Zeitschrift: Vermessung, Photogrammetrie, Kulturtechnik : VPK = Mensuration, photogrammétrie, génie rural

Herausgeber: Schweizerischer Verein für Vermessung und Kulturtechnik (SVVK) = Société suisse des mensurations et améliorations foncières (SSMAF)

Band: 84 (1986)

Heft: 2

Artikel: Hash-Dateien für raschen Zugriff auf Punktkoordinaten : praktische Erfahrungen

Autor: Gnägi, H.R. / Link, F.

DOI: https://doi.org/10.5169/seals-233029

ler Ebene sind noch erste Schwierigkeiten beim Aufstellen der Richtpläne zu meistern. Die Gemeinden bemühen sich, die Bauzonen zu redimensionieren und den Schutz der landwirtschaftlichen Vorranggebiete (mit Einschluss der Fruchtfolgeflächen) sicherzustellen. Besondere Aufmerksamkeit wird in die Zukunft hinein den Städten zu schenken sein. Diese erweisen sich immer mehr als Problemgebiete. Die Prozesse der Agglomerationsbildung durch Desurbanisation, Suburbanisation usw. sind im Gange und schwächen die Kraft der Kernstadt. Erfahrungsgemäss wird von Zeit zu Zeit die Bodenfrage neu diskutiert, doch sind grundsätzliche Änderungen nicht zu erwarten. Es erweist sich als ein Vorteil, dass die Schweiz auf nationaler Ebene über ein relativ junges Raumplanungsgesetz verfügt.

Adresse des Verfassers:
Martin Lendi, Dr. jur. Professor für Rechtswissenschaft ETH Zürich.
Mitglied der Leitung des Instituts für Orts-, Regional- und Landesplanung ETH-Hönggerberg, CH-8093 Zürich

Anhang

1. Literatur

Eidg. Justiz- und Polizeidepartement / Bundesamt für Raumplanung: Erläuterungen zum Bundesgesetz über die Raumplanung, Bern 1981

Lendi Martin: Recht und Politik der Raumplanung, Zürich 1984

Lendi Martin (Herausgeber): Raumplanung Vademecum, ORL-Institut ETH Zürich, Zürich 1985

Lendi Martin / Elsasser Hans: Raumplanung in der Schweiz – Eine Einführung, Zürich 1985

Schürmann Leo: Bau- und Planungsrecht, 2. A., Bern 1984

2. Materialien

Botschaft des Bundesrates an die Bundesversammlung über die Ergänzung der Bundesverfassung durch die Art. 22ter und 22^{quater} vom 15. August 1967, BBl 1967 II, S. 133 ff.

Botschaft des Bundesrates an die Bundesversammlung zum Bundesgesetz über die Raumplanung (RPG) vom 27. Februar 1978, BBl 1978 I, S. 1006 ff.

Arbeitsgruppe des Bundes für Raumplanung: Raumplanung Schweiz, Hauptbericht, Bern 1970

ORL-Institut: Landesplanerische Leitbilder, Bd. I - III und Plankassette, Schriftenreihe zur Orts-, Regional- und Landesplanung Nr. 10, Zürich 1971

Delegierter für Raumplanung: Raumplanerisches Leitbild CK-73, Bern 1973

ORL-Institut: Raumordnungskonzept Schweiz gemäss den Randbedingungen der Chefbeamtenkonferenz, Studienunterlage zur Orts-, Regional- und Landesplanung Nr. 20, Zürich 1974

Eidg. Kommission für die Gesamtverkehrskonzeption: Schlussbericht, GVK-CH 1977, Bern 1977

Eidg. Kommission für die Gesamtenergiekonzeption: Das schweizerische Energiekonzept, Bd. I und II sowie Zusammenfassung. Bern 1978

