Consistent Digital Rays

Jinhee Chun[*] Matias Korman[*] Martin Nöllenburg[†]
Takeshi Tokuyama[*]

Abstract

Given a fixed origin o in the d-dimensional grid, we give a novel definition of *digital rays* $\mathrm{dig}(op)$ from o to each grid point p. Each digital ray $\mathrm{dig}(op)$ approximates the Euclidean line segment $\overline{op}$ between o and p. The set of all digital rays satisfies a set of axioms analogous to the Euclidean axioms. We measure the approximation quality by the maximum Hausdorff distance between a digital ray and its Euclidean counterpart and establish an asymptotically tight $\Theta(\log n)$ bound in the $n \times n$ grid. The proof of the bound is based on discrepancy theory and a simple construction algorithm. Without a monotonicity property for digital rays the bound is improved to $O(1)$. Digital rays enable us to define the family of digital star-shaped regions centered at o which we use to design efficient algorithms for image processing problems.

1 Introduction

The *digital line segment* $\mathrm{dig}(pq)$ between two grid points p and q is a fundamental digital geometric object, but still its definition is not that obvious. Indeed, the digital representation of line segments has been an active subject of research for almost half a century now (see, for example, the excellent survey of Klette and Rosenfeld [9]). In digital geometry, a geometric object is represented by a set of d-dimensional grid points in a digital grid $\mathbf{G} = \mathbb{Z}^d$ and its topological properties are considered under a grid topology defined by a graph on the grid. In two dimensions, it is common to consider the *orthogonal* (or *4-neighbor*) grid topology, where each point $p = (x, y)$ is connected to its four vertical and horizontal neighbors $(x, y \pm 1)$ and $(x \pm 1, y)$, and we focus on this topology; as a variant, however, we may consider the *octagonal* (or *8-neighbor*) grid topology that connects each grid point $p = (x, y)$ to its 4-neighbors and additionally to its diagonal neighbors $(x + 1, y \pm 1)$ and $(x - 1, y \pm 1)$. Given a grid topology, the digital line segment $\mathrm{dig}(pq)$ between two grid points p and q is a path between p and q in this topology.

Since a digital line segment $\mathrm{dig}(pq)$ is a representation of a line segment $\overline{pq}$ in Euclidean geometry, it is natural (at least from a mathematical perspective) to set up the following axioms that a digital line segment should satisfy:

[*] Graduate School of Information Sciences, Tohoku University, Japan. `(jinhee,mati,tokuyama)@dais.is.tohoku.ac.jp`

[†] Faculty of Informatics, Karlsruhe University and Karlsruhe Institute of Technology (KIT), Germany. `noellenburg@iti.uka.de`

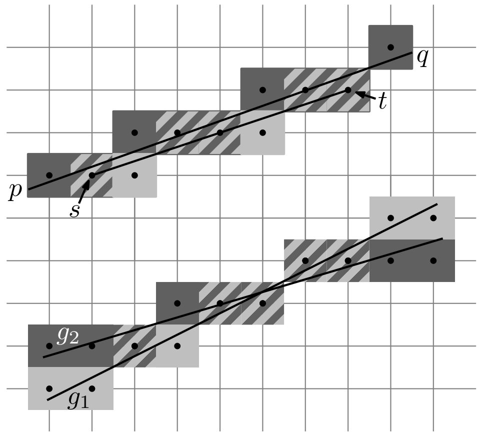


Figure 1: Euclidean line segments and their DSSs. Intersections are indicated by bicolored pixels. Axiom (S3) is violated since $s, t \in \text{dig}(pq)$ but $\text{dig}(st) \nsubseteq \text{dig}(pq)$ (top); the intersection of the DSSs g_1 and g_2 is not connected (bottom).

(S1) A digital line segment $\text{dig}(pq)$ is a connected path between p and q under the grid topology.

(S2) For any two grid points p and q there is a unique digital line segment $\text{dig}(pq) = \text{dig}(qp)$.

(S3) For a digital line segment $\text{dig}(pq)$ and two grid points $s, t \in \text{dig}(pq)$, it holds that $\text{dig}(st) \subseteq \text{dig}(pq)$.

(S4) For any two grid points p and q there is a grid point $r \notin \text{dig}(pq)$ such that $\text{dig}(pq) \subset \text{dig}(pr)$.

Note that axiom (S3) implies that a non-empty intersection of two digital line segments is either a grid point or a digital line segment. Axiom (S4) implies that any digital line segment can be extended to a digital line. We often identify a path in a grid with its vertex set if the correspondence is clear. Accordingly, if we say a grid point p is in a path P, it means that p is a vertex of P.

Unfortunately, popular definitions of two-dimensional (2D) digital line segments in computer vision do not satisfy these axioms. For example, in the standard definition of a *digital straight segment (DSS)* [9], a digital line segment (in the octagonal topology) that corresponds to the Euclidean line segment given by $y = mx + b$, $x_0 \leq x \leq x_1$ is defined as the set of grid points $\{(i, \lfloor mi + b + 0.5 \rfloor) \mid x_0 \leq i \leq x_1\}$ for $|m| \leq 1$. Using this definition connectivity of segments is certified only in the 8-neighbor topology. Moreover, the intersection of two DSSs is not always connected, and axiom (S3) is violated in some cases as depicted in Figure 1.

In the 2D grid, another possibility to define digital line segments would be to use the system of L- and Γ-shaped shortest paths. An L- or Γ-shaped path between two points $p = (x_p, y_p)$ and $q = (x_q, y_q)$ such that $x_p \leq x_q$, is the (at most) 2-link path that consists of the grid points on the vertical segment pp' and on the horizontal segment $p'q$ where $p' = (x_p, y_q)$. It is easy to confirm that the system of these paths satisfies axioms (S1)–(S4) for digital line segments.

A clear drawback is that an L-shaped path is visually very different from the Euclidean line segment, and the Hausdorff distance from $\overline{pq}$ to the L-shaped path between p and q becomes $n/\sqrt{2}$ for $p = (0, n)$ and $q = (n, 0)$. If, on the other hand, one accepts to use a non-planar graph structure to define the topology on the grid points, Pach, Pollack, and Spencer [12] show that the shortest-path distance (using Euclidean distance for the edge lengths) in the grid topology given by a suitable sparse graph is at most $(1 + \epsilon)$ times the Euclidean distance. Accordingly, the polygonal path consisting of the edge set of the shortest path between p and q in the graph gives a nice approximation of the line segment $\overline{pq}$. However, the graph structure is a union of many randomly chosen lattice structures on the grid points using long edges with a variety of slopes; thus, the vertex set of the polygonal path is too sparse for direct use as a digital line segment. Also, the method does not guarantee an $o(n)$ bound for the Hausdorff distance.

