

*Hermann Tropf, Februar 2021*

***Wer bis zu dem Vorfall mit dem berühmten Professor Bayer weiterliest, wird meinen nachträglich großen Fleiß in dieser Angelegenheit verstehen.***

1981 haben Helmut Herzog und ich am Fraunhofer-Institut für Informations- und Datenverarbeitung, Karlsruhe, einen kleinen Artikel veröffentlicht.

H Tropf, H, Herzog: Multidimensional Range Search in Dynamically Balanced Trees.

Angewandte Informatik, pp. 71 - 77 (2. Febr. 1981). Link:

<https://hermanntropf.de/media/multidimensionalrangequery.pdf>

Der Artikel wird heute, nach 40 Jahren, noch zitiert. In der schnelllebigen Datenverarbeitung für einen Artikel, von zwei Nobodys, in einer nicht wirklich bedeutenden Zeitschrift, schon bemerkenswert, wie ich finde. Google Scholar findet derzeit allein für 2019 und 2020 noch 22 Zitate. Die Arbeit wurde in Oktober / November 1980 so nebenbei innerhalb von 2 Monaten realisiert, einschließlich Artikel-Schreiben. Anlässlich des 40-Jahre „Jubiläums“ habe ich mir mal die Mühe gemacht, nach der Wirkung des Artikels zu recherchieren. Und ich finde, das Rechercheergebnis ist beachtlich:

Die Methode ist in diversen Programmbibliotheken integriert, hat Eingang in kommerzielle Datenbanken gefunden, wird in unterschiedlichsten technischen Anwendungen eingesetzt und ist in Patentanmeldungen und wissenschaftlichen Veröffentlichungen Ausgangspunkt für weitere Entwicklungen. Und wer weiß, wo sie sonst noch überall verwendet wird, ohne Zitat.

## **Für näher Interessierte:**

Es geht um das allgemeine Problem, in einer großen Datenbank mit mehreren Dimensionen (z.B. Ort: *x-y-z* oder *Gewicht-Höhe-Breite-Länge* oder *Einkommen-Postleitzahl-Lebensalter*) effizient nach Datensätzen zu suchen, deren Einträge sich in einem vorgegebenen mehrdimensionalen Wertebereich befinden (z.B. *Einkommen von 2000 bis 4000, Postleitzahl von 68000 bis 69000, Alter von 50 bis 60*).

Grundidee des Verfahrens ist das bitweise Verschränken der Attribute eines Records zu einem einzigen Schlüssel (Bit Interleaving), nach dem die Daten sortiert werden. Für wirklich effizientes Durchsuchen haben wir zwei wichtige, zueinander spiegelbildliche Funktionen realisiert, von Helmut Herzog LITMAX und BIGMIN genannt. Die Bezeichner haben wir in den Artikel übernommen, sie haben Eingang in die Computer-literatur gefunden.

Diverse Fundstellen hier:

### **[Technische Anwendungen](#)**

### **[Datenbanksysteme](#)**

### **[Kommentare](#)**

### **[LITMAX/BIGMIN Quellprogramme im Internet](#)**

## **Was vor und nach unserer Veröffentlichung geschah:**

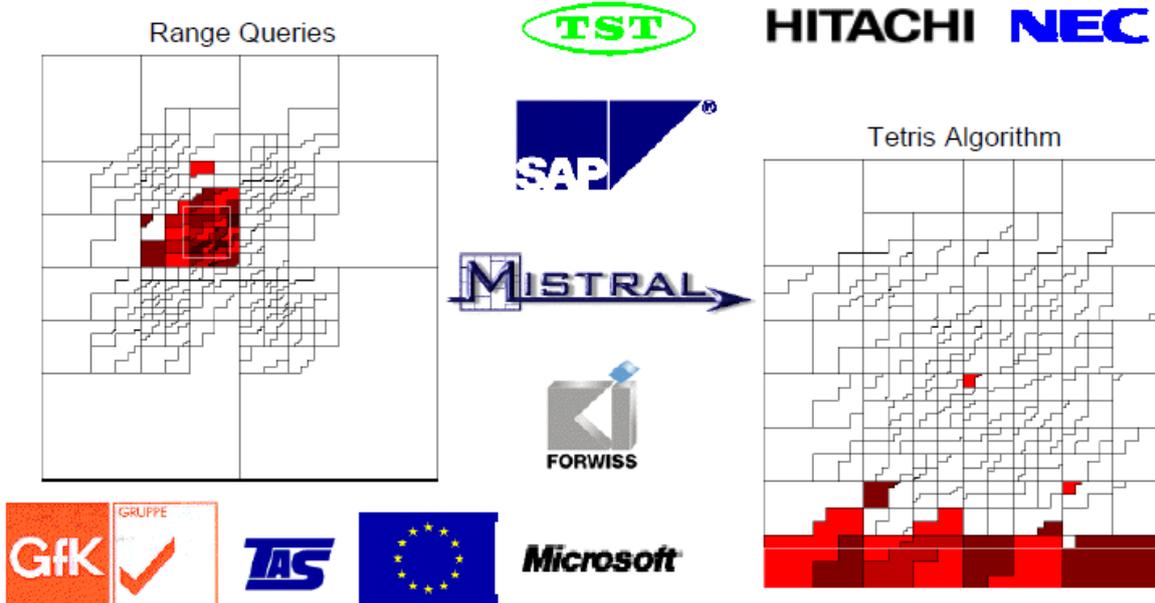
15 Jahre vor uns (1966), uns unbekannt, hatte G.M.Morton in einem IBM-Bericht für das Durchsuchen geografischer Daten (zwei Dimensionen *x* und *y*) eine Speicherreihenfolge vorgeschlagen, die der bitverschränkten Reihenfolge entspricht (ohne die Möglichkeit des Bitverschränkung zu erwähnen). Dadurch wurde die Wahrscheinlichkeit erhöht, dass man, ausgehend von einem Eintrag, geografisch benachbarte Einträge mit einem oder wenigen kleinen Suchschritten erreicht. Es handelte sich damals natürlich um statische Daten und es gab auch keine