Hash-Dateien für raschen Zugriff auf Punktkoordinaten. Praktische Erfahrungen

H.R. Gnägi, F. Link

Eine Dateiorganisation, die ermöglicht, Daten mit minimaler Anzahl von Zugriffen vom externen Datenträger zu holen, wird für das Vermessungswesen besonders interessant, wenn in der Datei Punkte abgespeichert sind und diese nach ihren Koordinaten gesucht werden sollen, besonders im Hinblick auf graphisch interaktive Bearbeitung von Vermessungsdaten. Wenn etwa die Diskplatzadresse eines Punktes direkt aus seinen Koordinaten berechnet werden kann, ist das Zugriffsverhalten der Datei besser, als wenn zunächst auf eine Indexdatei zugegriffen werden muss, um dort die Adresse zu holen für den effektiven Zugriff auf die Punktdatei. Die Vorschrift, nach welcher Speicheradressen aus Daten berechnet werden, heisst Zugriffsfunktion oder Hashfunktion. Wir beschreiben die von uns für Punktdateien verwendete Hashfunktion und zeigen an Beispielen aus der Praxis, dass ein Punkt im Mittel mit wenig mehr als einem Zugriff gefunden oder versorgt werden kann. Diese mittlere Anzahl Zugriffe ist im wesentlichen unabhängig von der Grösse der Dateien. Die vorliegende Arbeit ist eine erweiterte Version des Kurzvortrages von H.R. Gnägi und F. Link [1985] an der Tagung Datenbanksysteme in Büro, Technik und Wissenschaft.

Pour les applications en mensuration il est intéressant de disposer d'une gestion efficace de fichier sur disque, qui permet de retrouver les informations avec un minimum de pas d'accès, surtout si le fichier est un registre de points et que les points y sont cherchés d'après leurs coordonnées, cas fréquent lors du traitement graphique interactif.
La disposition proposée est telle que l'adresse de l'emplacement d'un point sur disque est déterminée avec les valeurs numériques de ses propres coordonnées.
Cela améliore nettement l'accessibilité par rapport à la version où, pour accéder au fichier de point, nous passons d'abord par un fichier d'index qui nous indique l'emplacement de l'information dans le fichier de point. Cette manière de déterminer l'adresse de stockage des données est appelée fonction d'accès ou «fonction hash».

1. Problem und Begriffe

Die Organisation von Punktdateien auf externen Speichermedien hat entscheidenden Einfluss auf Programmlaufzeiten, wenn Punkte nach Koordinaten gesucht werden sollen. Wir denken besonders an umfangreiche Auswertungen wie Stapelverarbeitung grosser Datenmengen oder an zeitkritische Auswertungen wie Displaygraphik und numerisch- oder graphisch-interaktive Bearbeitung. Damit Auswertezeiten, Antwortzeiten und Wartezeiten erträglich bleiben, muss die Anzahl Zugriffe auf das externe Speichermedium auf ein Minimum reduziert werden. Wie ist diese Forderung zu realisieren?

Um die Frage in der Datenbankterminologie zu formulieren, brauchen wir einige Definitionen: **Datei** (File) heisst eine Menge von Sätzen. **Satz** (Record, Datensatz) heisst eine endliche Folge, d.h. ein Tupel von Feldern. Ein **Feld** (Attribute) enthält den Wert einer Variablen. **Schlüssel** (Key) heisst ein Feld, das zur Identifikation des Satzes dient und wird mit k_j bezeichnet. **Identifikation** (Identifikation key) heisst die endliche Folge $(k_1, \ldots, k_d)$ von $d \geqslant 1$ Schlüsseln, die den Satz in der Datei eindeutig identifizieren. Die übrigen Felder, die nicht Schlüssel sind, heissen auch etwa **Datenfelder.** (Über den Zusammenhang dieser Begriffe der physischen Datenorganisation zur Festlegung des internen Schemas einer Datenbank und den entsprechenden Begriffen logischer Datenmodelle, z.B. des Relationenmodells, zur Festlegung des konzeptionellen Schemas

Notre version de la «fonction hash» et son application dans la pratique est décrite par la suite. Nous montrons à l'aide d'exemples que la recherche et la mise en mémoire d'un point demandent en général une manipulation d'accès. Ce nombre moyen d'accès est en principe indepéndent de la gradeur des fichiers.
Cet article est une version amplifiée d'un exposé présenté par H.R. Gnägi et F. Link [1985] lors des journées «Datenbanksysteme in Büro, Technik und Wissenschaft».