Thus, it seems that there is a trade-off between the axiomatic requirements and the visual quality of digital line segments. It is a challenging problem to find a system of digital line segments that satisfies the axioms and is visually alike Euclidean line segments at the same time.

In this paper we study a less ambitious but important subproblem, motivated by geometric optimization applications: we consider only digital line segments that have the origin o as one of their endpoints. In other words, we consider digital halflines emanating from o. Then $\mathrm{dig}(op)$ is defined as the unique portion of the halfline that has p as its second endpoint. We call such segments *digital ray segments* or simply *digital rays* emanating from o.

For digital rays, the axioms (S1)–(S4) for digital line segments are adapted as follows:

(R1) A digital ray $\mathrm{dig}(op)$ is a connected path between o and p under the grid topology.

(R2) There is a unique digital ray $\mathrm{dig}(op)$ between o and any grid point p.

(R3) For a digital ray $\mathrm{dig}(op)$ and a grid point $r \in \mathrm{dig}(op)$, it holds that $\mathrm{dig}(or) \subseteq \mathrm{dig}(op)$.

(R4) For any grid point p there is a grid point $r \notin \mathrm{dig}(op)$ such that $\mathrm{dig}(op) \subset \mathrm{dig}(or)$.

We also give an additional *monotonicity* axiom, which is not combinatorial, but a reasonable condition for a digital ray:

(R5) For any $r \in \mathrm{dig}(op)$, $|\overline{or}| \leq |\overline{op}|$, where $|\overline{ab}|$ is the length of the Euclidean segment $\overline{ab}$.

Note that in the orthogonal grid topology the monotonicity axiom (R5) implies that any ray $\mathrm{dig}(op)$ is a shortest path in the grid.

A system of digital rays is called *consistent* if it satisfies axioms (R1)–(R5). From these axioms, it follows that the union of all digital rays forms an infinite spanning tree T of the grid graph on $\mathbf{G}$ rooted at o, such that $\mathrm{dig}(op)$ is the unique path between o and p in the tree. Because of axiom (R4), T cannot have leaves. Thus, the problem is basically to embed the infinite "star" consisting of the halflines emanating from o in the d-dimensional Euclidean space as a tree in the d-dimensional grid. Although embedding a tree in a grid is a popular topic in metric embedding and graph drawing, it is a novel and interesting problem to geometrically approximate ray segments by paths.

Main result. The main result of the paper is the asymptotically tight $\Theta(\log n)$ bound for the maximum Hausdorff distance between $\mathrm{dig}(op)$ and $\overline{op}$ among all points p in an $n \times n$ grid. The lower bound argument is based on discrepancy theory, and the upper bound is attained by a simple and systematic construction of a tree T that is extended to the d-dimensional case. Surprisingly, if we do not include the monotonicity axiom (R5), the bound can be reduced to $O(1)$.

2 Motivation and related work

Our motivation comes from handling digital analogues of star-shaped regions for optimization problems in a pixel grid. A square pixel grid is a subdivision of an $n \times n$ square region into $N = n^2$ unit squares called *pixels*. We have a canonical one-to-one correspondence between pixels in a pixel grid $\mathbf{P}$ and grid points in our two-dimensional grid $\mathbf{G}$ restricted to an $n \times n$ subgrid. Thus, we can translate the definitions of digital rays and digital star-shaped regions in $\mathbf{G}$ to those in $\mathbf{P}$. A pixel grid image is an assignment of a color to each pixel: A monochromatic image can be considered as a function from the set $\mathbf{P}$ of all pixels to real values in $[0, 1]$ called *gray levels*, while a color image can be considered as a triple of functions from $\mathbf{P}$ to real values in $[0, 1]$ corresponding to the color levels of red, green, and blue. For example, a picture taken with a 1-megapixel digital camera is a color image in a pixel grid of size 1024×1024.

Image segmentation is an important problem in computer vision, which separates an object from the background in the picture. Asano, Chen, Katoh, and Tokuyama [1] formulated the problem as a least-square optimization problem and gave an efficient algorithm if the object is a region bounded by two x-monotone curves. Several improved results such as controlling smoothness of curves and higher dimensional extensions were given by Wu and Chen [19], and the optimal-ratio formulation was given by Wu [18]. It was further pointed out that image segmentation problems appear in medical applications [18]: Tumors can be approximated by a layer of three dimensional star-shaped annuli, where a star-shaped annulus is the set difference of two star-shaped regions with a shared center o. If an image transformed by the central projection from o is given as the input by using a mechanism such as optical coherence tomography, then a star-shaped annulus is transformed to a region bounded by two x-monotone surfaces, which can be naturally digitized. Wu [18] considered the case where such an input is given and applied his algorithm to extract a tumor region from a medical image. A remaining question is how to directly segment a star-shaped annulus from a pixel grid (in two or three dimensions). In other words, how to extract a tumor in a digital image that is not generated/preprocessed by using a central projection method about o.

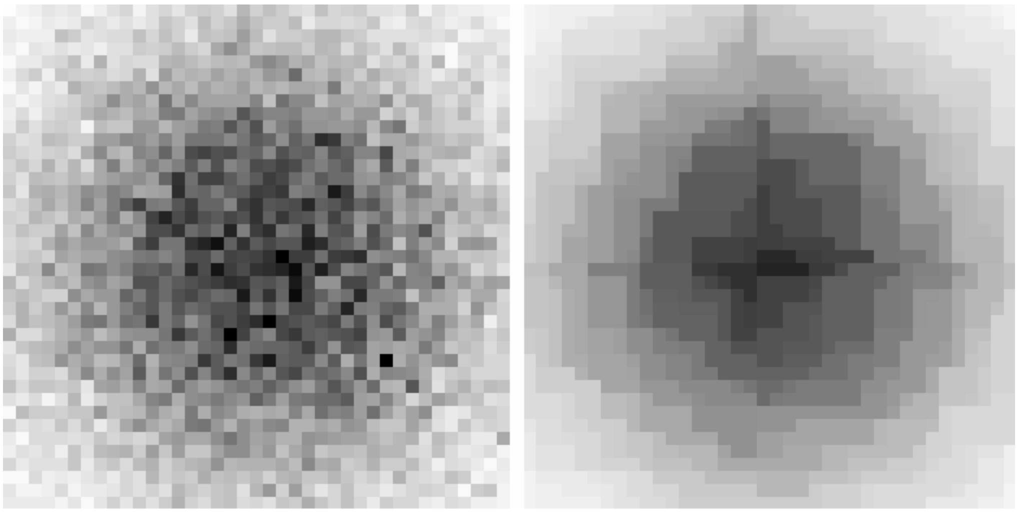


Figure 2: Input terrain (left) and output pyramid (right).

