Reading club - Space-filling curves

Timothée Guédon

Concordia University
Big Data infrastructures for neuroinformatics laboratory

October 1, 2019

Table of Contents

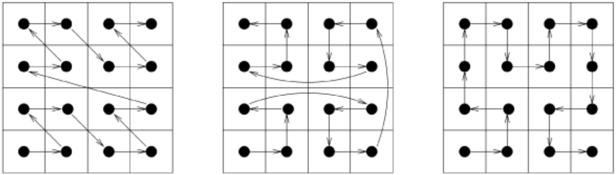
Table of Contents

Space-filling curves

What is a space-filling curve ?

- How can a curve fill space ?
- Need to setup a grid, i.e. a "resolution".
- A space filling curve is a **continous curve** that passes through every points of the grid.
- Some examples of space-filling curves include:

Figure: Illustration of space-filling curves (Moon et al. [2001])



(a) (b) (c)

Fig. 1. Illustration of space-filling curves.

Presentation of the paper

Figure: Analysis of the Clustering Properties of the Hilbert Space-Filling Curve. (2001) (Moon et al. [2001])

IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 13, NO. 1, JANUARY/FEBRUARY 2001

Analysis of the Clustering Properties of the Hilbert Space-Filling Curve

Bongki Moon, H.V. Jagadish, Christos Faloutsos, *Member*, *IEEE*, and
Joel H. Saltz, *Member*, *IEEE*

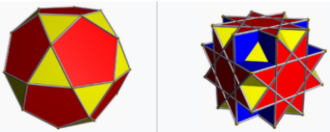
- Analysis of the Clustering Properties of the Hilbert Space-Filling Curve. (2001)
- Contributions include asymptotic and exact formulas for clustering performance.
- Comparison against other popular space-filling curves.

Polyhedrons

Queries of polyhedral shape:
Polyhedron: “In geometry, a polyhedron is a solid in three dimensions with flat polygonal faces, straight edges and sharp corners or vertices.” (Wikipedia)

- Rectilinear polyhedron: The faces are orthogonal to one axis and have right angles.
- Defines an interior and an exterior.
- In two dimensions ?

Figure: Examples of polyhedra (Wikipedia)



Icosidodecahedron
Archimedean solid

Great cubicuboctahedron
Uniform star-polyhedron

Hilbert curve

- Defined by two parameters: d and k
- d: dimension of the space
- k: order of the curve

To construct the H_k^d curve: take the H_1^d curve and replace each point of the grid by the H_{k-1}^d curve.

Figure: The first three steps of the Hilbert filling curve (Moon et al. [2001])

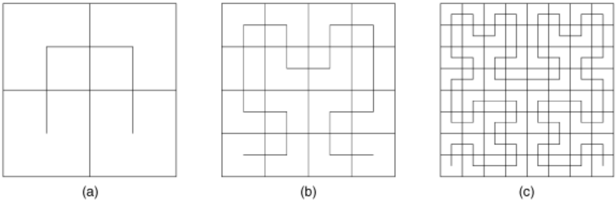


Fig. 3. The first three steps of the Hilbert space-filling curve: (a) first step, (b) second step, and (c) third step.

Table of Contents

Clustering

Definition 1: *Given a d-dimensional query, a cluster is defined to be a group of grid points inside the query that are consecutively connected by a mapping (or a curve).* (Moon et al. [2001])

Figure: Illustration of clustering (Moon et al. [2001])

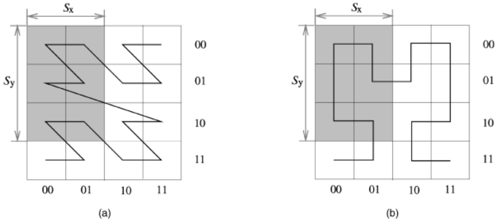


Fig. 2. Illustration of (a) two clusters for the z curve and (b) one cluster for the Hilbert curve.

