Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Locality and bounding-box quality of two-dimensional space-filling curves

Herman Haverkort*, Freek van Walderveen

Dept. of Computer Science, Eindhoven University of Technology, The Netherlands

ARTICLE INFO

Article history:
Received 12 July 2008
Received in revised form 24 April 2009
Accepted 2 June 2009
Available online 6 June 2009
Communicated by T. Asano

Keywords:
Space-filling curves
Spatial data structures
R-trees

ABSTRACT

Space-filling curves can be used to organise points in the plane into bounding-box hierarchies (such as R-trees). We develop measures of the *bounding-box quality* of space-filling curves that express how effective different space-filling curves are for this purpose. We give general lower bounds on the bounding-box quality measures and on locality according to Gotsman and Lindenbaum for a large class of space-filling curves. We describe a generic algorithm to approximate these and similar quality measures for any given curve. Using our algorithm we find good approximations of the locality and the bounding-box quality of several known and new space-filling curves. Surprisingly, some curves with relatively bad locality by Gotsman and Lindenbaum's measure, have good bounding-box quality, while the curve with the best-known locality has relatively bad bounding-box quality.

1. Introduction

A space-filling curve is a continuous, surjective mapping from $\mathbb{R}$ to $\mathbb{R}^d$. It was not always clear that such a mapping would exist for $d > 1$, but in the late 19th century Peano showed that it is possible for $d = 2$ and $d = 3$ [26]. Since then, quite a number of space-filling curves have appeared in the literature. Sagan wrote an extensive treatise on space-filling curves [27], which discusses most curves included in our study. During the early days space-filling curves were primarily seen as a mathematical curiosity. Today however, space-filling curves are applied in areas as diverse as load balancing for grid computing, colour space dimension reduction, small antenna design, I/O-efficient computations on massive matrices, and the creation of spatial data indexes. In this paper, we focus on the application of space-filling curves to the creation of query-efficient spatial data indexes such as R-trees.

Bounding-box hierarchies We consider the following type of spatial data indexes for points in the plane. The data points are organised in blocks of at most B points, for some parameter B, such that each point is stored in one block. With each block we associate a bounding box, which is the smallest axis-aligned rectangle that contains all points stored in the block. The block bounding boxes are then organised in an index structure. *Intersection queries* are answered as follows: to find all points intersecting a query window Q, we query the index structure for all bounding boxes that intersect Q; then we retrieve the corresponding blocks, and check the points stored in those blocks one by one. To find the *nearest neighbour* to a query point q, one can use the index to search blocks in order of increasing distance from q. Thus one retrieves exactly the blocks whose bounding boxes intersect the largest empty circle around q.

* Corresponding author.
E-mail addresses: cs.herman@haverkort.net (H. Haverkort), freek@vanwal.nl (F. van Walderveen).

0925-7721/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comgeo.2009.06.002

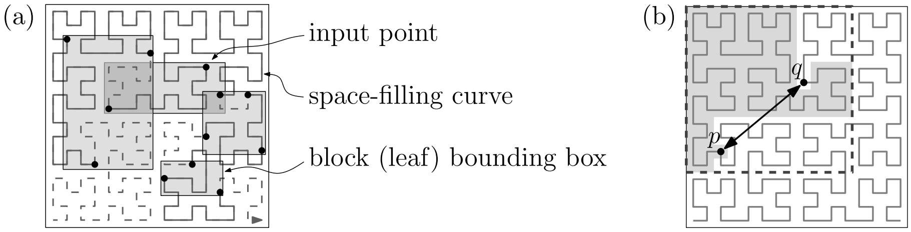


Fig. 1. (a) Leaves of an R-tree with $B = 3$. (b) Measuring locality for a particular curve section. L_2-locality ratio between p and q = squared Euclidean distance between p and q, divided by the area covered by the curve section between p and q: $(6^2 + 5^2)/87 \approx 0.70$. Bounding-box area ratio between p and q = area of the bounding-box of the curve section S between p and q, divided by the area covered by S: $12 \cdot 12/87 \approx 1.66$. (WBA is the maximum over all pairs p and q.)

An R-tree [19] is an example of the type of structure described above: the blocks constitute the leaves of the tree, and the higher levels of the tree act as an index structure for the block bounding boxes. In practice the query response time is mainly determined by the number of blocks that need to be retrieved: this is because the bounding box index structure can often be cached in main memory, while the blocks (leaves) with data points have to be stored on slow external memory (for example a hard disk needing 10 ms for each seek).

An R-tree is not uniquely defined by a set of data points. Any distribution of the input points over the leaves (blocks) may be used as the basis of an R-tree, as long as each point is put in exactly one block, and each block contains at most B points. One way of making the distribution is by ordering the input points along a space-filling curve [15] and then putting each next group of B points together in a block (see Fig. 1(a)).

Since the number of blocks retrieved to answer a query is simply the number of bounding boxes intersected, it is important that the ordering induced by the space-filling curve makes us fill each block with points that lie close to each other and thus have a small bounding box. In fact we can make some more precise observations for intersection queries with query windows that are points or lines. For point-intersection queries we observe that if the data and query points lie within a square of area 1, the average number of blocks retrieved for uniformly distributed point queries is simply the total area of the bounding boxes. For line-intersection queries with uniformly distributed orientation (between 0 and π) and signed distance from the centre of the square (in an interval containing at least $[-\frac{1}{2}\sqrt{2}, \frac{1}{2}\sqrt{2}]$), the chance of any particular block being retrieved is proportional to $\int_0^\pi w(\phi)\,d\phi$, where $w(\phi)$ is the width of the bounding box in the orthogonal projection on a line with orientation ϕ. By the Crofton formula $\int_0^\pi w(\phi)\,d\phi$ is simply the perimeter of the bounding box of the block, so for uniformly distributed line queries, the average number of blocks retrieved is proportional to the total perimeter of the bounding boxes. Therefore our goal is to have bounding boxes with small (total) area and small (total) perimeter.

Our results We investigate which space-filling curves best achieve the above-mentioned goal: sorting points into bounding boxes with small (total) area and small (total) perimeter. To this end we propose new quality measures of space-filling curves that express how effective different space-filling curves are in this context. We also provide an algorithm to compute approximations of these and similar quality measures for any given curve. We used this algorithm to compute approximations of known measures of so-called curve-to-plane locality and of our new *bounding-box quality* measures for several well-known and new space-filling curves.

The known locality measures considered are the maximum, over all contiguous sections S of a space-filling curve, of the squared L_∞-, L_2- or L_1-distance between the endpoints of S divided by the area covered by S (studied by Gotsman and Lindenbaum [10] and many other authors [1,5,6,17,24,25]), see Fig. 1(b) for an example.

Our first new measure is the maximum, over all contiguous sections S of a space-filling curve, of the area of the bounding box of S divided by the area covered by S. We call this measure the *worst-case bounding-box area ratio* (WBA, Fig. 1(b)). Our second new measure considers (1/16)th of the squared perimeter instead of the area, and we call it *worst-case bounding-box squared perimeter ratio* (WBP).

We prove that WBA and WBP are at least 2 for a large class of space-filling curves. We also show that this class of curves has L_2-locality at least 4, thus complementing earlier results by Niedermeier et al. [24] who proved this for another class of space-filling curves (more restricted in one way, more general in another way).

We found that Peano's original curve achieves a WBA-value of less than 2.0001; the exact value is probably exactly 2, which is optimal for this class of curves. Other well-known curves have WBA-values ranging from 2.400 to 3.000. However, on the WBP measure Peano's curve is not that good, with WBP = 2.722. Considering both WBA and WBP, the best curve we found in the literature is the $\beta\Omega$-curve [28], with WBA = 2.222 and WBP = 2.250. However, in this paper we present a new variation on Peano's curve with even better scores: a WBA-value of 2.000 and a WBP-value of 2.155. This variation also performs very well on L_∞-, L_2- and L_1-locality.

Both WBA and WBP consider the worst case over all possible subsections of the curve. However, in the context of our application, it may be more relevant to study the total bounding box area and perimeter of a set of disjoint subsections of the curve that together cover the complete curve. We can argue that in the limit, we may have the worst case for all subsections in such a cover, but this seems to be unlikely to happen in practice. Therefore we study the total bounding box area and perimeter of random subdivisions of the curve into subsections. Here we find that many curves perform roughly equally well, but those with particularly bad WBA- or WBP-values, such as the Sierpiński–Knopp curve [27] or H-order [24], the AR^2W^2-curve [4], or Peano's original (unbalanced) curve (regarding bounding box perimeters), are clearly suboptimal in this sense.

We also estimate the total diameter of the subsections in random subdivisions of the curve and present results for octagonal bounding boxes rather than rectangular bounding boxes.

Below we first explain how different space-filling curves can be described and how they can be used to order points. We also describe some new curves. Next we define the locality and bounding-box quality measures, and prove lower bounds. After that we present our approximation algorithm, and present the results of our computations.

2. Describing and using space-filling curves

There are many ways to define space-filling curves, for example algebraic, like in Peano's paper [26], or by describing an approximation of the curve by a polyline, with a rule on how to refine each segment of the approximation recursively [9]; many authors do this by specifying the regions filled by sections of the curve together with the location of the endpoints of such sections on the region boundaries (for example [4,27,28]). Since we are concerned with the use of space-filling curves as a way to order points in the plane, we choose a method to describe space-filling curves that is based on defining how to order the space inside a (usually square) unit region. We will see later how such a description is also a description of a curve.

2.1. How to define and use a scanning order

We define an order (*scanning order*) $\prec$ of points in the plane as follows. We give a set of rules, each of which specifies (i) how to subdivide a region in the plane into subregions; (ii) what is the order of those subregions; and (iii) for each subregion, which rule is to be applied to establish the order within that subregion. We also specify a unit region of area 1 for each order (usually the unit square), and we indicate what rule is used to subdivide and order it. Technically it would be possible to extend the orders to the full plane, but for simplicity we rather assume that all data that should be ordered is first scaled to lie within the unit region.

The definitions of the scanning orders discussed in this paper are shown in Fig. 2. Each rule is identified by a letter, and pictured by showing a region, its subdivision into subregions, the scanning order of the subregions (by numbers $\{0, 1, 2, \ldots\}$), and the rules applied to the subregions (by letters). Variations of rules that consist of simply rotating or mirroring the order of and within subregions, are indicated by rotating or mirroring the letter identifying that rule. Variations that consist of reversing the order of and within the subregion are indicated by an overscore (Fig. 2(k, l, m))—making such reversals explicit is the main difference between our notation and the notation of, for example, Asano et al. [4] or Wierum [28].

We can now see how we can implement a comparison operator that allows us to sort points according to a given scanning order. To decide whether p precedes q in the order, we determine in which subregions of the unit region p and q lie. If they are in different regions, p precedes q if and only if p lies in the lowest-numbered region of the two. If p and q lie in the same region, we compare them according to the rule for that subregion recursively. Ambiguity on the region boundaries can be resolved as follows: horizontal and diagonal region boundaries are always assumed to be included in the region above them; vertical boundaries are assumed to be included in the region to the right.