Chen, Chun, Katoh, and Tokuyama [3] and Chun, Sadakane, and Tokuyama [4] considered the *pyramid approximation* problem to compute the least-square approximation of an input digital terrain (given as a function on $\mathbf{P}$) where each horizontal slice (i.e., a region bounded by a contour line) of the output terrain is a special kind of rectilinear convex region as shown in Figure 2, where heights are given by gray-levels. It was desired to solve the analogous *mountain approximation* problem where each horizontal slice is a star shape, since it will be useful in applications to computer vision and geographic data processing.

A natural definition of a digital star-shaped region is the set of all pixels intersecting a given Euclidean star-shaped region. However, such a family of regions does not satisfy the condition that the intersection of two digital star-shaped regions centered at o is again a digital star-shaped region. This causes difficulty for solving the above mentioned problems. We give the following definition of a digital star-shaped region that satisfies the above condition:

Definition 2.1. *Given a system of digital rays from a center o, a region R is a* digital star-shaped region *centered at o if and only if* $\mathrm{dig}(op) \subseteq R$ *for any grid point $p \in R$.*

This definition and theory naturally can be extended to higher dimensional grids. The quality of a digital star-shaped region is assured by the following theorem, which follows immediately from our main results:

Theorem 2.2. *For any Euclidean star-shaped region R with center o in the $n \times n$ pixel grid $\mathbf{P}$, $R' = \bigcup_{p \in \mathbf{P} \cap R} \mathrm{dig}(op)$ is a digital star-shaped region such that the Hausdorff distance $H(R, R')$ between R and R' is $O(\log n)$. Conversely, given any digital star-shaped region Q, let Q' be the union of segments $\overline{ox}$ over all points x in the plane covered by pixels in Q. Then, Q' is a Euclidean star-shaped region such that $H(Q, Q') = O(\log n)$. The $O(\log n)$ bound improves to $O(1)$ if we use a system of digital rays without the monotonicity axiom.*

We can define the inverse digital central projection $\mathbf{D}$ from $\mathbf{P}$ to $\mathbf{P}$ along digital rays, such that a region below an x-monotone curve is canonically mapped to a digital star-shaped region: We use the spanning tree of the grid graph underlying $\mathbf{P}$ that will be given in Section 3.3 (Section 4 for its higher dimensional analogue) to define digital rays. Then a pixel $p = (i, j)$ is mapped to the pixel $\mathbf{D}(p)$ corresponding to the node of depth j on the path in the tree from the origin towards $(i, n - i)$. Thus, we can solve the segmentation problem for star-shaped annuli by using the inverse digital central projection combined with Wu's algorithm [18]. Instead of using $\mathbf{D}$ explicitly, we may also implement the algorithm by using our digital rays directly; we can control smoothness of the contour of the region by using techniques given in [18, 19] (omitted here). Section 5 gives our mountain approximation algorithm.

Relation to digital computational geometry. In computational geometry, the problem of representing geometric objects in digital geometry without causing topological and combinatorial inconsistencies is a major concern, and algorithmic solutions have been considered from the viewpoint of robust finite-precision geometric computation [8, 16].

Suppose that we would like to represent a set S of line segments digitally. Although ideally one would like to give a precisely defined and consistent system of digital line segments, the above mentioned difficulties prevent us from doing so. Rather, it is popular to use a *dynamic* method to digitize the line segments; that is, the digital approximation of a line segment ℓ is affected by the configuration of the other line segments of S. In particular, it is required to construct the arrangement of S in the digital plane without changing the combinatorial structure of the arrangement, while all vertices of the arrangement are located at grid points and each line segment is visually alike the original line segment. It is known that a grid of exponential size is necessary to represent all the combinatorial types of arrangements of n straight lines [6]; hence we need to bend lines if we want to use a polynomial-size grid. In the pioneering paper of Greene and Yao [8] and its following research by Goodrich, Guibas, Hershberger, and Tanenbaum [7], each line segment is represented by a polygonal chain consisting of edges of the arrangement. It is necessary to carefully round each vertex of the arrangement to a grid point in order to avoid combinatorial inconsistencies, and a method named *snap rounding* is proposed. Since no pair of edges of the arrangement intersect each other, we can draw edges by using a popular method like DSS once we have such a representation of the arrangement. We note that the snap rounding idea is important not only in the theory of robust computation but also in the practical design of geometric editors or systems: for example, the Ipe editor [15] is a pioneering system that uses snap rounding; the idea has also been implemented in the CGAL project [2]. This dynamic approach is different from our static approach in which each digital line segment is defined irrespective to the existence of other lines in the arrangement. Still, we think that it is important to investigate how well line segments can be digitized statically and to consider the combination of static and dynamic methods to design efficient systems and algorithms in digital geometry.

3 Digital rays in the plane grid

3.1 Preliminaries

The Hausdorff distance $H(A, B)$ of two objects A and B is defined as $H(A, B) = \max\{h(A, B), h(B, A)\}$, where $h(A, B) = \max_{a \in A} \min_{b \in B} d(a, b)$ and $d(a, b)$ is some distance between the points a and b. Although it is most natural to consider the Euclidean distance for $d(a, b)$, we will use the L_∞-metric in the following for technical convenience. Since the ratio of the Euclidean distance to the L_∞ distance in d-dimensional space is bounded by $1/\sqrt{d}$ and d, the choice of the metric is irrelevant in a constant-dimensional space when considering the bounds in big-O and big-Ω notations. Recall that the monotonicity axiom (R5) has been stated in terms of the Euclidean metric. Note that using the L_∞-metric to measure distances between Euclidean and digital ray segments does not affect the definition of axiom (R5).