"From a practical point of view, it is important to predict and minimize the number of clusters because it determines the number of nonconsecutive disk accesses, which, in turn, incur additional seek time." (Moon et al. [2001])

Predecessors

Given a point y in the grid:

- This point has $2d$ neighbours.
- One of those neighbours is a predecessor.
- A predecessor is the point preceding y in the curve order.
- Probability of a neighbour j parallel to dimension i to be the predecessor (Moon et al. [2001]):

$$p_{ij} = p_i * \frac{1}{2} = \frac{d}{2}$$

- p_i is the probability of the neighbour to be parallel to the i_{th} dimension.

Surfaces

- "Border cell": Cell of the interior, close to a face.
- "Potential predecessor": Cell of the exterior, close to a face.
- "Surface": Aggregate number of potential predecessors.
- Number of entry points: $N \approx S^{\frac{1}{2d}}$

Figure: Illustration of faces and surfaces (Moon et al. [2001])

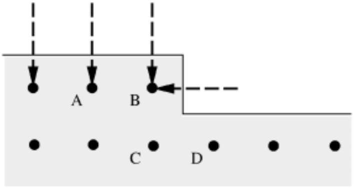


Fig. 6. Illustration of grid points facing surfaces.

Theorem 1 - the asymptotic formula

Missed one parameter: Order of the curve (resolution).

Theorem 1 (Moon et al. [2001]): *In a sufficiently large d-dimensional grid space mapped by H_k^d, let S be the total surface area of a given rectilinear polyhedral query q. Then,*

$$\lim_{k \to \infty} N_d = \frac{S}{2d}$$

Corollary (Moon et al. [2001]): *Given a hypercube of side length s:*

$$\lim_{k \to \infty} N_d = s^{d-1}$$

Table of Contents

Experiments

Experiments to test exact and asymptotic formulas
Range queries of various sizes and shapes
Shapes tested:

- 2D: square, circle, concave polygon
- 3D: cube, shere, concave polyhedra
- 4+D: hypercube, hypersphere: simpler formulas for the simulations

Figure: Shapes tested during the experiments (Moon et al. [2001])

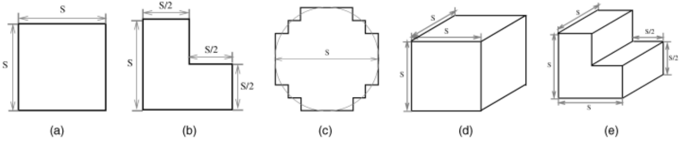


Fig. 10. Illustration of sample query shapes: (a) square, (b) polygon, (c) circle, (d) cube, and (e) polyhedron.

Protocol

For each query:

- Run the query on every possible points on the grid
- Average the numbers of clusters found

Limits:

- Not possible to do it when the dimensionality increases (Exponential):

$$N^d - volume\_of\_query$$

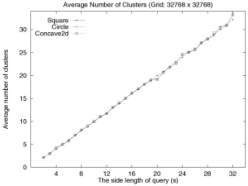
- Test all possibilities in 2D and 3D only + small queries
- Random sampling for higher dimensional space.

Results in 2D

Figure: Results of the experiments in 2D (Moon et al. [2001])

query	empirical	asymptotic	exact
$2^1 \times 2^1$	1.998534	2	2091524/1046529
$2^2 \times 2^2$	3.996328	4	4165936/1042441
$2^3 \times 2^3$	7.992257	8	8266304/1034289
$2^4 \times 2^4$	15.984206	16	16273216/1018081
$2^5 \times 2^5$	31.967807	32	31521824/986049

(a)



(b)

Fig. 11. Average number of clusters for two-dimensional queries: (a) exhaustive simulation (grid: $1,024 \times 1,024$) and (b) statistic simulation (grid: $32K \times 32K$).

- (a) 1K x 1K grid space, square query: Results consistent with formulas
- (b) 32K x 32K grid space, 200 random queries
- (b) Average number of clusters in the same no matter the shape
- (b) The number of clusters is proportional to the query size (consistent with formula)

Results in 3D

Figure: Results of the experiments in 3D and highest dimensions (Moon et al. [2001])

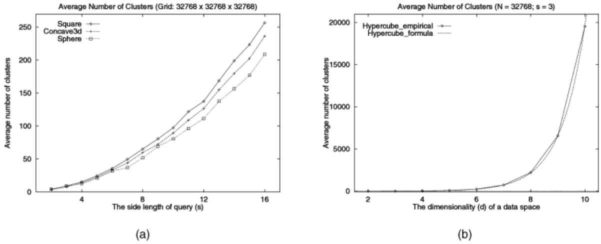


Fig. 12. Average number of clusters for higher-dimensional queries: (a) three-dimensional queries and (b) *d*-dimensional hypercubic queries.

