteme die Leistungsfähigkeit der Rasterzugriffe wesentlich mit. Im Rahmen studentischer Arbeiten wurde daher ein Benchmark für Blobs entwickelt, der im nächsten Schritt auf mehrere kommerzielle und kostenfrei verfügbare Datenbanksysteme angewendet werden soll. Auf dieser Basis können dann auch Tuning-Massnahmen gezielt angegangen werden, z.B. die Einstellung der jeweils für das Ziel-Datenbanksystem optimalen Kachelgröße. Auch können ggf. Optimierungen im PostgreSQL-Code vorgenommen werden, unter Einbeziehung der Entwickler-Gemeinschaft.

Insgesamt sollen als nächste Schritte die vorgestellten ersten experimentellen Ergebnisse konsolidiert und durch den praktischen Einsatz in den beschriebenen Feldern validiert werden.

8 Danksagung

Für die hervorragende Zusammenarbeit bedankt sich Peter Baumann herzlich bei Didier Richard, Projektleiter Geodatenbank, IGN France.

9 Literatur

BAUMANN P. (1994): On the Management of Multidimensional Discrete Data. – VLDB Journal, Special Issue on Spatial Database Systems, **4**(3): 401—444.

BAUMANN P. (2004): Die Verwaltung großvolumiger Geo-Rasterdaten in Standard-Datenbanken. – Wiss. Mitt., 25: 15—22; Bergmännische Tage, Freiberg, 2004.

BAUMANN P. (2005a): Modellierung und Analyse von 3D-Rasterdaten in Geodatenbanken. – In: COORS V., ZIPF A. (Hrsg.): 3D-Geoinformationssysteme. Hüthig Verlag.

BAUMANN P. (2005b): Web-gestützte Analysetechniken für mehrdimensionale Geo-Rasterdatenbanken. – In: COORS V., ZIPF A. (Hrsg.): 3D-Geoinformationssysteme. Hüthig Verlag.

HAHN K., REINER B. (2002): Parallel Query Support for Multidimensional Data: Inter-object Parallelism. – Proc. DEXA, Aix en Provence, France, 2002.

LIBKIN L., MACHLIN R., WONG L. (1996): A Query Language for Multidimensional Arrays: Design, Implementation, and Optimization Techniques. – Proc. ACM SIGMOD'96, Montreal, Canada, 1996: 228—239.

LORIE R.A. (1982): Issues in Databases for Design Transactions. – In: ENCARNAÇAO J., KRAUSE F.L. (Eds.): File Structures and Databases for CAD. North-Holland Publishing, IFIP, 1982.

MARATHE A.P., SALEM K. (1999): Query Processing Techniques for Arrays. – Proc. ACM SIGMOD '99, Philadelphia, USA, 1999: 323—334.

SARAWAGI S., STONEBRAKER M. (1994): Efficient Organization of Large Multidimensional Arrays. – Proc. ICDE'94, Houston, USA, 1994: 328—336.

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

REINER B., HAHN K., HÖFLING G., BAUMANN P. (2002): Hierarchical Storage Support and Management for Large-Scale Multidimensional Array Database Management Systems. – 13th International Conference on Database and Expert Systems Applications (DEXA), September 2-6, 2002, Aix en Provence, France.

WCS (2003) – In: EVANS J. (Ed.): Web Coverage Service (WCS), Version 1.0.0. OGC document 03-065r6, August 2003; verfügbar von www. opengis.org

WMS (2002) – In: DE LA BEAUJARDIERE J. (Ed.): Web Map Service Implementation Specification, Version 1.1.1, OGC document 01-068r3, January 2002; verfügbar von www. opengis.org