

Sonderdruck aus:

Nachrichten aus dem Karten- und Vermessungswesen, Reihe 1: Originalbeiträge - Heft Nr. 85 Verlag des
Instituts für Angewandte Geodäsie Frankfurt a. M. 1981
DK 519.256

Datenspeicherung für schnellen Zugriff auf Daten räumlich benachbarter Objekte

(mit 7 Figuren)

Von André Frank, Zürich

ZUSAMMENFASSUNG: In einem Landinformationssystem werden große Mengen von Daten, die sich auf im Raum fixierte Objekte beziehen, gespeichert. Für die graphische Darstellung dieser Objekte in Form von Kartenausschnitten auf einem Bildschirm sind Zugriffsmethoden und Speicherstrukturen erforderlich, die raschen Zugriff auf die Daten benachbarter Objekte erlauben. Es wird ein solches Verfahren vorgestellt, das auf der Kontrolle der Platzierung der Daten auf dem Massenspeicher aufbaut.

Versuche mit einer Implementierung, die auf einem Netzwerk-Datenbanksystem aufbaut, haben Antwortzeiten von 20 Sekunden bis 1 Minute, je nach dargestellter Datenmenge, ergeben; dies erlaubt interaktives Arbeiten.

Wesentlich am vorgestellten Verfahren ist, daß die Zugriffszeit von der Menge der gespeicherten Daten praktisch unabhängig ist.

SUMMARY: A great number of space-related data will be stored in a Land Information System (LIS). In order to show the data graphically, in the form of a map on a CRT- screen, data structures and retrieval algorithms are necessary which permit fast access for the data on adjoining objects in space. Such a method relying on the control of placement of data on mass storage device is presented. Using this method, retrieval time is almost independent of the number of data stored and linear in the number of data retrieved. Tests based on a network-type data-base-management-system showed response time between 20 sec and 1 min depending on the number of details required. This seems fast enough to permit interactive work.

RÉSUMÉ: Un système d'information du territoire est appelé à mémoriser de grandes quantités de données se rapportant à des objets déterminés dans l'espace. La représentation graphique de ces objets, sous la forme d'extraits de cartes visualisés sur écran, nécessite des méthodes d'appel et des structures de mémorisation qui permettent un accès rapide aux données d'objets voisins.

L'auteur présente un tel procédé fondé sur le contrôle du placement des données dans la mémoire principale. La chose essentielle de ce procédé est que le temps d'accès ne dépend pratiquement pas de la quantité des données mémorisées. Des essais effectués avec un système de banque de données fondé sur un modèle ramifié, ont donné, selon la quantité de données représentées, des temps de réponse allant de 20 secondes à une minute. Ceci permet effectivement un travail interactif.

1 Einleitung

Seit einigen Jahren laufen am Institut für Geodäsie und Photogrammetrie der Eidgenössischen Technischen Hochschule Zürich unter der Leitung von Prof. R. Conzett Forschungsarbeiten im Bereich Landinformationssysteme.

Ein wichtiger Punkt in diesem Problemkreis ist bestimmt die Realisierung rascher Zugriffsmethoden auf die gespeicherten Daten, um Auskünfte im interaktiven Betrieb erteilen zu können.

Über die bis heute vorliegenden Resultate zur Lösung dieser Aufgabe soll hier berichtet werden.

2 Fragestellung

Bei EDV-unterstützten Informationssystemen, die im interaktiven Betrieb Auskünfte erteilen sollen, spielt, neben anderen Problemen, die Antwortzeit eine große Rolle.

Besteht die Antwort aus einem oder wenigen Datensätzen, wie das in kommerziellen Anwendungen üblich ist, so können die von verschiedenen Herstellern angebotenen Standard-Datenbanksysteme [2], [3] eingesetzt werden. Diese erlauben in solchen Fällen Zugriffspfade vorzubereiten, die rasche Antworten gewährleisten.

Handelt es sich hingegen um raumbezogene Informationssysteme - verstanden als Oberbegriff für Landinformationssysteme und Geographische Informationssysteme [10] - und besteht die Antwort dementsprechend aus einer graphischen Darstellung, so bietet die rasche Antwort Probleme.

Die Antwort setzt sich nicht mehr aus wenigen Datensätzen zusammen, sondern eine größere Datenmenge muß herausgesucht werden. Die Daten, die einer Karte im Bildschirmformat zu Grunde liegen, umfassen unserer Schätzung nach um die tausend Datensätze. Diese in einem Massenspeicher zu finden, benötigt schon bei nur einem Zugriff pro Datensatz eine beträchtliche Zeitspanne ($\sim \frac{1}{2}$ bis 1 Minute). Es gilt also, ein Verfahren zu entwickeln, die benötigten Daten mit wenigen Zugriffen rasch zu finden.