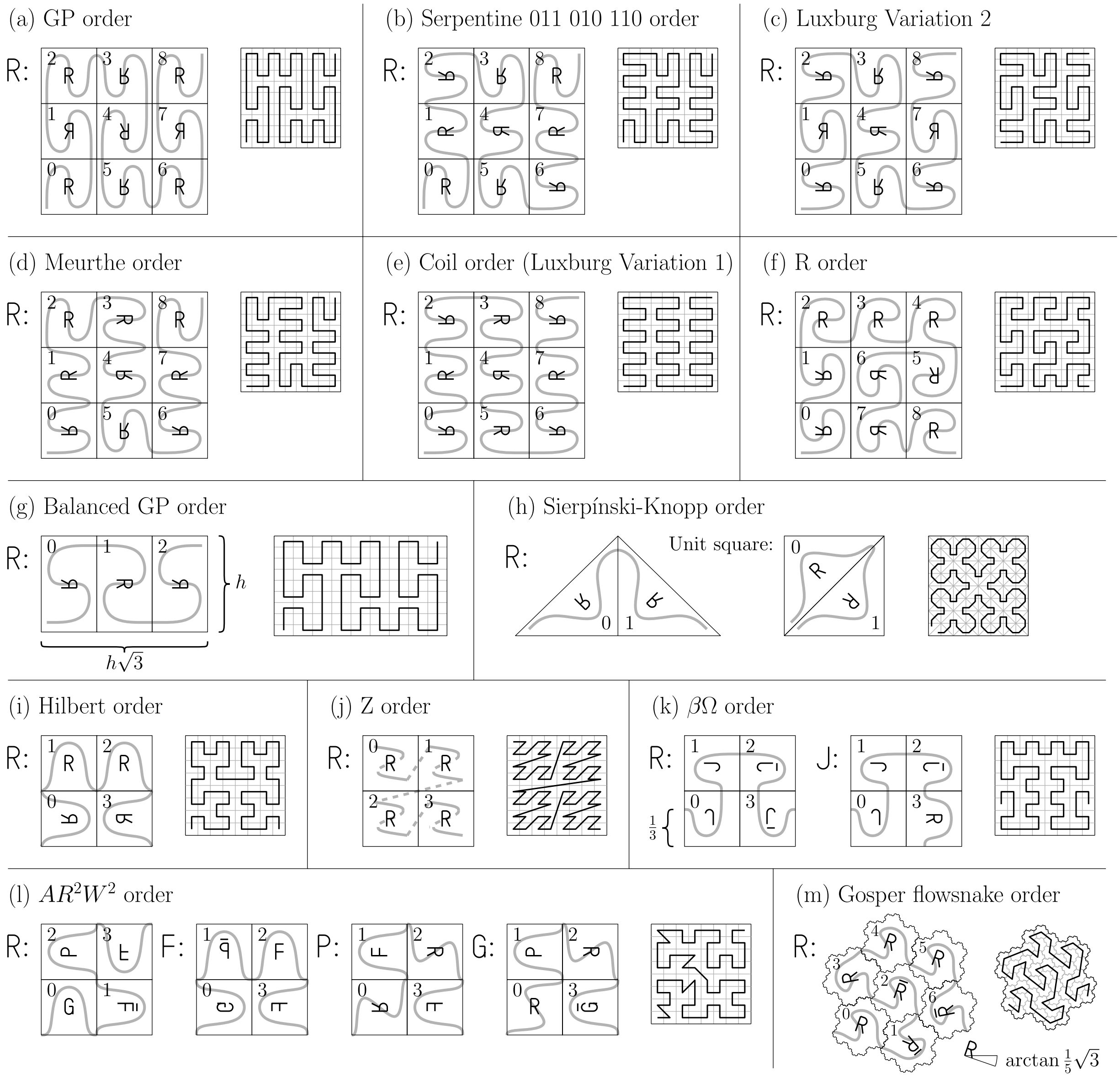
Each drawing in Fig. 2 includes a curve that roughly indicates the scanning order within the subdivisions. To obtain an arbitrarily fine approximation of a space-filling curve corresponding to a given scanning order, we may compute the subdivision of the unit region into subregions recursively to the desired depth of recursion, and connect the centre points of the resulting subregions by a polygonal curve in the order specified by the rules. Fig. 2 includes a small example for each scanning order.

In the rest of this paper, whenever we write "space-filling curve", what we really mean is the scanning order that defines it.

2.2. The curves

The traditional scanning orders considered in this paper are the following.

- *GP-order,* producing the space-filling curve described in detail by Giuseppe Peano [22,26] (Fig. 2(a)). We call this order *GP-order* instead of *Peano order* to avoid confusion with other curves that have also been referred to as the Peano curve by other authors.
- *Sierpiński–Knopp-order,* producing the Sierpiński–Knopp curve [27] (Fig. 2(h)). It orders triangular regions, and can be used to order points as described in Section 2.1. Niedermeier et al. [24] describe how to use it to order squares in a $2^k \times 2^k$ grid for any $k \in \mathbb{N}$, their variation is called H-order. For all purposes in our paper, Sierpiński–Knopp order and H-order are equivalent.
- *Hilbert order,* producing Hilbert's curve [12] (Fig. 2(i)). One could say that Peano had already suggested that such a curve would exist [26].



(a) GP order

(b) Serpentine 011 010 110 order

(c) Luxburg Variation 2

(d) Meurthe order

(e) Coil order (Luxburg Variation 1)

(f) R order

(g) Balanced GP order

(h) Sierpínski-Knopp order

(i) Hilbert order

(j) Z order

(k) $\beta\Omega$ order

(l) AR^2W^2 order

(m) Gosper flowsnake order

Fig. 2. Space-filling curve definitions and example approximating polylines.

- *Z-order*, which follows a space-filling curve by Lebesgue [16] (Fig. 2(j)); Morton is credited with introducing Z-order to computer science [23].
- *Gosper flowsnake order*, producing the Gosper curve, also known as the flowsnake [2,9] (Fig. 2(m)). It involves scaling with a factor $1/\sqrt{7}$ and rotations with angles of $\{0, \frac{2}{3}\pi, \frac{4}{3}\pi\} - \arctan \frac{1}{5}\sqrt{3}$.

In addition to the above we include a number of variations on these orders. Wunderlich [30] defined a class of orders that satisfy certain restrictions, including:

- *simplicity*: the order is defined by only one rule, and transformed versions of it;
- *order-preservation*: transformations are rotations and/or reflections (no reversals);
- *edge-connectivity*: considering the set of regions obtained by applying the rule to any depth of recursion, we find that any two consecutive regions in the order share an edge;
- *uniformity*: all subregions have the same size.

Wunderlich categorises all different simple, order-preserving, edge-connected, uniform orders based on subdividing the unit square in a grid of 2 × 2 or 3 × 3 squares (modulo rotations, reflections and reversals). There is only one such order on a

2×2 grid (Hilbert order), and there are only 273 on a 3×3 grid: one which we call *R-order*[1] (Fig. 2(f)), and 272 different orders that put the nine grid cells in the order of the 'serpentine' pattern of the GP-order (these differ in the rotations and reflections of the cells). We investigated all of these orders to some extent, and based on the results we studied some of them in more depth, particularly those depicted in Figs. 2(a)–(e). By Wunderlich's numbering scheme these orders are *Serpentine* orders 000 000 000, 011 010 110, 101 010 101, 110 110 110, and 111 111 111, respectively. The first of these is simply GP-order. Luxburg [17] examined the third order calling it *Variation 2*; we call the next order *Meurthe order* after the river in whose watershed we first presented it, and we call the last order *coil order* (a variation also commonly found on the Internet, and studied by Luxburg as *Variation 1*). We explain in Section 6 why we chose exactly these orders.

When some of the above-mentioned restrictions are dropped, more orders are possible. An example from the literature is Wierum's *$\beta\Omega$-order*[2] [28], which is not simple, and unlike the other orders, it does not start and end in the corners of the unit square (Fig. 2(k)). Another example is the AR^2W^2-order which is not simple and not edge-connected, but still *vertex-connected* (Fig. 2(l)). It was designed to have the special property that any axis-parallel square query window can be covered by *three* contiguous sections of the curve that together cover an area of at most a constant times that of the query window [4]—from most other curves in our study one would need at least *four* sections to get such a constant-ratio cover.

Most orders discussed so far cannot be scaled in only one dimension, because their definitions involve rotations. For GP-order this is not the case. In fact, as we will see later, a scaling in the horizontal dimension by a factor $\sqrt{3}$ results in an order with much better locality properties. We call the scaled order *balanced GP-order* (Fig. 2(g)).

3. Quality measures for space-filling curves

Several locality measures, or more generally, quality measures of space-filling curves have been considered in the literature. These include:

- bounds on the (average) distance between two points along the curve as a function of their distance in the plane [8,20,29] (non-trivial worst-case bounds are not possible [10]);
- bounds on the (worst-case or average) distance between two points in the plane as a function of their distance along the curve [7,10,24];
- bounds on the (worst-case or average) number of contiguous sections of the curve that is needed to cover an axis-parallel query window, without covering too much space outside the query window [4,7,14,21];
- bounds on the (worst-case or average) perimeter or diameter of sections of the curve as a function of their area [13,28].

Not all methods of analysis considered in the literature can easily be applied to compare all curves in our study. Some calculations or experiments in the literature are based on how the scanning order sorts the points of a regular square grid whose size is a fixed integral power of the number of subregions in the rule(s) that define(s) the order. Such measures may vary with the grid size, which prevents a fair comparison between, for example, GP-order (for which we would have to use a grid size that is a power of nine) and Hilbert order (for which we would have to use a grid size that is a power of four). To enable a comparison between a broad range of curves, we need measures that can be computed efficiently for large grids and converge when the grid size goes to infinity.

3.1. Notation

Before we can discuss and analyse quality measures for space-filling curves in detail, we need to introduce some notation. For ease of writing, we assume for now that if a scanning order is defined by more than one rule, then each rule contains the same number of subregions.

A rule of a scanning order defines how to subdivide a unit region C of size (area) 1 into n subregions, numbered $0, \ldots, n-1$. The scanning order inside subregion i is given by applying a transformation $\tau(i)$ to the unit region C and the way it is ordered by the ordering rule. For any base-n number a we use a' to denote its first digit, and a'' to denote the remaining digits. We use $C(a)$ as a shorthand for $\tau(a')(C(a''))$, where $C(\emptyset) = C$. For example, $C(538)$ is subregion 8 of subregion 3 of subregion 5, and it is found by applying transformation $\tau(5)$ to $C(38)$. Similarly, we use $\tau(a)$ as a shorthand for $\tau(a') \circ \tau(a'')$, where $\tau(\emptyset)$ is the identity transformation.

By $|A|$ we denote the size of a region A. We have $0 < |C(i)| < 1$ for all $0 \leqslant i < n$ (there are no empty subregions in the rules), and $\sum_{0 \leqslant i < n} |C(i)| = |C| = 1$.

Let N_k denote the set of k-digit base-n numbers. We write $a \prec b$ if, in base-n notation, a and b have the same number of digits and $a < b$. By $C(\preceq b)$ we denote the union of subregion b and its predecessors, that is, $\bigcup_{i=0}^{b'-1} C(i) \cup \tau(b')(C(\preceq b''))$, where $C(\preceq \emptyset) = C$. Define $C(\prec b) := C(\preceq b) \setminus C(b)$, $C(\succeq a) := C \setminus C(\prec a)$, $C(\succ a) := C \setminus C(\preceq a)$, and $C(a, b) := C(\prec b) \setminus C(\prec a)$.

Above we talked about the distance between two points along the curve, which may be a somewhat counter-intuitive concept for a curve that can be refined and therefore lengthened indefinitely. However, the distance between two points p

[1] Wunderlich calls it *Meander*, but because other authors have used that name to describe other orders, we rather use *R-order*.