Consider the set $V = \{(i, j) \mid i, j \in \mathbb{Z}\}$ of integer grid points. We define a planar graph G on V that represents the adjacency relations of a 4-neighborhood pixel grid. In $G = (V, E)$ each vertex (i, j) is connected to its four neighbors $(i, j-1)$, $(i-1, j)$, $(i+1, j)$, and $(i, j+1)$. This also defines the orthogonal topology of the grid $\mathbf{G}$. A subset of V is *connected* in this topology if its induced subgraph in G is connected.

3.2 The lower bound result

We focus on the part $\mathbf{G}(n)$ of the planar orthogonal grid restricted to the region defined by $x + y \leq n$ in the first quadrant. The remaining quadrants are handled analogously by rotating $\mathbf{G}(n)$ around the origin. From the monotonicity axiom (R5) it follows that $\operatorname{dig}(op) \subset \mathbf{G}(n)$ for any $p \in \mathbf{G}(n)$ and that $\operatorname{dig}(op)$ is a shortest path in the grid. We show that there exists a point $p \in \mathbf{G}(n)$ such that the Hausdorff distance $H(\operatorname{dig}(op), \overline{op})$ is $\Omega(\log n)$. Let T be the spanning tree of $\mathbf{G}(n)$ that is the union of $\operatorname{dig}(op)$ for all $p \in \mathbf{G}(n)$. Figure 3 shows the spanning tree induced by a set of consistent rays.

We use a classical result on pseudo-random number generation [10, 11, 14]. The following historical summary is according to Schmidt's textbook [14]. Consider a sequence $X = (x_0, x_1, x_2, \ldots)$ of real numbers in $[0, 1]$. For any given $a \in [0, 1]$ and $m \in \mathbb{N}$ define $X_m(a) = |\{0 \leq i \leq m \mid x_i \in [0, a]\}|$. The *discrepancy* of the sequence $X_m = (x_0, x_1, \ldots, x_m)$ is defined as $D(X_m) = \sup_{a \in [0,1]} |am - X_m(a)|$. Van der Corput conjectured in 1935 that no sequence X in the unit interval can have bounded discrepancy, that is, $D(X_m) \in \omega(1)$ for infinitely many integers m. This was affirmatively answered by van Aardenne-Ehrenfest in 1945. Roth gave an $\Omega(\sqrt{\log m})$ bound in 1954, and the correct order of magnitude of the discrepancy is $\Theta(\log m)$ given by Schmidt in 1972. We make use of discrepancy theory in the form of Theorem 3.1 below. We remark that a slightly stronger version of the conjecture was given in a list of favorite questions of Erdős [5]: He conjectured that there is an a such that $\max_{m<n} |am - X_m(a)|$ is an unbounded function in n, for which Schmidt's method also gives a $\Theta(\log n)$ bound.

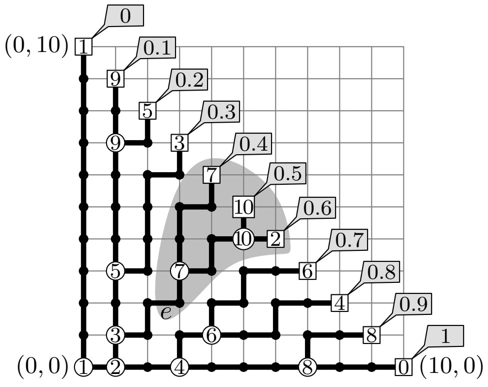


Figure 3: A spanning tree T of $\mathbf{G}(n)$ for $n = 10$. The labels attached to the leaves as callouts are the elements of the low-discrepancy sequence $X(T)$ in the interval $[0, 1]$, while the inner labels are used in the construction of that sequence and determine the order of the elements.

Theorem 3.1 (Schmidt [13]). *Given a sequence $X = (x_0, x_1, x_2, \ldots)$ of real numbers in $[0, 1]$ and a sufficiently large integer n, there are an integer $m < n$ and a real number $a \in [0, 1]$ such that the subsequence $X_m = (x_0, x_1, \ldots, x_m)$ satisfies that $|am - X_m(a)| > c \log n$, where c is a positive constant independent of n.*

We will apply Theorem 3.1 to a sequence constructed from the spanning tree T to obtain a lower bound for the Hausdorff distance between $\mathrm{dig}(op)$ and $\overline{op}$. Let us start with some preparations to construct that sequence.

For $m = 1, 2, \ldots n + 1$, let $L(m) = \{(i, m - 1 - i) \mid i = 0, \ldots, m - 1\}$ be the subset of $\mathbf{G}(n)$ satisfying $x + y = m - 1$.

Lemma 3.2. *For any integer m, $1 \le m \le n$, the spanning tree T has a unique node of degree 3 in $L(m)$.*

Proof. There are m grid points in $L(m)$ and $m + 1$ grid points in $L(m + 1)$. Since T does not have any leaves in $L(m)$ for $m \le n$ and since all paths from the origin o to any tree node in T are shortest paths in the grid $\mathbf{G}$ (due to the monotonicity axiom (R5)), each of the m points of $L(m)$ must be connected to a point in $L(m + 1)$ and, conversely, each of the $m + 1$ points of $L(m + 1)$ must be connected to a point in $L(m)$. In the 4-neighbor grid topology, this is only possible if there are exactly one node of degree 3 and $m - 1$ nodes of degree 2 in $L(m)$. □

We denote this unique degree-3 node in $L(m)$ as the *branching node* of $L(m)$. There is a vertical and a horizontal edge incident to the branching node that lead to its two children in $L(m+1)$. We denote these two edges as the *branching edges* of $L(m)$.

We associate the number j/n to the leaf $(j, n - j) \in L(n + 1)$ in order to obtain the set $N = \{j/n : j = 0, 1, 2, \ldots n\} \subset [0, 1]$ of leaf labels, see Figure 3. For an edge $e = uv$ in T, where u is the parent of v, we define the *subtree rooted* *at* e to be the subtree of T rooted at the child node v of e. Then for each edge e of T in $\mathbf{G}(n)$, the set of vertices of $L(n+1)$ in the subtree rooted at e are consecutive and their associated numbers form an interval $I(e) \subset N$. Let $x(e)$ denote the largest element in $I(e)$. An example for an edge e is given in Figure 3, where $I(e) = \{0.4, 0.5, 0.6\}$ and $x(e) = 0.6$.