Standard-Datenbanksysteme bieten für die Verwaltung großer Datenmengen wesentliche Vorteile [12]; Zugriffsmethoden für ein raumbezogenes Informationssystem sollten so entworfen werden, daß sie auf Standard-Datenbanksystemen aufbauen können. Damit können deren Leistungen (Datenschutz, Datensicherheit) für Landinformationssysteme genutzt werden und die Programme können leichter wechselnden Anforderungen angepaßt werden.

3 Ziel

Ziel unserer Arbeit war, Möglichkeiten zu finden, mit Standard-Datenbanksystemen raumbezogene Datenbanken aufzubauen und zu verwalten, ohne unzumutbare Wartezeiten für Antworten in graphischer Form in Kauf zu nehmen.

Wir haben DBMS-10 der Firma Digital Equipment Corp. [7], [8] auf der DEC-10-Anlage des Zentrums für Interaktives Rechnen der ETH eingesetzt. Es handelt sich dabei um eine Implementierung des CODASYL-DBTG-Standards, entsprechende Programme sind auch auf anderen Anlagen erhältlich (z. B. UDS von Siemens [14], von Data General [5]).

4 Analyse des Problems

Die Antwortzeit wird bei modernen Systemen hauptsächlich durch die Zugriffszeit bestimmt. Der Zeitbedarf für Rechenoperationen kann dagegen vernachlässigt werden.

Zugriffe auf Massenspeicher bringen bei modernen Betriebssystemen meist mehr als einen Datensatz in den Hauptspeicher, da der Zugriff des Speicherverwaltungssystems in Einheiten fester Größe erfolgt und erst im Hauptspeicher der gewünschte Datensatz herausgesucht wird. Gelingt es, pro Zugriff mehr als einen Datensatz, der zur Antwort verwendet werden kann, in den Hauptspeicher zu bringen, so läßt sich die Zahl der notwendigen Zugriffe verkleinern und die Antwortzeit verkürzen. Dies wird in der kommerziellen Datenverarbeitung ausgenutzt, indem Daten, die häufig gleichzeitig gebraucht werden, auch am gleichen Ort auf dem Massenspeicher gespeichert werden und so mit einem Zugriff in den Hauptspeicher verbracht werden. Natürlich läßt sich das nicht für alle Zugriffswege gleichzeitig erreichen, sondern es können nur die häufigsten Zugriffe so beschleunigt werden.

Der wichtigste Zugriffsweg in einem raumbezogenen Informationssystem ist bestimmt die Lage im Raum.

Im Normalfall sollen ja Auskünfte in Form einer Karte auf einem Bildschirm präsentiert werden. Dabei darf nicht nur das direkt interessierende Objekt dargestellt werden, sondern es muß eine bestimmte Umgebung ebenfalls abgebildet werden.

Die Frage wird also im allgemeinen etwa von folgender Form sein:

ZEIGE *{objekt-typ}* INNERHALB *{fenster}*

wo

objekt-typ die darzustellenden Objekte (Gebäude, Straßen etc.)

und

fenster den darzustellenden Ausschnitt
($X_{min.}$, $X_{max.}$, $Y_{min.}$, $Y_{max.}$) beschreibt.

Zusätzliche Angaben über Darstellung etc. können im hier betrachteten Zusammenhang vernachlässigt werden.

Unsere Erfahrungen lassen vermuten, daß im allgemeinen auf einem Bildschirm 500 bis 2000 Objekte, die meistens als Punkte zur Liniendefinition, dargestellt werden müssen. Bilder mit weniger Information scheinen leer und sind schwierig zu interpretieren.

Typisch für Fragen dieser Art ist, daß benachbarte Objekte verlangt werden. In der Informatik-Literatur werden Fragen dieser Art meist als zweidimensionales Abschnitt-Suchen bezeichnet (two-dimensional range queries) [9], [6], [1], [13].

Die bekannten Verfahren sind vor allem für punktförmige Objekte geeignet; hier soll eine Erweiterung auf zweidimensional ausgedehnte Objekte vorgeschlagen werden.

Für die rasche Beantwortung solcher Fragen muß versucht werden, räumlich benachbarte Objekte auch im Massenspeicher benachbart zu speichern, so daß mit einem Zugriff ein größerer Teil einer Karte gefunden wird.

Bildlich läßt sich das etwa so darstellen (Fig. 1): Statt daß wir Daten gestreut über das ganze Speichermedium speichern, sind sie nachbarschaftstreu auf das Speichermedium abgebildet.



**Gleichmässige Verteilung
interessierender Daten
über den gesamten
Speicherbereich:**

Viele Zugriffe = lange Wartezeit



**Daten sind im Speicher
gruppiert gespeichert:**

Wenige Zugriffe = schnelle Antwort

Figur 1:

5 Generelle Lösungsidee

Standardisierte Datenbanksysteme lassen eine Beeinflussung des Ortes der physischen Speicherung eines Datensatzes im Massenspeicher zu [4]. Dies muß ausgenutzt werden, wenn ein unverändertes Datenbanksystem angewendet werden soll [11].