[2] For simplicity we take an Ω-shaped section of the curve. Wierum adds a special rule for the unit square so that he gets a closed (cyclic) curve, but for the purposes of our discussion this would be an unnecessary complication.

and q along the curve is well-defined as the area filled by the section of the curve that runs from p to q, or more precisely, as:

$$\left|C(p,q)\right| := \lim_{k\to\infty} \min_{a,b\in N_k \text{ s.t. } p\in C(a), q\in C(b)} \left|C(a,b)\right|.$$

3.2. Pairwise locality measures

From the quality measures mentioned above, the most relevant and applicable to the construction of bounding-box hierarchies seem to be those that bound the (worst-case or average) distance between two points in the plane as a function of their distance along the curve. This is, intuitively, because points that lie close to each other along the curve are likely to be put together in a block. Then, if the distance between those points in the plane is small too, the block may have a small bounding box.

Gotsman and Lindenbaum [10] defined the following class of locality measures:

$$\lim_{m\to\infty} \max_{1\leqslant i<j\leqslant m} \frac{d_r(S(i), S(j))^2}{(j-i)/m},$$

where i and j are integers, $S(i)$ is the ith square along the curve in a subdivision of the unit square into a regular grid of m squares, and $d_r(S,T)$ is the L_r-distance between the centre point (S_x, S_y) of S and the centre point (T_x, T_y) of T. Thus $d_r(S,T) = (|S_x - T_x|^r + |S_y - T_y|^r)^{1/r}$ for $r \in \mathbb{N}$, and $d_\infty(S,T) = \max(|S_x - T_x|, |S_y - T_y|)$. We note that the measure is easily generalised to scanning orders that are not based on regular grids, by defining it as:

$$\mathrm{WL}_r := \lim_{k\to\infty} \sup_{a,b\in N_k} \frac{d_r(C(a), C(b))^2}{|C(a,b)|}.$$

We call this measure WL_r for *Worst-case Locality*, as it indicates for points that lie close to each other on the curve how far from each other they might get in the plane. Since we have $d_1(p,q) \geqslant d_2(p,q) \geqslant d_\infty(p,q)$ for any pair of points p and q, we have $\mathrm{WL}_1 \geqslant \mathrm{WL}_2 \geqslant \mathrm{WL}_\infty$ for any space-filling curve.

3.3. Pairwise bounding box measures

Intuitively, one may expect a relation between locality and bounding box size, as explained above. However, we may also try to measure bounding box size directly. We define two measures to do so. The first is the *worst-case bounding box area ratio* (WBA):

$$\mathrm{WBA} := \lim_{k\to\infty} \sup_{a,b\in N_k} \frac{|\mathrm{bbox}(C(a,b))|}{|C(a,b)|},$$

where bbox(S) is the smallest axis-aligned rectangle that contains S. The second measure is the *worst-case bounding box square perimeter ratio* (WBP):

$$\mathrm{WBP} := \frac{1}{16} \cdot \lim_{k\to\infty} \sup_{a,b\in N_k} \frac{\mathrm{peri}(\mathrm{bbox}(C(a,b)))^2}{|C(a,b)|},$$

where peri(S) is the perimeter of S in the L_2 metric. Taking the square of the perimeter is necessary, because otherwise the measure would be unbounded as k (the resolution of the "grid") goes to infinity. Because the rectangle of smallest perimeter that has any given area is a square, the "ideal" bounding box has squared perimeter 16. The division by 16 gives an easy relation between WBP and WBA: we have $\mathrm{WBP} \geqslant \frac{1}{16}(4\sqrt{\mathrm{WBA}})^2 = \mathrm{WBA}$. Furthermore, since the perimeter of the bounding box of two points p and q is simply twice their L_1-distance, we have $\mathrm{WBP} \geqslant \frac{1}{16}(2\sqrt{\mathrm{WL}_1})^2 = \frac{1}{4}\mathrm{WL}_1$.

We can define measures similar to WBA and WBP when the bounding boxes used are not axis-parallel rectangles, but convex octagons whose sides have normals at angles of 0, $\frac{1}{4}\pi$, $\frac{1}{2}\pi$, $\frac{3}{4}\pi$, π, $\frac{5}{4}\pi$, $\frac{3}{2}\pi$, and $\frac{7}{4}\pi$ with the positive x-axis. We call these measures *worst-case bounding octagon area ratio* (WOA) and *worst-case bounding octagon square perimeter ratio* (WOP). In the definition of WOP we still use the factor 16 to allow a direct comparison with WBP and WBA. Because the octagon of smallest perimeter that has area 1 has squared perimeter $32/(1+\sqrt{2}) \approx 0.828 \cdot 16$, we have $\mathrm{WOP} \geqslant 0.828 \cdot \mathrm{WOA}$.

3.4. Total bounding box measures

Worst-case For our application we argued that the average query response time is related to the total area and perimeter of the bounding boxes formed by grouping data points according to a given scanning order. When the points are sufficiently densely distributed in the unit region, the gap in the scanning order between the last point of a group and the first point in the next group will typically be small. Thus the grouping practically corresponds to subdividing the complete unit region

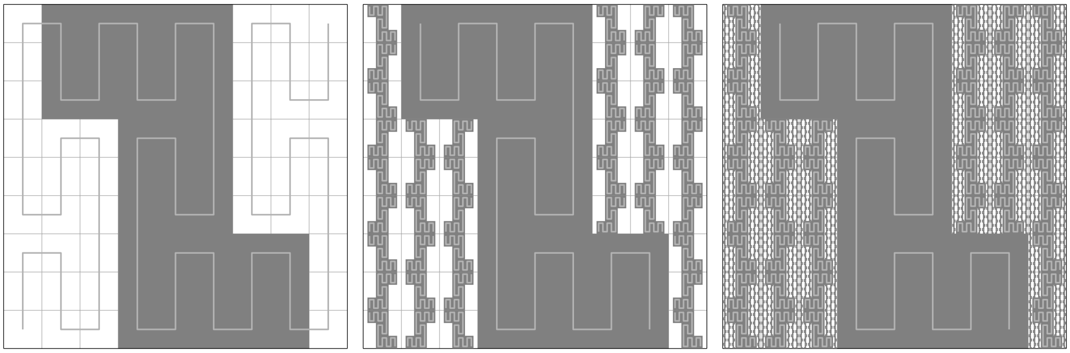


Fig. 3. Left: an approximation of a section S of the GP curve with $|\mathrm{bbox}(S)|/|S| = \mathrm{WBA}$. Middle and right: we can cover an arbitrarily large part of the unit region with such worst-case sections.

into curve sections, of which we store the bounding boxes. To assess the quality of the order, we can define the *worst-case total bounding box area* (TBA) as follows:

$$\mathrm{TBA} := \lim_{k\to\infty} \sup_{a_1 \prec a_2 \prec \cdots \prec a_{m-1} \in N_k} \left(\sum_{i=1}^{m} \bigl|\mathrm{bbox}\bigl(C(a_{i-1}, a_i)\bigr)\bigr| \right),$$

where a_0 is defined as 0 and a_m is defined as $\emptyset$. Since the bounding box area of each section $C(a_{i-1}, a_i)$ is at most $\mathrm{WBA} \cdot |C(a_{i-1}, a_i)|$ and the area of all sections together sum up to 1, the total bounding box area is clearly at most WBA. But in fact we can prove the following.

Lemma 1. *For any uniform scanning order, we have* TBA = WBA*.*

Proof. Consider a section S of the curve such that $|\mathrm{bbox}(S)|$ is equal to, or close to, $\mathrm{WBA} \cdot |S|$. Now consider a recursive subdivision, following the rules of the scanning order, of the unit region into a set D of m subregions (cells). Let S' be an approximation of S that consists of all the cells of D that are completely covered by S, and let D' be the remaining cells of D. Note that we can repeat the construction within each of the cells of D' (Fig. 3).

$$\mathrm{TBA} \geqslant \bigl|\mathrm{bbox}(S')\bigr| + |D'|/m \cdot \mathrm{TBA} \geqslant \bigl|\mathrm{bbox}(S')\bigr| + \bigl(1 - |S|\bigr) \cdot \mathrm{TBA}.$$

By choosing m big enough we can let $|\mathrm{bbox}(S')|$ be arbitrarily close to $\mathrm{WBA} \cdot |S|$. Thus we get:

$$\mathrm{TBA} \geqslant \mathrm{WBA} \cdot |S| + \bigl(1 - |S|\bigr) \cdot \mathrm{TBA},$$

which solves to $\mathrm{TBA} \geqslant \mathrm{WBA}$. □

Average As we showed above, the worst-case total bounding box area is not very informative. Of greater practical relevance may be the *average total bounding box area* (ABA):

$$\mathrm{ABA} := \lim_{k\to\infty} \mathrm{avg}_{a_1 \prec a_2 \prec \cdots \prec a_{m-1} \in N_k} \left(\sum_{i=1}^{m} \bigl|\mathrm{bbox}\bigl(C(a_{i-1}, a_i)\bigr)\bigr| \right),$$

where the average is taken over sets of $m - 1$ cutting points $a_1, \ldots, a_{m-1}$ uniformly chosen from the unit region. The above equation may serve as a complete definition of the ABA measure for *fixed m*, but this is not completely satisfactory. Experimental results with the scanning orders described in this paper indicate that asymptotically, the measure does not grow or shrink with m, but it exhibits a small fluctuation which repeats itself as m increases by a factor 3, 4 or 9 (depending on the curve)—in other words, the measure tends to be periodic in $\log m$. Therefore any fixed choice of m is likely to give an advantage to some scanning orders and a disadvantage to others. Therefore, we define the average total bounding box area more precisely as the above measure, averaged over a range of values of m, such that m is large enough and $\log m$ is uniformly distributed in a range that covers an integral number of periods of fluctuation of the curve under consideration. We define an *average total bounding octagon area* (AOA) analogously.

We could define an *average total bounding box square perimeter* in a similar way. However, we are ultimately interested in the average perimeter, not the square perimeter. We have to be more careful with the effect of m now: we cannot expect to keep roughly constant total bounding box perimeter as m increases. To cut up a unit region into m sections such that their total bounding box perimeter is minimum, we would have to cut it up into squares of area $1/m$ each, and their total bounding box perimeter would be $4\sqrt{m}$. Therefore the total bounding box perimeter should be considered relative to $4\sqrt{m}$, and we define the *square average relative total bounding box perimeter* (ABP) as:

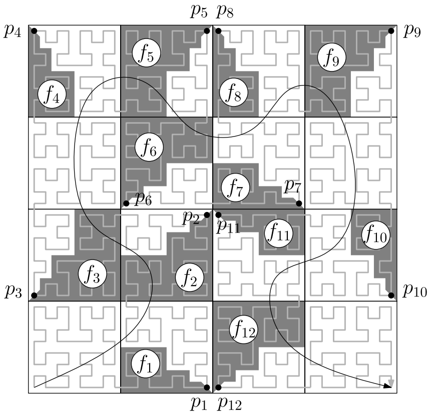


Fig. 4. Corner rectangles in a grid. The smooth curve illustrates the order of the rectangles along the curve. In each corner rectangle, we marked the outer corner and we shaded the front part.

$$\mathrm{ABP} := \lim_{k\to\infty}\left(\mathrm{avg}_{a_1\prec a_2\prec\cdots\prec a_{m-1}\in N_k}\frac{1}{4\sqrt{m}}\left(\sum_{i=1}^{m}\mathrm{peri}\big(\mathrm{bbox}\big(C(a_{i-1},a_i)\big)\big)\right)\right)^2.$$

In the above definition we still take the square in the end, to allow a direct comparison between ABP and WBP. The reader may verify that we must now have $1 \leqslant \mathrm{ABA} \leqslant \mathrm{WBA}$ and $1 \leqslant \mathrm{ABP} \leqslant \mathrm{WBP}$.