We create a sequence $X(T) \subset [0,1]$ as follows: We set $x_0 = 1$, and for $m = 1, \ldots, n$ we set $x_m = x(e_m)$, where e_m is the upper (vertical) branching edge in $L(m)$. Note that for any two vertical branching edges e and e' the numbers $x(e)$ and $x(e')$ differ since the path from e to the leaf with the largest associated value in $I(e)$ always uses the horizontal branching edge at each encountered branching node. Thus the obtained sequence $X(T) = (x_0, x_1, \ldots, x_n)$, is a permutation of N that depends only on T. For example, the tree T in Figure 3 creates the sequence $X(T) = (1, 0, 0.6, 0.3, 0.8, 0.2, 0.7, 0.4, 0.9, 0.1, 0.5)$. The labels inside the nodes in Figure 3 show the correspondence between the unique internal branching node in $L(i)$ and the leaf located at $(nx_i, n - nx_i)$ in $L(n+1)$ that is associated with the number x_i shown in the callouts. For each $i = 1, \ldots, n$ the corresponding nodes are labeled by i. In other words, each branching node and the rightmost leaf in the subtree rooted at the upper branching edge of that node have the same label.

Let $E(m)$ be the set of edges in T going from $L(m)$ towards $L(m+1)$. We get the following two lemmas.

Lemma 3.3. *Let e and f be edges in $E(m)$. If e is to the left of f (that is, the endpoint of e in $L(m+1)$ has smaller x-coordinate than the endpoint of f in $L(m+1)$), we have $x(e) < x(f)$.*

Proof. Assume to the contrary that $x(e) \geq x(f)$. The case $x(e) = x(f)$ contradicts the fact that T is a tree since we would have two different paths from the root to the same leaf. In the case $x(e) > x(f)$ the paths from e and f to their largest leaves must cross. But since the grid topology allows only horizontal and vertical edges between adjacent grid points the paths must cross in a common tree node, which again contradicts the fact that T is a tree. □

Lemma 3.4. *The set $\{x(e) : e \in E(m)\}$ equals the set $\{x_0, x_1, x_2, \ldots, x_m\}$.*

Proof. A simple induction shows the lemma. First, let $m = 1$ and thus in all valid spanning trees $E(m)$ consists of the vertical and horizontal edge leaving the origin. We denote these edges by e_v and e_h, respectively. By definition we have $x_1 = x(e_v)$. For e_h, $x(e_h) = 1$ is the largest element (attached to the leaf $(n, 0)$) in $I(e_h)$. Hence we indeed have $x(e_h) = x_0$.

Now assume that the statement holds for some m. All edges in the set $E(m+1)$, except the two branching edges, just continue the corresponding predecessor edge in $E(m)$ and thus have the same x-values as their predecessor edges. It remains to consider the branching node u in $L(m+1)$. Let e be the edge incident to u in $E(m)$ and e_v and e_h the vertical and the horizontal branching edge incident to u in $E(m+1)$. By definition $x_{m+1} = x(e_v)$; furthermore, we have $x(e) = x(e_h)$ since $I(e_h) \subset I(e)$ contains (by Lemma 3.3) the largest element $x(e)$ of $I(e)$. Thus the statement also holds for $m+1$. □

The following theorem shows our lower bound.

Theorem 3.5. *For any spanning tree T there is a grid point $p \in L(n+1)$ and q in $\mathbf{G}(n)$ such that q is on the path* $\mathrm{dig}(op)$ *in T and the L_∞-distance from q to the line segment $\overline{op}$ exceeds $c \log n - 1$, where c is the constant considered in Theorem 3.1.*

Proof. To prove the theorem we consider the discrepancy of the sequence $X(T)$. From Theorem 3.1, we have a real number $0 \leq a \leq 1$ and two integers $m < n$ for n large enough such that $|am - X_m(a)| > c \log n$. The following two cases should be considered:

Case 1: $X_m(a) > am + c \log n$. Consider the node q in $L(m+1)$ located at the grid point $(X_m(a) - 1, m - (X_m(a) - 1))$, and let e be the edge between q and its parent in T. By definition, q is on the path $\mathrm{dig}(op)$ from o to the node $p = (x(e)n, n - x(e)n) \in L(n+1)$. Because of the definition of $X_m(a)$ and Lemma 3.4, we have exactly $X_m(a)$ edges $f \in E(m)$ for which $x(f) \leq a$. However, there are also exactly $X_m(a)$ edges of $E(m)$ to the left of e, including e itself, since q is the $X_m(a)$-th node in $L(m+1)$ counted from the left. Lemma 3.3 implies that no edge g to the right of e can attain $x(g) \leq a$. Thus, e itself must satisfy $x(e) \leq a$. Now, consider the L_∞ distance of the line segment $\overline{op}$ and q. The line segment $\overline{op}$ goes through $(x(e)m, m - x(e)m)$, which is the L_∞-nearest point from q on $\overline{op}$. The L_∞ distance is $(X_m(a) - 1 - x(e)m) \geq (X_m(a) - 1 - am) > c \log n - 1$.

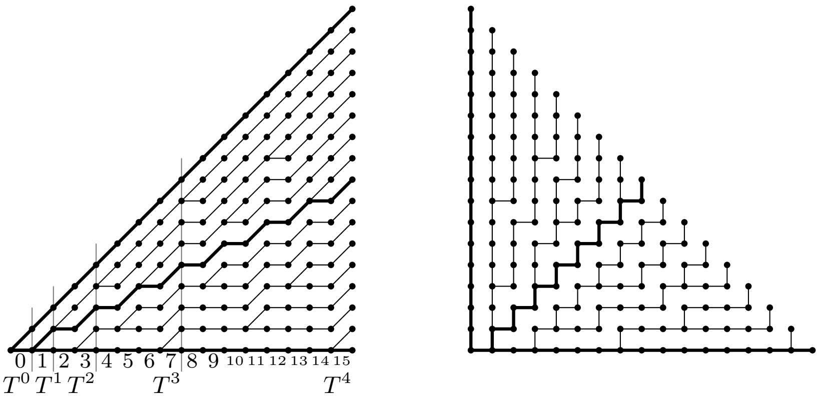
Case 2: $X_m(a) < am - c \log n$. Consider the node q in $L(m+1)$ located at the grid point $(X_m(a), m - X_m(a))$ and the edge e between q and its parent. Since there are only $X_m(a)$ edges $f \in E(m)$ for which $x(f) \leq a$ we have $x(e) > a$ (again, from Lemma 3.3). Node q is on the path $\mathrm{dig}(op)$ to the node $p = (x(e)n, n - x(e)n)$. Similarly to Case 1, we can show that the L_∞-distance from q to $\overline{op}$ is greater than $c \log n$. This proves the theorem. □

3.3 The upper bound results

We deterministically construct a 2D spanning tree DT(2) of G (which we generalize to a d-dimensional tree DT(d) in Section 4), such that for every $p = (i, j) \in V$, the unique path from o to p in DT(2) defines the digital ray $\mathrm{dig}(op)$ that represents the Euclidean line segment $\overline{op}$. By the monotonicity axiom (R5), $\mathrm{dig}(op)$ is always a shortest path in the orthogonal grid.