Beschleunigung durch eine Funktion, die LITMAX / BIGMIN entspricht. Irgendwann entstand der Begriff „Morton Order“. Wenige Jahre nach uns (1986) erschien die Grundidee bei Orenstein / Merret "Spatial Query Processing in an Object-Oriented Database System". Das dort vorgeschlagene Verfahren, das im Ergebnis unseren Funktionen LITMAX und BIGMIN entspricht, ist jedoch wesentlich aufwändiger (Erläuterung siehe US-Patent 7321890B2). Orenstein prägte für die Grundidee die heute übliche Bezeichnung Z-Order.

So weit so gut. Aber das Folgende ist schon seltsam: Ganze 16 Jahre nach uns (1997) präsentierte der berühmte Professor Rudolf Bayer, München, genau dasselbe, nur für seine B+-Bäume statt binärer Suchbäume (unsere Methode funktioniert mit allen eindimensional sortierten Datenstrukturen - steht so ausdrücklich im Artikel - also selbstredend auch mit B+ Bäumen). Er hatte aber keine gute LITMAX/BIGMIN-Funktion. Sein Ansatz war exponentiell mit der Anzahl der Dimensionen und somit für mehr als 4 Dimensionen nicht zu gebrauchen. Er lancierte dann ein riesiges EU-Forschungsprojekt (MISTRAL), bei dem ein Algorithmus "GetNextZ-address" herauskam, der identisch ist mit unserem LITMAX-Algorithmus! Das war das wesentliche Ergebnis eines riesig aufgeblähten EU-Projekts (mit zahlreichen Institutionen und namhaften Firmen als Kooperationspartnern).



<http://mistral.in.tum.de>



Wir hatten das 15 Jahre vorher schon gefunden und auf wenigen Seiten beschrieben.

Wie das? Schlummerte unsere Arbeit lange Zeit unerkannt? Eigentlich nicht. Unsere Arbeit wurde nach Erscheinen durchaus zitiert, wenn auch anfänglich selten. z.B. ab 1984 von Hanan Samet, dann auch in seinem umfangreichen Lehrbuch „Foundations on Multidimensional and Metric Data Structures“, in dem sie nicht nur zitiert, sondern auch ausführlich beschrieben und gewürdigt wurde.

### **Ein Hinweis für Leute, die es direkt verwenden wollen:**

Beim Bitinterleaving eines Datenbank-Records mit den Attributen z.B.  $x$ ,  $y$ , und  $z$  entsteht eine Folge von Bits  $xyzxyzxyz...$ . Diese u.U. sehr lange Bitfolge wird als Binärzahl interpretiert und die Daten werden nach dieser Binärzahl sortiert oder indiziert.

Vor ein paar Jahren wurde mir klar, dass das Interleaving nicht explizit durchgeführt werden muss: es genügt, die Daten so stehen zu lassen wie sie sind, sie aber in der dem Interleaving

entsprechenden Reihenfolge zu verarbeiten. Die Erläuterung der LITMAX/BIGMIN Berechnung wird dadurch auch leichter verständlich.

Hier stelle ich ein entsprechendes Pascal-Programm zur Verfügung:

**[DBCode mit Erläuterung](#)**