Umweltdatenbanken 2003
(Workshop AK „Umweltdatenbanken“, GI-Fachgruppe 4.6.1 „Informatik im Umweltschutz“),
May 19-20, 2003, Berlin, Germany

Integrierte Meß- und Bilddatenhaltung für Umweltdatenbanken

- Vortrag mit Live-Demonstration -

Peter Baumann

Rasdaman GmbH, Bunsenstr. 4, D-81735 München
E-Mail: baumann@rasdaman.com

Kurzfassung

Traditionell werden Luft/Satellitenbilder und andere Rasterdaten im Dateisystem gehalten, während die Ablage von Meta- und Vektordaten in Datenbanken heutzutage mehr oder weniger Standard ist. Erst jüngst wird Datenbankunterstützung auch für große Rasterdaten in größerem Umfang diskutiert und gefordert.
Im EU-Projekt EuroClim findet erstmalig eine datenbankgestützte Integration unterschiedlich dimensionaler Rasterobjekte statt. Als Datenhaltungsplattform für die Rasterdaten dient rasdaman, welches die klassischen Datenbankvorteile Anfragesprache, serverseitige Optimierung, Transaktions- und Speicherverwaltung für Rasterdaten aller Art und Dimension bereitstellt. Die Datenbank-Modellierung und Beispielsanfragen darauf werden in diesem Beitrag vorgestellt.

1 Motivation

Traditionell werden Luft/Satellitenbilder und andere Rasterdaten im Dateisystem gehalten, während die Ablage von Meta- und Vektordaten in Datenbanken heutzutage mehr oder weniger Standard ist. Erst jüngst wird Datenbankunterstützung auch für große Rasterdaten in größerem Umfang diskutiert und gefordert. Es steht zu erwarten, dass sich in der nächsten Zeit Rasterdatenbanken auch kommerziell durchsetzen. So sieht Oracle nach eigener Aussage auf der GITA Annual Conference 2003 in der 2D- und 3D-Rasterunterstützung eine der beiden Kerninnovationen im Bereich Geodatenbanken in den nächsten Jahren.
Derzeit verfügbar am Markt ist allerdings hauptsächlich dateibasierte Rasterdatenhaltung. Entsprechende datenbankbasierte Produkte von (z.B.) ESRI und Oracle sind in Vorbereitung.
Bereits seit mehreren Jahren verfügbar und auch im praktischen Einsatz befindlich ist dagegen der Rasterserver rasdaman. Rasdaman ist Client/Server-Middleware mit einer Anfragesprache, welche ISO-SQL92 um deklarative Raster-Operatoren erweitert. Die parallele Anfrageauswertung benutzt transparente Mechanismen zur Verbesserung der Performance und Speicherausnutzung, z.B. regelbasierte Optimierung, automatische Partitionierung (Kachelung), Geo-Indexierung, Kompression. Die Ablage erfolgt in einer relationalen Datenbank, etwa Oracle, IBM DB2 oder IBM Informix. Anfragesprache, Optimierung und Speicherverwaltung basieren auf der rasdaman Array-Algebra als mathematischem Fundament.
In mehreren EU-Projekten wurde rasdaman eingesetzt bzw. weiterentwickelt, u.a. RasDaMan (www.forwiss.tu-muenchen.de/~rasdaman), ESTEDI (www.estedi.org), FOREMMS (foremms.nr.no) EuroClim (euroclim.nr.no). Derzeit existieren rasdaman-Installationen in 12 Nationen (einschl. USA und GUS) eingesetzt, hauptsächlich für Geo-Rasterserver[1]. Der holistische Ansatz, der bei der Entwicklung von rasdaman verfolgt wurde, macht das System außerdem zu einer geeigneten Basis für die Untersuchung aller Aspekte datenbankgestützter Rasterdatenhaltung. Im vorliegenden Beitrag wird der Einsatz von rasdaman in den vorgenanntenen Projekten EuroClim und FOREMMS präsentiert. Der Schwerpunkt liegt auf konzeptueller Ebene, da sich EuroClim derzeit in der Spezifikationsphase befindet. Insbesondere wird gezeigt, wie Meß- und Bilddaten übergreifend abgefragt werden können.