We give the construction of DT(2) restricted to $\mathbf{G}(n)$ for $n = 2^k$. By creating rotated copies in the other quadrants and extending them to the infinite grid we get DT(2). To simplify the description (especially, when we generalize to higher dimensions later), we transform the grid by a linear map Φ that maps the lattice base $\{(1,0),(0,1)\}$ to $\{(1,0),(1,1)\}$, respectively. The linear map Φ transforms the quadrant containing $\mathbf{G}(n)$ to the the first octant and maps $\mathbf{G}(n)$ to a skew-grid with the base $\{(1,0),(1,1)\}$ in the triangular region defined by $0 \leq y \leq x \leq n$. The set $L(m)$ is mapped to the m-th column of the transformed grid. Figure 4 shows the tree T that we will construct in the skew grid as well as the corresponding tree $\Phi^{-1}(T)$ in $\mathbf{G}(n)$.

In the transformed grid $\Phi(\mathbf{G}(n))$, all edges are horizontal or diagonal with positive unit slope. An edge connecting a vertex (i, j) and a vertex $(i+1, j)$ or



(a) (b)

Figure 4: The spanning tree $T = T^4$ (a) and the corresponding tree $\Phi^{-1}(T)$ in $\mathbf{G}(n)$ (b). The center path and the two boundary paths are highlighted in bold.

$(i+1, j+1)$ is called an edge in the i-th edge-column. The i-th edge-column is called an even (odd) edge-column if i is even (odd). Note that the column index starts from 0.

Since the infinite tree DT(2) cannot have leaves, the set of leaves of T restricted to $\Phi(\mathbf{G}(n))$ must be the right endpoints of the edges in the rightmost edge-column, that is, the set $\{(n, b) \mid b = 0, 1, 2, \ldots, n\}$. Any such spanning tree, and thus also the one we will construct, must satisfy the following lemma.

Lemma 3.6. *If an edge $e \in T$ is horizontal (resp. diagonal), all the edges in T in the same edge-column below e (resp. above e) must be horizontal (resp. diagonal).*

Proof. If e is horizontal and there is a diagonal edge below e then two edges in that column must share their right endpoint by the pigeon hole principle. This creates a cycle in T, which contradicts the fact that T is a tree. If e is diagonal a similar argument holds. □

This lemma implies that there is not much freedom for defining T, and it is also a crucial observation for generalizing the construction to higher dimensions.

We give a procedure to construct all paths from the root to the leaves of T. This suffices to define T. For sake of convenience, we denote the spanning tree restricted to the subgrid $\Phi(\mathbf{G}(2^k))$ by T^k. We have two *boundary paths*: The path towards $(2^k, 0)$ uses only horizontal edges, and the path towards $(2^k, 2^k)$ uses only diagonal edges. These are the only paths for $k = 0$ and uniquely define T^0. If $k \geq 1$, we first give the path towards $(2^k, 2^{k-1})$, which we call the *center path* (see Figure 4). The center path is the alternating chain of horizontal and diagonal edges, starting with the horizontal edge connecting the origin $o = (0, 0)$ and $(1, 0)$. Thus, the center path has a horizontal edge in every even column and a diagonal one in every odd column. We observe that the left endpoint of an edge of the center path in an even column is on the diagonal line $y = x/2$, while its right endpoint is below this line. The following lemma is a straightforward consequence of Lemma 3.6:

Lemma 3.7. *In the tree T^k, all the edges in an even column below the center path are horizontal and all the edges in an odd column above the center path are diagonal.*

Let us first consider the part of T^k below and including the center path. The even columns are determined by Lemma 3.7 and consist of horizontal edges only. The number of edges between the upper and lower boundary paths in the i-th column of $\Phi(\mathbf{G}(2^{k-1}))$ equals the number of edges between the center path and the lower boundary path in the $(2i+1)$-th column of $\Phi(\mathbf{G}(2^k))$. So we can simply copy the i-th column of T^{k-1} to the lower half of the $(2i+1)$-th column of T^k. Similarly, we know the odd columns of the part of T^k above the center path and fill the even columns by copying the i-th column of T^{k-1} to the upper half of the $2i$-th column for $i = 0, 1, \dots, 2^{k-1} - 1$. These copies do not conflict with the boundary paths and the center path of T^k.

This recursively constructs the tree T^k for $k \in \mathbb{N}$, and we can generate a spanning tree T of the first octant of the whole infinite grid such that T^k is the restriction of T to $\Phi(\mathbf{G}(2^k))$. Our tree in the orthogonal grid $\mathbf{G}(2^k)$ is $\Phi^{-1}(T^k)$, which we can obviously extend to DT(2), the tree on the whole orthogonal grid $\mathbf{G}$.

Theorem 3.8. *The set of digital rays defined by* DT(2) *is consistent. For any grid point $p \in \mathbf{G}(n)$, the L_∞-Hausdorff distance between* $\mathrm{dig}(op)$ *and $\overline{op}$ is less than $1 + \log n$.*

Proof. It is easy to verify that the set of digital rays defined by DT(2) is consistent, that is, it satisfies axioms (R1)–(R5). It remains to bound the distance between $\mathrm{dig}(op)$ and $\overline{op}$. Let $p = (x_p, y_p)$ be any vertex in $T = T^k$, and let $q = (x_q, y_q)$ be any vertex on $\mathrm{dig}(op)$, the path from o to p in T. We claim that the vertical distance between $\overline{op}$ and q is at most k. The proof is by induction on k. If $k \leq 1$, the claim is trivial. Thus, assume that the claim holds for T_{k-1}. We can further assume that $x_q \leq x_p - 2$, as we can check the claim directly otherwise.

