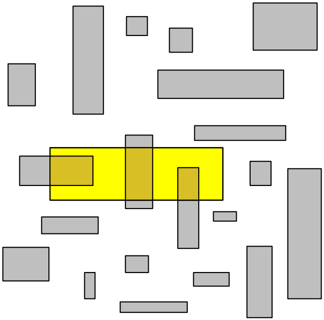
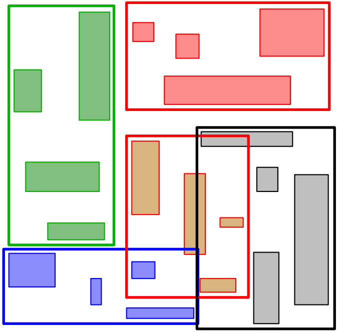
Space-filling curves for spatial data structures

Herman Haverkort & Freek van Walderveen, TU Eindhoven



Report all rectangles that intersect rectangular query range

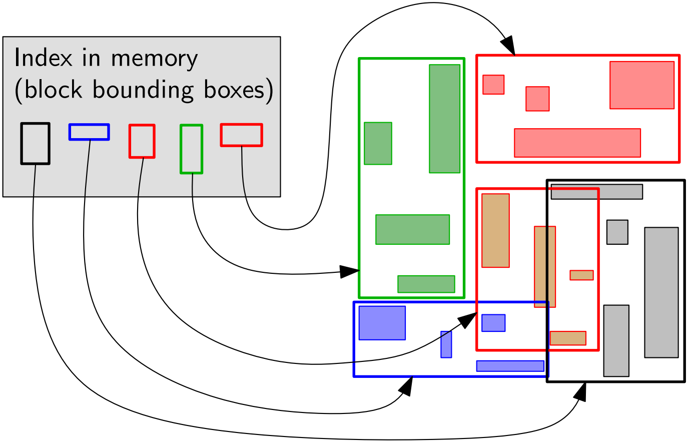
Space-filling curves for spatial data structures



Report all rectangles that intersect rectangular query range

Query time ≈ seek time ≈ disk blocks retrieved

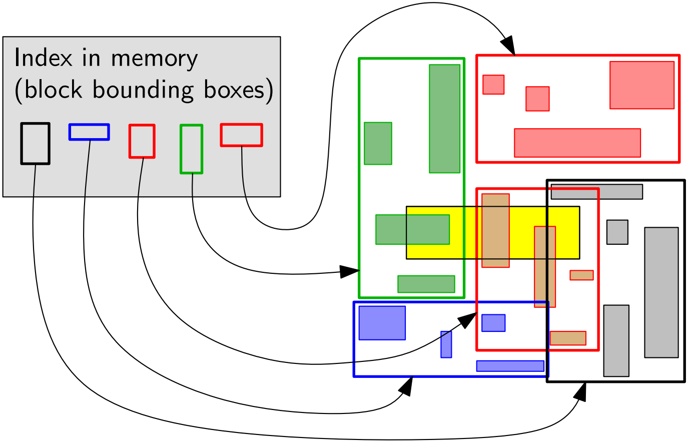
Space-filling curves for spatial data structures



Report all rectangles that intersect rectangular query range

Query time $\approx$ seek time $\approx$ disk blocks retrieved

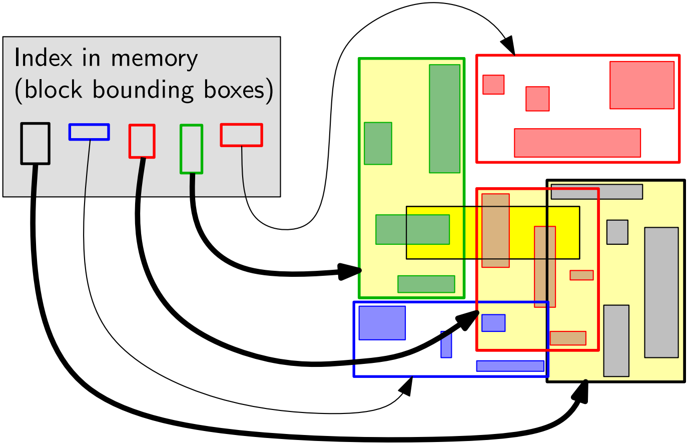
Space-filling curves for spatial data structures



Report all rectangles that intersect rectangular query range

Query time ≈ seek time ≈ disk blocks retrieved

Space-filling curves for spatial data structures

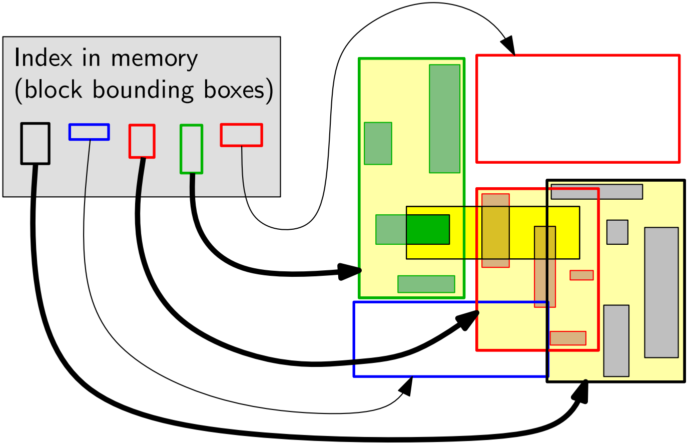


Report all rectangles that intersect rectangular query range

Query time ≈ seek time ≈ disk blocks retrieved

Only read blocks whose bounding boxes intersect query range

Space-filling curves for spatial data structures

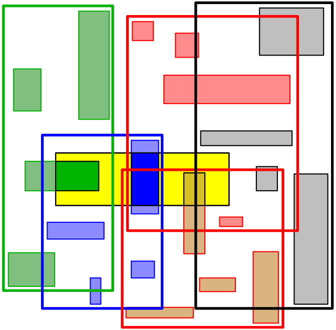


Report all rectangles that intersect rectangular query range

Query time ≈ seek time ≈ disk blocks retrieved

Only read blocks whose bounding boxes intersect query range

Space-filling curves for spatial data structures



Report all rectangles that intersect rectangular query range

Query time $\approx$ seek time $\approx$ disk blocks retrieved

Only read blocks whose bounding boxes intersect query range

Algorithms to make blocks

Theory:

blocks visited in ... for ...	rectangle query	point query
KDB-trees (point data)	$\Theta(\sqrt{n/B} + t/B)$	1
PR-trees (rectangle data)	$\Theta(\sqrt{n/B} + t/B)$	$\Theta(\sqrt{n/B})$

