Hermann Tropf, February 2021

Those who read on until the incident with the famous Professor Bayer, Munich, will understand my subsequently great diligence in this matter.

In 1981, Helmut Herzog and I, working at the Fraunhofer-Institute for Information and Data Processing (IITB) , Karlsruhe, Germany, published a small article

H Tropf, H, Herzog: Multidimensional Range Search in Dynamically Balanced Trees.
Angewandte Informatik, pp. 71 - 77 (2. Febr. 1981). Link:

https://www.vision-tools.com/h-tropf/multidimensionalrangequery.pdf

The article is still cited today, after 40 years. For an article in the fast evolving data processing area, published by two nobodies in a not really important journal, I think this is quite remarkable. Google Scholar currently still finds 22 citations for 2019 and 2020 alone. The work was realized in October / November 1980 so incidentally within 2 months, including article writing. On the occasion of the "40th anniversary" I took the trouble to research the impact of the article. The method is integrated into various program libraries, has found its way into commercial databases, is used in a wide variety of technical applications, and it is the starting point for further developments in patent applications and scientific publications. And who knows where else it is used, without citation.

For those who are interested in more detail:

The general problem is to search efficiently in a large database with multiple dimensions (examples: location x-y-z | weight-height-width-length | income-postcode-age) for records whose entries are in a given multidimensional range of values (for instance in the last example search records with income from 2000 to 4000, postcode from 68000 to 69000, age from 50 to 60).
The first idea of the procedure is the bitwise interleaving of the attributes of a record to a single key, according to which the data are sorted. As an advantage, there is free choice of structuring the

data, so well known methods such as balanced trees can be used to cope with dynamic data (in contrast for example to R-trees where special considerations are necessary). Similarly, this independence makes it easier to incorporate the method into existing databases.

However, binary searching in these data turned out to be not really efficient. For really efficient searching, we have defined and implemented two important balanced functions, called LITMAX and BIGMIN in Helmut Herzog's (Fortran) program code. We have adopted his identifiers in the article, the names have found their way into the computer literature.

Various references here:

Technical Applications

Database Systems

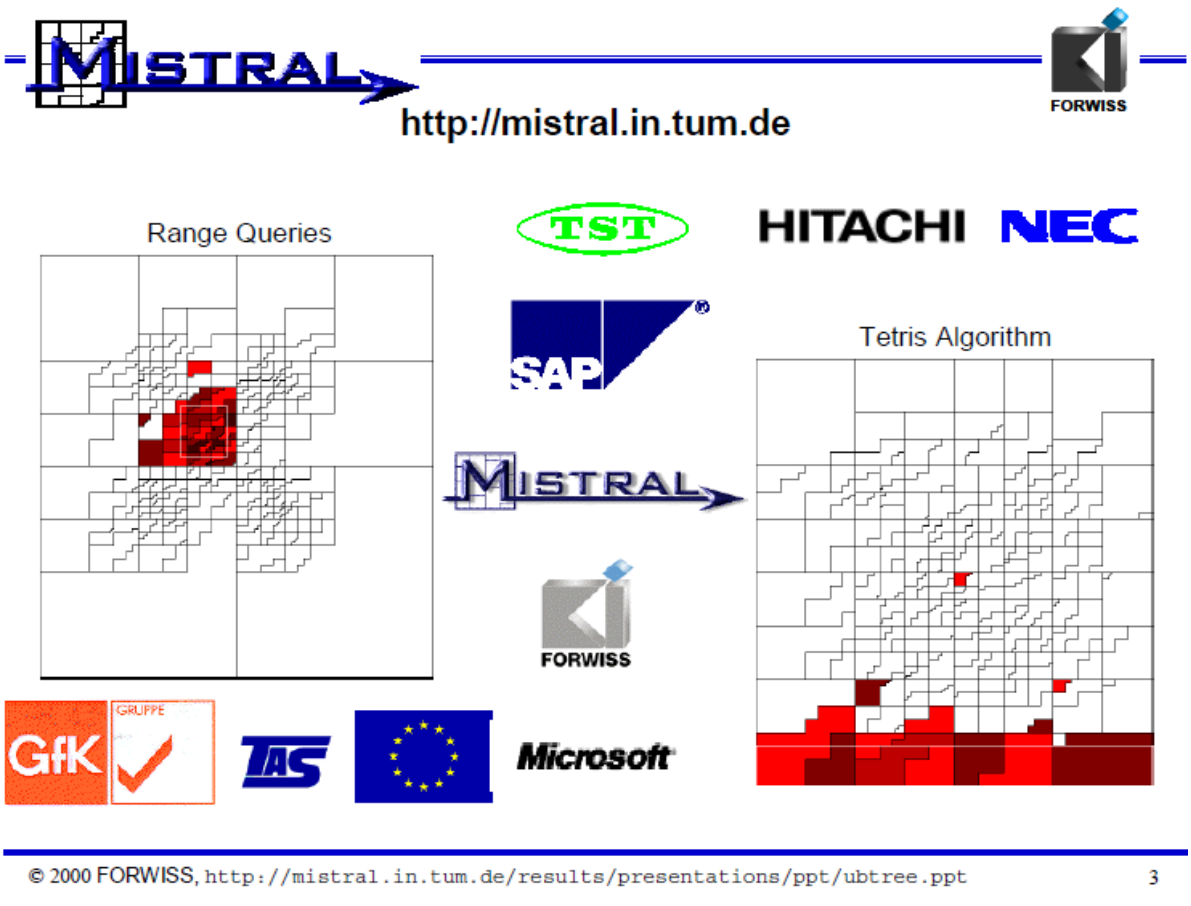
Comments

LITMAX/BIGMIN Source Code in the Internet

What happened before and after our publication:

15 years before us (1966), unknown to us, G.M.Morton had proposed in an IBM report for searching geographical data (two dimensions x and y) a memory order corresponding to the bit interleaved order (without mentioning the possibility of bit interleaving). Thus the probability was increased that, starting from one record found, geographically neighboring records could be reached with one or a few small search steps. Of course, at that time it was static data and there was no acceleration by a function corresponding to LITMAX / BIGMIN. At some time, the term "Morton Order" came into existence. A few years after us (1986) the basic idea appeared in Orenstein / Merret "Spatial Query Processing in an Object-Oriented Database System". However, the procedure proposed there, which corresponds in the result to our functions LITMAX and BIGMIN, is substantially more complex (for explanation see US patent 7321890B2). Orenstein coined the now common name Z-Order.

So far so good. But the following is strange: 16 years after us (1997) the famous professor Rudolf Bayer, Munich, presented exactly the same, only for his B+trees instead of (dynamically balanced) binary search trees (our method works with any one-dimensional sorted data structure - it is stated explicitly in our article - so of course it works also with B+trees). But he did not have a good LITMAX/BIGMIN function. His approach was exponential with the number of dimensions and thus not usable for more than 4 dimensions. He then launched a huge EU research project (MISTRAL) where an algorithm "GetNextZ-address" came out, which is identical to our LITMAX algorithm! This was the main result of a hugely inflated EU project (with numerous institutions and well-known companies as cooperation partners).



We had found this 15 years before and described it on a few pages. How so? Did our work lie dormant for a long time unrecognized? Actually not. Our work was cited after its publication, even if rarely in the beginning, e.g. from 1984 by Hanan Samet, then also in his extensive textbook "Foundations on

Multidimensional and Metric Data Structures", in which it was not only cited, but also described and discussed in detail.

A note for people who want to use it directly:

When bit interleaving a database record with the attributes (e.g. X, Y, Z) a sequence of bits (xyzxyzxyz... starting with the most significant bit of X in the example) is created. This possibly very long bit sequence is interpreted as binary number and the data is sorted or indexed by this binary number. A few years ago I realized that interleaving does not have to be done explicitly: it is sufficient to leave the data as they are, but process them in the order corresponding to interleaving. This also makes the explanation of the LITMAX/BIGMIN calculation easier to understand.

Here I provide a detailed description and a corresponding Pascal program:

Code with Explanation