

Anfrageformulierung und Ablage dimensionsbehafteter Daten in RasDaMan* (Kurzbeitrag)

Peter Baumann, Paula Furtado, Roland Ritsch, Norbert Widmann

Bayerisches Forschungszentrum für Wissensbasierte Systeme (FORWISS)
Orleansstr. 34, D-81667 München
email: {baumann,furtado,ritsch,widmann}@forwiss.tu-muenchen.de
Tel.: (89) 48095-200, Fax: (89) 48095-203

Kurzfassung

Im ESPRIT-Projekt *RasDaMan* (**R**aster **D**ata **M**anagement in Databases) entsteht Datenbanktechnologie für die Verwaltung Multidimensionaler Diskreter Daten (MDD) beliebiger Größe und Dimension. Besonderes Ziel ist, volle Datenunabhängigkeit für MDD zu erreichen. Dazu erlaubt die ODMG-konforme MDD-Definitionssprache *RasDL* eine semantisch vollständige Definition, auf deren Basis die deklarative, algebraisch optimierbare SQL-Erweiterung *RasML* assoziative MDD-Anfragen unterstützt. Die Anfragebearbeitung geschieht damit durch den Server, wo ein spezieller Speicherverwalter für beliebig große Arrays unter Einbeziehung von Tertiärspeichermedien für einen effizienten MDD-Zugriff sorgt.

Obwohl das System anwendungsneutral konzipiert ist, sieht das Projekt eine intensive Evaluierung des Systems vor: Prototypapplikationen in den Bereichen GIS und Medizin werden von Endanwendern im praktischen Einsatz getestet, um die Brauchbarkeit des Datenbanksystems für diese Gebiete nachzuweisen.

Im vorliegenden Beitrag geben wir einen Überblick über den RasDaMan-Ansatz im Kontext der im Projekt untersuchten Anwendungsgebiete.

1 Einleitung

Wiewohl Rasterbilder immer noch das Standardbeispiel für Rasterdaten darstellen, wird im Prinzip jedes natürliche Phänomen auf raum-zeitliche Rasterdaten einer bestimmten Dimension abgebildet, sobald es für die Speicherung und Bearbeitung in Computersystemen durch Sensoren erfaßt und diskretisiert wird; darüberhinaus produziert auch eine Vielzahl künstlicher Datenquellen wie Simulatoren, Rendering-Programme und OLAP-Software Rasterdaten unterschiedlicher Dimensionalität. Die verbindende Eigenschaft derartiger Informationstypen ist, daß eine große Zahl üblicherweise sehr großer multidimensionaler Felder (Arrays) zu verwalten ist; daher die Sammelbezeichnung *Multidimensionale Diskrete Daten* oder *MDD* [Baum-92].

Im dreijährigen Forschungsprojekt *RasDaMan*, das hier vorgestellt werden soll, entsteht ein anwendungsneutrales MDD-DBS gemäß [Baum-94]. Aufbauend auf einem kommerziellen ODBS entwickelt FORWISS - unter Berücksichtigung der relevanten Standards SQL-92 [ISO-92, Cann-93] und ODMG [Catt-96] - eine MDD-Definitions- und Anfragesprache sowie eine C++-Programmierschnittstelle. Von der Firma STI in Spanien und einem weiteren Partner wer-

* von der Europäischen Kommission im ESPRIT-Bereich Langzeitforschung unter Nr. 20073 gefördert.

den parallel dazu Pilotapplikationen in den Bereichen Medizin und Satellitenbilddauswertung implementiert, die anschließend in der katalanischen Klinik Hospital General de Manresa und dem Nationalen Geographischen Institut in Spanien einer praktischen Evaluierung unterzogen werden.

Im folgenden diskutieren wir in Abschnitt 2 verwandte Arbeiten; in Abschnitt 3 präsentieren wir den Ansatz von RasDaMan, um dann in Abschnitt 4 die im Projekt betrachteten Anwendungsfelder zu skizzieren. Abschnitt 5 gibt eine Zusammenfassung.