[1] in Deutschland u.a. als offizieller Rasterserver mehrerer Landesvermessungsämter, etwa Thüringen und Bayern

Der Rest dieses Beitrags gliedert sich wie folgt. In Kapitel 2 wird ein Überblick über konzeptuelles Modell und Architektur von rasdaman gegeben. In Kapitel 3 werden Meß-, Simulations- und Bilddaten in 1D-, 2D-, 3D- und 4D diskutiert. In Kapitel 4 wird die Live-Demo des Vortrags skizziert. Kapitel 5 gibt eine Zusammenfassung.

2 Das rasdaman-System

2.1 Konzeptuelles Modell

Das konzeptuelle Modell von rasdaman basiert auf n-dimensionalen Arrays (im Programmiersprachensinn, also Rasterdaten), welche von beliebiger Dimension und Ausdehnung sein können [Bau94]. Das dynamische Typsystem [Wid00] unterstützt alle C/C++-Typen als Raster-Zellentyp („Pixeltyp"), einschließlich geschachtelter `structs` und ausschließlich Zeigertypen (welche in der Datenbank keinen Sinn machen). Die Ober- und Untergrenze jeder Dimension kann zur Typdefinitionszeit festgelegt oder variabel gelassen werden. Die Typdefinition geschieht durch die Definitionssprache rasdl, welche auf der ODMG ODL (Object Definition Language) basiert [Cat96]. Ein Rastertyp wird definiert mittels eines Templates

```
marray<b,d>
```

welches mit einem Zellentyp `b` und der Raster-Ausdehnung (*spatial domain*) `d`, instantiiert wird. Die Ausdehnung wird angegeben durch die Unter- und Obergrenze pro Dimension, wobei jeder Wert durch "*" ersetzt werden kann, um dynamische Feldgrenzen fest zu legen. Beispielsweise kann ein blattschnittfreies Rasterbild unbegrenzter Ausdehnung definiert werden durch

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

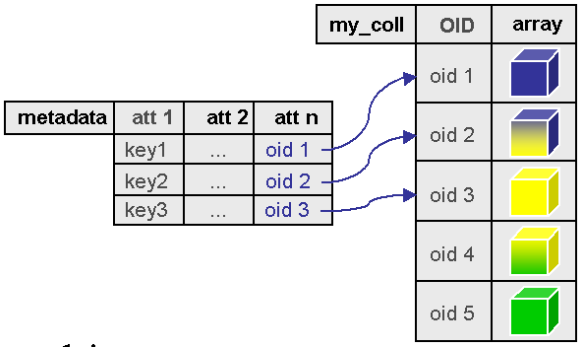

Abb. 1: Verweise von relationalen Metadaten auf Rasterobjekte in rasdaman

Durch Angabe von Punktkoordinaten, Begrenzungsboxen oder –polygonen lassen sich Regionen auch innerhalb eines Rasterobjekts referenzieren. Selbstverständlich ist aber auch inhaltsbezogene Referenzierung möglich, in dem mit dem Geoobjekt eine Anfrage abgelegt wird, die beim Zugriff entsprechende Informationsextraktion bewirkt. Beispielsweise könnte dies sein „*Alle Gebiete im DEM mit Höhen zwischen 0 und 100 m über NN*".

2.2 Raster-Anfragen

Die rasdaman-Anfragesprache rasql bietet Rasterprimitive eingebettet in SQL92; wie üblich gibt eine `select`-Anfrage eine homogene Resultatmenge zurück. Raster-wertige Ausdrücke können sowohl in der `select`- als auch in der `where`-Klausel verwendet werden.
Trimming liefert einen rechteckigen Ausschnitt.
Beispiel: "Ein Ausschnitt mit Eckpunkten (1000,1000) und (2000,2000) aus den DOPs":

```
select OrthoCollection[1000:2000,1000:2000]
from   OrthoCollection
```