3.5. Total diameter measures

Because for some applications it may be interesting to keep the diameter of curve sections small [28] and because our software was easy to adapt to it, our results in Section 6 also include estimations of the *square average relative total curve section diameter* (AD_∞), defined as:

$$\mathrm{AD}_\infty := \lim_{k\to\infty}\left(\mathrm{avg}_{a_1\prec a_2\prec\cdots\prec a_{m-1}\in N_k}\frac{1}{\sqrt{m}}\left(\sum_{i=1}^{m}\mathrm{diam}_\infty\big(C(a_{i-1},a_i)\big)\right)\right)^2,$$

where $\mathrm{diam}_\infty(S)$ is the diameter of S in the L_∞-metric. We also compute AD_1: the same measure based on the L_1-metric.

4. Lower bounds

4.1. Worst-case bounding box area

Theorem 1. *Any scanning order with a rule that contains a triangle has* $\mathrm{WBA} \geqslant 2$.

Proof. The area of a bounding rectangle of any triangle $\triangle$ is at least twice the area of $\triangle$. □

Theorem 2. *Any scanning order based on recursively subdividing an axis-aligned rectangle into a regular grid of rectangles has* $\mathrm{WBA} \geqslant 2$.

Proof. Consider a subdivision of the unit rectangle into a regular grid of m rectangles, following the rules of the scanning order recursively to the depth where a grid of $\sqrt{m} \times \sqrt{m}$ rectangles is obtained. We distinguish two cases: either there is a pair of rectangles that are consecutive in the scanning order and do not share an edge, or all pairs of consecutive rectangles share an edge.

In the first case, the bounding box of such a pair contains at least four rectangles, and thus the curve section that covers that pair results in $\mathrm{WBA} \geqslant 2$.

In the second case we argue as follows. Let $s_1, \ldots, s_m$ be these rectangles in order along the space-filling curve. For each rectangle s_i $(1 < i \leqslant m)$, let the edge of entry be the edge shared with s_{i-1}, and for each rectangle s_i $(1 \leqslant i < m)$, let the edge of departure be the edge shared with s_{i+1}. Among rectangles $s_2, s_3, \ldots, s_{m-1}$, we distinguish two types of rectangles: straight rectangles and corner rectangles. A *straight* rectangle is a rectangle whose edges of entry and departure are not adjacent. A *corner* rectangle is a rectangle s_i whose edges of entry and departure share a vertex—we call this vertex the inner corner of s_i, and the opposite vertex is the outer corner of s_i. The *front part* of s_i is the part of s_i that appears before the outer corner in the order.

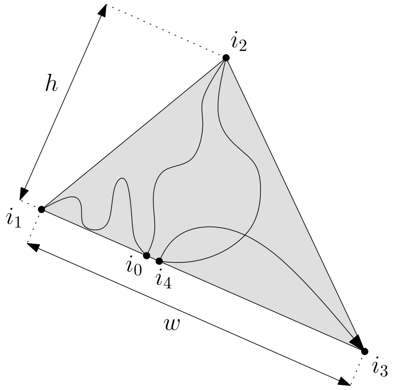


Fig. 5. Definition of the points i_0, i_1, i_2, i_3 and i_4 on the boundary of a triangle, and a sketch of an order in which the space-filling curve may visit them.

Now we number the corner rectangles $t_1, t_2, \ldots, t_k$ in the order in which they appear on the curve, let $p_1, p_2, \ldots, p_k$ be their outer corners, and $f_1, f_2, \ldots, f_k$ be the areas of their front parts (Fig. 4). Note that any sequence of at least $\sqrt{m}$ rectangles must include a corner rectangle, so $k \geqslant \sqrt{m}$. Consider the curve section from p_i to p_{i+2}, for any $i = 1, 2, \ldots, k-2$. Let the width of this section (by number of rectangles) be w, let the height be h, and let $n \geqslant 3$ be the number of rectangles from t_i to t_{i+2} inclusive. Observe that because there is exactly one corner rectangle between t_i and t_{i+2}, namely t_{i+1}, we have $w \geqslant 2$, $h \geqslant 2$, and $w + h = n + 1$ (the $+1$ is because t_{i+1} counts towards both w and h). Now the area of the curve section between p_i and p_{i+2} is $n - 1 + f_{i+2} - f_i$, and the area of its bounding box is $w \cdot h \geqslant 2(n-1)$. Hence we have $\mathrm{WBA} \geqslant 2(n-1)/(n-1+f_{i+2}-f_i)$, or equivalently, $f_{i+2} - f_i \geqslant (2/\mathrm{WBA} - 1)(n-1) \geqslant 2 \cdot (2/\mathrm{WBA} - 1)$.

For the sake of contradiction, suppose $\mathrm{WBA} < 2$. From the above we get $f_{2i+2} - f_{2i} \geqslant 2 \cdot (2/\mathrm{WBA} - 1)$ for all $i \in \{1, 2, 3, \ldots, m'\}$, where m' is $\lfloor \frac{1}{2}\sqrt{m} \rfloor - 1$. Therefore $2/\mathrm{WBA} - 1 \leqslant \frac{1}{2m'} \sum_{i=1}^{m'} (f_{2i+2} - f_{2i}) = \frac{1}{2m'}(f_{2m'+2} - f_2) < \frac{1}{2m'}$. This must be true for any grid of rectangles that is obtained by refining the subdivision recursively, following the rules of the scanning order. So we must have $\lim_{m\to\infty}(2/\mathrm{WBA} - 1) = 0$ and thus $\lim_{m\to\infty} \mathrm{WBA} = 2$. □

4.2. Worst-case locality

Niedermeier et al. [24] prove $\mathrm{WL}_2 \geqslant \mathrm{WL}_\infty \geqslant 3\frac{1}{2}$ for scanning orders that contain a section whose perimeter is an axis-aligned square. The proof is based on defining six points on the boundary of the square that need to be visited by the curve. For each possible order in which these points may be visited, they add up the squares of the distances between each pair of consecutive points in the order. Thus they derive a lower bound on this sum that holds for all orders in which the points can be visited. For the L_∞- and L_2-metric this lower bound is at least $3\frac{1}{2}$ times the area of the square, and the lower bounds on WL_∞ and WL_2 follow.

Below we show how to apply this technique to triangular curve sections. Unfortunately it does not work that well for rectangular curve sections. But there our new proof technique of Theorem 2 comes to rescue, leading to better bounds.

Theorem 3. *Any scanning order with a rule that contains a triangle has* $\mathrm{WL}_2 \geqslant 4$.

Proof. Consider a triangle that constitutes a contiguous section of the curve. Let i_1, i_2 and i_3 be its three vertices, in the order in which they appear on the curve. Let i_0 be the point on the edge $i_1 i_3$ that appears before i_2 and is closest to i_3 among such points. Let i_4 be a point on the segment $i_0 i_3$, arbitrarily close to i_0. Let w be the length of the edge $i_1 i_3$, and let h be the height of the triangle relative to this edge, that is, the distance between i_2 and the line containing $i_1 i_3$, see Fig. 5.

The curve may pass through the points just defined in four different orders, it must be $i_0 i_1 i_2 i_3 i_4$, $i_1 i_0 i_2 i_3 i_4$, $i_0 i_1 i_2 i_4 i_3$, or $i_1 i_0 i_2 i_4 i_3$.

We will now analyse the last possibility in detail (the others can be checked in a similar way). Consider the first leg of this path. When going from i_1 to i_0, the area $C(i_1, i_0)$ filled by the curve must satisfy $d_2^2(i_1, i_0)/|C(i_1, i_0)| \leqslant \mathrm{WL}_2$, that is:

$$\left|C(i_1, i_0)\right| \geqslant d_2^2(i_1, i_0)/\mathrm{WL}_2.$$

Similarly we get:

$$\left|C(i_0, i_2)\right| \geqslant d_2^2(i_0, i_2)/\mathrm{WL}_2 \geqslant h^2/\mathrm{WL}_2,$$
$$\left|C(i_2, i_4)\right| \geqslant d_2^2(i_2, i_4)/\mathrm{WL}_2 \geqslant h^2/\mathrm{WL}_2,$$
$$\left|C(i_4, i_3)\right| \geqslant d_2^2(i_4, i_3)/\mathrm{WL}_2 = \left(w^2 - 2wd_2(i_1, i_0) + d_2^2(i_1, i_0)\right)/\mathrm{WL}_2.$$

Adding these up we get:

$$\left|C(i_1,i_3)\right| \geqslant \left(2h^2 + w^2 - 2wd_2(i_1,i_0) + 2d_2^2(i_1,i_0)\right)/\mathrm{WL}_2 \geqslant \left(2h^2 + w^2/2\right)/\mathrm{WL}_2.$$

Note that $C(i_1,i_0) \cup C(i_0,i_2) \cup C(i_2,i_4) \cup C(i_4,i_3) \cup C(i_1,i_3)$ is at most the complete triangle, with $|C(i_1,i_3)| \leqslant hw/2$. Thus we get $hw/2 \geqslant (2h^2 + w^2/2)/\mathrm{WL}_2$, which solves to $\mathrm{WL}_2 \geqslant 4h/w + w/h \geqslant 4$.

The other possible orders of i_0, i_1, i_2, i_3 and i_4 can be analysed in a similar way, leading to the same result. □

Note that in fact we get $\mathrm{WL}_2 \geqslant 4\alpha + 1/\alpha$, where α is the height/width ratio that minimises the right-hand side. Somewhat surprisingly, this implies that optimal locality cannot be achieved with equilateral triangles: with $\alpha = \frac{1}{2}\sqrt{3}$ they are subject to a lower bound of $\mathrm{WL}_2 \geqslant 8/\sqrt{3}$, while the triangles of the Sierpiński–Knopp order, with $\alpha = 1/2$, give $\mathrm{WL}_2 = 4$ [24].

The proof technique of Niedermeier et al. could also be modified for scanning orders that contain rectangular (but not necessarily square) sections. However, in that case the lower bound will drop below 3 (consider rectangles of aspect ratio $\sqrt{5}$). Below we show how to use the proof technique of Theorem 2 to get a lower bound of 4, not only for WL_2 but also for WL_∞.

Theorem 4. *Any scanning order based on recursively subdividing an axis-aligned rectangle into a regular grid of rectangles has* $\mathrm{WL}_2 \geqslant \mathrm{WL}_\infty \geqslant 4$.

Proof. We follow the same approach as with Theorem 2, but to get a good bound on WL_∞ (not only WL_2), we need to be a bit more careful in the definition of the corners.

Consider a subdivision of the unit rectangle into a regular grid of rectangles, following the rules of the scanning order recursively to the depth where a grid of $\sqrt{m} \times \sqrt{m}$ rectangles is obtained. Let $s_1, \ldots, s_m$ be these rectangles in the order in which the space-filling curve visits them. Note that each rectangle touches the next one, at least in a vertex, otherwise WL_2 and WL_∞ would be unbounded. Assume that the height/width ratio of each rectangle is $\alpha \geqslant 1$ (otherwise we swap the coordinate axes). Within this proof, we define the width of a single rectangle to be $1/\sqrt{\alpha}$, and so its height is $\sqrt{\alpha}$ and its area is 1.

Among rectangles $s_2, s_3, \ldots, s_{m-1}$, we distinguish three types of rectangles.

A rectangle s_i is *straight* when s_{i-1}, s_i and s_{i+1} lie either in the same row, or in three different rows.

A rectangle s_i is a *corner* rectangle when exactly one rectangle out of s_{i-1} and s_{i+1} lies in the same row as s_i. The *outer corner* of s_i is the vertex that lies farthest from s_{i-1} and s_{i+1}; more precisely, if s_{i-1} is in the same row as s_i, the outer corner is the vertex of s_i that touches neither the column of s_{i-1} nor the row of s_{i+1}; otherwise, that is, if s_{i+1} is in the same row as s_i, the outer corner is the vertex of s_i that touches neither the row of s_{i-1} nor the column of s_{i+1}.