2 Verwandte Arbeiten

Zwar bilden MDD einen in vielen Anwendungen anzutreffenden Informationstyp mit wohldefinierter Datenstruktur, dennoch werden sie üblicherweise (sowohl in relationalen als auch in objektorientierten DBSen) linearisiert und in einem mehr oder weniger willkürlich gewählten Datenformat codiert in BLOBs, **binary large objects**, abgelegt, die auf die von Lorie bereits vor 15 Jahren vorgeschlagenen Langfelder zurückdatieren [Lori-82]. Da bei diesem Vorgehen das DBS keinerlei Information über die Semantik der Daten besitzt, fehlen flexible, deklarative Anfragemöglichkeiten, wie sie auf den konventionellen alphanumerischen Datentypen Standard sind. Darüberhinaus führt die spezielle Linearisierungsabbildung zu beträchtlichen Leistungseinbußen bei allen Zugriffsmustern jenseits sequentiellen, zeilenweisen Zugriffs.

Modellierungsmethoden objektorientierter [Kim-95] und objektrelationaler DBSe [Ston-95] erlauben die Definition eines Abstrakten Datentyps (ADTs) für MDD mit feststehender Dimensionalität. Das wohl bekannteste System dieser Kategorie ist Illustra [Ston-95]. Das DBS kann durch sogenannte DataBlades um zusätzliche Typen erweitert werden, wovon u.a. für eindimensionale Zeitreihen und zweidimensionale Bilder Erweiterungen existieren. Intern werden die Rasterdaten jedoch weiterhin auf ganze BLOBs abgebildet, wodurch der Zugriff ineffizient bleibt. Ein einheitlicher Datentyp für MDD sowie eine deklarative Anfragesprache mit speziellen Konstrukten zur MDD-Bearbeitung sind nicht vorhanden.

MDD-Speichersysteme ermöglichen die Speicherung großer Datenmengen durch eine achsenparallele Partitionierung der Rasterdaten in n-dimensionale Rechtecke, sogenannte *Kacheln*, die die räumliche Nachbarschaft der Pixelwerte erhalten. Eine Aufteilung in ausgerichtete Kacheln wurde bereits in [Tamu-80] beschrieben. Eine flexiblere Kachelungsmethode in nicht notwendigerweise ausgerichtete multidimensionale Rechtecke wird in [Furt-93, Furt-94] vorgestellt. Wiederum auf ausgerichteter Kachelung basierend, diskutiert [Sara-94] verschiedene Strategien zur Speicherung großer, multidimensionaler Felder auf Sekundär- und Tertiärspeichermedien. Das Speichersystem OPTIMASS [Chen-95] partitioniert multidimensionale Datensätze in Cluster, deren Aufteilung und Größe von gerätespezifischen Zugriffsgrößen und Anfragemustern abhängt. Keines der Systeme ist in ein DBS mit allgemeiner MDD-Anfragefunktionalität integriert.

Spezialisierte DBSe sind meist proprietäre Lösungen mit komplexen, anwendungsorientierten Operationen. Das Paradise-System [DeWi-94] ist ein Beispiel für ein DBS zur Verwaltung von 2D MDD im Bereich GIS. Zur effizienten Speicherung werden die Rasterdaten in eine Menge von Objekten aufgeteilt, die dann im Speichersystem SHORE [Care-94] abgelegt werden. Im objektorientierten Datenmodell von SHORE werden MDD als ADT modelliert. Paradise besitzt weder die Funktionalität zur Modellierung von MDD höherer Dimensionalität, noch ist eine allgemeine Anfragesprache für MDD vorhanden.