If $\mathrm{dig}(op)$ is the center path, the claim holds by construction of the center path. Thus we assume this is not the case. Since two paths in T cannot cross each other, both p and q must be on the same side of the center path. We distinguish the following two cases:

Case 1. If $p = (x_p, y_p)$ is below the center path (that is, $y_p < \lfloor x_p/2 \rfloor$), then $q = (x_q, y_q)$ satisfies that $y_q \leq \lfloor x_q/2 \rfloor$. From the recursive definition of T we know that the odd columns below the center path are copied from T^{k-1} and the even columns contain only horizontal edges. Thus p is a copy of $p' = (\lfloor x_p/2 \rfloor, y_p)$ and q is a copy of $q' = (\lfloor x_q/2 \rfloor, y_q)$.

Since the claim holds for T^{k-1}, the vertical distance from q' to the line segment $\overline{op'}$ is at most $k-1$, that is,

$$d_y(q', \overline{op'}) = |y_q - y_p(\lfloor x_q/2 \rfloor)/(\lfloor x_p/2 \rfloor)| \leq k - 1.$$

Now, consider the vertical distance $d_y(q, \overline{op}) = |y_q - y_p x_q/x_p|$ from q to $\overline{op}$. We have the following inequality

$$\left| y_p \frac{\lfloor x_q/2 \rfloor}{\lfloor x_p/2 \rfloor} - y_p \frac{x_q}{x_p} \right| \leq y_p \left| \frac{x_q+1}{x_p-1} - \frac{x_q}{x_p} \right| = y_p \left| \frac{1}{x_p} + \frac{x_q+1}{(x_p-1)x_p} \right| \leq y_p \left| \frac{2}{x_p} \right| < 1 \tag{1}$$

and thus

$$d_y(q, \overline{op}) \leq \left| y_q - y_p \frac{\lfloor x_q/2 \rfloor}{\lfloor x_p/2 \rfloor} \right| + \left| y_p \frac{\lfloor x_q/2 \rfloor}{\lfloor x_p/2 \rfloor} - y_p \frac{x_q}{x_p} \right| \leq (k-1) + 1 = k. \tag{2}$$

Case 2. If $p = (x_p, y_p)$ is above the center path (that is, $y_p > \lfloor x_p/2 \rfloor$), then $q = (x_q, y_q)$ satisfies that $y_q \geq \lfloor x_q/2 \rfloor$. The even columns above the center path are copied from T_{k-1} and the odd columns contain only diagonal edges. Thus p is a copy of $p' = (\lfloor x_p/2 \rfloor, y_p - \lfloor x_p/2 \rfloor)$ and q is a copy of $q' = (\lfloor x_q/2 \rfloor, y_q - \lfloor x_q/2 \rfloor)$.

Since the claim holds for T^{k-1}, the vertical distance from q' to the line segment $\overline{op'}$ is

$$d_y(q', \overline{op'}) = \left| y_q - \left\lfloor \frac{x_q}{2} \right\rfloor - (y_p - \left\lfloor \frac{x_p}{2} \right\rfloor) \frac{\lfloor x_q/2 \rfloor}{\lfloor x_p/2 \rfloor} \right| = \left| y_q - y_p \frac{\lfloor x_q/2 \rfloor}{\lfloor x_p/2 \rfloor} \right| \leq k-1, \tag{3}$$

which is exactly the same expression as in Case 1. Hence, by (1) and the same argument as above, we get $d_y(q, \overline{op}) \leq k$.

Analogously, we can pick any point q on the Euclidean line segment $\overline{op}$ and show that its vertical distance $d_y(q, \mathrm{dig}(op))$ from the digital line segment digop is at most $1 + \log n$. Note that in this case we need to add 1 to the bound since the grid point of $\mathrm{dig}(op)$ that is closest to q has a distance of at most 1 from the vertical line through q.

Since Φ^{-1} maps the vector $(1, 0)$ to $(1, 0)$ and the vector $(0, 1)$ to $(-1, 1)$, the L_∞ distance of q and a line op (with a positive slope) in $\mathbf{G}(n)$ is the same as the vertical distance $d_y(\Phi(q), \Phi(op))$ between the corresponding point and line in $\Phi(\mathbf{G}(n))$. □

Note that the tree DT(2) is related to a famous low discrepancy sequence called van der Corput sequence [17]. Assume that n is a power of 2, and construct the sequence $X(\mathrm{DT}(2))$ using the method of Section 3.2 (ignoring $x_0 = 1$). Then, we obtain $X(\mathrm{DT}(2)) = (0, 1/2, 1/4, 3/4, 1/8, 5/8, 3/8, 7/8, \ldots)$, where in general for the 2-adic expansion $b_1 b_2 b_3 \ldots b_s$ of $i - 1$ we have $x_i = 0.b_s b_{s-1} \ldots b_1$ for $1 \leq i \leq n$. This sequence is indeed the van der Corput sequence.

It is also an interesting observation that DT(2) has a quite uniform structure. Indeed, for any grid point $p = (x, y)$, the path from o to p has $\lfloor \log(|x| + |y|) \rfloor$ or $\lceil \log(|x| + |y|) \rceil$ branching vertices (excluding o) in DT(2).

3.4 Constant distance bound for non-monotonic rays

Surprisingly, if we omit the monotonicity axiom (R5), the lower bound does not hold. We instead give a constant upper bound on the Hausdorff distance in Theorem 3.9. The same bound holds for the Fréchet distance if we regard a digital ray as the corresponding connected path in the graph G defining the

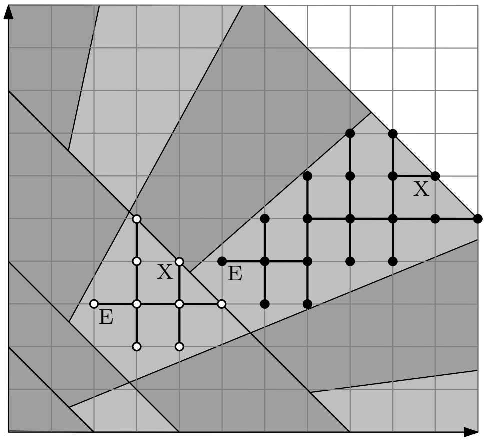


Figure 5: Trapezoid decomposition and two trees of the forest T_1.

grid topology. The digital rays that we construct are locally snake-like almost everywhere; but their bird's eye views can approximate the respective Euclidean line segments fairly well.