A rectangle s_i is a *double corner* rectangle when s_{i-1} and s_{i+1} lie in the same row, but not in the same row as s_i—implying that the curve makes something of a U-turn in s_i. Such a rectangle s_i has two outer corners, namely the vertices of s_i that do not touch s_{i-1} or s_{i+1}. We distinguish a first corner and a second corner, by the order in which they appear in the scanning order between s_{i-1} and s_{i+1} (Fig. 6).

Now number the corner rectangles $t_1, t_2, \ldots, t_k$ in the order in which they appear on the curve, with each double corner rectangle listed twice. Let $p_1, p_2, \ldots, p_k$ be the outer corners of these rectangles. Where t_i and t_{i+1} are the two copies of a double corner rectangle, p_i is the first corner and p_{i+1} is the second corner. The *front part* of t_j is the part of t_j that appears before p_j in the order. As before, we have $k \geqslant \sqrt{m}$.

As before, we will argue about curve sections between two corners p_i to p_{i+2}, for even i. In addition we assume that for even i, the rectangles t_i and t_{i+1} lie in the same row (if this is not the case, we could simply do the calculations given below for odd i instead of even i).

For the sake of contradiction, assume that $\mathrm{WL}_\infty < 4$.

Let $w \geqslant 1$ be the number of rectangles in the order from t_i to t_{i+1} inclusive (w could be 1 if $t_i = t_{i+1}$ is a double corner rectangle), and $h \geqslant 2$ the number of rectangles in the order from t_{i+1} to t_{i+2} inclusive.

The L_∞-distance between p_i and p_{i+1} is at least $w/\sqrt{\alpha}$, hence we get $\mathrm{WL}_\infty \geqslant \frac{1}{\alpha}w^2/(w - 1 + f_{i+1} - f_i)$, which we rewrite as:

$$f_{i+1} - f_i \geqslant \frac{w^2/\alpha}{\mathrm{WL}_\infty} - (w-1). \tag{1}$$

The L_∞-distance between p_{i+1} and p_{i+2} is at least $\sqrt{\alpha}h$. Hence we get:

$$\mathrm{WL}_\infty \geqslant \frac{\alpha h^2}{h - 1 + f_{i+2} - f_{i+1}}, \tag{2}$$

which we rewrite as:

$$f_{i+2} - f_{i+1} \geqslant \frac{\alpha h^2}{\mathrm{WL}_\infty} - (h-1). \tag{3}$$

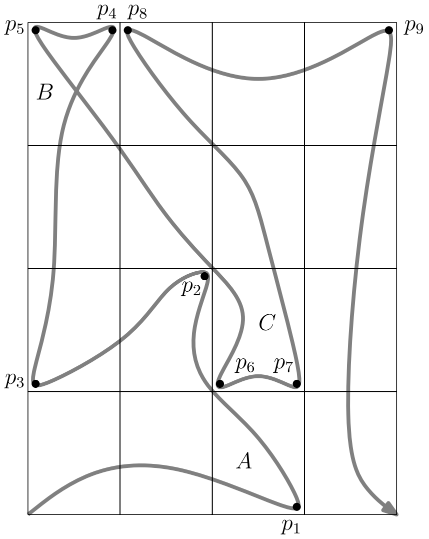


Fig. 6. Corners of a scanning order that is not edge-connected on a rectangular grid. Note how horizontal and mainly vertical sections alternate between the corners. Although both the predecessor and the successor of rectangle A lie to the left, it is *not* a double corner rectangle, since the path to the predecessor is horizontal and a path to a rectangle in another row (such as the successor A) is always considered to be vertical. Rectangles B and C are double corner rectangles.

Adding up Eqs. (1) and (3) we get:

$$f_{i+2} - f_i \geqslant \frac{w^2/\alpha + \alpha h^2}{\mathrm{WL}_\infty} - (w + h - 2). \tag{4}$$

Furthermore Eq. (2) gives us $4 > \mathrm{WL}_\infty \geqslant \alpha h^2/(h - 1 + f_{i+2} - f_{i+1}) \geqslant \alpha h \geqslant h$. From this we get that h must be either 2 or 3. In the case of $h = 2$ Eq. (4) becomes:

$$f_{i+2} - f_i \geqslant \frac{w^2/\alpha + 4\alpha}{\mathrm{WL}_\infty} - w = \left(\frac{w/\alpha + 4\alpha/w}{\mathrm{WL}_\infty} - 1\right) w \geqslant \frac{4}{\mathrm{WL}_\infty} - 1.$$

In the case of $h = 3$ and $w = 1$ Eq. (4) becomes:

$$f_{i+2} - f_i \geqslant \frac{1/\alpha + 9\alpha}{\mathrm{WL}_\infty} - 2 \geqslant \frac{10}{\mathrm{WL}_\infty} - 2 > 2\left(\frac{4}{\mathrm{WL}_\infty} - 1\right) > \frac{4}{\mathrm{WL}_\infty} - 1.$$

In the case of $h = 3$ and $w \geqslant 2$ Eq. (4) becomes:

$$f_{i+2} - f_i \geqslant \frac{\frac{w^2}{\alpha} + 9\alpha}{\mathrm{WL}_\infty} - (w+1) \geqslant \frac{\frac{4(w+1)^2}{9\alpha} + 9\alpha}{\mathrm{WL}_\infty} - (w+1) = \left(\frac{4\frac{w+1}{9\alpha} + \frac{9\alpha}{w+1}}{\mathrm{WL}_\infty} - 1\right)(w+1) \geqslant \frac{4}{\mathrm{WL}_\infty} - 1.$$

Thus we get $f_{i+2} - f_i \geqslant 4/\mathrm{WL}_\infty - 1$ in all cases. The proof now concludes as for Theorem 2, leading to the conclusion $\lim_{m\to\infty} \mathrm{WL}_\infty = 4$, which also implies $\mathrm{WL}_2 \geqslant 4$. □

Niedermeier et al. [24] also proved $\mathrm{WL}_2 \geqslant \mathrm{WL}_\infty \geqslant 4$, but for another class of scanning orders, namely those that (i) contain a section whose perimeter is an axis-aligned square and (ii) are cyclic, that is, the end of that section touches its beginning. Our proof does not need those conditions, but needs others. To prove $\mathrm{WL}_2 \geqslant 4$ we require that the curve has a triangular section, or that it has a rectangular section subdivided recursively into a regular grid of rectangles.

Regarding L_1-locality, Niedermeier et al. proved $\mathrm{WL}_1 \geqslant 8$ if both conditions (i) and (ii) hold, and $\mathrm{WL}_1 \geqslant 6\frac{1}{2}$ if only condition (i) holds. We have no results to complement this: it seems hard to use our technique to prove any lower bound on WL_1 which is significantly better than 4.

5. Approximating the worst-case measures

In this section we describe how we can obtain upper and lower bounds on the quality measures such as the worst-case locality and the worst-case bounding box quality measures defined in Section 3. For ease of description, we assume that the scanning order is defined by a single recursive rule without reversals. The techniques described below can easily be extended to multiple-rule scanning orders or scanning orders with reversals (in fact that is what we implemented).

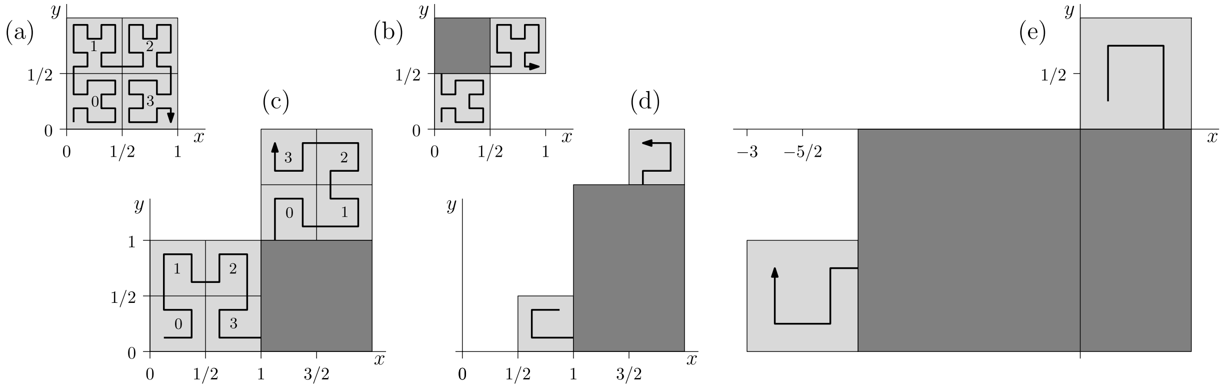


Fig. 7. (a) The Hilbert order. (b) Base probe B_{02} of the Hilbert order. The front transformation $\alpha(B_{02})$ consists of scaling with factor 1/2 and a reflection in the line $x = y$. The midsection $M(B_{02})$ is the square $[0, \frac{1}{2}] \times [\frac{1}{2}, 1]$ (the dark area in the figure). The tail transformation $\omega(B_{02})$ consists of scaling with factor 1/2 and a translation over $(\frac{1}{2}, \frac{1}{2})$. (c) Canonical form $\mathcal{P}$ of B_{02}. The canonical form is obtained from B_{02} by applying the transformation $\alpha(B_{02})^{-1}$, that is, reflection in the line $x = y$ and scaling with factor 2. (d) Refinement $\mathrm{r}_{32}(\mathcal{P})$ of $\mathcal{P}$. (e) The same refinement in canonical form: child $\mathcal{P}_{32}$ of $\mathcal{P}$.

Let μ be a mapping from regions to real numbers in a way that is invariant under all transformations $\tau(i)$ involved in the recursive definition of the scanning order. For example, $\mu(R)$ could be $|\mathrm{bbox}(R)|/|R|$, or the square of the diameter of R divided by $|R|$. Our goal is to approximate $\mu^* = \lim_{k\to\infty} \sup_{i \prec j \in N_k} \mu(C(i, j))$. (We may also let μ depend on $C(i)$ and/or $C(j)$.) The mapping μ must be well-defined when $C(i, j)$ is not empty; when $|C(i, j)| = 0$ we may assume $\mu(C(i, j)) = \infty$.

5.1. Representing curve sections by probes

We will compute the approximation of μ^* by exploring *probes*. A probe P is specified by three consecutive subsections of the order: a front section, a midsection, and a tail section. The probe P thus describes a set of contiguous subsections of the scanning order, namely all those subsections S that start somewhere in the front section of P and end somewhere in the tail section of P. For any probe P, let $\alpha(P)$ be the transformation that transforms C into the front section of P; let $M(P)$ be the midsection of P; and let $\omega(P)$ be the transformation that transforms C into the tail section of P. A section $P(i, j)$ of a probe P is the region $\alpha(P)C(\succeq i) \cup M(P) \cup \omega(P)C(\prec j)$. Let $\mu^*(P)$ be the maximum value of $\mu(S)$ over all subsections S covered by the probe, that is, $\mu^*(P) = \lim_{k\to\infty} \sup_{i,j\in N_k} \mu(P(i, j))$. A probe P may be rotated, mirrored, scaled and/or reversed: this does not affect the value of $\mu^*(P)$.

All subsections of the scanning order can be captured by a set of probes as follows. For $0 \leqslant i < k < n$, let *base probe* B_{ik} be the probe with front transformation $\tau(i)$, midsection $\bigcup_{i<j<k} C(j)$, and tail transformation $\tau(k)$. For an example, see Fig. 7(a), (b).

Lemma 2. $\mu^* = \max_{0\leqslant i<k<n} \mu^*(B_{ik})$.