3 Der RasDaMan-Ansatz

Ein wesentliches Ziel von RasDaMan ist es, durch eine klare Trennung der logischen und physischen Ebene Datenunabhängigkeit für MDD zu erreichen. Zur konzeptuellen Behandlung wurde die deklarative Schnittstelle RasQL (Raster Query Language) entwickelt, welche sich

aus den Komponenten RasDL (Raster Definition Language) und RasML (Raster Manipulation Language) zusammensetzt. RasDL ist eine Erweiterung der ODMG Object Definition Language (ODL) um Elemente zur MDD-Definition. RasML erweitert die ODMG Object Manipulation Language (OML) - welche sich ihrerseits auf SQL-92 abstützt - um MDD-Anfrageprimitive (siehe Abschnitt 3.1). Auf der physischen Ebene wurde eine Architektur basierend auf MDD-Kachelung entworfen, welche speziell für die Bearbeitung beliebig großer MDD-Objekte geeignet ist (siehe die Abschnitte 3.2 und 3.3); die C++-Klassenbibliothek RasLib realisiert die Spracheinbettung.

3.1 Das konzeptuelle MDD-Modell

Im folgenden geben wir einen informellen Überblick über die RasDaMan-Anfrageschnittstelle; eine formale Behandlung, aufbauend auf der AFATL Image Algebra [Ritt-90], findet sich in [Baum-94]. MDD-Objekte (wir verwenden den Begriff *Objekt* hier in naiver Weise) in RasDaMan sind Felder mit einer beliebigen Anzahl von Dimensionen über einem beliebigen (unter Umständen zusammengesetzten) Basistyp. Dimension und Basistyp werden zur Definitionszeit festgelegt, während Ober- und Untergrenzen in der Typdefinition wahlweise fixiert werden oder variabel bleiben können. Dazu dient die Spezifikation der Raum-Domäne (*spatial domain*), die für ein n-dimensionales Feld mit Unter- und Obergrenzen lo_i und hi_i für jede Dimension $i \in \{1..n\}$ die syntaktische Form

$$[lo_1:hi_1, lo_2:hi_2, \dots, lo_n:hi_n]$$

annimmt; ein "*" anstelle von lo_i oder hi_i spezifiziert eine variable Grenze.

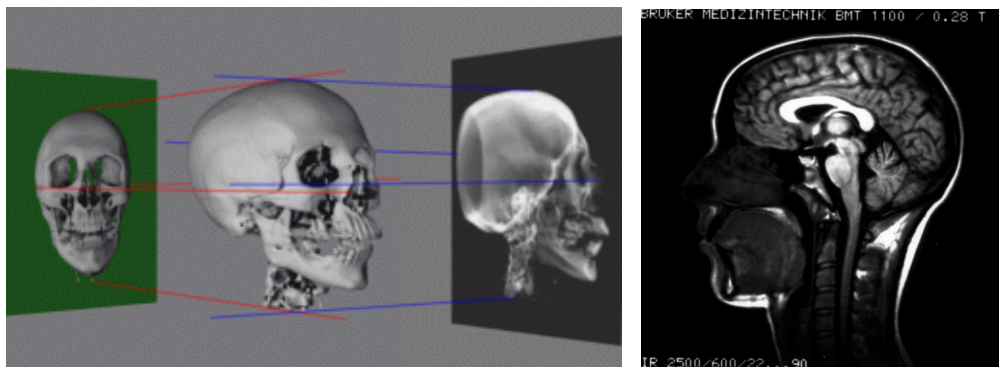


Abb. 1: 3D-Rekonstruktion eines menschlichen Schädels und 2D-Projektion

Die grundlegenden Konzepte der Anfragesprache werden anhand zweier Beispiele vorgestellt. Das erste entstammt dem klinischen Bereich, wo 3D-Tomogramme (Volumetomogramme, VT) schichtweise durch Abtastung eines Patienten mit einem 2D-bildgebenden Tomographen gewonnen werden. Typischerweise ist die x/y-Ausdehnung 256x256, während in z-Richtung eine variable Anzahl von Schnitten möglich ist (Abb. 1). Der Basistyp ist integer.