(n is input size; t is output size; B is disk block size)

Practice:

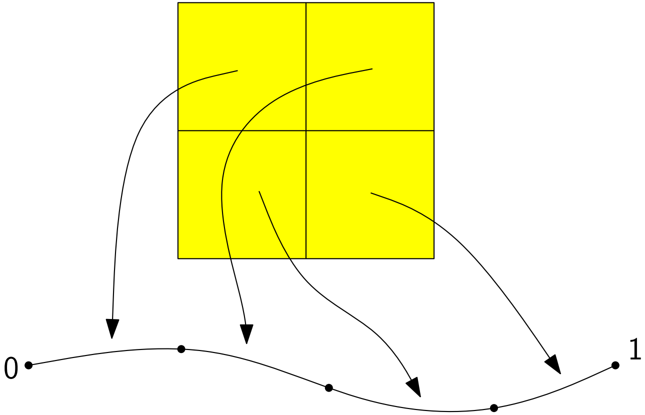
Use space-filling curve.

Example: Z-order for point data

Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE

Example: Z-order for point data

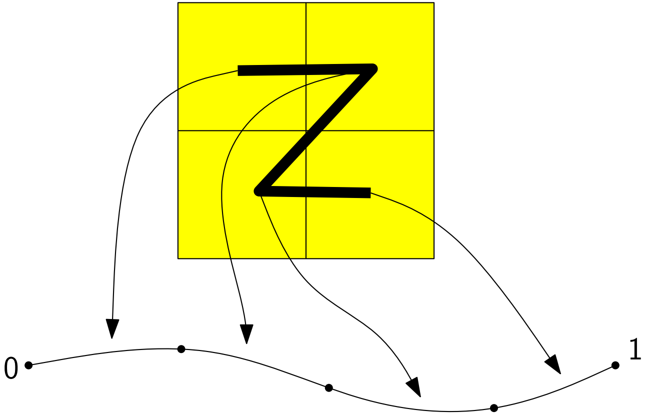
Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE



Example: Z-order for point data

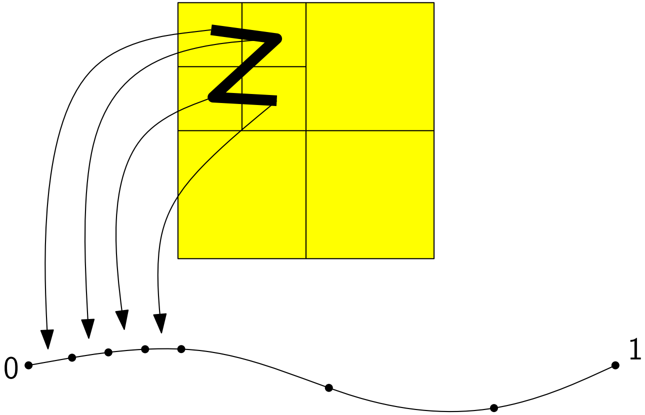
Space-filling curve: mapping from unit square to unit interval

Z-order: map quadrants recursively in order NW, NE, SW, SE



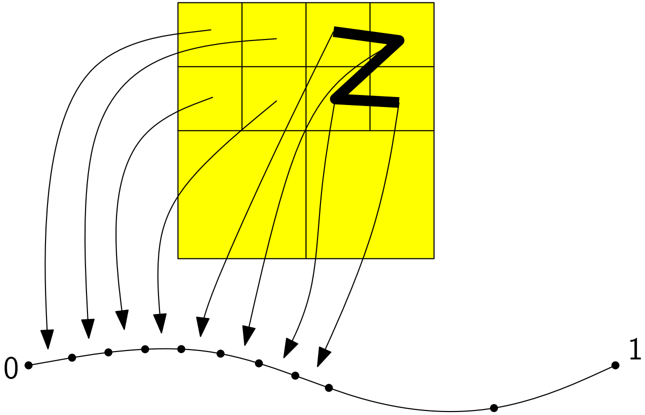
Example: Z-order for point data

Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE



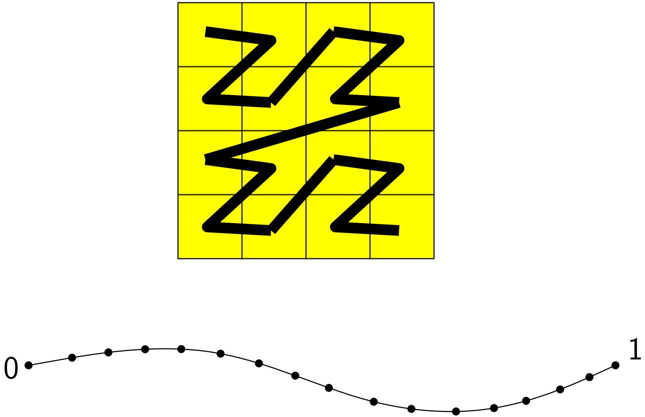
Example: Z-order for point data

Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE



Example: Z-order for point data

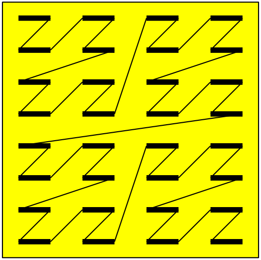
Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE

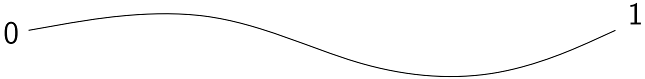


Example: Z-order for point data

Space-filling curve: mapping from unit square to unit interval

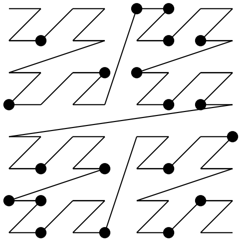
Z-order: map quadrants recursively in order NW, NE, SW, SE

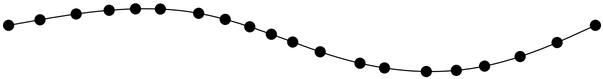




Example: Z-order for point data

Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE

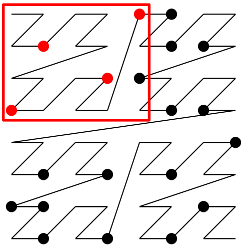


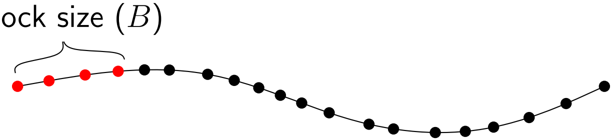


Group data into blocks in order along curve (example: block size = 4)