Proof. For any $x < \mu^*$, let a, b be any pair such that $a \prec b$ and $\mu(C(a, b)) > x$. Now let c be the longest common prefix of a and b. Note that a and b have the same number of digits, so c must be a proper prefix of both a and b. Let $\hat{a}$ be the digit of a following the prefix c, and let $\hat{b}$ be the digit of b following the prefix c. Thus $C(a)$ lies in subregion $\hat{a}$ of $C(c)$, and $C(b)$ lies in subregion $\hat{b}$ of $C(c)$. Since μ is invariant under $\tau(c)$, it follows that $\mu^*(B_{\hat{a}\hat{b}}) \geqslant \mu(C(a, b)) > x$. Thus, for any $x < \mu^*$, there are $0 \leqslant i < k < n$ such that $\mu^*(B_{ik}) > x$. This implies there are $0 \leqslant i < k < n$ such that $\mu^* \leqslant \mu^*(B_{ik})$, which proves the lemma. □

Let *refinement* $\mathrm{r}_{ij}(P)$ of probe P, with $i, j \in \{0, \ldots, n-1\}$, be the probe with front transformation $\alpha(P) \circ \tau(i)$, tail transformation $\omega(P) \circ \tau(j)$, and midsection $\alpha(P)(C(\succ i)) \cup M(P) \cup \omega(P)(C(\prec j))$; see Fig. 7(c), (d). Since $P = \bigcup \mathrm{r}_{ij}(P)$, we have $\mu^*(P) = \max \mu^*(\mathrm{r}_{ij}(P))$.

We say a probe P is in canonical form if $\alpha(P)$ is the identity transformation. We can construct a canonical form $\mathcal{P}$ of any probe P by setting $\alpha(\mathcal{P})$ to the identity transformation, $M(\mathcal{P}) := \alpha(P)^{-1}(M(P))$, and $\omega(\mathcal{P}) := \alpha(P)^{-1} \circ \omega(P)$; see Fig. 7(c), (e). Since μ is invariant under all transformations involved, we have $\mu^*(P) = \mu^*(\mathcal{P})$. Therefore it suffices to work only with probes in canonical form, where the *children* of a canonical probe $\mathcal{P}$ are the canonical forms of its refinements. So child $\mathcal{P}_{ij}$ is the canonical probe with midsection $M(\mathcal{P}_{ij}) := \tau(i)^{-1}(C(\succ i) \cup M(\mathcal{P}) \cup \omega(\mathcal{P})(C(\prec j)))$ and tail transformation $\omega(\mathcal{P}_{ij}) := \tau(i)^{-1} \circ \omega(\mathcal{P}) \circ \tau(j)$ (Fig. 7(e)). Observe that $\tau(i)^{-1}$ always includes scaling with a factor greater than 1, so for any canonical probe $\mathcal{P}$ and any canonical child $\mathcal{P}_{ij}$ we have $|M(\mathcal{P}_{ij})| > |M(\mathcal{P})|$, unless $i = n-1$, $M(\mathcal{P}) = \emptyset$ and $j = 0$.

Note that while computing $\mu^*(\mathcal{P})$ may be difficult, it may be easy to get a lower bound $\mu^-(\mathcal{P})$ and an upper bound $\mu^+(\mathcal{P})$ on $\mu^*(\mathcal{P})$. For example, if $\mu(A)$ is defined as $|\mathrm{bbox}(A)|/|A|$, then $|\mathrm{bbox}(M(\mathcal{P}))|/|C \cup M(\mathcal{P}) \cup \omega(\mathcal{P})(C)|$ would be a

```
ComputeCurveQuality
 1 $Q \leftarrow$ an empty first-in-first-out queue
 2 $R \leftarrow$ an empty dictionary
 3 Insert the canonical forms of all base probes $B_{ik}$ in $Q$ and in $R$
 4 $lowerBound \leftarrow \max_{\mathcal{P} \in Q} \mu^-(\mathcal{P})$
 5 while $\max_{\mathcal{P} \in Q} \mu^+(\mathcal{P}) - lowerBound >$ the desired precision
 6     do Extract a probe $\mathcal{P}$ from the head of $Q$
 7         for all canonical children $\mathcal{P}_{ij}$ of $\mathcal{P}$
 8             do if $\mu^+(\mathcal{P}_{ij}) \geqslant lowerBound$ and $R$ does not contain $\mathcal{P}_{ij}$
 9                 then Add $\mathcal{P}_{ij}$ to $Q$ and $R$
10                     $lowerBound \leftarrow \max(lowerBound, \mu^-(\mathcal{P}_{ij}))$
11 Report that $\mu^*$ lies in the interval $[lowerBound, \max_{\mathcal{P} \in Q} \mu^+(\mathcal{P})]$.
```

Fig. 8. Algorithm to compute an approximation of a curve quality measure.

lower bound on $\mu^*(\mathcal{P})$, and $|\text{bbox}(C \cup M(\mathcal{P}) \cup \omega(\mathcal{P})(C))|/|M(\mathcal{P})|$ would be an upper bound on $\mu^*(\mathcal{P})$ (provided $|M(\mathcal{P})| > 0$, otherwise we define $\mu^*(\mathcal{P}) := \infty$).

5.2. Searching probes

Our general algorithm to approximate μ^* is shown in Fig. 8. The main idea of the algorithm is that it keeps replacing probes by their refinements to get tighter lower and upper bounds on μ^*. It is easy to see that the algorithm produces a correct lower bound. We will prove that the algorithm also produces a correct upper bound by proving that the following invariant holds after every iteration of the while loop: for every probe $\mathcal{P}$ that was ever added to the queue and has $\mu^*(\mathcal{P}) = \mu^*$, there is a descendant $\mathcal{P}'$ of $\mathcal{P}$ in Q such that $\mu^*(\mathcal{P}') = \mu^*$.

In fact the algorithm would be trivial to prove correct if we would omit the check if $\mathcal{P}_{ij}$ is contained in R on line 8. However, the algorithm would be useless, because the queue would continue to contain *degenerate probes*, that is, probes $\mathcal{P}$ with $M(\mathcal{P}) = \emptyset$. This is because some degenerate probes are inserted on line 3 (namely all base probes B_{ik} with $k = i + 1$), and whenever a degenerate probe $\mathcal{P}$ is extracted from the queue, its degenerate child $\mathcal{P}_{n-1,0}$ would be added to the queue. Since $\mu^+(\mathcal{P}) = \infty$ for degenerate probes, the algorithm would never find an actual upper bound on μ^*.

Fortunately, it is easy to see that for most space-filling curves the algorithm can only generate a small number of *different* degenerate probes: typically for any degenerate canonical probe the transformation ω has scale factor one, rotations and reflections form a small closed set, and the translation is fixed because the tail section has to connect to the front section. Moreover, non-degenerate probes have only non-degenerate children. Therefore, making sure that no probe is inserted in Q more than once, guarantees that Q soon becomes and remains free of degenerate probes, and the algorithm soon finds an upper bound on μ^*. We have to prove, however, that this upper bound is indeed correct.

Theorem 5. *Algorithm* ComputeCurveQuality *returns correct lower and upper bounds on μ^*.*

Proof. We need to prove that, despite the fact that the algorithm refuses to insert probes in Q that were inserted before, the invariant still holds after every iteration: for every probe $\mathcal{P}$ that was ever added to the queue and has $\mu^*(\mathcal{P}) = \mu^*$, there is a descendant $\mathcal{P}'$ of $\mathcal{P}$ in Q such that $\mu^*(\mathcal{P}') = \mu^*$.

For the sake of contradiction, let $\mathcal{P}$ be the probe that is removed from Q in the first iteration that violates the invariant and does not restore it by inserting children of $\mathcal{P}$. Since $\mu^*(\mathcal{P}) = \mu^*$, the probe $\mathcal{P}$ must have a child $\mathcal{P}_{ij}$ with $\mu^*(\mathcal{P}_{ij}) = \mu^* \geqslant lowerBound$. If this child is not inserted in Q (restoring the invariant), it must be because R contains $\mathcal{P}_{ij}$ already, which implies that at the beginning of the iteration Q contained a descendant $\mathcal{P}''$ of $\mathcal{P}_{ij}$ with $\mu^*(\mathcal{P}'') = \mu^*$. Since $\mathcal{P}''$ is also a descendant of $\mathcal{P}$, the invariant that Q holds a descendant $\mathcal{P}'$ of $\mathcal{P}$ with $\mu^*(\mathcal{P}') = \mu^*$ can only be violated by also removing (at least) $\mathcal{P}''$. But we remove only one probe from the queue, so this implies $\mathcal{P}'' = \mathcal{P}$, and we have that $\mathcal{P}_{ij}$ is a descendant of $\mathcal{P}$ and vice versa. Now as observed above, from canonical parent to canonical child the size of the midsection is either zero or strictly increasing. So if $\mathcal{P}$ is a descendant of $\mathcal{P}_{ij}$, we must have $M(\mathcal{P}) = M(\mathcal{P}_{ij}) = \emptyset$, $i = n - 1$, and $j = 0$.

Consider a full line of ancestry of $\mathcal{P}_{ij}$ down from itself, that is, a sequence of canonical probes $\mathcal{P}^0, \ldots, \mathcal{P}^k$, where $\mathcal{P}^0 = \mathcal{P}_{ij}, \mathcal{P}^{k-1} = \mathcal{P}$, $\mathcal{P}^k = \mathcal{P}^0$, and each probe $\mathcal{P}^{m+1}$ is a child of $\mathcal{P}^m$, for $0 \leqslant m < k$. We have $\mu^*(\mathcal{P}^m) = \mu^*$ for all $0 \leqslant m \leqslant k$. We prove by induction on m that at the end of the current iteration, we have for all $0 \leqslant m < k$ that $\mathcal{P}^m$ is in R but not in Q. For $\mathcal{P}^0 = \mathcal{P}_{ij}$ this is true because the violation of the invariant is caused by not inserting $\mathcal{P}_{ij}$ in Q, which must be because it is in R already. Now, given that $\mathcal{P}^0, \ldots, \mathcal{P}^m$ are in R but not in Q, consider $\mathcal{P}^{m+1}$. The probe $\mathcal{P}^{m+1}$ is in R, because $\mathcal{P}^m$ is in R and not in Q; when $\mathcal{P}^m$ was removed from Q, its child $\mathcal{P}^{m+1}$, with $\mu^*(\mathcal{P}^{m+1}) = \mu^* \geqslant lowerBound$, must have been added to R if it was not in R already. But again $\mathcal{P}^{m+1}$ is not in Q, otherwise Q would contain a descendant of $\mathcal{P}$ and the invariant would not be violated.

Therefore $\mathcal{P}^0, \ldots, \mathcal{P}^k$ are all in R but not in Q, that is, they were all once added to the queue and have been extracted since. It follows that any non-degenerate children $\mathcal{P}^*$ of $\mathcal{P}^0, \ldots, \mathcal{P}^k$ with $\mu^*(\mathcal{P}^*) = \mu^*$ were once inserted in R and Q, so Q must still contain descendants of any such non-degenerate children $\mathcal{P}^*$. The removal of $\mathcal{P}$, a degenerate probe, did not change that, so if there was ever such a non-degenerate child $\mathcal{P}^*$ with $\mu^*(\mathcal{P}^*) = \mu^*$, then a descendant $\mathcal{P}'$ of $\mathcal{P}^*$, and

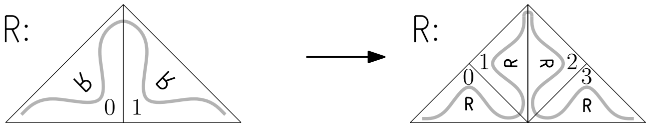


Fig. 9. Expanding the ordering rule of the Sierpiński–Knopp order by one level results in rotations by multiples of $\pi/2$.

thus, of $\mathcal{P}$, with $\mu^*(\mathcal{P}') = \mu^*$, would still exist in Q and the invariant would not be violated. So we must conclude that for $0 \leqslant m < k$, the degenerate probe $\mathcal{P}^{m+1}$ is the *only* child $\mathcal{P}^*$ of $\mathcal{P}^m$ with $\mu^*(\mathcal{P}^*) = \mu^*$. It follows that $\mathcal{P}$ in particular does not have any non-degenerate descendant probes $\mathcal{P}^*$ with $\mu^*(\mathcal{P}^*) = \mu^*$.

