

GEOMESA

GeoMesa is an open source suite of tools that enables large-scale geospatial querying and analytics on distributed computing systems.

<https://www.geomesa.org/>

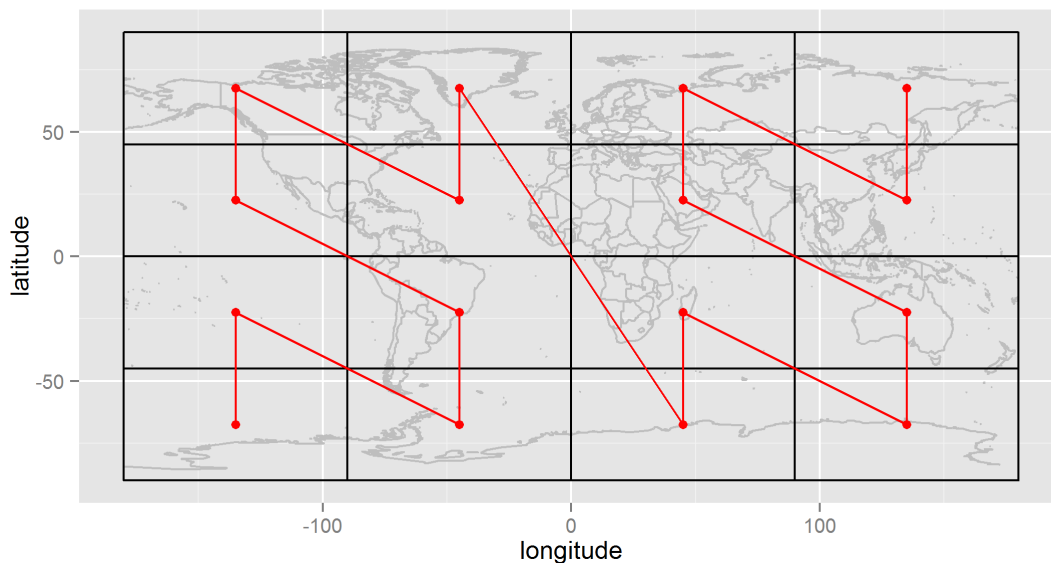
.....

(für 3 Dimensionen, Anwendung: Ort+Zeit)

```
/**
 * Returns (litmax, bigmin) for the given range and point
 */
def zdivide(p: Z3, rmin: Z3, rmax: Z3): (Z3, Z3) = {
  val (litmax, bigmin) = zdiv(load, MAX_DIM)(p.z, rmin.z, rmax.z)
  (new Z3(litmax), new Z3(bigmin))
}
```

<https://git.jingtao.fun/mirror/geomesa/commit/e8c4078928869741be4ab6b2e2bfd0ac08852fd>

At four bits precision, this space-filling curve will divide the flattened globe into 16 separate regions, and the sequence of values from 0000 to 1111 will trace out a Z-order curve as shown in Figure 1



https://www.eclipse.org/community/eclipse_newsletter/2014/march/article3.php

Center for Statistical Genetics

The Center for Statistical Genetics at the University of Michigan School of Public Health <https://github.com/statgen>

statgen / LDServer LDServer is a fast implementation of various metrics of linkage disequilibrium (LD) between genetic variants.
(Zu LD bin ich bei wiki „Assoziation (Genetik)“ fündig geworden)

1:1 aus unserem Artikel („decision table“) nachprogrammiert:

```
uint64_t compute_bigmin(uint64_t xd, uint64_t z_min, uint64_t z_max) {
    uint64_t bigmin = 0u;
    uint64_t mask = 0x8000000000000000u;
    uint32_t bit_position = 63u;
    do {
        uint64_t z_min_bit = z_min & mask;
        uint64_t z_max_bit = z_max & mask;
        uint64_t xd_bit = xd & mask;
        uint32_t dim = bit_position % 2u;
        uint64_t bit_mask = 0x1 << (bit_position / 2u);
        if (xd_bit == 0u && z_min_bit == 0u && z_max_bit > 0u) {
            bigmin = load_bits(bit_mask, bit_position, z_min, dim);
            z_max = load_bits(bit_mask - 1u, bit_position, z_max, dim);
        } else if (xd_bit == 0u && z_min_bit > 0u && z_max_bit == 0u) {
            // not possible because min <= max
            throw logic_error("Error while computing BIGMIN");
        } else if (xd_bit == 0u && z_min_bit > 0u && z_max_bit > 0u) {
            bigmin = z_min;
            return bigmin;
        } else if (xd_bit > 0u && z_min_bit == 0u && z_max_bit == 0u) {
            return bigmin;
        } else if (xd_bit > 0u && z_min_bit == 0u && z_max_bit > 0u) {
            z_min = load_bits(bit_mask, bit_position, z_min, dim);
        } else if (xd_bit > 0u && z_min_bit > 0u && z_max_bit == 0u) {
            // not possible because min <= max
            throw logic_error("Error while computing BIGMIN");
        }
        --bit_position;
        mask >>= 1;
    } while (mask != 0u);
    return bigmin;
}
```