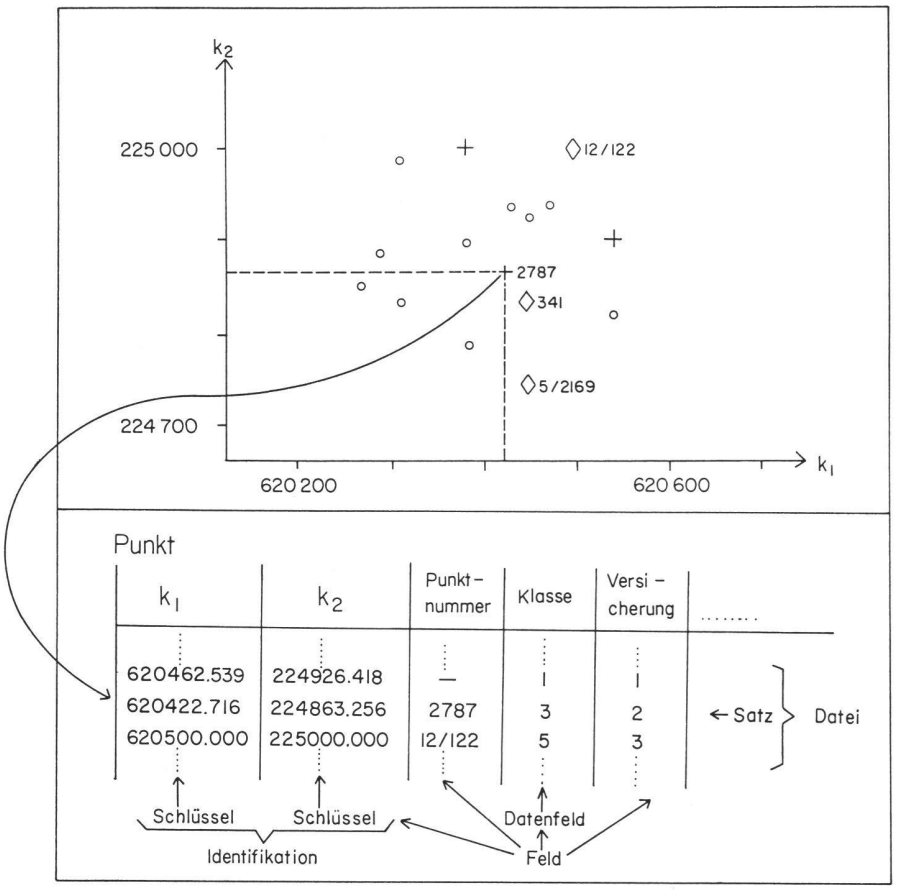

Abb. 1: Datenbankbegriffe am Beispiel einer Punktdatei.

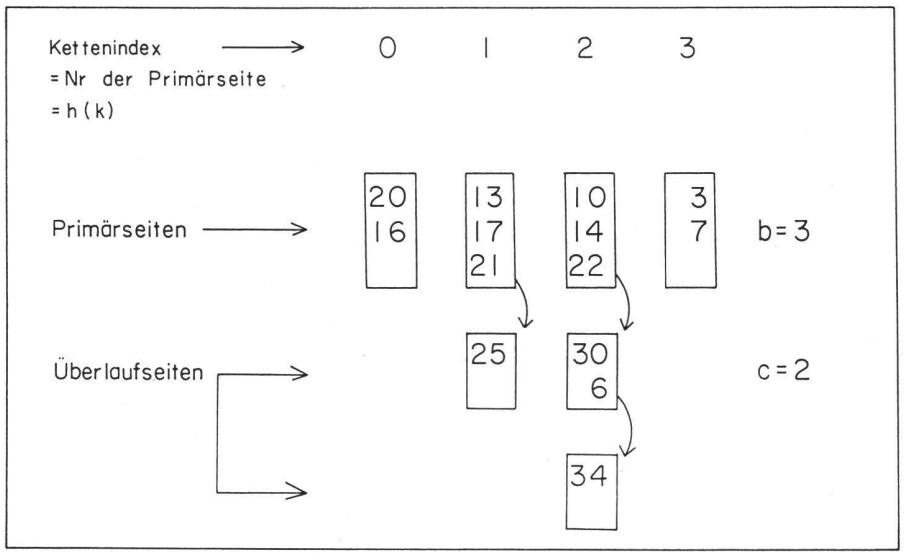

Abb. 2: Aufbau einer einfachen Hash-Datei durch Einfügen der Sätze mit Schlüssel k = 10,20,...; verwendete Hashfunktionen ist h(k) = k mod 4, maximale. Anzahl Sätze je Primärseite ist b = 3 und je Überlaufseite c = 2.

einer Datenbank vgl. C.A. Zehnder [1985] Abschnitte 2 und 5 und A. Meier [1982]).