Now, for any value $x < \mu^*$, let a, b be any pair such that $\mu(\mathcal{P}(a,b)) > x$. Let $\hat{a}$ be a followed by a zero, and let $\hat{b}$ be b followed by a zero. Now we have $\mu(\mathcal{P}(\hat{a},\hat{b})) = \mu(\mathcal{P}(a,b)) > x$, and $C(\succ \hat{a}) \neq \emptyset$. The probe $\mathcal{P}^*$ with midsection $\tau(a)^{-1}(C(\succ \hat{a}) \cup M(\mathcal{P}) \cup \omega(\mathcal{P})(C(\prec \hat{b})))$, and tail transformation $\tau(a)^{-1} \circ \omega(\mathcal{P}) \circ \tau(\hat{b})$, is a non-degenerate descendant probe of $\mathcal{P}$ with $\mu(\mathcal{P}^*) > x$. Since we can find such a non-degenerate descendant probe of $\mathcal{P}$ for every $x < \mu^*$, this contradicts the conclusion of the previous paragraph that $\mathcal{P}$ does not have any non-degenerate descendant probes $\mathcal{P}^*$ with $\mu^*(\mathcal{P}^*) = \mu^*$.

Hence there cannot be such a probe $\mathcal{P}$ whose removal from Q leads to the invariant being violated at the end of an iteration. Therefore the invariant is maintained, and the algorithm is correct. □

5.3. Implementation

To implement algorithm ComputeCurveQuality for a particular measure, one needs to come up with a representation of midsections of canonical probes $\mathcal{P}$ that enables a quick, correct, and reasonably tight approximation of $\mu^*(\mathcal{P})$ from above and below. If the approximations are tight, the algorithm may zoom in on the worst-case sections of the curve quickly, keeping a small probe queue Q and refining only where necessary. If the approximations are not tight enough, the algorithm may degenerate into a search of a complete grid, taking long to converge or not converging at all. For an efficient operation of the algorithm the representation of the midsections should also be easy to maintain under the transformations in the rules that describe the scanning order.

For some curves and measures a good implementation is easier to accomplish than for others. To compute WL_∞, WL_2, and WL_1 we only need to know the size (area) of the midsection; for WBA and WBP we also need to know the minimum rectangular bounding box of the midsection; for WOA and WOP we need to know the minimum octagonal bounding box. For curves that only use axis-parallel reflections and rotations with angles of $\frac{1}{2}\pi$, π and $\frac{3}{2}\pi$, these midsection properties can be maintained easily. Fortunately, most curves presented in Section 2.2 fulfill these requirements. However, the Sierpiński–Knopp order and the Gosper flowsnake use rotations that are not multiples of $\pi/2$, and as a result the WL_∞, WL_1, WBA, WBP, WOA and WOP measures are not invariant under rotation as required by the algorithm.

For the Sierpiński–Knopp order this is easily solved by expanding the definition by one level of recursion, see Fig. 9.

For the Gosper flowsnake we observe that WL_2 is invariant under rotation and can still be computed by our algorithm. Because the rotations of the curve are not a rational fraction of π, any pair of points that defines the worst-case L_2-locality will appear scaled down and turned arbitrarily close to horizontal somewhere in the curve, so that we have $\mathrm{WL}_\infty \geqslant \mathrm{WL}_2$. Since the L_2-distance between any pair of points is at least their L_∞-distance, it follows that $\mathrm{WL}_\infty = \mathrm{WL}_2$. Likewise, any pair of points that defines the worst-case L_2-locality will appear scaled down and turned arbitrarily close to making a 45 degree angle with the horizontal, so that we have $\mathrm{WL}_1 \geqslant 2\mathrm{WL}_2$ and $\mathrm{WBP} \geqslant \mathrm{WBA} \geqslant \mathrm{WL}_2/2$. Since the squared L_1-distance between any pair of points is at most twice their squared L_2-distance, it follows that $\mathrm{WL}_1 = 2\mathrm{WL}_2$. For the computation of WBA, WBP, WOA and WOP similar arguments could be used, provided the boundaries of unit regions and midsections are represented in such a way that one can determine good lower and upper bounds on the maximum, over all possible rotations, of the size of the minimum bounding box. This seems hard to accomplish due to the fractalic nature of the region boundaries.

6. Computational results

Computing pairwise worst-case measures We implemented the approximation algorithm described above, specifically to compute the worst-case locality measures WL_∞, WL_2, and WL_1, the worst-case bounding box quality measures WBA and WBP, and the corresponding measures for bounding octagons WOA and WOP. We ran our algorithm on the curves mentioned in Section 2.2, except that we did not try to compute WBA, WBP, WOA and WOP for the Gosper flowsnake, because of the reasons explained in Section 5.3.

The running times of the computations varied with the curves. When the algorithm succeeded in having a probe queue of constant size, we would have an approximation with precision 0.0001 within a fraction of a second (higher precision being prevented by the number of bits used to represent numbers); when the probe queue kept growing, the computation could take a few minutes and the precision would be limited by memory requirements. WBA, WBP, WOA and WOP were always computed fast and precise; WL_1 was fast and precise for all orders except Sierpiński–Knopp order; WL_2 and WL_∞ were generally computed only to a precision of 0.0005.

Table 1
Bounds for different measures and curves. New curves printed in bold. For the A-measures the standard deviation is indicated behind the number: no symbol when less than 0.5%; one mark when between 0.5% and 1.0%, two marks when between 1.0% and 2.0%.

Order	WL_∞	WL_2	WL_1	WBA	ABA	WBP	ABP	WOA	AOA	WOP	AD_∞
Sierpiński–Knopp order	4	4	8	3.000	1.78	3.000	1.42	1.789	1.25	1.629	$1.77'$
Balanced GP	4.619	4.619	8.619	2.000	1.44	2.155	1.19	1.769	1.31	1.807	$1.72'$
GP (Serp. 000 000 000)	8	8	$10\frac{2}{3}$	2.000	1.44	2.722	1.28	1.835	1.32	2.395	$2.13'$
Serpentine 011 010 110	5.625	6.250	10.000	2.500	$1.44''$	2.500	1.20	2.222	$1.32'$	2.036	$1.71'$
Luxburg 2 (101 010 101)	$5\frac{5}{8}$	$6\frac{1}{4}$	10	2.500	$1.49'$	2.500	1.24	2.222	$1.35'$	2.036	$1.81'$
Meurthe (110 110 110)	5.333	5.667	10.667	2.500	$1.41''$	2.667	1.17	2.000	$1.30'$	2.018	$1.64'$
Coil (Serp. 111 111 111)	$6\frac{2}{3}$	$6\frac{2}{3}$	$10\frac{2}{3}$	2.500	$1.41'$	2.667	1.17	2.222	$1.29'$	2.424	$1.63'$
Hilbert	6	6	9	2.400	1.44	2.400	1.19	1.929	1.30	1.955	$1.67'$
$\beta\Omega$	5.000	5.000	9.000	2.222	1.42	2.250	1.17	1.800	1.29	1.933	$1.64'$
AR^2W^2	5.400	6.046	12.000	3.055	$1.49'$	3.125	1.22	2.344	1.33	2.255	$1.70'$
Z-order	∞	∞	∞	∞	2.92	∞	$2.40'$	∞	2.46	∞	$3.80''$
Gosper flowsnake	6.35	6.35	12.70	$\geqslant 3.18$		$\geqslant 3.18$					

We tested the 278 different orders that fit Wunderlich's scheme and are defined on a 2 × 2-grid (that is, Hilbert order), a 3 × 3-grid (R-order and the 272 Serpentine orders), or a 4 × 4-grid (4 different orders, not counting Hilbert order). From these orders five curves $\mathcal{C}$ turned out to be "dominant" in the sense that there was no curve that was better than $\mathcal{C}$ on at least one measure and at least as good as $\mathcal{C}$ on the other measures. These five curves are the Hilbert order (best on WL_1, WBP and WOP), Serpentine 000 000 000 (that is, GP-order, best on WBA and WOA), Serpentine 011 010 110 and 101 010 101 (with equal scores, not best on anything but not dominated either), and Serpentine 110 110 110 (Meurthe, best on WL_2 and (WL_∞).

Computing average total measures We examined the five dominant curves further, together with coil order, balanced GP-order, $\beta\Omega$-order, AR^2W^2-order, Z-order and Sierpiński–Knopp order. For these curves we estimated the average total bounding box area and the square average relative total bounding box perimeter (also for bounding octagons) by random sampling as follows: we generated 100 sets of numbers chosen uniformly between 0 and 1 that subdivide the curve, where the logarithm of the size of each set was chosen uniformly between log 500 and log 18 000. Thus we cover an integral number of periods of fluctuation for each scanning order. We estimated the average total area (or relative perimeter) by taking the average over these 100 sample subdivisions. Because in some applications it is useful to keep the L_1-diameter of curve sections small and our software was easy to adapt to it, we used the same method to also estimate the square average relative total curve section diameter in the L_∞ metric (AD_∞) and in the L_1 metric (AD_1).

Results The results of our computations are shown in Table 1. Note that for some scanning orders the exact worst-case locality measures were already known: tight lower and upper bounds for the Hilbert order were proven one by one in several papers: lower bounds on WL_∞, WL_2, and WL_1 by Alber and Niedermeier [1], Gotsman and Lindenbaum [10], and Niedermeier and Sanders [25], respectively; upper bounds by Bauman [5] and Chochia et al. [6], respectively. Tight bounds for GP-order, coil order (Luxburg 1), and Luxburg 2 were proven by Luxburg [17], and tight bounds for Sierpiński–Knopp order were proven by Niedermeier et al. [24], confirming earlier observations on WL_2 [18] (with respect to these measures, the H-order described by Niedermeier et al. is equivalent to Sierpiński–Knopp order and to Cesàro's variant of the Von Koch curve as mentioned by Mandelbrot [18]). Our computations confirmed all of these results. The other bounds have been computed by us. The bounds on the W-measures are the average of lower and upper bounds which have a gap of at most 0.0005; all printed numbers are less than 0.001 off from the real values. Only the bounds for the Gosper flowsnake are less precise (this order involves rotations by angles of $\arctan \frac{1}{5}\sqrt{3}$, which makes it more challenging to get bounds with high precision).

The bounds on the A-measures result from our experiments with random subdivisions of curves. We omit the results for AD_1, because for all curves we found that $2\mathrm{AD}_\infty \leqslant \mathrm{AD}_1$. This means that the best total L_1-diameter is obtained by rotating the curve by 45 degrees, so that the square total L_1-diameter becomes twice the original square total L_∞-diameter. The Sierpiński–Knopp order was the only one not affected by the rotation.

Regarding worst-case locality in the L_∞- and L_2-metrics, we see that the best order in Wunderlich's scheme had not yet been found: Meurthe (Serpentine 110 110 110) turns out to have even better locality in these measures than Luxburg's second variant (Serpentine 101 010 101). Even better locality is achieved by Wierum's $\beta\Omega$-curve (matching or improving on Hilbert's curve in all measures) and still better by our new Peano variant: balanced GP. The latter approaches the locality of the Sierpiński–Knopp order, which is still conjectured to be optimal.