Dito für Litmax

<https://github.com/statgen/LDServer/blob/master/core/src/Morton.cpp>

ähnlich:

<https://github.com/locationtech/sfcurve/blob/master/zorder/src/main/scala/org/locationtech/sfcurve/zorder/ZN.scala>

INRIA France

Institut national de recherche en informatique et en automatique

```
void spadas_get_LITMAX_BIGMIN(spadas_t *s, int min_code, int max_code,
position_t *min_pos, position_t *max_pos,
- int *BIGMIN, position_t *BIGMIN_pos, int
*LITMAX, position_t *LITMAX_pos, int code);
+void spadas_get_LITMAX_BIGMIN(spadas_t *s, int min_code, int max_code,
position_t *min_pos, position_t *max_pos, int *BIGMIN, position_t
*BIGMIN_pos, int *LITMAX, position_t *LITMAX_pos, int code);
```

<https://lists.gforge.inria.fr/pipermail/wsnet-commits/2008-February/000741.html>

```
function litMax(minz, maxz, zcode) {
  var litmax = minz;
  for (var p = BITMAX; p > 0; p--) {
    var mask = 1 << p;
    var v = (zcode & mask) && _100_ || _000_;
    if (minz & mask) v |= _010_
    if (maxz & mask) v |= _001_

    if (v === _001_)      maxz = setbits(p, maxz);
    else if (v === _011_) return litmax;
    else if (v === _100_) return maxz;
    else if (v === _101_) {
      litmax = setbits(p, maxz);
      minz   = unsetbits(p, minz);
    }
  }

  return litmax;
}
```

Dto bigMin:

```
function bigMin(minz, maxz, zcode)
..
```

Alexander Milevski <https://github.com/w8r>

<https://bl.ocks.org/w8r/d4660e6acbf56a0ad6b06e491afeec65>

next_zorder_index is re-implementation of Tropf, H.; Herzog, H. (1981), "Multidimensional Range Search in Dynamically Balanced Trees" in Python.

next_zorder_index is shown as BIGMIN in the paper.

<https://pypi.org/project/pyzorder/>

Little script to play around with the code and data of H. Tropf, H. Herzog's paper 1981 paper "Multidimensional Range Search in Dynamically Balanced Trees" Thanks to Bernhard Herzog (unrelated to H. Herzog)

<https://gist.github.com/4014664>

=

<https://gist.github.com/s-l-teichmann/4014664>

<http://restlebenszeit.blogspot.com/2012/11/raumfullend-von-z-kurven-und-leveldb.html>

(Mit Bigmin/Litmax Code)

.....Diese neue Position kann mittels eines von [H. Tropf und H. Herzog 1981](#) beschriebenen Verfahren (BIGMIN) im Programmcode effizient errechnet werden.

Folgende Programme testen diesen Ansatz:

[multidimrangesearch.py](#) Ist ein einfaches Python-Programm, das die Ergebnis von Tropf und Herzog nachvollzieht.

[bigmintest.go](#) ist eine Go-Implementation des BIGMIN-Algorithmus nach Tropf und Herzog. Es wird die Korrektheit mit einem Ergebnisvergleich mit einer naiven Implementation überprüft.

<http://restlebenszeit.blogspot.de/>

Nicht mehr gefunden!!!