Der Speicherraum einer Datenbank ist im allgemeinen in eine variable Zahl von Datenbankseiten fester Länge eingeteilt. Diese Seiten bilden auch die Einheit des physischen Zugriffes auf das Speichermedium.

Gelingt es, Datensätze benachbarter Objekte auf der gleichen Datenbankseite zu speichern, so können wir erwarten, daß der Zugriff im Nachbarschaftsbereich wesentlich beschleunigt wird.

Ziel der Optimierung des Abfrage-Algorithmus ist also, die Anzahl der Zugriffe auf den Massenspeicher zu reduzieren.

6 Lösungsweg

Mit einem schrittweise verfeinerten Modell wird das Verfahren hierzu erläutert:

6.1 Grundidee

In einem sehr einfachen raumbezogenen Informationssystem, das nur Datensätze einer Art über punktförmige Objekte, die gleichmäßig im Raum verteilt sind, enthält, läßt sich die Zusammenfassung benachbarter Objekte leicht erreichen: Der zweidimensionale Koordinatenraum wird in z. B. quadratische Felder eingeteilt. Jedem Feld wird eine Datenbankseite zugeordnet. Jeder Datensatz wird auf der Seite gespeichert, die dem Feld zugeordnet ist, in dem das Objekt liegt.

Für den Zugriff wird berechnet, welche Felder in Frage kommen und dann wird nur auf die entsprechenden Seiten zugegriffen.

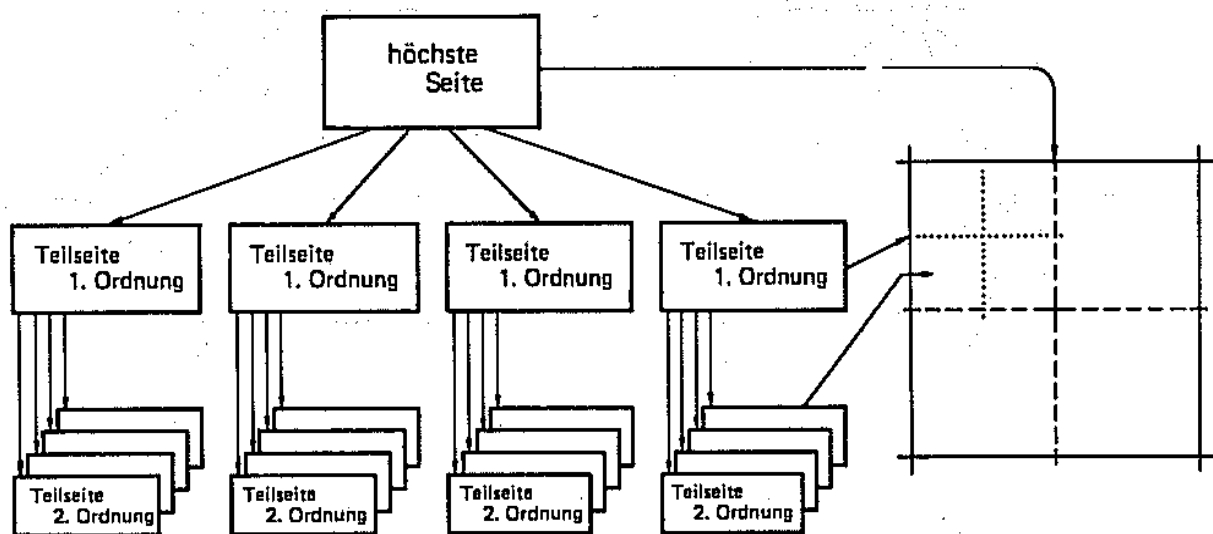
6.2 Ungleichmäßig im Raum verteilte Punkte

In einem raumbezogenen Informationssystem werden sehr unterschiedliche Dichten von Objekten pro Flächeneinheit vorkommen. Zwischen dicht überbauten Stadtgebieten und extensiv genutzten ländlichen Räumen dürfen sich Unterschiede in der Datenmenge pro Flächeneinheit bis 1:1000 ergeben.

Es ist also nicht möglich, die Seiten so groß zu wählen, daß alle Daten eines Gebietes mit hoher Datendichte untergebracht werden können, ohne daß enorm Speicherplatz verschwendet, würde.

Die Speicherstruktur muß sich an solche Unterschiede anpassen können, und zwar sollte dies dynamisch immer dann erfolgen, wenn mehr Daten gespeichert werden sollen und darf nicht bei Eröffnung der Datenbank festgelegt werden müssen.