- (a) grid space: 32K x 32K x 32K
- (a) Approximate the quadratic formulas using least squares method
- (a) Consistent with corollary $\lim_{k\to\infty} N_d = s^{d-1}$ for the hypercube
- (a) Different equations for the other shapes:

Results in 3D

Figure: Results of the experiments in 3D and highest dimensions (Moon et al. [2001])

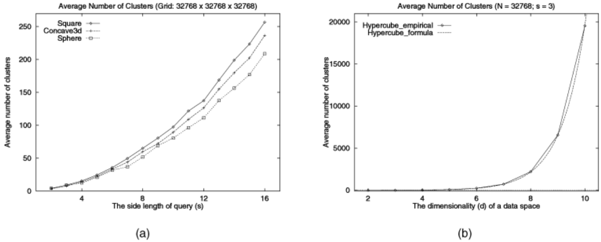
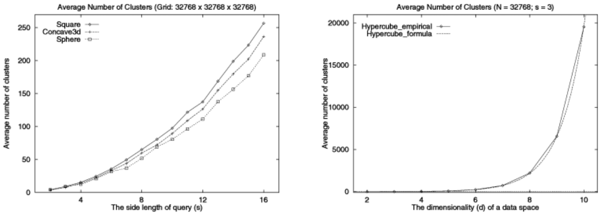


Fig. 12. Average number of clusters for higher-dimensional queries: (a) three-dimensional queries and (b) *d*-dimensional hypercubic queries.

[...] Unlike in the two-dimensional case, the surface area of a concave polyhedron or a sphere is smaller than that of its minimum bounding cube. (Moon et al. [2001])

Results in higher dimensions

Figure: Results of the experiments in 3D and highest dimensions (Moon et al. [2001])



(a) (b)

Fig. 12. Average number of clusters for higher-dimensional queries: (a) three-dimensional queries and (b) *d*-dimensional hypercubic queries.

- (b) grid space: 32K x 32K x ... x 32K
- (b) queries of size 3 x 3 x ... x 3
- (b) Formula coincide with the asymptotic formula even in higher dimensional spaces.

Worst case comparisons

Figure: Worst case comparisons between the Z, the Hilbert and the Grey-coded curves. (Moon et al. [2001])

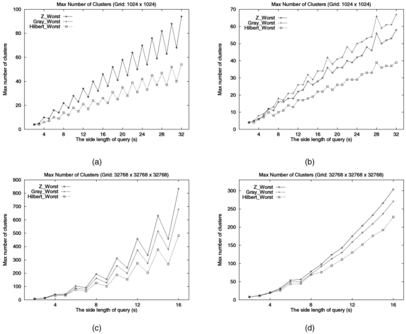


Fig. 13. Worst-case number of clusters for three different space-filling curves: (a) two-dimensional square queries, (b) two-dimensional circular queries, (c) three-dimensional cubic queries, and (d) three-dimensional spherical queries.

Average case comparisons

Figure: Average case comparisons between the Z, the Hilbert and the Grey-coded curves. (Moon et al. [2001])

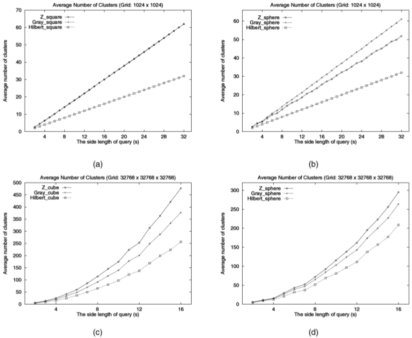


Fig. 14. Average number of clusters for three different space-filling curves: (a) two-dimensional square queries, (b) two-dimensional circular queries, (c) three-dimensional cubic queries, and (d) three-dimensional spherical queries.