Für jede Operation auf dem Zelltyp wird vom Server automatisch eine entsprechende *induzierte Operation* bereitgestellt, welche diese Operation simultan auf alle Zellen eines Objekts anwendet. Sowohl unäre (z.B. Komponentenzugriff) als auch binäre Operationen (z.B. Jmaskeirung und Überlagerung) lassen sich induzieren.
Beispiel: "Topographical map bit layer 3 oveölaid with the (grayscale) ortho image":

```
select OrthoCollection overlay bit( Map, 3 ) * 255c
from   Map, OrthoCollection
```

Im allgemeinen können Rasterausdrücke im `select`-Teil einer Anfrage auftreten und, wenn der Ergebnistyp der äußersten Operation vom Typ Bool ist, auch im `where`-Teil. Die Mächtigkeit von rasql umfasst viele Signalverarbeitungs-, Bildverarbeitungs- und Statistikoperationen, bis hin zu beispielsweise Fourier-Transformationen; sie endet bei rekursiven Operationen, so dass jede Anfrage terminiert, mithin rasql sicher in der Auswertung ist. Details der unterliegenden rasdaman Array-Algebra sind in [Bau99] beschrieben.

2.3 Anfrageverarbeitung

In rasdaman stehen etwa 150 Ersetzungsregeln für heuristische Optimierung zur Verfügung [Rit99]. Beispiele sind "*pull out disjunctions while aggregating cell values of a raster object using logical or*" oder "*push down geometric operations to expression leaves*". Die letztere Regel erreicht, dass nur die minimale Datenmenge gelesen wird, welche für die Berechnung des Anfrageunterbaums erforderlich ist. Das Prinzip der *Common Subexpression Elimination* wurde um räumliche Überlappung erweitert, um Gebiete jeweils nur einmal zu laden. Die Kachel-basierte Abarbeitung des Anfragebaums bewirkt, dass pro Zeitpunkt nur die minimale erforderliche Kachelanzahl im Hauptspeicher gehalten werden muss.
Der rasdaman-Server unterstützt bereits Inter-Objekt-Parallelität, in dem eine Anfrage an einen Server im Pool zugewiesen wird. Derzeit erfolgt die Erweiterung um Intra-Objekt-Parallelität, so dass auch eine einzelne Anfrage auf mehrere CPUs bzw. Rechner verteilt werden kann [Hah02].

2.4 Speicherverwaltung

Raster-Objekte werden intern partitioniert in sog. Kacheln. Die Partitionierung ist nicht auf reguläre Gitter beschränkt. Eine Reihe von Kachelungsstrategien steht für den Datenbankprogrammierer zur Verfügung, um eine für bestimmte Anfragesets optimale Kachelung einzurichten. Nähere Beschreibungen und Benchmarks finden sich in [Fur99]. Jede Kachel wird in ein relationales BLOB gespeichert. Geoindexverfahren (u.a. R+-Baum) sorgen für die schnelle Selektion der für eine Anfrage benötigten Kacheln.

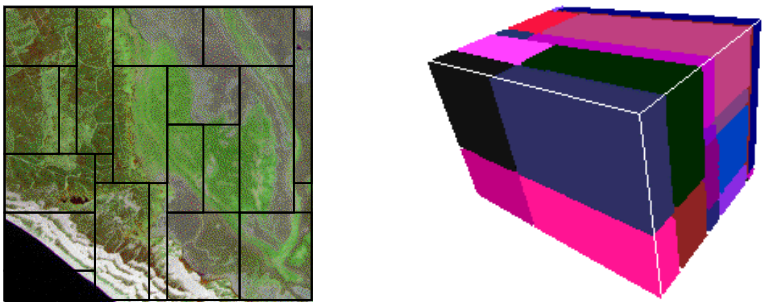

Abb. 2: irreguläre 2D- und 3D-Kachelungen

Heutige Plattensysteme erlauben bereits Terabyte-Archive; der Bedarf entwickelt sich jedoch rasant weiter, so dass für großvolumige Daten wie Satellitenarchive auch weiterhin Bandspeicher zum Einsatz kommen werden. Daher wurde für rasdaman eine Tertiärspeicher-Anbindung entwickelt [Rei02]. Die Ein- und Auslagerung erfolgt wahlweise automatisch oder manuell, wobei der Speicherverwalter räumliche Clusterung auf dem Tertiärspeicher berücksichtigt.