Die Grund-; Größe der Felder und Seiten wird so eingerichtet, daß im Fall weniger Daten diese gerade knapp voll werden. Sind mehr Daten zu speichern, so wird mehr Speicherraum beschafft, indem zusätzliche Datenbankseiten eröffnet werden. Dazu wird das Feld in Teilfelder unterteilt und jedem Teilfeld eine Teilseite zugeordnet. Das kann mehrmals wiederholt werden, bis genügend Raum für die Daten gefunden ist (Fig. 2).



Figur 2:

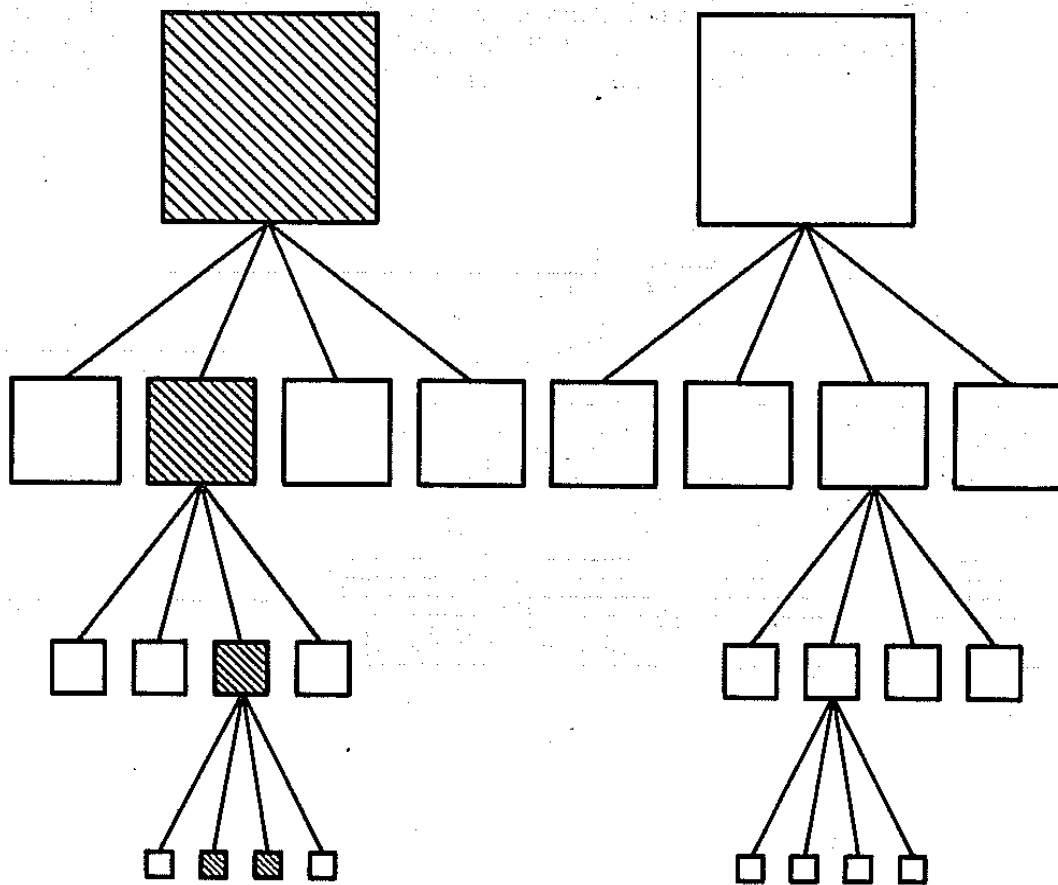
Die Felder und Teilfelder bilden, graphentheoretisch gesehen, einen Wald voll Bäumen.

Um auf alle Daten innerhalb eines Ausschnittes zuzugreifen, müssen alle Grundfelder bestimmt werden und nachher jeweils den in den Ausschnitt fallenden Teil des Baumes abgesucht werden.

6.3 Verschiedene Objekt-Typen

Es ist natürlich nicht richtig anzunehmen, daß ein Landinformationssystem nur eine Art von Datensätzen enthalte: Es handelt sich in der Praxis vielleicht um Häuser, Straßen, Grenzen usw., die je mit einem speziellen Datensatz-Typ gespeichert werden.

Werden die verschiedenen Datensätze alle miteinander in der vorher beschriebenen Hierarchie von Feldern gespeichert, so ist die Datenstruktur geeignet für die Beantwortung von Fragen, die die unterste Stufe der Detaillierung betreffen. Da auf dem Bildschirm nur eine bestimmte Menge von Daten dargestellt werden kann, umfaßt die Antwort immer etwa die gleiche Menge von Datensätzen; es müssen daher auch immer etwa gleich viele Datenbankseiten abgesehen werden. Für ein kleines Gebiet mit hoher Informationsdichte müssen von wenigen Grundfeldern ausgehend viele Teilfelder über mehrere Stufen abgesehen werden (Fig. 3). Für ein großes Gebiet mit niedriger Informationsdichte werden viele Grundfelder, aber kaum Teilfelder untersucht werden. Die Zugriffe nehmen etwa gleich viel Zeit in Anspruch.