Source Code for geotrellis.spark.io.index.zcurve.package.scala
[http://snacktrace.com/artifacts/com.azavea.geotrellis/geotrellis-spark_2.11/0.10.0-RC2/geotrellis.spark.io.index.zcurve.package\\$](http://snacktrace.com/artifacts/com.azavea.geotrellis/geotrellis-spark_2.11/0.10.0-RC2/geotrellis.spark.io.index.zcurve.package$)

Best [Java](#) code snippets using [com.bigdata.service.geospatial.ZOrderRangeScanUtil](#).

Class providing utility functions for efficient zOrder-based multi-dimensional range scans, such as efficient range checks and functionality for BigMin calculation. The latter follows the logics defined in the BIGMIN decision table as provided in <https://hermannthropf.de/media/multidimensionalrangequery.pdf>, page 76.

<https://www.codota.com/code/java/classes/com.bigdata.service.geospatial.ZOrderRangeScanUtil>

<https://blazegraph.github.io/database/apidocs/com/bigdata/service/geospatial/ZOrderRangeScanUtil.html>

(Mit Bigmin Code)

_Introducing a new Open Source C++ library for Spatial Representations (Nov. 2018).doc

ähnlich

<http://bl.ocks.org/daniarleagk/278709dedf09451b794ff72c4c05cda1>

Spatial hash on morton curve

<https://bl.ocks.org/w8r/d4660e6acbf56a0ad6b06e491afeec65>

(Mit Bigmin/Litmax Code)

Class ZCurve<T>

Z-Curve encoder/decoder and optimized bbox range extraction for arbitrary dimensions (≥ 2). Supports max. 32bit per-component value range and resulting Morton codes encoded as BigInt.

bigMin(zcurr: bigint, zmin: bigint, zmax: bigint): bigint

Defined in [zcurve.ts:190](#)

Computes the next valid Z index in bbox defined by zmin / zmax and greater than zcurr. Returns -1 if no further indices are in the box.

<https://docs.thi.ng/umbrella/morton/classes/zcurve.html#bigmin>

Alexander Milevski <https://github.com/w8r>

```
function litMax(minz, maxz, zcode) {
  var litmax = minz;
  for (var p = BITMAX; p > 0; p--) {
    var mask = 1 << p;
    var v = (zcode & mask) && _100_ || _000_;
    if (minz & mask) v |= _010_
    if (maxz & mask) v |= _001_

    if (v === _001_)      maxz = setbits(p, maxz);
    else if (v === _011_) return litmax;
    else if (v === _100_) return maxz;
    else if (v === _101_) {
      litmax = setbits(p, maxz);
      minz   = unsetbits(p, minz);
    }
  }

  return litmax;
}
```

Dto bigMin:

```
function bigMin(minz, maxz, zcode)
....
```

<https://bl.ocks.org/w8r/d4660e6acbf56a0ad6b06e491afeec65>

Project description

zCurve

zCurve is a Python module with methods to efficiently map multidimensional data to a single dimension while preserving locality of the data points. (...)

For range searching, zCurve offers two functions for calculating the necessary LITMAX and BIGMIN values: (..) This implementation is based on the following paper

H.Tropf, H. Herzog. "Multidimensional Range Search in Dynamically Balanced Trees."
ANGEWANDTE INFO. 2 (1981): 71-77.

and it makes heavy use of the excellent [gmpy2 module](#).

	x value													
	0	1	2	3	4	5	6	7	8					
0	0	2	8	10	32	34	40	42	128					
1	1	3	9	11	33	35	41	43	129					
2	4	6	12	14	36	38	44	46	132					
3	5	7	13	15	37	39	45	47	133					
4	16	18	24	26	48	50	56	58	144					
5	17	19	25	27	49	51	57	59	145					
6	20	22	28	30	52	54	60	62	148					
7	21	23	29	31	53	55	61	63	149					
8	64	66	72	74	96	98	104	106	192					
9	65	67	73	75	97	99	105	107	193					
10	68	70	76	78	100	102	108	110	196					
11	69	71	77	79	101	103	109	111	197					
12	80	82	88	90	112	114	120	122	208					
13	81	83	89	91	113	115	121	123	209					

When range searching, we can prune the search space by calculating BIGMIN (aka "GetNextZ-address") and LITMAX (aka "GetPrevZ-address") values.

<https://pypi.org/project/zCurve/>