2.5 Architektur

Der rasdaman-Server ist als komplettes Client/Server-Datenbanksystem implementiert, mit eigenen Komponenten für Client/Server-Kommunikation, Anfrageanalyse und –optimierung, Katalogverwaltung etc. Die unterste Schicht nimmt eine Speicherabbildung auf relationale Blobs vor; Adapter existieren derzeit für Oracle, IBM DB2 und IBM Informix.
Ein verteilbarer Server-Manager teilt den Clients jeweils einen Serverprozess zu. Jeder Server kann unterschiedliche relationale Datenbanken ansprechen. Damit ist eine hohe Skalierbarkeit gegeben: während Datenbanken bereits auf PC-Servern mit 256 MB Hauptspeicher Zugriffe unter einer Sekunde erlauben, lassen sich ebenso große Serverfarmen aufbauen (Abb. 3).

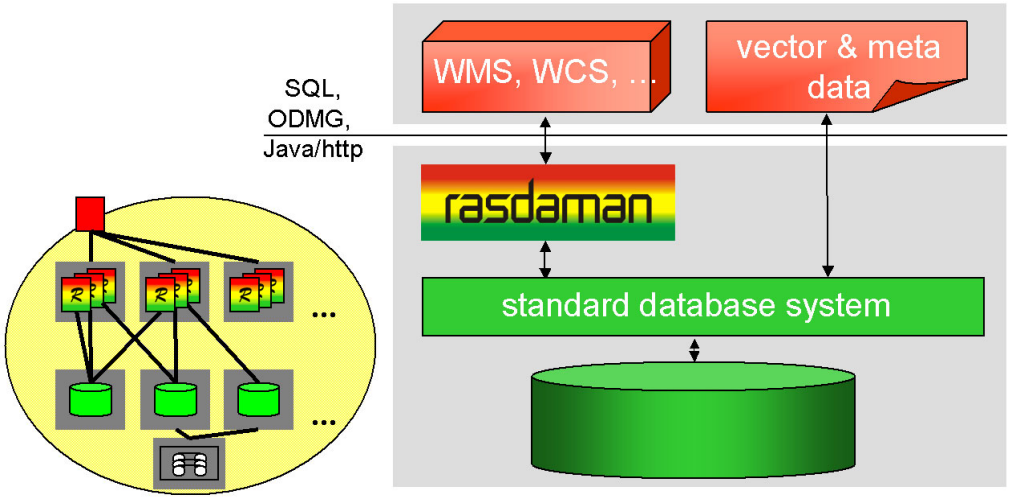

Abb. 3: rasdaman-Architektur
(rechts: Blockschaltbild, links: hochparallele Serverfarm mit zentralem Einsprungspunkt)

3 EuroClim

Im Rahmen des EU-IST-Projekts EuroClim (euroclim.nr.no) wird ein europaweites kryosphärisches Klima-Informationssystem, also mit Schwerpunkt aufEis-, Schnee- und Gletscherbedeckung, entwickelt. Unter der Leitung des Norwegian Computing Institute in Oslo arbeiten hauptsächlich nordeuropäische Teilnehmer aus den Bereichen Klimaforschung, Erdbeobachtung und Geo-IT zusammen. Derzeit befindet sich das Projekt in der Spezifikationsphase, i.w. ist das Schema für Satellitendaten entwickelt. Trotzdem lassen sich bereits die prinzipiellen Schemata darstellen.
Vorausgeschickt sei, dass die Datenaufbereitung (L1 bis L3) von eigener Software erledigt wird. Zum Zeitpunkt des Einfügens liegen die Daten radiometrisch korrigiert und georeferenziert vor. Das Einfügeformat bei Messreihen