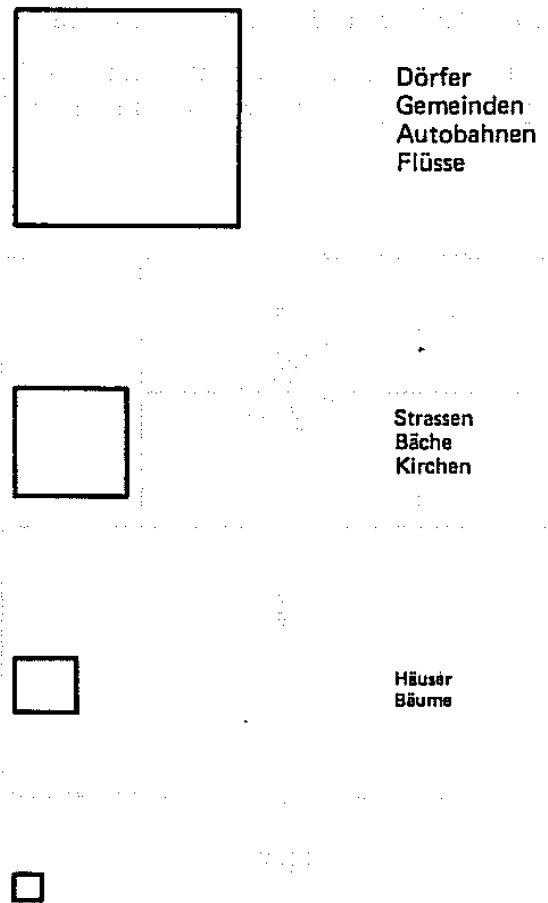


Figur 3:

In vielen Fällen werden aber Übersichtsdarstellungen verlangt: Es sollen nur die Dörfer in einem größeren Gebiet, nicht aber Details, wie die Häuser und Bäume dargestellt werden.

Es sollte in diesem Fall möglich sein, die Datensätze schneller zu finden, ohne alle Seiten, die die Details enthalten, abzusuchen.

Dazu dient folgendes Vorgehen: Es wird festgelegt, in welchem Bereich der Felder-Hierarchie gewisse Datensätze gespeichert werden dürfen. Datensätze von wichtigen Objekten, die für größere Gebiete maßgebend sind, werden in großen Feldern gespeichert, Details in kleineren (Fig. 4).



Figur 4:

Um eine Übersichtsdarstellung über ein großes Gebiet mit den wenigen wichtigsten Datensatz-Typen zu erstellen, muß nur die Hierarchie bis zu einer bestimmten Tiefe abgesucht werden; die Antwort kann rasch gegeben werden. Muß hingegen für ein kleines Gebiet eine Detaildarstellung angegeben werden, so muß ein kleiner Teil der Hierarchie bis zum Grunde abgesucht werden; auch das geht rasch.

Natürlich dauert es länger, eine detaillierte Karte über ein großes Gebiet zu erstellen, aber diese wird ja ohnehin auf einem Plotter gezeichnet und dann spielt die Zugriffszeit keine wesentliche Rolle mehr.

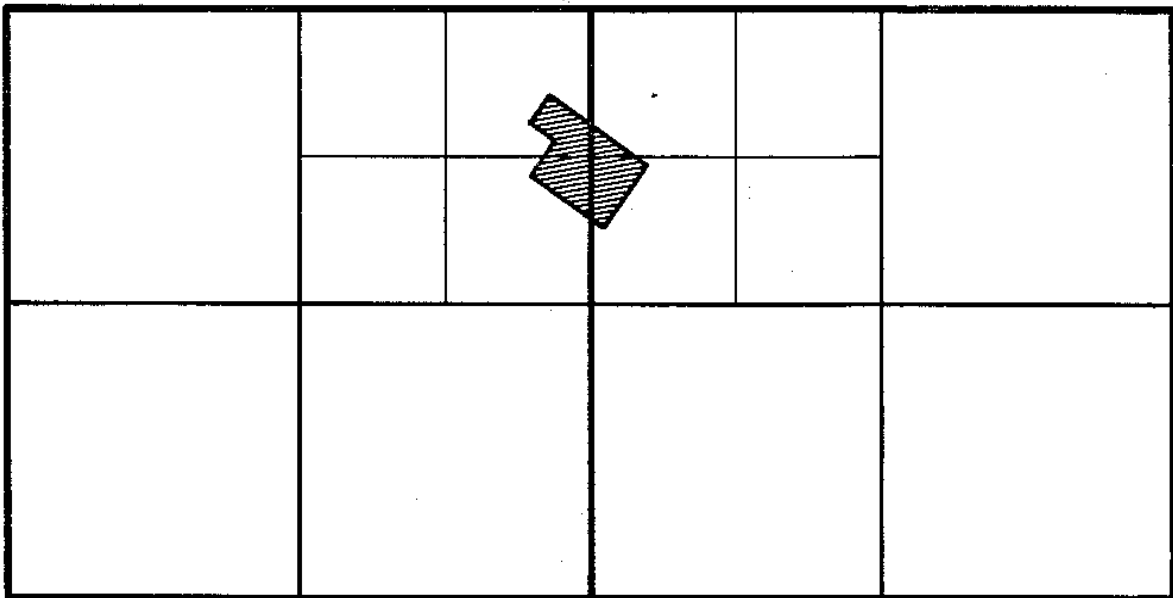
6.4 Erweiterung auf zwei-dimensional ausgedehnte Objekte