Das zweite Beispiel ist dem Bereich Fernerkundung/Photogrammetrie entnommen. Die Satellitenfamilie Landsat TM liefert Bilder der Größe 7020x7560 in acht Frequenzbereichen (Abb. 2); die Pixel bestehen somit aus einem Verbund von acht Grauwerten. Es ergibt sich eine Gesamtgröße (unkomprimiert) von 283 MB pro Bild.

Im folgenden setzen wir die Klassen `VolumeTomogram` und `LandsatImage` voraus, deren Instanzen im Attribut `data` die MDD-Information enthalten, und zeigen daran exemplarisch die Einbettung von MDD-Anfrageprimitiven in SQL.

Trimmen extrahiert einen Sub-Würfel aus einem MDD-Objekt, wobei die Dimension unverändert bleibt.

Beispiel 1: “Von allen Landsat-Bildern den Ausschnitt zwischen den Punkten (x_0, y_0) und (x_1, y_1) .”

```
select  li.data [ x0:x1,  y0:y1 ]
from    LandsatImage as li
```

Nahe verwandt dazu ist die *Projektion*, die eine achsenparallele Schicht der Dicke 1 ausschneidet.

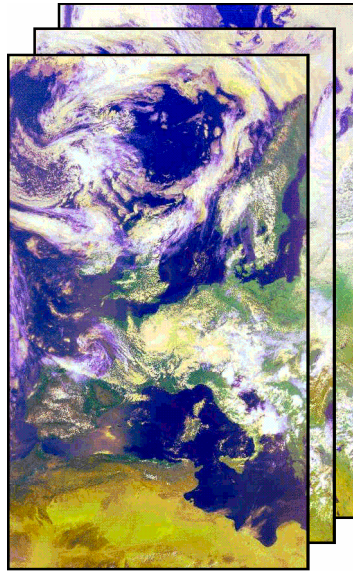


Abb. 2: Satellitenbild-Sequenz

Beispiel 2: “Für jedes VT drei achsenparallele Schnitte durch den Punkt (x_0, y_0, z_0) .” Der ***-Operator steht für die aktuelle Feldgrenze in der jeweiligen Dimension.

```
select  vt.data [ x0,  ** ,  ** ],
        vt.data [ ** ,  y0,  ** ],
        vt.data [ ** ,  ** ,  z0 ]
from    VolumeTomogram as vt
```

Operationen dieser Art verändern den MDD-Koordinatenbereich, jedoch nicht die Zellwerte. MDD-Inhalte lassen sich durch *induzierte Operationen* [Ritt-90] global verändern. Induzierte Operationen ergeben sich aus der simultanen Anwendung einer Operation, die über dem Basistyp definiert ist, auf sämtliche Zellen eines Feldes, wobei sowohl einstellige als auch binäre Induktion möglich ist.

Beispiel 3: “Von allen Landsat-Bildern ein Graubild, das aus den drei sichtbaren Kanälen gemittelt ist.” Verbund-Zugriff ist auf dem (zusammengesetzten) Zelltyp definiert und kann daher per Induktion zur Extraktion eines Kanals benutzt werden. Abschließend wird die integer-Division auf alle Ergebnispixel angewendet.

```
select  ( li.data.red + li.data.green + li.data.blue ) / 3
from    LandsatImage as li
```

Ein verallgemeinerter Aggregationsoperator kollabiert dimensionale Daten zu niedrigdimensionaleren MDD bis hin zu Skalaren. Ein *Kondensierer*-Ausdruck

```
condense o over x in K using e(x)
```

bindet x an den Indexbereich K und kombiniert für jede darin enthaltene Koordinate x die Ausdrücke $e(x)$ gemäß der Operation o . Die Reihenfolge der Index-Iteration ist undefiniert, wo-

durch interne Optimierung - insbesondere Parallelisierung - möglich wird.

Beispiel 4: *”Für alle Volumetomogramme die maximal im Würfel auftretende Intensität.”* Die Funktion `spatial_domain()` liefert die aktuelle Koordinatenmenge eines MDD-Objekts.

```
select  condense max over x in spatial_domain(vt.data)
        using vt.data[x]
from    VolumeTomogram as vt
```

Das Kondensierer-Konstrukt erlaubt eine drastische Reduzierung des Datentransfers zwischen Anwendung und Server, insbesondere, wenn es im where-Teil der Anfrage erscheint.