Example: Z-order for point data

Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE

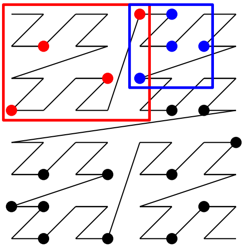


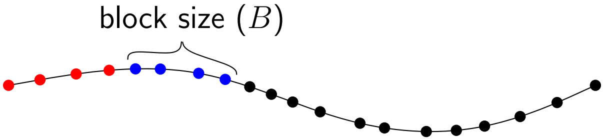


Group data into blocks in order along curve (example: block size $= 4$)

Example: Z-order for point data

Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE

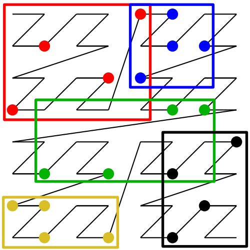


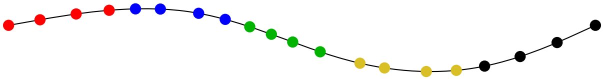


Group data into blocks in order along curve (example: block size = 4)

Example: Z-order for point data

Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE

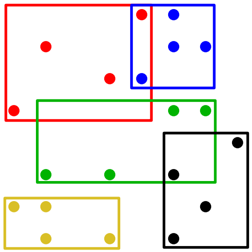




Group data into blocks in order along curve (example: block size = 4)

Example: Z-order for point data

Space-filling curve: mapping from unit square to unit interval
Z-order: map quadrants recursively in order NW, NE, SW, SE



Group data into blocks in order along curve (example: block size = 4)

Question: what to do with rectangle data?
Question: what is the best curve?

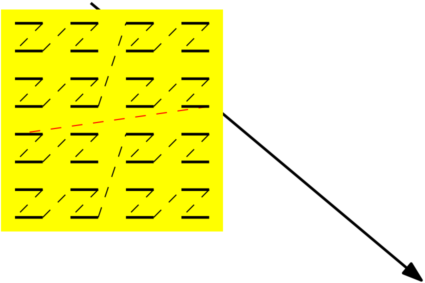
What curve?

Author A: *"Use the Peano curve!"*

Giuseppe Peano:
Sur une courbe, qui remplit toute une aire plane.
Math. Ann., 96(1):157–160, 1890

What curve?

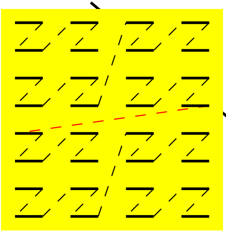
Author A: *"Use the Peano curve!"*



Giuseppe Peano:
Sur une courbe, qui remplit toute une aire plane.
Math. Ann., 96(1):157–160, 1890

What curve?

Author A: *"Use the Peano curve!"*

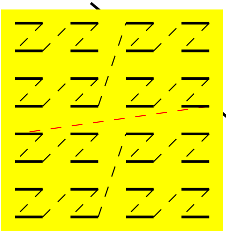


Author B: *"Use the Peano curve!"*

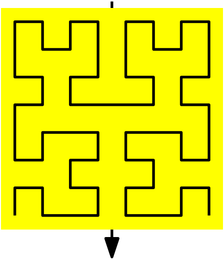
Giuseppe Peano:
Sur une courbe, qui remplit toute une aire plane.
Math. Ann., 96(1):157–160, 1890

What curve?

Author A: *"Use the Peano curve!"*



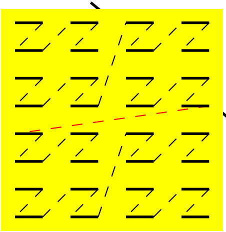
Author B: *"Use the Peano curve!"*



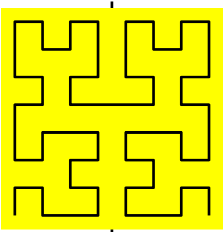
Giuseppe Peano:
Sur une courbe, qui remplit toute une aire plane.
Math. Ann., 96(1):157–160, 1890

What curve?

Author A: *"Use the Peano curve!"*



Author B: *"Use the Peano curve!"*

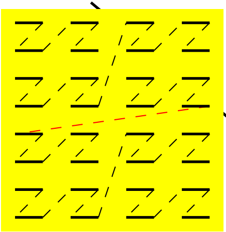


Giuseppe Peano:
Sur une courbe, qui remplit toute une aire plane.
Math. Ann., 96(1):157–160, 1890

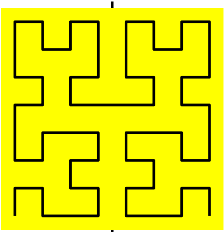
Author C: *"Use the Peano curve!"*

What curve?

Author A: *"Use the Peano curve!"*

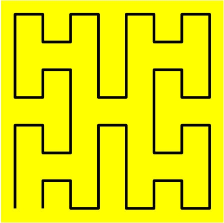


Author B: *"Use the Peano curve!"*



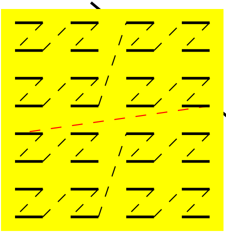
Giuseppe Peano:
Sur une courbe, qui remplit toute une aire plane.
Math. Ann., 96(1):157–160, 1890

Author C: *"Use the Peano curve!"*

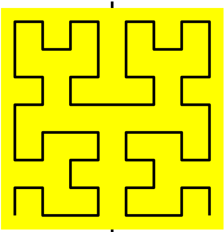


What curve?

Author A: *“Use the Peano curve!”*

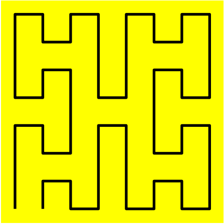


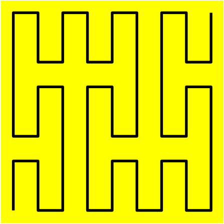
Author B: *“Use the Peano curve!”*



Giuseppe Peano:
Sur une courbe, qui remplit toute une aire plane.
Math. Ann., 96(1):157–160, 1890

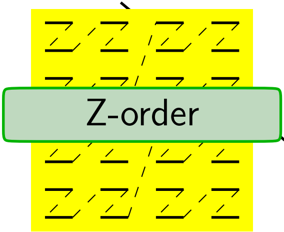
Author C: *“Use the Peano curve!”*





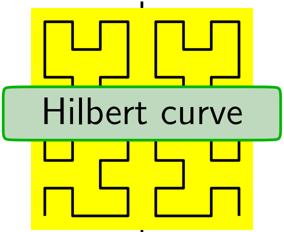
What curve?