3.1 Meßreihen (1D)

Eindimensionale Messreihen lassen sich folgendermassen modellieren. Der Zellentyp besteht aus einem Messwert und einem Indikator zur Validität; darüber wird ein eindimensionales Array mit offener Obergrenze definiert, entsprechend einer zeitlich unbegrenzten Laufzeit. Die rasdl-Definition lautet

```
typedef marray< struct { float value; char flag; }, [0:*] > TimeSeries;
```

Entsprechende Metadatenstrukturen sind im konventionellen Katalog zu definieren, um die zeitliche Auflösung, den Anfangszeitpunkt der Messungen, Sensorspezifikationen u.a. zu bestimmen.
Für `value` werden Gleitpunktzahlen entgegengenommen, die z.B. per Telemetrie automatisch in die Datenbank eingefügt werden können. Das entsprechende rasql-Update-Statement wäre etwa wie folgt.

```
update Sensor_X
set    SensorX[ sdom(Sensor_X)[0].hi+1 ].value assign $1,
       SensorX[ sdom(Sensor_X)[0].hi+1 ].flag assign $2
```

Hierbei sind die Positionsparameter `$i` durch die jeweiligen Eingangswerte zu substituieren. Die Standardfunktion `sdom(X)` liefert die aktuelle Begrenzungsbox zu Objekt X; hiervon wird Dimension 0 abgegriffen und davon wiederum die Obergrenze; durch Inkrementierung um 1 wird das Rasterobjekt erweitert, der übergebene Wert wird sodann der neuen Position zugewiesen.
Aus einem solcherart aufgebauten Sensorobjekt lassen sich Kenngrößen wie z.B. der bisher aufgetretene Maximalwert abfragen, wobei nur gültige Werte berücksichtigt werden (gekennzeichnet durch leeres Flag).

```
select max_cells( Sensor_X.value * Sensor_X.flag = ' ' )
from   SensorX
```

Der Einfachheit halber wurde hier angenommen, dass nur positive Meßwerte auftreten können. Damit lassen sich mit der Multiplikation über den Booleschen `flag`-Wert alle ungültigen Werte auf 0 setzen.

3.2 Satellitenbilder (2D)

Die im Projekt ausgewählten MODIS-Daten sind hyperspektraler Natur, d.h. ein aufgenommener Punkt hat eine große Zahl (hier: mehrere Dutzend) von spektralen Kanälen. Unter Ausnutzung räumlicher Clusterung lässt sich dies in rasdl definieren als:

```
typedef marray
< struct{ char band1, …, bandn; },
  [xmin:xmax,ymin:ymax] >
ModisImage;
```

mit entsprechend gewählten Begrenzungskoordinaten für Europa. Ziel ist, eine blattschnittfreie Datenbank zu erhalten, auf der beliebiges Zoom und Pan möglich ist. Auf einer solchen Datenbank lässt sich beispielsweise der normalisierte Vegetationsindex (NDVI) folgendermassen gewinnen.

```
select abs( m.near_infrared - m.red ) / ( m.near_infrared + m.red )
from   ModisCollection as m
```

Dabei wird angenommen, dass die Kanalbezeichnungen für nahes Infrarot bzw. Rot gegeben seien durch `near_infrared` und `red`.

3.3 Satellitenbild-Zeitreihen (3D)

Nach dem Schema der oben definierten Modis-Kollektion lässt sich mit einer kleinen Erweiterung ein blattschnittfreier 3D-Würfel aufbauen, dessen dritte Koordinate die Zeit bildet:

```
typedef marray
< struct{ char band1, …, bandn; },
  [xmin:xmax,ymin:ymax,tmin:*] >
ModisTimeseries;
```

Auf diesem 3D-Würfel lassen sich Schnitte entlang aller Dimensionen legen, beispielsweise ein Kartenausschnitt zum Zeitpunkt T:

```
select m[x0:x1,y0:y1,T]
from   ModisTimeseriesCollection as m
```

Aber auch ein Zugriff entlang der Zeitachse zu einer bestimmten Position x/y ist möglich; in diesem Fall ist das Resultat eindimensional; abgegriffen wird der gesamte verfügbare Zeitbereich:

```
select m[x,y,*:*]
from   ModisTimeseriesCollection as m
```