Beispiel 5: *”Die OIDs aller Landsat-Bilder, in denen im Infrarotkanal die Intensität den Schwellwert 140 übersteigt.”* Durch Induktion über den integer-Vergleichsoperator `>` entsteht eine Boolesche Matrix, welche durch `or` zu einem einzelnen Wert zusammengefaßt wird. Der `some`-Operator als Spezialfall eines Kondensierers erlaubt eine prägnante Formulierung, wobei die Semantik analog zum SQL-Mengenoperator erklärt ist.

```
select  li
from    LandsatImage as li
where   some( li.data[x].infrared > 140 )
```

Angesichts des Datenumfangs bei MDD ist es angebracht, partielle Update-Operationen zuzulassen. Zu diesem Zweck wird die linke Seite der Zuweisung im SQL-Update-Statement um Trimm- und Projektionsausdrücke erweitert.

Beispiel 6: Die folgende Anweisung fügt eine weitere x/y-Tomogrammschicht in das Volumetomogramm ein; der Ausdruck `”spatial_domain(data)[2].hi+1”` erweitert gleichzeitig den Würfel in z-Richtung um eine Schicht und legt diese als Ziel fest.

```
update  VolumeTomogram
set     data[ ***, ***, spatial_domain(data)[2].hi+1 ] =
        <scan data>
```

Gearbeitet wird derzeit u.a. an Operationen, die allgemeine Koordinatentransformationen erlauben (z.B. affine Transformationen) sowie Operationen, die eine Umgebung jeder Ausgangszelle in die Berechnung des neuen Zellwerts einbeziehen; damit lassen sich z.B. Diskrete Fouriertransformationen [Bune-93] formulieren.

3.2 Die Architektur von RasDaMan

Abb. 3 zeigt ein vereinfachtes Architekturschema des RasDaMan-Systems. Die Schnittstelle zu RasDaMan besteht aus der Abfragesprache RasQL und der C++ Sprachanbindung RasLib, einer ODMG-konformen Klassenbibliothek, die Klassen zur Definition, Speicherung und Wiedergewinnung von MDD-Objekten aus einer RasDaMan-Datenbank enthält.

Die Client/Server-Kommunikation basiert auf dem Distributed Computing Environment (DCE) der Open Software Foundation (OSF) [Shir-95]. Der Client ruft Funktionen des Servers über DCE Remote Procedure Calls auf. Anfragen werden in zwei Schritten optimiert: Algebraische Optimierung mit Regeln der AFATL Image Algebra und physische Optimierung mit Informationen über Kachelung, Clusterung und Geräteeigenschaften. Die Anfrageauswertung erfolgt auf den einzelnen Kacheln eines MDD; die Indexverwaltung bestimmt die relevanten Kacheln und liefert Hinweise bezüglich ihrer Speicherung für den Optimierer. Schemainformation findet sich in der Katalogverwaltung; die Geräteverwaltung kennt die Eigenschaften der verschiedenen Speichermedien. Der optimierte Ausführungsplan wird bearbeitet, indem Kacheln von der Kachelverwaltung angefordert und dann verarbeitet werden. Portierbarkeit zwischen verschiedenen DBMS soll der Speicherverwaltungsadapter sicherstellen.