~~Author A: *“Use the Peano curve!”*~~



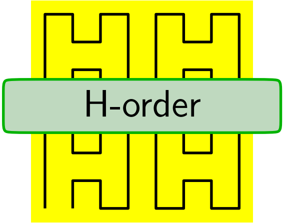
a.k.a. Lebesgue order, Morton indexing

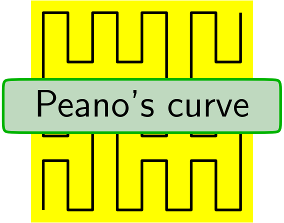
~~Author B: *“Use the Peano curve!”*~~



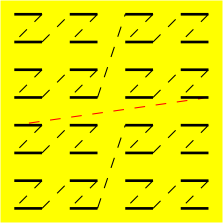
Giuseppe Peano:
Sur une courbe, qui remplit toute une aire plane.
Math. Ann., 96(1):157–160, 1890

~~Author C: *“Use the Peano curve!”*~~

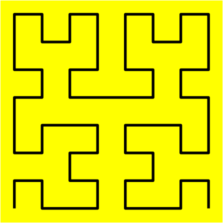




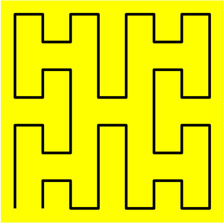
What curve?



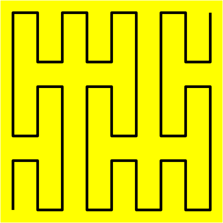
Z-order



Hilbert curve

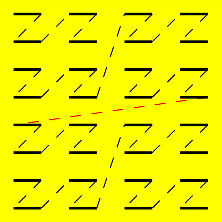


H-order

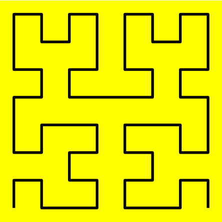


Peano's curve

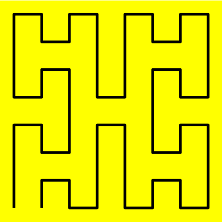
What curve?



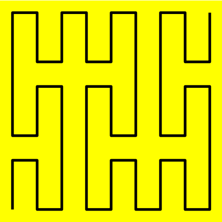
Z-order



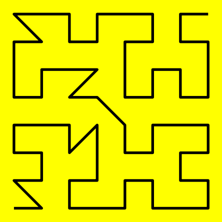
Hilbert curve



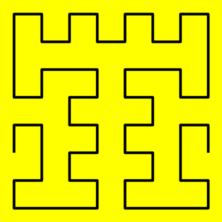
H-order



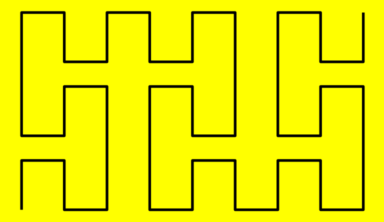
Peano's curve



AR^{2}W^2-curve



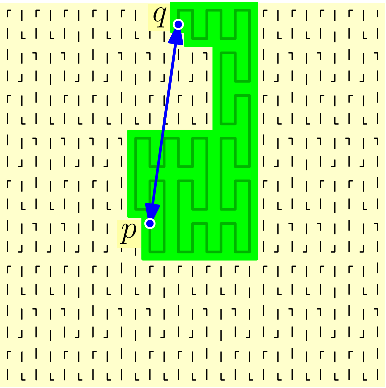
$\beta\Omega$-curve



Balanced Peano

Measuring curves

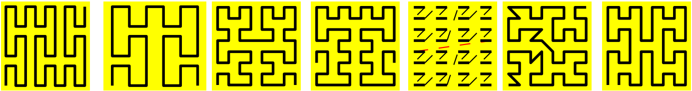
$$\text{worst-case dilation} := \max_{p,q \in \text{unit}\square} \frac{\text{squared distance between } p \text{ and } q}{\text{area filled by curve between } p \text{ and } q}$$

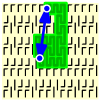


For a curve section of fixed size, how far can the endpoints be apart?

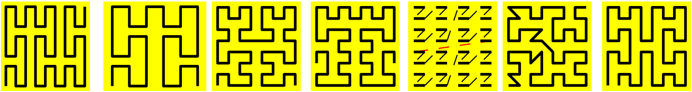
$\rightsquigarrow$ how big can the bounding box be?

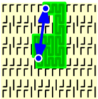
Measuring curves



		Peano	Balanced	Hilbert	$\beta\Omega$	Z-order	AR^{2}W^2	H-order	lower bound
worst dilation 	L_∞	8		6		∞		4	closed: 4
	L_2	8		6		∞		4	closed: 4
	L_1	$10\frac{2}{3}$		9		∞		8	closed: 8

Measuring curves



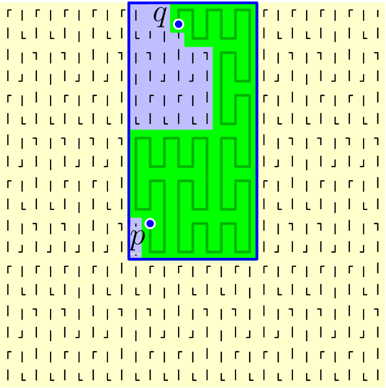


worst dilation

	Peano	Balanced	Hilbert	$\beta\Omega$	Z-order	AR^2W^2	H-order	lower bound
L_∞	8	4.62	6	5.00	∞	5.40	4	closed: 4
L_2	8	4.62	6	5.00	∞	6.05	4	closed: 4
L_1	$10\frac{2}{3}$	8.62	9	9.00	∞	12.00	8	closed: 8

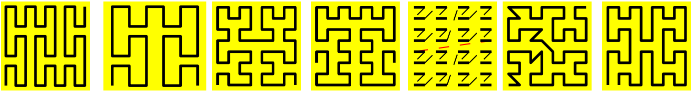
Measuring curves

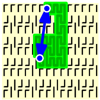
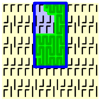
worst-case bbox area ratio $:= \max_{p,q \in \text{unit}\square} \frac{\text{bbox area of curve between } p \text{ and } q}{\text{area filled by curve between } p \text{ and } q}$



For a curve section of fixed size, how big can the bounding box be?