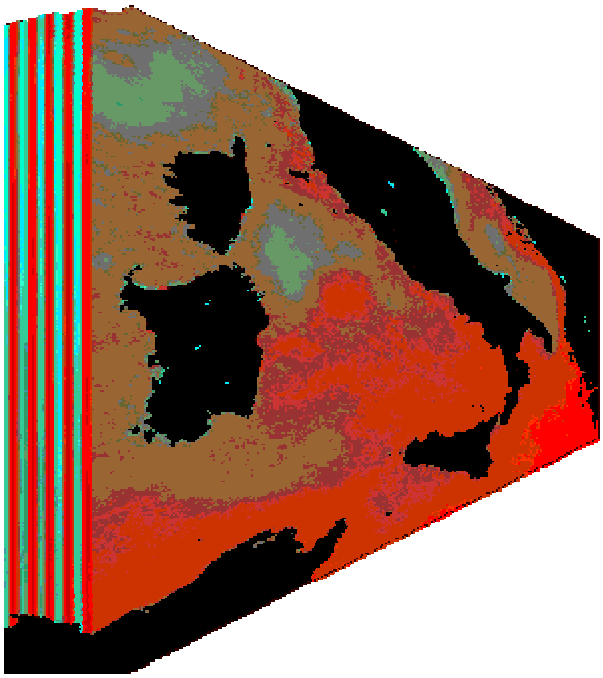

Abb. 4: blattschnittfreie Satellitenbild-Zeitreihe als 3D-Würfel
(mit frdl. Genehmigung DFD/DLR)

3.4 Dimensionsübergreifende Anfragen

Aufbauend auf den vorangehenden Definitionen lassen sich Rasterobjekte in Anfragen verknüpfen. Beispiel: „*den maximalen NDVI für alle Werte, an denen der Sensor an Position P=(x,y) eine Überschreitung des Schwellwerts T gemeldet hat*".
Dazu wird zuerst aus den (relational abgelegten) Metadaten über eine Standard-SQL-Anfrage der Sensor zu Position P selektiert:

```
select rasdaman_id
from   Sensors
where  posx = x and posy = y
```

Das Ergebnis ist der Identifikator des gewünschten Sensors in der rasdaman-Kollektion. Damit kann im zweiten Schritt die rasdaman-Anfrage formuliert werden:

```
select max_cells(
         abs( m.near_infrared - m.red ) / ( m.near_infrared + m.red )
         *
         ( s.flag = ' ' and s.value > T )
       )
from   ModisTimeseriesCollection as m, Sensors as s
where  s.id = $1
```

4 Demonstration

Die Demonstration von rasdaman erfolgt über das graphisch-interaktive Frontend rView und über Web-Browser. Die C++-Applikation rView erlaubt das Absetzen von rasql-Anfragen an eine Datenbank und die Visualisierung von 1D- bis 4D-Anfrageergebnissen. Das Java-Servlet rasgeo generiert puren HTML-Zugang für Standard-Webbrowser, um auf 2D-Karten zu navigieren, sie zu überlagern und einzufärben. Die Beispielsdatensätze umfassen mehrere Dutzend GB große 2D-Georaster (Orthophotos, TK, DHM), 3D-Zeitreihen und eine 4D-Klimasimulation.

5 Zusammenfassung

Rasterdaten bilden einen der letzten „weißen Flecken" auf der Landkarte der Internet-Geodienste. Die Integration der Rasterdatenhaltung in eine allgemeine, datenbankgestützte Verwaltung sämtlicher