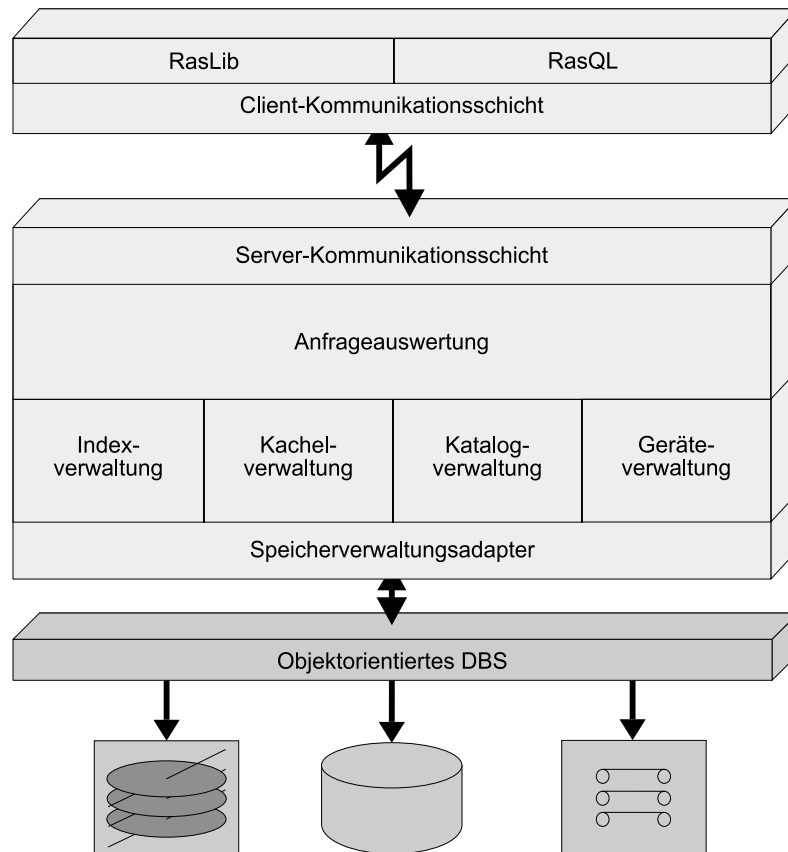


Abb. 3: Vereinfachte RasDaMan-Architektur

3.3 Speicherverwaltung

Damit trotz des potentiell sehr hohen Platzbedarfs von MDD-Objekten der effiziente Zugriff auf MDD-Objekte und insbesondere Teile davon möglich ist, benötigt RasDaMan intern eine gekachelte Speicherung. Dabei sollte die Menge der Transfers zwischen Sekundär- und Hauptspeicher bei der Ausführung von Operationen minimiert werden. Der RasDaMan-Speicherverwalter unterstützt verschiedene Speichermedien transparent für den Benutzer.

Es gibt keine optimale Speicherung von MDD für alle Operationen; eine flexible Aufteilung in mehrdimensionale Kacheln von jeweils beliebiger Größe erlaubt jedoch Anpassung an die Zugriffscharakteristiken der häufigsten Operationen [Furt-94]. Damit kann RasDaMan mit Hilfe Statistiken oder nach Hinweisen des Benutzers die individuell beste Kachelung für ein Objekt benutzen. Um irreguläre Kachelung zu unterstützen, wird für jedes MDD-Objekt ein räumlicher Index verwaltet. Bei Bedarf können einzelne Kacheln individuell mit verschiedenen Verfahren komprimiert werden.

RasDaMan verwaltet MDD Objekte transparent auf Sekundär- oder verschiedenen Tertiärspeichern. Die Verteilung und der Transfer der Daten zwischen den Speichersystemen wird mit Informationen über Eigenschaften der jeweiligen Speichermedien und typische Zugriffsmuster der Benutzer optimiert [Sara-94, Chen-95].

Das Speichermanagement wird auf Basis des kommerziellen ODBS O₂ von O₂ Technology implementiert [Banc-92], das nach einer Evaluierung mehrerer ODBSe ausgewählt wurde. Einige Konzepte von RasDaMan wurden bereits experimentell mit einem RDBS implementiert [Furt-93].

4 Anwendungsgebiete

Zwei Anwendungsgebiete werden im Rahmen des Projekts genauer untersucht: Die Nutzung digitaler Daten in medizinischen Bildarchiven (PACS) und geographischen Informationssystemen.