Beispiel (siehe Abb. 1):
Ein Satz unserer Datei, in der Punkte der Ebene abgespeichert und nach Koordinaten gesucht werden sollen, hat zwei Schlüssel (erste und zweite Koordinate des Punktes) und beispielsweise drei Datenfelder Punktnummer, Klasse, Versicherung.

Unsere Frage lautet allgemein: Wie ist eine Datei zu organisieren, wenn nach mehreren Schlüsseln gleichzeitig mit minimalem Aufwand gesucht werden soll (file organisation for multi-key-access)? Als Hash-Datei, lautet die Antwort, die im folgenden vorgeführt werden soll. Dazu werden weitere Definitionen benötigt: **Hash-Funktion** (Adressberechnungsfunktionen) heisst eine Funktion, die jedem Satz auf Grund seiner Schlüssel eine Kette zuordnet, in der er abgespeichert wird. **Kette** (chain) heisst eine lineare Liste von Seiten. **Seite** (page, bucket) nennt man eine Satzfolge, die durch einen Zugriff auf das externe Speichermedium (Platte/Trommel) gelesen oder geschrieben werden kann. Die erste Seite einer Kette heisst **Primärseite** (primary page) mit Platz für maximal b Sätze, die Folgeseiten einer Kette heissen **Überlaufseiten** (overflow pages) mit Platz für maximal c Sätze.

Beispiel: Einfügen der folgenden Sätze (1 Schlüssel k, 0 Datenfelder)

k = 10, 20, 3, 7, 13, 14, 17,
21, 25, 16, 22, 30, 6, 34

in eine Datei mit b = 3 und c = 2 mit Hilfe der Hash-Funktion

$$h(k) = k \bmod 4.$$

(siehe Abb. 2)

Unsere Hash-Dateien sind – wie im obigen Beispiel – charakterisiert durch feste Seitengrösse, festen Primärbereich und getrennte Verkettung für Behandlung der Überlaufseiten. Solche Dateien heissen **traditionelle Hash-Dateien** nach P.-Å. Larson [1983]. (Vgl. auch D.E. Knuth [1973] und W.A. Burkhard [1983]).

2. Realisierung der Koordinaten Hash-Datei (kurz KHD)

Wir erläutern den Aufbau einer Koordinaten Hash-Datei (kurz KHD) für Punkte der Ebene. Erster Schritt ist die Zerlegung der Ebene in Zellen (Abb. 3).

Setzen wir in die Ebene ein orthogonales Geradengitter mit Gitterdistanz δ in beiden Richtungen, dann ist sie lückenlos überdeckt durch Gitterquadrate mit linkem und unterem Rand, aber ohne rechten und oberen Rand. Diese Gitterquadrate heissen **Zellen**. Die Datensätze aller Punkte im Innern oder auf dem linken oder unteren Rand einer Zelle werden in dieselbe Kette der KHD abgespeichert. In seiner Übersichtsarbeit beschreibt J.L. Bentley [1979] sechs Datenstrukturen für Bereich-Suche. Das Zellenkonzept unserer KHD entspricht seiner Variante 3.