Leider sind die Objekte, über die Informationen gespeichert werden sollen, nicht alle punktförmig. Flächenhaft ausgedehnte Objekte, wie z. B. Häuser, Wälder, auf Felder aufzuteilen, ist schwieriger.

Wiederum wird die Hierarchie ausgenützt: jedes Objekt soll auf dem kleinsten Feld gespeichert werden, auf dem es vollständig Platz findet. - Ein Objekt wird also nie zerschnitten. Damit werden alle bekannten Probleme des Zerschneidens und Wiederaussetzens von Figuren umgangen. Hingegen werden der soeben gemachten Zuordnung der Daten zu Feldgrößen gewisse Grenzen gesetzt: Diese müssen so gewählt werden, daß die entsprechenden Objekte auf den Feldern auch Platz finden.

Das scheint im erwähnten Verfahren für die Feldereinteilung nicht einfach:

Auch ein winziges Objekt, das auf der Grenze zwischen den größten Feldern liegt, kann in keinem Feld ganz gespeichert werden (Fig. 5). Abhilfe bringt eine kompliziertere Feldereinteilung:

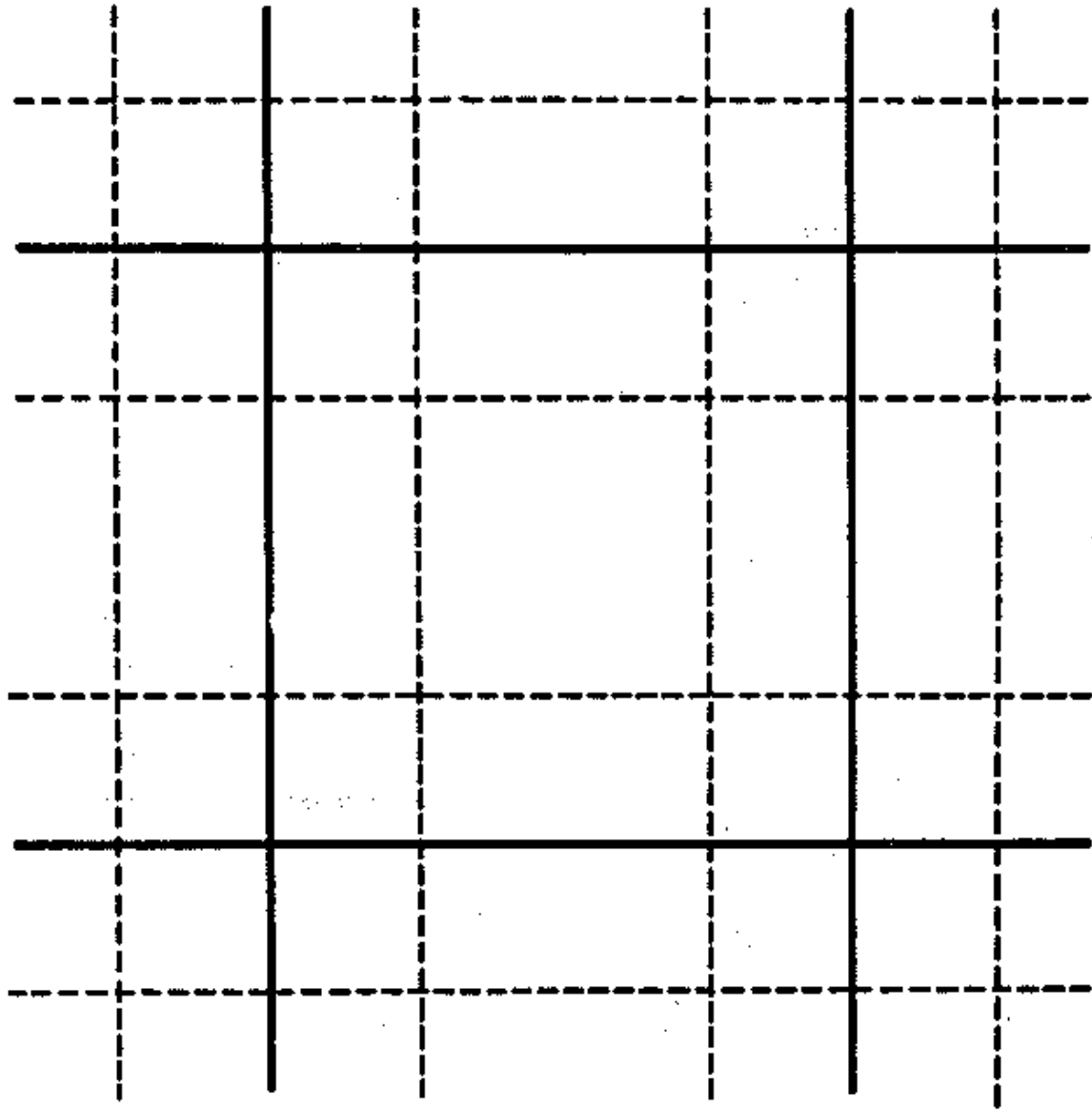


Figur 5:

Die Grenzen der Teilfelder fallen nie mit den Grenzen der Hauptfelder zusammen, sondern liegen symmetrisch zu diesen (Fig. 6).

Es kann für diesen Fall gezeigt werden, daß ein Objekt in einem Feld mit einer Seitenlänge kleiner als das 16fache der größten Ausdehnung des Objektes Platz findet. Wird das größte Feld genügend groß gewählt, so können wir alle Objekte unterbringen.

Leider bringt dieses Verfahren eine wesentlich kompliziertere Datenstruktur für die Felder mit sich: ein Netz, in dem jedes Feld vier Oberfelder und neun Teilfelder haben kann.



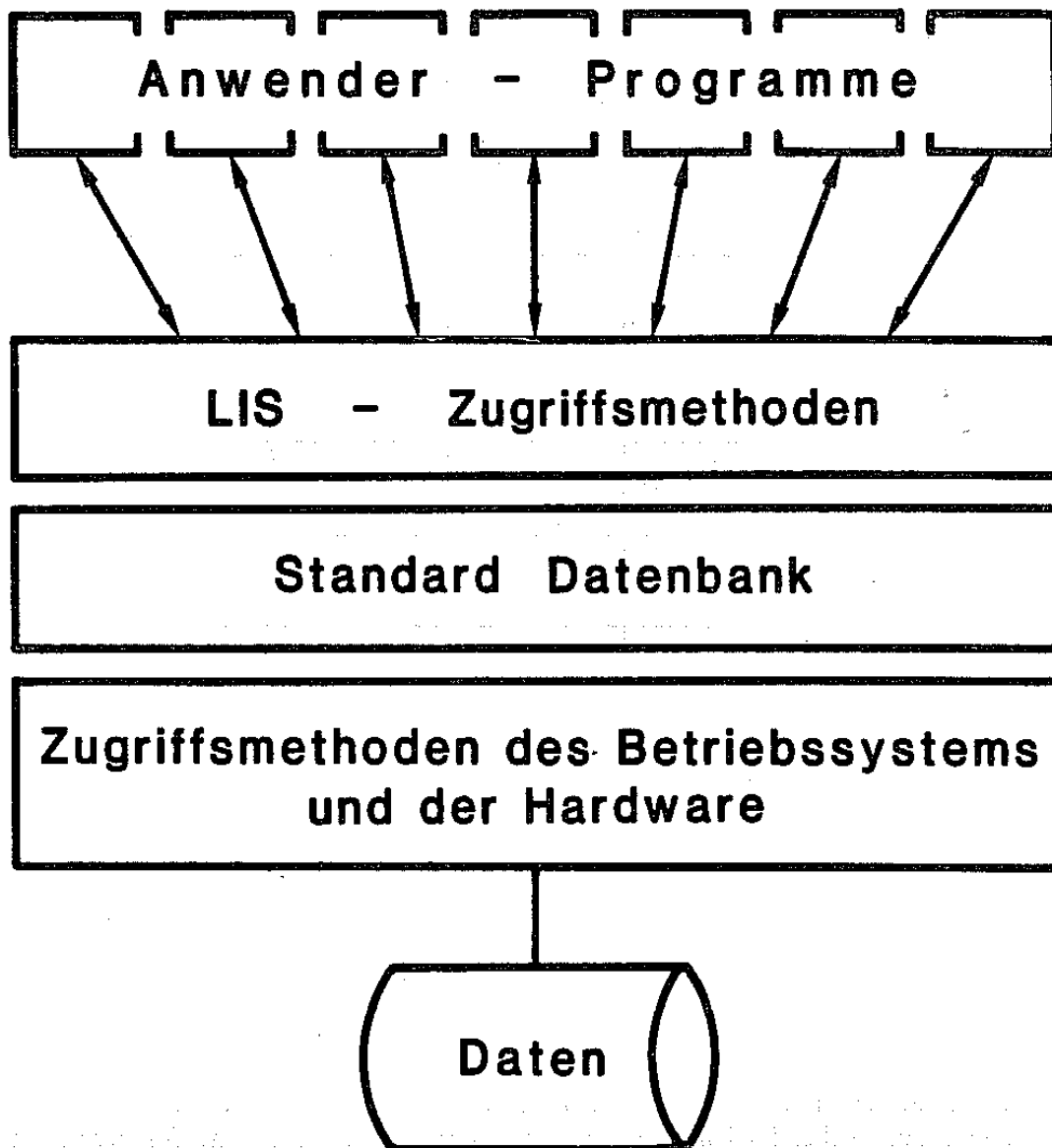
Figur 6:

7 Realisierung

Wir haben, auf einem Standard-Datenbanksystem (DBMS- 10 der Firma DEC Corp.) aufbauend, ein solches Speicherverfahren realisiert. Dabei wurde eine Gliederung in Schichten angestrebt, wobei jeweils die obere Schicht auf den Dienstleistungen der unteren Schicht aufbaut (Fig. 7).

Die bis heute durchgeführten Tests zeigen Antwortzeiten zwischen 20 Sekunden und einer Minute, je nach Größe des Gebietes, das dargestellt werden soll. Wesentlich ist, daß die Zugriffszeit von der Menge der gespeicherten Daten praktisch unabhängig ist.

Prinzipiell scheint das Verfahren richtig; weitere Arbeiten sind noch erforderlich, um Verbesserungen im Detail zu erreichen. Wir erwarten dabei noch wesentliche Leistungssteigerungen.



Figur 7:

8 Zusammenfassung

Das hier vorgestellte Verfahren zur Speicherung von Daten im Raum fixierter Objekte, kurz: raumbezogener Daten, zeigt einen Weg zur raschen Beantwortung von zweidimensionalen Abschnitts-Fragen. Es wird angenommen, daß Fragen dieses Types in der Form

ZEIGE {objekt-typ} INNERHALB {fenster}

für raumbezogene Informationssysteme dominierend sind.

Die Methode basiert auf der Kontrolle der physischen Speicherung der Daten innerhalb eines Datenbank-Systems und ist damit entfernt den index-sequentiellen Zugriffsverfahren für ein-dimensionale Daten vergleichbar.

Literaturverzeichnis

- [1] *Bentley, J. L.; Friedman J. H.*: Data structures for range searching. - ACM Computing Surveys 11 (4) Dez. 1979, S. 397-409.
- [2] CODASYL Data Base Task Group: April 1971 Report.
- [3] CODASYL Data Base Task Group: June 1973 Report.
- [4] CODASYL Data Base Task Group: DDL Journal of Development 1978.
- [5] Data General. Data General Data Base Management System (DG/DBMS). - Reference Manual Westboro, Mass. 1980.
- [6] *Denert, E.; Franck, R.*: Datenstrukturen. - Bibliographisches Institut, Mannheim 1977, (Reihe Informatik 22).
- [7] Digital Equipment Corp.: Data Base Management System (DBMS- 1 0). - Administrator's Procedures Manual, Maynard, Mass. 1977.
- [8] Digital Equipm-ent Corp.: Data Base Management System (DBMS-10). - Programmer's Procedures Manual, Maynard, Mass. 1977.
- [9] *Finkel, R. A.; Bentley,j. L.*: Quad Trees; A Data Structure for Retrieval on Composite Trees. - Acta Informatica Vol. 4, Fase. 1 (1 974), S. 1-9.
- [10] *Frank, A.*: Landinformationssysteme - Ein Versuch zu einer Abgrenzung. - In: Nachr. Kt. u. Vermess.-wes., R. 1, H. 8 1, Frankfurt a. M. 1980, S. 23.-30.
- [11] *Frank, A.*: Datenstrukturen und Speicherung, - 1. Teil von: Probleme der Realisierung von Landinformationssystemen. - Institut für Geodäsie und Photogrammetrie, ETH Zürich 1979, Bericht Nr. 26.
- [12] *Frank, A.*: Datenbanksysteme für Landinformationssysteme. - Institut für Geodäsie und Photogrammetrie, ETH Zürich 1980, Bericht Nr. 44.
- [13] *Lee, D. T.; Wong, C. K.*: Quintary Trees: A file structure for Multidimensional Database Systems. - ACM Transactions on Database Systems, Vol. 5, No. 3, Sept. 1980, S. 339.
- [14] Siemens- Universelles Datenbanksystem (UDS). - Allgemeine Beschreibung, München 1977.

Manuskript eingegangen am 16. 3.1981