Im medizinischen Bereich [Mart-93, Garc-95] wird mit MDD unterschiedlicher Dimensionalität gearbeitet: z.B. sind EKGs 1D, Ultraschallbilder 2D und Volumentomogramme 3D. Zur interaktiven Diagnose am Bildschirm werden umfangreiche Abfragemöglichkeiten benötigt wie Projektion an verschiedenen Achsen, Ausschnittsbildung, Vergrößerung oder Möglichkeiten zur inhaltsbasierten Suche. Das spanische Systemhaus STI integriert RasDaMan mit Global PACS [MKNS-93]; das System wird dann im katalanischen Hospital General de Manresa eingesetzt.

Im geographischen Bereich [Toml-90] machen die Fortschritte in Speichertechnologie und Rechenleistung die Speicherung großer Mengen an Bilddaten, also 2D-MDD, billiger als deren Vektorisierung. Rasterdaten höherer Dimensionalität erhält man z.B. bei Klimasimulationen, wenn auch Zeit und Höhe berücksichtigt werden müssen. Typische Operationen sind inhaltsbasierte Suche (“weniger als 10% Wolken”) oder Zusammenfassen von Information (“stelle Gebiete mit einer Temperatur > 40°C in Rot dar, <= 40°C in Grün”). Ein weiterer Partner wird RasDaMan in ein GIS für das spanische Nationale Geographische Institut einbinden.

5 Zusammenfassung

In diesem Papier wurde das Projekt RasDaMan zur Speicherung und Verwaltung mehrdimensionaler diskreter Daten vorgestellt. Basierend auf einem mathematischen Modell aus der Bildverarbeitung wurde ein Datenmodell für MDD entwickelt, das sich an ODMG-93 und SQL-92 anlehnt. Die Implementierung von Feldern in der Datenbank ermöglicht deren Benutzung in einer deklarativen, algebraisch optimierbaren Abfragesprache. Effiziente Speicherung wird durch die Verwendung von Kachelungstechniken aus der Bildverarbeitung kombiniert mit den Indizierungsverfahren räumlicher Datenbanken erreicht.

Die Modellierungs- und Abfragemöglichkeiten zusammen mit dem effizienten Speichersystem von RasDaMan erlauben eine einfache und damit schnelle, unkomplizierte Anwendungsentwicklung im geographischen und medizinischen Bereich. Durch den allgemeinen MDD-Ansatz sind Einsatzmöglichkeiten in vielen anderen Gebieten denkbar; so arbeiten z.B. OLAP-Anwendungen mit großen, hochdimensionalen Datenbeständen und benötigen ähnliche Funktionalität, etwa Ausschnittsbildung und Aggregation.

Der Kern des Rasterdatenbanksystems ist bereits implementiert und dessen Basisfunktionalität verfügbar. Derzeit werden die Pilotanwendungen für PACS und GIS realisiert und in Folge bei den Endbenutzern eingesetzt. Während der Evaluierungsphase werden unter Berücksichtigung der Erfahrungen mit RasDaMan im praktischen Einsatz die Abfragesprache weiterentwickelt und neue Speicherungstechniken integriert. Darüber hinaus wird ein allgemeiner Benchmark für multidimensionale Datenbanksysteme entwickelt und veröffentlicht werden, mit dem die Funktionalität und Leistung dieser Technologie objektiv beurteilt und verglichen werden kann.

Literatur

- [Banc-92] F. Bancilhon, C. Delobel, P. Kanellakis: *Building an Object-Oriented Database System*. Morgan Kaufmann Publishers, San Mateo, CA, 1992.
- [Baum-92] P. Baumann: *Language Support for Raster Image Manipulation in Databases*. Proc. Int. Workshop on Graphics Modeling and Visualization in Science & Technology, Darmstadt, Germany, April 1992.
- [Baum-94] P. Baumann: *On the Management of Multidimensional Discrete Data*. VLDB Journal, 4(3)1994, Special Issue on Spatial Database Systems, S. 401-444, 1994.