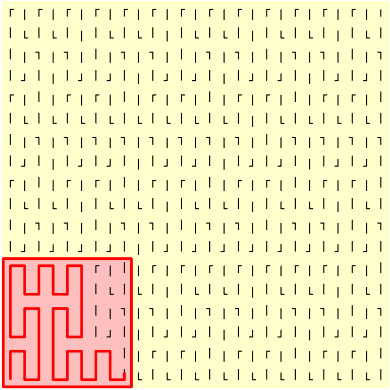
Measuring curves



		Peano	Balanced	Hilbert	$\beta\Omega$	Z-order	AR^2W^2	H-order	lower bound
 worst dilation	L_∞	8	4.62	6	5.00	∞	5.40	4	closed: 4
	L_2	8	4.62	6	5.00	∞	6.05	4	closed: 4
	L_1	$10\frac{2}{3}$	8.62	9	9.00	∞	12.00	8	closed: 8
 worst box area		2.00	2.00	2.40	2.22	∞	3.05	3.00	rectangles /triangles: 2

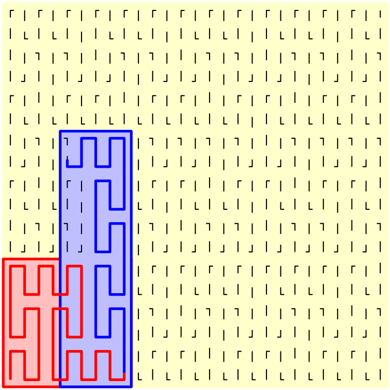
Measuring curves

random subdivision bbox area =
sum of section bbox areas of random subdivision of curve into sections



Measuring curves

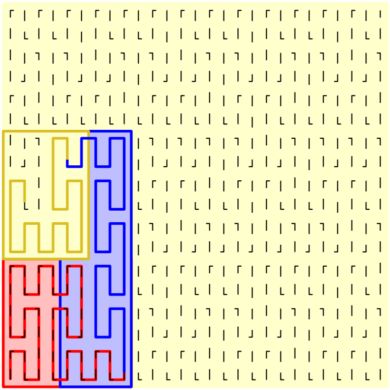
random subdivision bbox area =
sum of section bbox areas of random subdivision of curve into sections



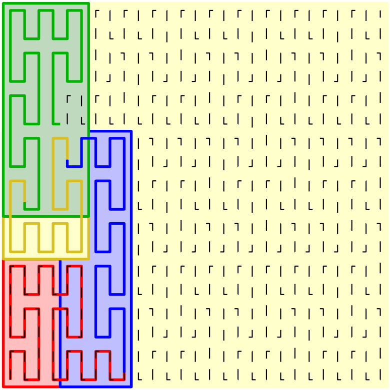
Measuring curves

random subdivision bbox area =

sum of section bbox areas of random subdivision of curve into sections

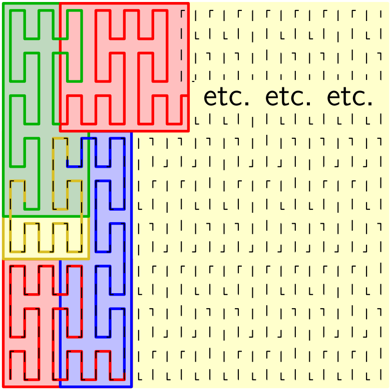


random subdivision bbox area =
sum of section bbox areas of random subdivision of curve into sections

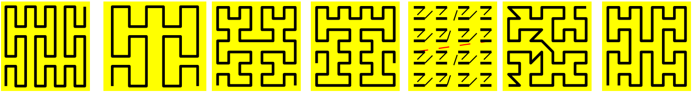


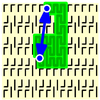
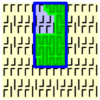
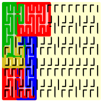
Measuring curves

random subdivision bbox area =
sum of section bbox areas of random subdivision of curve into sections

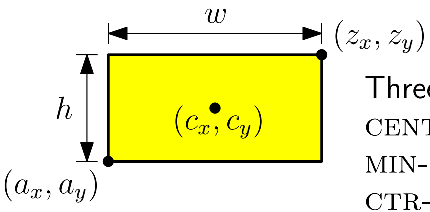


Measuring curves



		Peano	Balanced	Hilbert	$\beta\Omega$	Z-order	AR^{2}W^2	H-order	lower bound
 worst dilation	L_∞	8	4.62	6	5.00	∞	5.40	4	closed: 4
	L_2	8	4.62	6	5.00	∞	6.05	4	closed: 4
	L_1	$10\frac{2}{3}$	8.62	9	9.00	∞	12.00	8	closed: 8
 worst box area		2.00	2.00	2.40	2.22	∞	3.05	3.00	rectangles /triangles: 2
 total box area		1.42	1.42	1.41	1.40	2.86	1.47	1.69	

What to do with rectangle data on Hilbert curves?



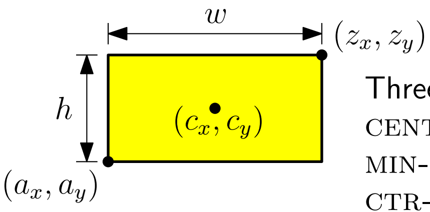
Three ways to order rectangles: map rectangle to:

CENTRE: (c_x, c_y), order along 2D curve

MIN-MAX: (a_x, a_y, z_x, z_y), order along 4D curve

CTR-SIZE: (c_x, c_y, w, h), order along 4D curve

What to do with rectangle data on Hilbert curves?