However, it appears that the optimal locality of the Sierpiński–Knopp order comes at a price: it results in high worst-case bounding box measures, and in our experiments on random subdivisions the resulting bounding boxes are about 25% worse than with most other orders. Only Z-order, which tends to result in bounding boxes twice as big as with the other orders, performs worse. The best performer on the worst-case bounding box measures is our balanced GP-order, which also

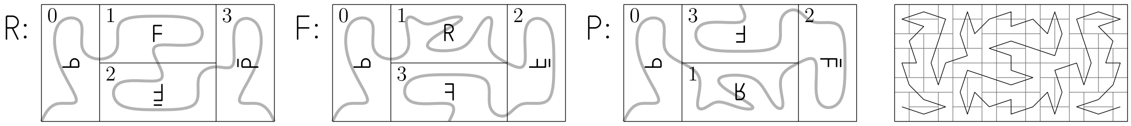


Fig. 10. Example of a scanning order based on rectangles, but not in a standard grid pattern. The locality and bounding-box quality of this particular scanning order are not very interesting: $\mathrm{WL}_\infty = 6.000$, $\mathrm{WL}_2 = 6.667$, $\mathrm{WL}_1 = 12.656$, $\mathrm{WBA} = 3.125$, $\mathrm{WBP} = 3.164$.

performs well in the experiments on random subdivisions (similar to Hilbert), but coil order, Meurthe order and $\beta\Omega$-order perform even better in the experiments.

We also computed worst-case and average total bounding box measures for rectangular bounding boxes with edges at 45 degrees' angles with the coordinate axes. This was (very) harmful with all scanning orders except Sierpiński–Knopp order, where it had no effect (this was to be expected, because the definition of Sierpiński–Knopp order involves rotations by all multiples of 45 degrees). The Sierpiński–Knopp order benefits more than any other from using octagonal bounding boxes instead of rectangular bounding boxes, as we can see in the right columns: here Sierpiński–Knopp order is the best performer. However, all things considered the advantages of bounding octagons may not be worth the effort. Such octagonal bounding boxes need twice the description size of rectangular bounding boxes, but the savings are small: in total bounding octagons constructed by Sierpiński–Knopp order are only 13% smaller than bounding rectangles constructed by Hilbert or GP-order.

7. Conclusions

Pairwise worst-case measures Known locality measures of space-filling curves do not predict well how effective they are when used to group points into bounding boxes. Therefore we proposed new measures of bounding-box quality of space-filling curves. We presented new scanning orders that perform well on these measures, most notably the *balanced GP-order*, which has worst-case bounding box area ratio (WBA) 2.000, and worst-case bounding box square perimeter ratio (WBP) 2.155. On worst-case locality measures this curve also scores very well, much better than Peano's original curve, and beaten only by Sierpiński–Knopp order.

We conjecture that a worst-case bounding box area ratio (WBA) of 2 is in fact optimal and cannot be improved by any (recursively defined) space-filling curve. More provocatively we conjecture that the optimal worst-case bounding box square perimeter ratio (WBP) is also 2 (note that we have not actually found a curve with WBP less than 2.155). We add these conjectures to those by Niedermeier et al., who conjectured that the optimal WL_∞, WL_2, and WL_1 locality values are 4, 4, and 8, respectively (Niedermeier et al. posed this conjecture for curves filling a square, but we would like to drop this restriction).

For proving the lower bounds in these conjectures we have come a long way. Niedermeier et al. proved tight lower bounds on the worst-case locality for a certain class of space-filling curves. Unfortunately, strictly speaking almost none of the space-filling curves in our study belongs to that class. For L_2-locality and for worst-case bounding-box area and squared perimeter, we managed to prove the conjectured lower bounds for another class of space-filling curves, partly overlapping with the class covered by Niedermeier et al., and now including almost all space-filling curves mentioned in this paper.

Still we have not been able to prove these lower bounds for *all* space-filling curves. Our proof is restricted to scanning orders based on axis-aligned rectangles in a regular grid pattern and to scanning orders based on triangles. In this paper we mentioned the Gosper flowsnake curve, which is based on a subdivision of a unit region into subregions that all have fractalic boundaries, and is therefore not included in the class of scanning orders to which our lower bounds apply. Of course one may argue that from a practical point of view it is questionable if one would like to use such a curve anyway, but we can also come up with scanning orders based on tiling the plane with rectangles—but not in a standard grid pattern—or L-shapes, for example (see Fig. 10). Our bounds do not apply to such curves, and the bounds by Niedermeier et al. do not necessarily apply either.

Regarding lower bounds on L_1-locality we did not make any progress on Niedermeier et al.

Performance in practice(?) Our experiments on random points may give an impression of how effective the different curves would be in the application considered in this paper: a data structure for points in the plane, based on sorting the points into blocks of points that are consecutive along the curve. We see that it would be clearly suboptimal to use the order with the best $\mathrm{WL}_\infty, \mathrm{WL}_2$ and WL_1 locality (Sierpiński–Knopp order) for this application. It seems to be better indeed to choose a curve based on WBA and WBP (balanced GP-order).

Still the WBA and WBP measures do not predict performance on random points perfectly either: there are several curves with only moderate WBA and WBP values that seem to be as effective as the balanced GP-order (for example Hilbert order) or even slightly better (for example coil order or $\beta\Omega$-order) on random point data. It should also be noted that total bounding-box area and perimeter may not be the only factors that determine the performance of a curve in a data structure setting. Asano et al. [4] argued that in certain settings it would be good if any axis-parallel square query window can be covered by a small number of contiguous sections of the curve that together cover an area of at most a constant times that of the query window. It would be interesting to have an algorithm that can analyse a given curve automatically to compute the constants involved, in the worst case and on average.

Higher dimensions For what the WBA and WBP measures are worth, the conjectured near-optimality of the balanced GP-order suggests that there is little room for hope to find significantly more effective scanning orders in two dimensions. A first topic for further research is to determine the gap between our lower bound constructions and the performance of known space-filling curves when we consider generalisations to three dimensions. Gotsman and Lindenbaum [10] and Niedermeier et al. [24] have some results on locality, but the gap is large and the field is still wide open, especially with respect to bounding box quality.

Still higher dimensions can be interesting; four-dimensional space-filling curves can be particularly interesting to order rectangles in the plane (which are specified by four coordinates each) [3,11,15]. A first challenge in that context is to define appropriate quality measures that say something sensible about the quality of bounding boxes in two dimensions that are formed by grouping points in four dimensions.

References

[1] J. Alber, R. Niedermeier, On multidimensional curves with Hilbert property, Theory of Computing Systems 33 (4) (2000) 295–312.

[2] J. Akiyama, H. Fukuda, H. Ito, G. Nakamura, Infinite series of generalized Gosper space filling curves, in: China–Japan Conf. on Discrete Geometry, Combinatorics and Graph Theory 2005, LNCS, vol. 4381, 2007, pp 1–9.

[3] L. Arge, M. de Berg, H.J. Haverkort, K. Yi, The priority R-tree: A practically efficient and worst-case optimal R-tree, ACM Transactions on Algorithms 4 (1) (2008) 9.

[4] T. Asano, D. Ranjan, T. Roos, E. Welzl, P. Widmayer, Space-filling curves and their use in the design of geometric data structures, Theoretical Computer Science 181 (1) (1997) 3–15.

[5] K.E. Bauman, The dilation factor of the Peano–Hilbert curve, Math. Notes 80 (5) (2006) 609–620.

[6] G. Chochia, M. Cole, T. Heywood, Implementing the hierarchical PRAM on the 2D mesh: Analyses and experiments, in: Symp. on Parallel and Distributed Processing, 1995, pp. 587–595.

[7] C. Faloutsos, S. Roseman, Fractals for secondary key retrieval, in: Symp. on Principles of Database Systems, 1989, pp. 247–252.

[8] P. Fishburn, P. Tetali, P. Winkler, Optimal linear arrangement of a rectangular grid, Discrete Math. 213 (2000) 123–139.

[9] M. Gardner, Mathematical games—In which “monster” curves force redefinition of the word “curve”, Scientific American 235 (6) (1976) 124–133.

[10] C. Gotsman, M. Lindenbaum, On the metric properties of discrete space-filling curves, IEEE Trans. Image Processing 5 (5) (1996) 794–797.

[11] H. Haverkort, F. van Walderveen, Four-dimensional Hilbert curves for R-trees, in: Proc. 11th Workshop on Algorithm Engineering and Experiments (ALENEX), New York, 2009, pp. 63–73.

[12] D. Hilbert, Über die stetige Abbildung einer Linie auf ein Flächenstück, Math. Ann. 38 (3) (1891) 459–460.

[13] J. Hungershöfer, J.-M. Wierum, On the quality of partitions based on space-filling curves, in: Int. Conf. on Computational Science 2002, LNCS, vol. 2331, 2002, pp. 36–45.

[14] H.V. Jagadish, Linear clustering of objects with multiple attributes, in: ACM SIGMOD Conf. on Management of Data, 1990, pp. 332–342.

[15] I. Kamel, C. Faloutsos, On packing R-trees, in: Conf. on Information and Knowledge Management, 1993, pp. 490–499.

[16] H.L. Lebesgue, Leçons sur l'intégration et la recherche des fonctions primitives, Gauthier-Villars, 1904, pp. 44–45.

[17] U. von Luxburg, Lokalitätsmaße von Peanokurven, Student project report, Universität Tübingen, Wilhelm-Schickard-Institut für Informatik, 1998.

[18] B.B. Mandelbrot, The Fractal Geometry of Nature, W.H. Freeman and Company, 1982, p. 65.

[19] Y. Manolopoulos, A. Nanopoulos, A.N. Papadopoulos, Y. Theodoridis, R-trees: Theory and Applications, Springer, 2005.

[20] G. Mitchison, R. Durbin, Optimal numberings of an $N \times N$ array, SIAM J. Alg. Disc. Meth. 7 (4) (1986) 571–582.

[21] B.K. Moon, H.V. Jagadish, C. Faloutsos, J.H. Saltz, Analysis of the clustering properties of the Hilbert space-filling curve, IEEE Trans. on Knowledge and Data Engineering 13 (1) (2001) 124–141.

[22] E.H. Moore, On certain crinkly curves, Trans. Amer. Math. Soc. 1 (1900) 72–90.

[23] G.M. Morton, A computer oriented geodetic data base and a new technique in file sequencing, Technical report, IBM, Ottawa, 1966.

[24] R. Niedermeier, K. Reinhardt, P. Sanders, Towards optimal locality in mesh-indexings, Discrete Applied Mathematics 117 (2002) 211–237.

[25] R. Niedermeier, P. Sanders, On the Manhattan-distance between points on space-filling mesh-indexings, Technical Report IB 18/96, Karlsruhe University, Dept. of Computer Science, 1996.

[26] G. Peano, Sur une courbe, qui remplit toute une aire plane, Math. Ann. 36 (1) (1890) 157–160.

[27] H. Sagan, Space-Filling Curves, Universitext Series, Springer, 1994.

[28] J.-M. Wierum, Definition of a new circular space-filling curve: $\beta\Omega$-indexing, Technical Report TR-001-02, Paderborn Center for Parallel Computing (PC^2), 2002.

[29] J.-M. Wierum, Logarithmic path-length in space-filling curves, in: Canadian Conference on Computational Geometry, 2002, pp. 22–26.

[30] W. Wunderlich, Über Peano-Kurven, Elemente der Mathematik 28 (1) (1973) 1–10.