- [Bune-93] P. Buneman: *The Fast Fourier Transform as a Database Query*. Technical Report MS-CIS-93-37/L&C 60, University of Pennsylvania, 1993.
- [Cann-93] S. Cannan, G. Otten: *SQL - The Standard Handbook*. McGraw-Hill, 1993.
- [Care-94] M. Carey, D. DeWitt, M. Franklin, N. Hall, M. McAuliffe, J. Naughton, D. Schuh, M. Solomon, C. Tan, O. Tsatalos, S. White, M. Zwilling: *Shoring up Persistent Objects*. Proceedings of the 1994 ACM-SIGMOD Conference, Minneapolis-Minnesota, 1994.
- [Catt-96] R.G.G. Cattell: *The Object Database Standard: ODMG-93*. Morgan Kaufmann Publishers, 1996.
- [Chen-95] L. Chen, R. Drach, M. Keating, S. Louis, D. Rotem, A. Shoshani: *Efficient Organization and Access of Multi-dimensional Datasets on Tertiary Storage Systems*. Information Systems Journal, April 1995.
- [DeWi-94] D. DeWitt, N. Kabra, J. Luo, J. Patel, J. Yu: *Client-Server Paradise*. Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994.
- [Furt-93] P. Furtado, J. Teixeira: *Storage Support for Multidimensional Discrete Data in Databases*. Computer Graphics forum - Special Issue on Eurographics'93 Conference, vol. 12, no.3, Seiten 89-100, September 1993.
- [Furt-94] Paula Furtado: *Management of Multidimensional Discrete Data in Database Systems*. Master Thesis, Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 1994.
- [Garc-95] H. Garcia, D. Yun: *Intelligent Distributed Medical Image Management*. Proc. SPIE Medical Imaging Conference, Vol. 2435, Seiten 80-91, Februar 1995.
- [ISO-92] The International Organization for Standardization (ISO): *Database Language SQL*. ISO 9075, 1992(E), 1992.
- [Kim-95] W. Kim, Ed.: *Modern database Systems: the object model, interoperability, and beyond*. ACM Press, 1995.
- [Lori-82] R. Lorie: *Issues in Databases for Design Transactions*. in J. Encarnação, F. Krause (eds.): *File Structures and Databases for CAD*. North Holland Publishing, 1982.
- [Mart-93] R. Martinez, J. Kim, J. Nam, B. Sutaria: *Remote Consultation and Diagnosis in a Global PACS Environment*. Proc. SPIE Medical Imaging Conference, Seiten 296-307, Februar 1993.
- [MKNS-93] Ralph Martinez, Jinman Kim, Jiseung Nam, and Bijal Sutaria: *Remote Consultation and Diagnosis in a Global PACS Environment*. Proc. SPIE Medical Imaging Conference, Seiten 296-307, Februar 1993.
- [Ritt-90] G. Ritter, J. Wilson, J. Davidson: *Image Algebra: An Overview*. Computer Vision, Graphics, and Image Processing, 49(1)1990, S. 297-331, 1990.
- [Sara-94] S. Sarawagi, M. Stonebraker: *Efficient Organization of Large Multidimensional Arrays*. Tenth Int. Conf on Data Engineering, Seiten 328-336, Houston, Februar 1994.
- [Shir-95] John Shirley; Wei Hu; David Magid: *OSF Distributed Computing Environment: Guide to Writing DCE Applications, 2. edition*. O'Reilly & Associates, Sebastopol, CA, 1994.
- [Ston-95] M. Stonebraker, D. Moore: *Object-Relational DBMSs: The Next Great Wave*. Morgan Kaufmann Publishers, 1995.
- [Tamu-80] H. Tamura: *Image Database Management for Pattern Information Processing Studies*. In: S. Chang, K. Fu (ed): *Pictorial Information Systems*. Lecture Notes in Computer Science Vol. 80, S. 198-227, Springer 1980.
- [Toml-90] C. Tomlin: *Geographic Information Systems and Cartographic Modelling*. Prentice Hall, 1990.