Summary

"Hilbert curve achieves better clustering than the z curve in a two-dimensional space" (Moon et al. [2001])

In 2D: "The average number of clusters for the Hilbert curve is **one-fourth** of the perimeter of a query rectangle, while that of the z curve is one-third of the perimeter plus two-thirds of the side length of the rectangle in the unfavored direction" (Moon et al. [2001])

"we have shown that the Hilbert curve outperforms both the z and Gray-coded curves in two-dimensional and 3-dimensional spaces. We conjecture that this trend will hold even in higher-dimensional spaces." (Moon et al. [2001])

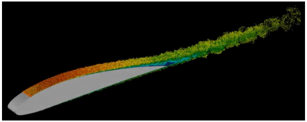
Table of Contents

Data exploration of turbulence simulations using a database cluster. (2007)

A "cluster of databases" (Perlman et al. [2007]) to store the history of direct numerical simulations (DNS) of turbulent flows. 1024 time samples of 1024^3 spatial points
"In fluid dynamics, turbulence or turbulent flow is any pattern of fluid motion characterized by chaotic changes in pressure and flow velocity." (Wikipedia)
"We provide real examples of how scientists use the systemto perform high-resolution turbulence research from standard desktop computing environments."

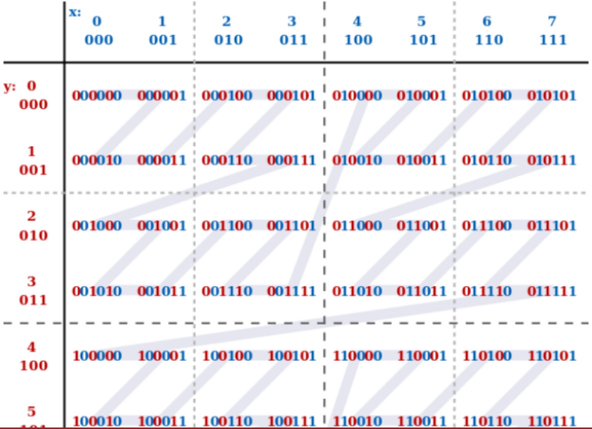
(Perlman et al. [2007]) Use of Morton code in a B+-tree (standard for databases indexing)

Figure: Screenshot from "Turbulent flow around a wing profile, a direct numerical simulation" (KTH Mechanics et al.)



Derivation of Morton code from the Z curve

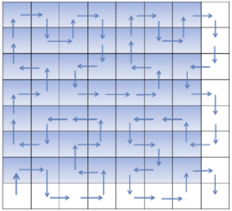
Figure: Illustration of how to find the Morton code. (Wikipedia)

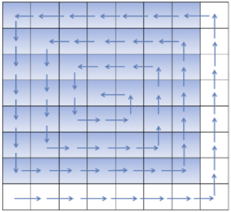


Z-curve VS Hilbert curve

- Hilbert curve has better clustering properties than the Z-curve
- The Z-curve is simpler to encode/decode than the Hilbert curve.
- More efficient implementations of Hilbert's curve encoding/decoding now available.
- A new player: onion curve? (Xu et al. [2018])

Figure: Comparison of Hilbert and Onion curves (Xu et al. [2018])





References I

Mohammad Hosseini KTH Mechanics, Ricardo Vinuesa KTH Mechanics, Ardeshir KTH Hanifi KTH Mechanics, Dan Henningson KTH Mechanics, and Philipp Schlatter KTH Mechanics. V0078: Turbulent flow around a wing profile, a direct numerical simulation. URL `https://doi.org/10.1103/APS.DFD.2015.GFM.V0078`.

B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. IEEE Transactions on Knowledge and Data Engineering, 13 (1):124–141, Jan 2001. doi: 10.1109/69.908985.

Eric Perlman, Randal Burns, Yi Li, and Charles Meneveau. Data exploration of turbulence simulations using a database cluster. page 23, 01 2007. doi: 10.1145/1362622.1362654.

Pan Xu, Cuong Nguyen, and Srikanta Tirthapura. Onion curve: A space filling curve with near-optimal clustering. CoRR, abs/1801.07399, 2018. URL `http://arxiv.org/abs/1801.07399`.