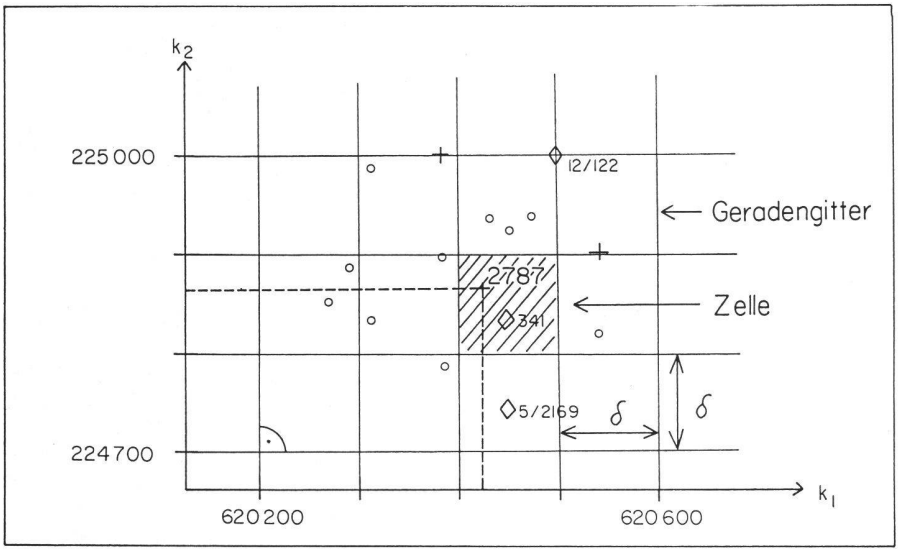

Abb. 3: Zelleneinteilung der Ebene.

Dateien	F1	F2	F3	F4	F5	F6
n	27956	10187	12459	9575	17869	11570
N	3517	1733	1733	1831	1733	3517
P	1978	811	1264	1070	1619	1903
E	1532	922	469	761	114	1614
S	1192	408	446	256	551	216
A	0.673	0.569	0.785	0.635	0.950	0.568
g	0.568	0.420	0.514	0.373	0.737	0.235
a	0.424	0.340	0.408	0.328	0.559	0.221

Tab. 1: Seitenbelegung A, Belegungsfaktor g und Satzbelegung a von 6 Koordinaten-Hash-Dateien.

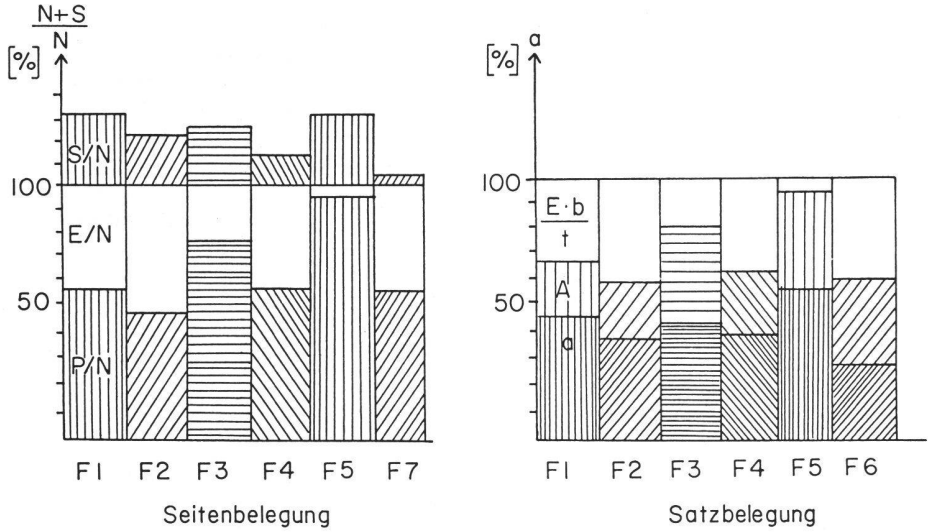

Abb. 4: Seitenbelegung und Satzbelegung von 6 Koordinaten-Hash-Dateien, Graphik zu Tabelle 1.

Die Adresse der Primärseite der Kette, in welcher der Punkt mit den Koordinaten (k_1, k_2) abgespeichert wird, berechnet sich durch folgende Hash-Funktionen:

$$(2.1)\quad h(k_1,k_2) = \left(10000\frac{k_1}{\lfloor\delta\rfloor} + \frac{k_2}{\lfloor\delta\rfloor}\right) \bmod N+1$$

Beispiel: Punkt 2787 mit
$(k_1,k_2)=(620422.716, 224863.256)$
soll in der KHD abgespeichert werden mit $N=7$ und $\delta=100$. Einsetzen aller Werte in (2.1) ergibt

$$h(k_1,k_2)=62042248 \bmod 7 + 1 = 3,$$

Punkt 2787 kommt also in Kette 3.