relevanter Daten bietet eine Reihe von Vorteilen: zuvorderst die Integration der Datenhaltung, welche die Konsistenz erhöht und die Verwaltung erleichtert. Weiterhin lassen sich sämtliche bereits vorhandenen Datenbank-Werkzeuge und Mechanismen nutzbringend einsetzen, u.a. Views, Trigger/Constraints, Stored Procedures, Optimierung/Tuning, Transaktionskonzept, Zugriffsschutz, Replikation, Online-Backup, Report-Generatoren, 4GL-Werkzeuge. Schließlich gewinnt man mit in hohem Maße Flexibilität durch die Fähigkeit der Datenbanksysteme zur ad-hoc Definition neuer Anfragen und Tabellenstrukturen.
Erlaubt ein solcher Rasterserver die gleichzeitige Haltung von Daten unterschiedlicher Dimension (von 1D-Zeitreihen bis 4D-Klimamodellen), dann ergibt sich ein besonders hoher Integrationsgrad. Anfragen werden damit möglich, welche eine Verknüpfung der diversen Sensordaten erlauben (z.B. Ergebnisse von Messbojen korreliert mit Satellitenbildern Klimasimulationsdaten) und damit neuartige Dienstequalitäten hervorbringen.
Rasdaman ist das erste System seiner Art, welches ein mathematisch fundiertes Modell zur Definition und Bearbeitung von multidimensionalen Rasterdaten anbietet und gleichzeitig mit seiner gesamten Architektur diese speziellen Strukturen unterstützt. Eine große Zahl untersuchter Anwendungen hat gezeigt, dass dieser Ansatz tragfähig ist – was sich nicht zuletzt darin zeigt, dass große GIS- und Datenbankhersteller mittlerweile an eigenen Varianten der datenbankgestützten Rasterverwaltung arbeiten.
Aktuelle Themen der Weiterentwicklung von rasdaman konzentrieren sich auf den effizienten Zugriff mit besonders komplexen Anfragen bzw. auf extrem große Datenvolumina. FORWISS München (http://www.wibas.forwiss.tu-muenchen.de) arbeitet an Intra-Query-Parallelität zur Verteilung von Anfragen auf mehrere CPUs bzw. Rechner in Clustern [Hah02] sowie weiteren Anfrageoptimierungen sowie an der Einbindung von Bandrobotern mit besonderem Schwerpunkt auf räumliche Clusterung der Daten im Bandarchiv [Rei02]. Der Autor arbeitet aktiv im OGC an der Definition der Web Coverage Services (WCS) mit.

Danksagung

Für wertvolle Beiträge, Diskussionen und Implementierungen gilt der besondere Dank des Autors Karl Hahn, Bernd Reiner, Steve Pester, Asgeir Finnseth, Jason Baragry, Rune Solberg, Bernhard Buckl, Erhard Diedrich, Clemens Glock, Bruno Roder.

Referenzen

[Bau99] P. Baumann: *A Database Array Algebra for Spatio-Temporal Data and Beyond.* Next Generation Information Technology and Systems (NGITS) '99, Zikhron Yaakov, Israel, 1999.

[Bau94] P. Baumann: *On the Management of Multidimensional Discrete Data*. VLDB Journal 4(3)1994, Special Issue on Spatial Database Systems, pp. 401-444.

[Cat96] R. Cattell: *The Object Database Standard: ODMG-93*. Morgan Kaufmann Publishers, 1996.

[Fur99] P. Furtado, P. Baumann: *Storage of Multidimensional Arrays Based on Arbitrary Tiling. ICDE'99*, Sidney - Australia 1999.

[Hah02] Karl Hahn, Bernd Reiner, Gabriele Höfling, Peter Baumann: *Parallel Query Support for Multidimensional Data: Inter-object Parallelism*. 13th International Conference on Database and Expert Systems Applications (DEXA), September 2-6, 2002, Aix en Provence, France.

[Rei02] Bernd Reiner, Karl Hahn, Gabriele Höfling, Peter Baumann: *Hierarchical Storage Support and Management for Large-Scale Multidimensional Array Database Management Systems*. 3th International Conference on Database and Expert Systems Applications (DEXA), September 2-6, 2002, Aix en Provence, France.

[Rit99] R. Ritsch: Optimization and Evaluation of Array Queries in Database Management Systems. PhD Thesis, Technische Universität München, 1999.

[Wid00] N. Widmann: *Efficient Operation Execution on Multidimensional Array Data.* PhD Thesis, Technische Universität München, 2000.