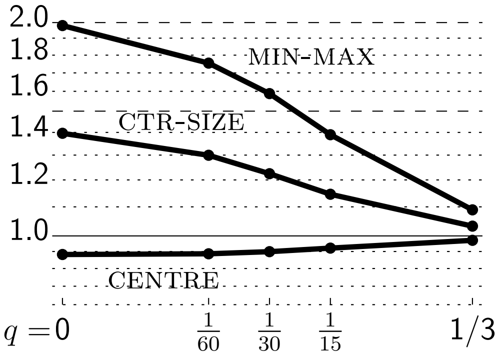
Three ways to order rectangles: map rectangle to:

CENTRE: (c_x, c_y), order along 2D curve

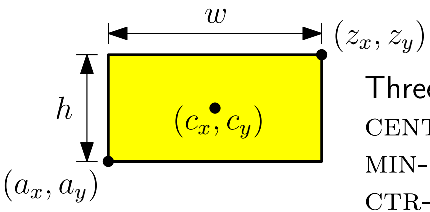
MIN-MAX: (a_x, a_y, z_x, z_y), order along 4D curve

CTR-SIZE: (c_x, c_y, w, h), order along 4D curve

Relative number of blocks visited by queries of size q in data Kamel and Faloutsos



What to do with rectangle data on Hilbert curves?



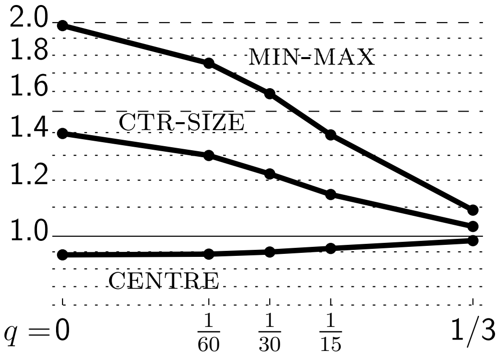
Three ways to order rectangles: map rectangle to:

CENTRE: (c_x, c_y), order along 2D curve

MIN-MAX: (a_x, a_y, z_x, z_y), order along 4D curve

CTR-SIZE: (c_x, c_y, w, h), order along 4D curve

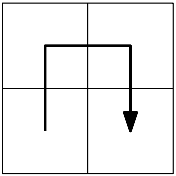
Relative number of blocks visited by queries of size q in data Kamel and Faloutsos



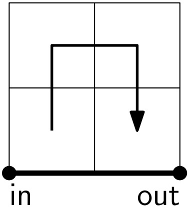
Their conclusion:
4D curves are no good

Our conclusion:
You used the wrong curve!