Zur Wahl der Dateiparameter δ und N: Um eine gleichmässige Belegung aller zur Verfügung stehenden Primärseiten zu gewährleisten, sollte N eine Primzahl sein (vgl. D.E.Knuth [1973]). Im übrigen können für die Bestimmung von δ und N nur heuristische Regeln angegeben werden. Falls in einer Primärseite b Sätze Platz finden können und für das ganze Operat etwa t Punkte zu erwarten sind, sollte $N \geqslant t/b$ gewählt werden. Die Gitterdistanz δ sollte so sein, dass im Gitterquadrat «selten» mehr als b Punkte liegen, dass aber auch nicht «zu viele» Gitterquadrate leer sind. Die Festlegung von δ und N nennen wir **Dimensionierung** der KHD.

3. Laufzeitverhalten grosser KHD

Von 6 KHD haben wir uns die folgenden Zahlen beschafft: Die Anzahl N reservierter Primärseiten, die Anzahl E leerer Primärseiten, die Anzahlen Q_λ ($\lambda=1,2,...$) von Ketten der Länge λ und die Gesamtzahl n effektiv gespeicherter Sätze, d.h. Punkte. Primär- und Überlaufseiten sind gleich gross: $b = c = 14$. Damit haben wir die Anzahl S der Sekundärseiten, die Anzahl $P \leqslant N$ der effektiv belegten Primärseiten, die Seitenbelegung A, den Belegungsfaktor g und die Satzbelegung a (auch etwa Speicherplatzausnützung genannt) berechnet gemäss den Formeln (3.1) bis (3.4). Ferner gestatten uns diese Daten eine grobe Abschätzung der mittleren Anzahl Zugriffe auf den externen Speicher pro Suchoperation gemäss Formeln (3.5) und (3.6).

Zunächst zur Speicherausnützung:

$$(3.1)\quad S = \sum_{\lambda \geqslant 2} Q_\lambda(\lambda-1)\ ;\ P = \sum_{\lambda \geqslant 1} Q_\lambda$$

$$(3.2)\quad g := n/(N\cdot b) \qquad \text{Belegungsfaktor}$$

$$(3.3)\quad a := n/t \qquad \text{Satzbelegung}$$
$$t := N\cdot b + S\cdot c$$

$$(3.4)\quad A := (P+S)/(N+S) \qquad \text{Seitenbelegung}$$

Die Ergebnisse sind in Tabelle 1 und Abb. 4 zusammengestellt.

Bei der Beurteilung dieser Belegungswerte müssen die Dateien F2 und F4 ausser Betracht gelassen werden, da sie noch nicht die Hälfte der Punkte enthalten, für welche sie dimensioniert wurden. D.h. sie enthalten weniger als t/2 Punkte (vgl. Abschnitt 2). Ihre Belegung ist untypisch klein. Auch F6 ist untypisch belegt. Diese Datei wurde irrtümlicherweise viel zu gross dimensioniert. Bei F1 und F3 fällt auf, dass die Zahl der unbenützten Primärseiten grösser ist als die Zahl der Überlaufseiten. Allgemein ist die Satzbelegung (Speicherplatzausnützung) maximal 56% (bei F5). Mögliche Gründe für die festgestellte unvorteilhafte Platzausnützung sind

- Lokal wesentlich mehr als 14 Punkte pro Zelle (das ist sicher bei Datei F1 der Fall, die teilweise Stadtgebiet umfasst)
- Hash-Funktion nicht optimal in dem Sinne, dass die Zellen nicht gleichmässig auf Seiten abgebildet werden

Gezielte Untersuchungen sollen zur Klärung dieser Fragen führen.

Dateien	F1	F2	F3	F4	F5	F6
$\overline{z}_m$	2.139	1.932	1.815	1.448	1.648	1.256
$\overline{z}_h$	1.570	1.464	1.408	1.224	1.324	1.128

Tab. 2: Mittlere Anzahl Zugriffe auf Disk bei erfolglosem Suchen ($\overline{z}_m$) bzw. erfolgreichem Suchen ($\overline{z}_h$).