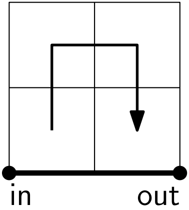
Making Hilbert curves in 2D

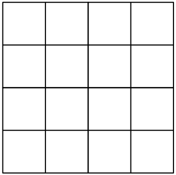


Making Hilbert curves in 2D

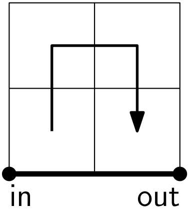


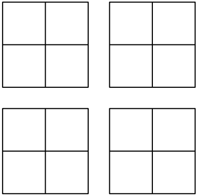
Making Hilbert curves in 2D



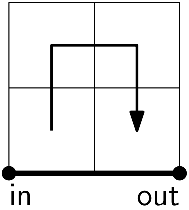


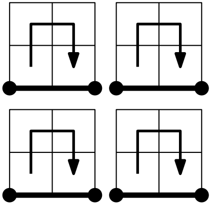
Making Hilbert curves in 2D



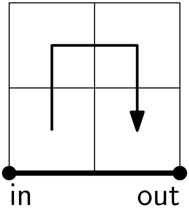


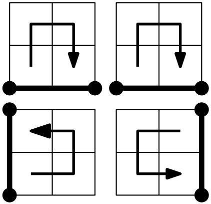
Making Hilbert curves in 2D



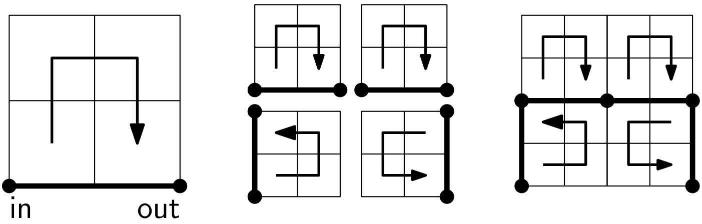


Making Hilbert curves in 2D

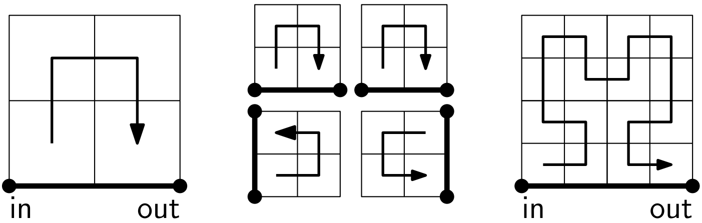




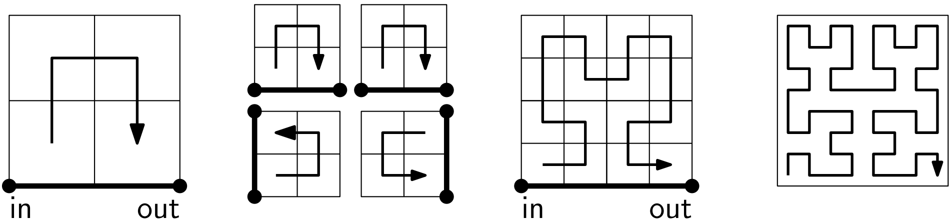
Making Hilbert curves in 2D



Making Hilbert curves in 2D

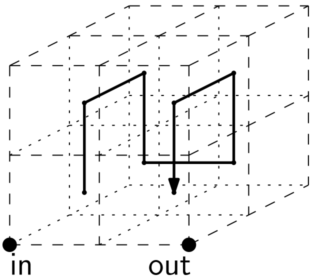


Making Hilbert curves in 2D

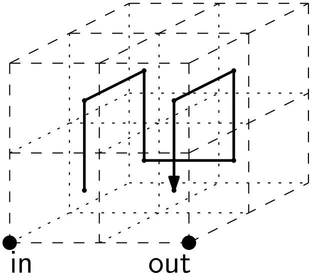


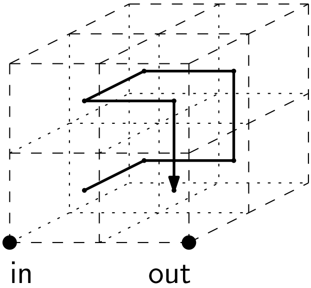
There is only one way to do this

Making Hilbert curves in 3D

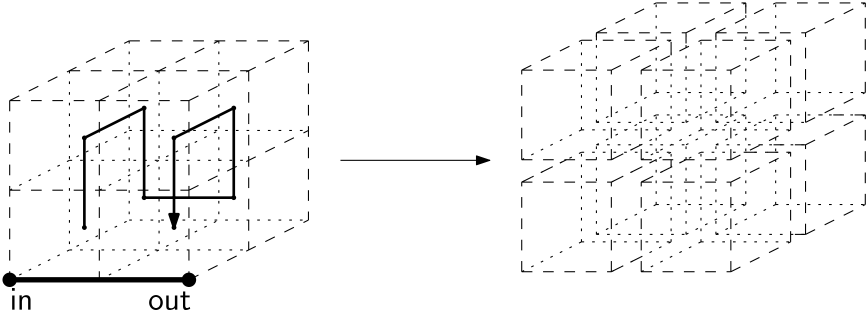


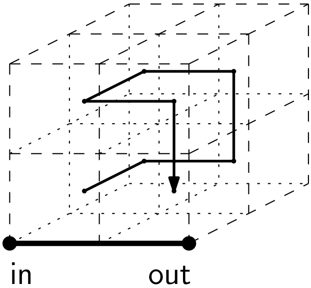
Making Hilbert curves in 3D



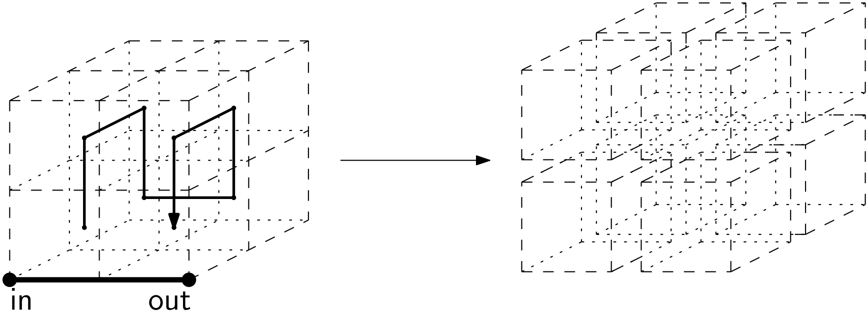


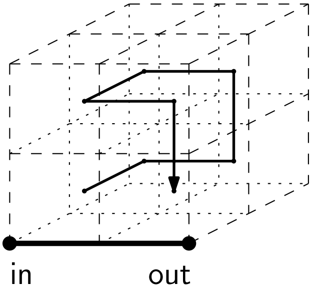
Making Hilbert curves in 3D

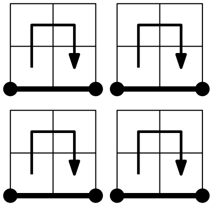




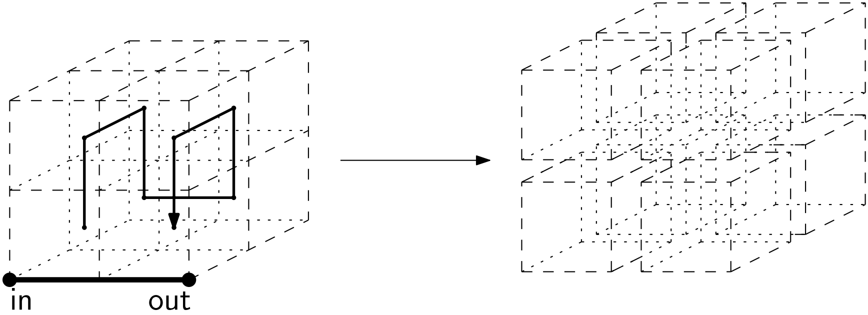
Making Hilbert curves in 3D

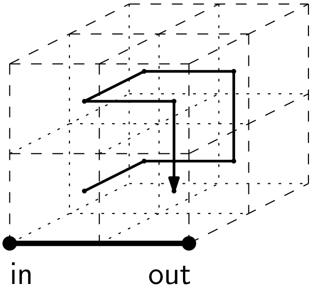


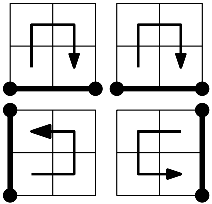




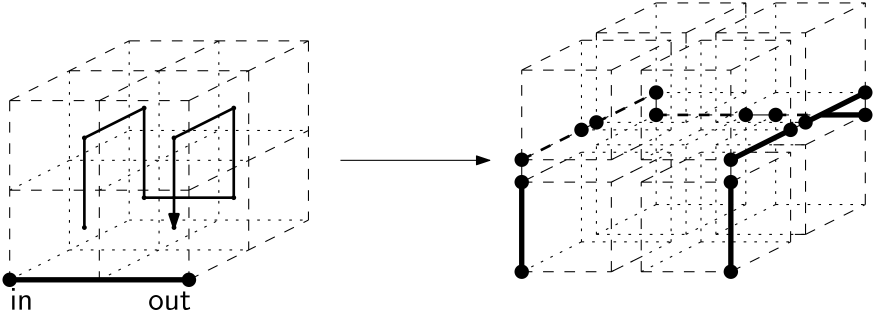
Making Hilbert curves in 3D

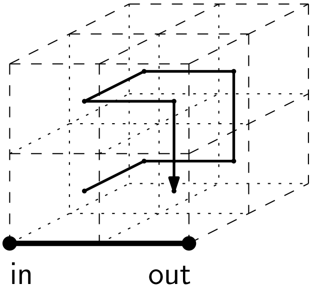


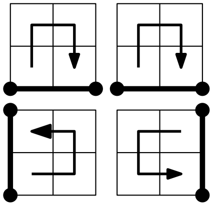




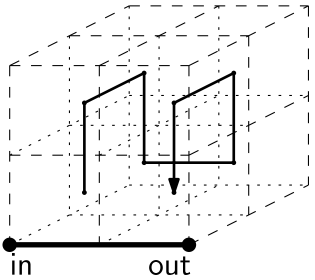
Making Hilbert curves in 3D

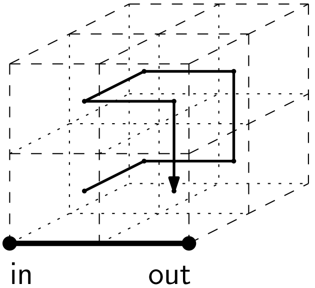


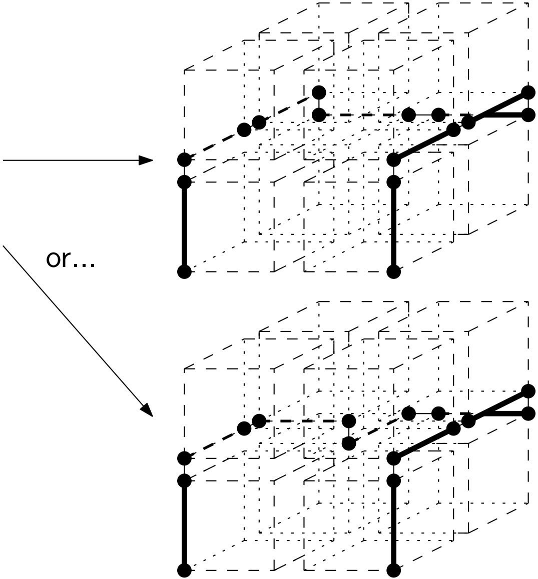




Making Hilbert curves in 3D







There are 1536 different 3D Hilbert curves...

Making Hilbert curves in 4D

There are 9782750684860313493469 different 4D Hilbert curves (more or less)

What does 4D implementation of Butz/Thomas/Moore do with point data?

- ordering by $(c_x, c_y, 0, 0)$ is not the same as by $(c_x, 0, c_y, 0)$ etc.
- ordering by $(c_x, c_y, 0, 0)$ is not as effective as by (c_x, c_y) on 2D curve

Making Hilbert curves in 4D

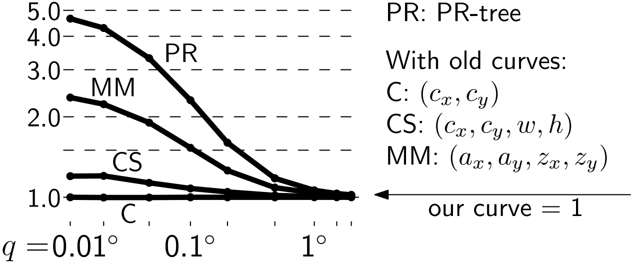
There are 9782750684860313493469 different 4D Hilbert curves (more or less)

What does 4D implementation of Butz/Thomas/Moore do with point data?

- ordering by $(c_x, c_y, 0, 0)$ is not the same as by $(c_x, 0, c_y, 0)$ etc.
- ordering by $(c_x, c_y, 0, 0)$ is not as effective as by (c_x, c_y) on 2D curve

Our curve: $(c_x, c_y, 0, 0)$ on 4D curve in same order as (c_x, c_y) on 2D curve

TIGER road data



Making Hilbert curves in 4D

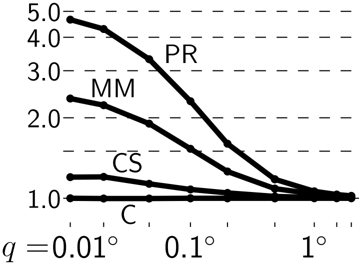
There are 9782750684860313493469 different 4D Hilbert curves (more or less)