Um die Anzahl der Diskzugriffe pro Suchoperation abzuschätzen, machen wir die folgenden vereinfachenden Annahmen: In jeder besetzten Seite der KHD sind genau b = c = 14 Sätze gespeichert, also die maximal mögliche Zahl. Die Wahrscheinlichkeit dafür, Anfangsseite der (mit oder ohne Erfolg) abzusuchenden Kette zu sein, sei für alle Primärseiten proportional zur Länge λ ihrer Kette, nämlich $\lambda/(P+S)$. Beim erfolgreichen Suchen sei die bedingte Wahrscheinlichkeit dafür, dass der gefundene Satz in einer bestimmten der λ Seiten seiner Kette liege, für alle Seiten der Kette gleich gross, nämlich $1/\lambda$. Damit lässt sich $\overline{z}_m$, die mittlere Anzahl Zugriffe bei erfolglosem Suchen (miss), und $\overline{z}_h$, die mittlere Anzahl Zugriffe bei erfolgreichem Suchen (hit) abschätzen mit Hilfe der Formeln (3.5) und (3.6)

$$(3.5)\quad \overline{z}_m = \frac{1}{P+S} \sum_{\lambda \geq 1} Q_\lambda \cdot \lambda^2$$

$$(3.6)\quad \overline{z}_h = \frac{1}{P+S} \sum_{\lambda \geq 1} Q_\lambda \cdot \frac{\lambda(\lambda+1)}{2}$$

Die Werte für unsere Testdateien sind in Tabelle 2 zusammengestellt.

Beurteilung: Die mittlere Anzahl Zugriffe ist im wesentlichen unabhängig von der Grösse der Dateien (vgl. F1 und F2). Auch bei ungünstigem Dateiaufbau, d.h. bei grossem Anteil an Überlaufseiten wie etwa bei F1, steigt die mittlere Anzahl Zugriffe nur knapp über 2, im allgemeinen ist sie deutlich unter 2. Diese Zahlen bestätigen unsere praktischen Erfahrungen des ausgezeichneten Zugriffsverhaltens. Es bleibt theoretisch zu klären, ob allgemein bei Hash-Dateien mit getrennter Verkettung gar nicht schlechteres Zugriffsverhalten zu erwarten ist.

4. Mit Hilfe von Koordinaten-Hash-Dateien gelöste Probleme

In unseren Parzellarvermessungsprogrammen setzt F. Link seit 1974 KHD ein. Damit werden heute folgende Aufgaben gelöst:

- Verwaltung der Punktdateien mit wahlweisem Zugriff über Punktenummern oder Koordinaten
- Automatische Numerierung von Punkten
- Automatische Mittelung oder Kontrolle mehrfach gemessener Punkte (für Grenzpunkte vorgeschrieben)
- Absuchen der Umgebung eines Punktes oder eines achsparallelen Rechtecksbereiches nach Punkten (Range querry)
- Berechnung von Absteckungselementen für Punkte, die nach geometrischen Kriterien ausgewählt werden können
- Darstellung von Punktfeldern und/oder Parzellengrenzen am graphischen Bildschirm, auf Wunsch mit Punktnummern und Parzellenidentifikation, und zwar planweise, bereichsweise oder parzellenweise (Display Graphik) mit Möglichkeit zur Vergrösserung von Ausschnitten (echtes Zooming).

Im Programmpaket zur Bearbeitung von Güterzusammenlegungen (Landumlegungen) kommen KHD zusätzlich zu Anwendungen der bereits besprochenen Art noch zum Einsatz für

- Herstellung kohärenter Netze von Parzellen mit automatischer Mittelung der digitalisierten Grenzpunkte und Erstellung der Umfahrungsdefinition von Parzellen der betreffenden Netze
- Verwaltung von Spezialdateien (z.B. Obstbäume) im Güterzusammenlegungsverfahren.

5. Verbesserungen und Verallgemeinerungen der KHD

Das Konzept der KHD kann natürlich ohne weiteres auf Punkte eines d-dimensionalen Raumes mit $d \geq 2$ übertragen werden und allgemeiner auf Dateien, deren Sätze nicht geometrische Punktdaten enthalten, die nach d-dimensionalen Koordinaten gesucht werden müssen, sondern irgendwelche Datenfelder und $d \geq 2$ Schlüssel, nach denen gleichzeitig zu suchen ist.