What does 4D implementation of Butz/Thomas/Moore do with point data?

- ordering by $(c_x, c_y, 0, 0)$ is not the same as by $(c_x, 0, c_y, 0)$ etc.
- ordering by $(c_x, c_y, 0, 0)$ is not as effective as by (c_x, c_y) on 2D curve

Our curve: $(c_x, c_y, 0, 0)$ on 4D curve in same order as (c_x, c_y) on 2D curve

TIGER road data

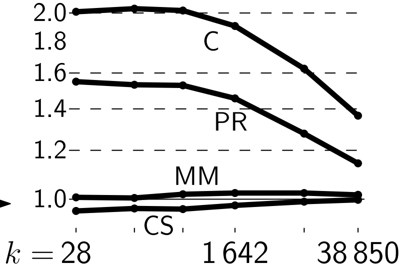


PR: PR-tree

With old curves:
C: (c_x, c_y)
CS: (c_x, c_y, w, h)
MM: (a_x, a_y, z_x, z_y)

our curve = 1

VLSI data



Ours is as good as the best for both types of data

Problems

2D: Can a recursive plane-filling curve have worst-case bbox area ratio < 2 ?

What if we rotate the bounding boxes?

What curve is best if we allow, say, octagonal bounding boxes?

Problems

2D: Can a recursive plane-filling curve have worst-case bbox area ratio < 2 ?

What if we rotate the bounding boxes?

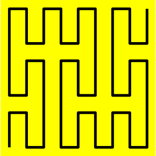
What curve is best if we allow, say, octagonal bounding boxes?

4D: Our 4D Hilbert curve is weird and a rather random choice. Why this one?

What (other) conditions should the curve fulfill?

To get good bounding boxes? And good caching?

Is there a better or a simpler curve? 4D-Peano?



Problems

2D: Can a recursive plane-filling curve have worst-case bbox area ratio < 2 ?

What if we rotate the bounding boxes?

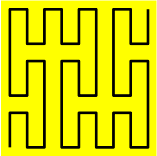
What curve is best if we allow, say, octagonal bounding boxes?

4D: Our 4D Hilbert curve is weird and a rather random choice. Why this one?

What (other) conditions should the curve fulfill?

To get good bounding boxes? And good caching?

Is there a better or a simpler curve? 4D-Peano?



References: mail cs.herman@haverkort.net and ask for our manuscripts:

Bulk-loading R-trees with four-dimensional space-filling curves

Space-filling curve properties for efficient spatial index structures (EuroCG 2008)

A comparison of space-filling curves for data structures (survey/notes)

Problems

2D: Can a recursive plane-filling curve have worst-case bbox area ratio < 2 ?

What if we rotate the bounding boxes?

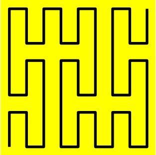
What curve is best if we allow, say, octagonal bounding boxes?

4D: Our 4D Hilbert curve is weird and a rather random choice. Why this one?

What (other) conditions should the curve fulfill?

To get good bounding boxes? And good caching?

Is there a better or a simpler curve? 4D-Peano?



References: mail cs.herman@haverkort.net and ask for our manuscripts:

Bulk-loading R-trees with four-dimensional space-filling curves

Space-filling curve properties for efficient spatial index structures (EuroCG 2008)

A comparison of space-fill... *ures* (survey/notes)

That's all folks