Wie in Abschnitt 3 notiert, möchten wir abklären, ob und wie die Satzbelegung der KHD verbessert werden kann. Für das «wie» scheinen uns drei Wege offen zu stehen: Einmal gilt es zu versuchen, Zellengrösse und/oder Hash-Funktion besser anzupassen an die Dateistruktur, d.h. an die Verteilung der tatsächlich vorkommenden Schlüsselwerte im d-dimensionalen Schlüsselraum. Zweitens könnte man versuchen, die Grösse des Primärbereiches dynamisch der Speicherplatzbelegung anzupassen durch Einsatz nicht nur einer einzigen Hash-Funktion h, sondern einer Folge $h_0, h_1, h_2, \ldots$ von Hash-Funktionen wie beim linearen Hashing von W. Litwin [1980] oder der von W.A. Burkhard [1983] vorgeschlagenen Verbesserung davon. Drittens müsste man sich fragen, ob der nicht benützte externe Speicherplatz nicht ausreichen würde, um die Datei zu ergänzen durch eine Indexdatei oder einen Indexarray zur seitenweisen Speicherung der je verwendeten Hash-Funktion wie beim virtuellen Hashing VHO oder VH1 von W. Litwin [1979] und beim dynamischen Hashing von P.-Å. Larson [1978] oder zur Speicherung einer Gitterstruktur zur Ermittlung der Seitenadresse (und damit effektiv Verzicht auf Hashing) wie beim Grid-File von J. Nievergelt et al. [1981]. Eine gute Übersicht über die zur Zeit existierenden Dateiorganisationen mit Hashing gibt P.-Å. Larson [1983]. Zu berücksichtigen wären ferner Quadtrees und weitere für geographische Daten geeignete Datenstrukturen (vgl. Übersichtsartikel von H. Samet [1964]). In jedem Fall ist genau abzuklären, ob die bessere Speicherplatzausnutzung mit einer wesentlich aufwendigeren Datenorganisation bezahlt werden muss, oder sogar mit einem Effizienzverlust beim Zugriff und ob sich das lohnt.

Zur Verbesserung der KHD-Umgebung sind wir schliesslich daran interessiert, diese Dateiorganisation in ein Datenbanksystem einzubetten, um die Vorteile des raschen Mehrfachschlüsselzugriffs mit Bequemlichkeit und Sicherheit der Datenbankverwaltung zu verbinden.

Adresse der Verfasser:
H.R. Gnägi, F. Link
Leupin AG
Dufourstrasse 45, CH-3005 Bern

Literatur

J.L. Bentley. J.H. Friedmann: Data Structures for Range Searching. Computing Surveys (4) 11, 1979 397 - 409.

W.A. Burkhard: Interpolation-Based Index Maintenance. BIT 23, 1983 274 - 294.

H.R. Gnägi, F. Link: Erfahrungen mit dem Einsatz grosser Hashdateien für raschen Zugriff auf Punktkoordinaten. Proc. GI-Fachtagung «Datenbank-Systeme für Büro, Technik und Wissenschaft», Karlsruhe, 20. - 22. 3. 1985, pp 474 - 475.

D.E. Knuth: The Art of Computer Programming. Vol 3: Sorting and Searching. Addison-Wesley, Reading Mass. 1973.

P.-Å. Larson: Dynamic Hashing. BIT (2) 18, 1978 184 - 201.

P.-Å. Larson: Dynamische Hashverfahren. Informatik-Spektrum (1) 6, 1983 7 - 19.

W. Litwin: Hachage virtuel: Une nouvelle technique d'adressage de mémoires. Thèse de Doctorat d'Etat, Univ. de Paris VI, 1979.

W. Litwin: Linear Virtuel Hashing: A New Tool for Files and Tables Implementation. Proc. 6th Conf. on Very Large Data bases, Montreal 1980, pp 212 - 223.

A. Meier: Semantisches Datenmodell für flächenbezogene Daten, Diss. ETH Nr. 7043, Zürich 1982.

J. Nievergelt, H. Hinterberger, K.C. Sevcik: The Grid File: An Adaptable Symmetric Multi-Key File Structure. Berichte des Instituts für Informatik ETH Nr. 46, Zürich 1981.

H. Samet: The Quadtree and Related Hierarchical Data Structures. Computing Surveys (2) 16, 1984, 187 - 260.

C.A. Zehnder: Informationssysteme und Datenbanken. B.G. Teubner, Stutgart 1985.