Theorem 3.9. *If the monotonicity axiom (R5) is not considered, there exists a system of digital rays in the plane grid such that the Hausdorff distance between each digital ray and its corresponding Euclidean line segment is $O(1)$.*

Proof. The idea is as follows: We first consider a coarser grid of width 2 that is additionally translated by $(0.5, 0.5)$, and construct a spanning forest T_1 of this grid, where internal leaves are allowed. Then, we replace each node v of this forest by four nodes in the original unit-width grid such that v is located in the center of gravity of these four nodes, that is, if v has coordinates $(2x+0.5, 2y+0.5)$ for two integers x and y then it is replaced by the four nodes $(2x, 2y)$, $(2x, 2y+1)$, $(2x+1, 2y)$, and $(2x+1, 2y+1)$ in the unit-width grid. Finally, we convert the forest T_1 into a tree T_2 in the unit-width grid.

Let $c > 1$ be an irrational constant. The forest T_1 is constructed as follows: We consider the *belt* $R(k) \supset \mathbf{G}(2^{k+1}) \setminus \mathbf{G}(2^k)$ defined by $2^k < x + y \leq 2^{k+1}$ in the first quadrant and subdivide it into trapezoids by lines $\ell_t : y = \frac{2^k - tc}{tc}x$ passing through the non-grid points $(tc, 2^k - tc)$ on the line $x + y = 2^k$ for $t = 1, 2, \ldots, \lfloor 2^k/c \rfloor$. The widths of the two parallel edges of each trapezoid are (at most) $\sqrt{2}c$ and $2\sqrt{2}c$, respectively. Further, each trapezoid F is adjacent to one trapezoid $p(F)$ in $R(k-1)$ called the parent of F and to two trapezoids $l(F)$ and $r(F)$ in the belt $R(k+1)$ that are called the left and right child, respectively. Let q be the intersection of $x + y = 2^{k+1}$ and the dividing line of $l(F)$ and $r(F)$. The nearest grid point to q in F is called the exit node of F, and the nearest grid points to q in $l(F)$ and $r(F)$ are called their entry nodes. Each trapezoid has exactly one entry and one exit node. In Figure 5, the entry node and the exit node of F are marked by "E" and "X", respectively.

By gathering these trapezoids for all $k \geq \lceil \log c \rceil$, we have a decomposition of the first quadrant of the plane. Since $c > 1$, each trapezoid is wide enough

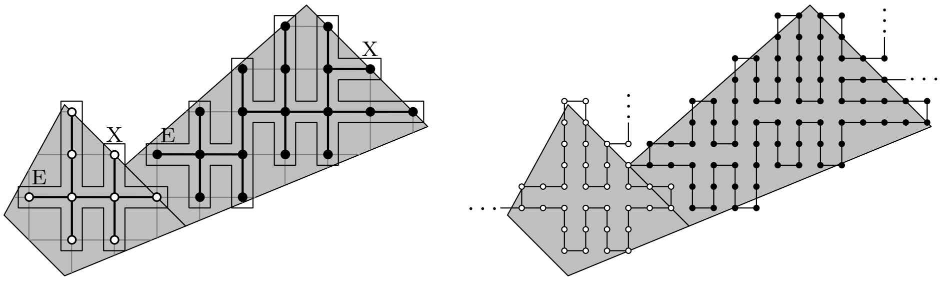


Figure 6: The walks around the two trees (a) and the corresponding part of the tree T_2 formed by connecting the two walks (b).

so that the induced subgraph of the grid points in a trapezoid is connected. It is easy to find a spanning tree of the vertices in each trapezoid consisting of a *trunk* that is a shortest path from its entry node to its exit node, together with branches such that the length of each branch (that is, the path length from the trunk to the furthest leaf) is at most $2c$ as seen in Figure 5. This gives a forest T_1 consisting of small trees, one in each trapezoid. Now, let us convert T_1 into T_2 as shown in Figure 6. Each node of T_1 is replaced by four nodes at the corners of the surrounding unit square. Thus, we can realize the walk around the subtree of T_1 in F as a Hamiltonian cycle in the finer grid. We cut the cycle at the exit node and connect to the entry nodes of the trees in the two child trapezoids as in Figure 6. We obtain a tree T_2 that has no internal leaves. For any grid point $p \in F$, the line segment $\overline{op}$ is contained in the union of the ancestor trapezoids of F, and also all ancestors of p in the tree T_2 are in the same union of trapezoids. Since the width of each trapezoid is at most $2\sqrt{2}c$, the distance from any point q in the path $\mathrm{dig}(op)$ in T_2 to the line op is at most $2\sqrt{2}c$. It might happen that the nearest point from q to the line op is not in the segment $\overline{op}$ since we do not assume the monotonicity axiom. However, since the length of each branch of a subtree in T_1 is at most $2c$, the Hausdorff distance between the segment $\overline{op}$ and the path from o to p in the tree is at most $(2\sqrt{2}+2)c$. □

4 Digital rays in higher-dimensional grids

We can give a d-dimensional analogue DT(d) of DT(2) to define digital rays in d-dimensional space. We utilize the fact that a line in d-dimensional space is uniquely determined by its projections to all two-dimensional subspaces spanned by the first coordinate and the i-th coordinate for $i = 2, 3, \ldots, d$. We first demonstrate the construction for the case $d = 3$ and discuss the general case later.

Analogously to the two-dimensional case, we first transform the orthogonal grid by a linear map that maps the base vectors $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$ to $(1,0,0)$, $(1,1,0)$ and $(1,1,1)$, respectively. Thus, the first octant of the orthogonal grid is mapped to the part $\mathbf{Q}(3)$ defined by $0 \le z \le y \le x$ of the skew grid spanned by three types of edges corresponding to the vectors $(1,0,0)$, $(1,1,0)$ and $(1,1,1)$. Next, we define a spanning tree $T(3)$ in this skew grid and transform it back to a spanning tree in the orthogonal grid.

To define $T(3)$, it suffices to define the parent of each vertex $(i,j,k) \in \mathbf{Q}(3)$. We use our previous two-dimensional tree in the skew-grid $\Phi(\mathbf{G})$ which covers the range $0 \le y \le x$ in the plane. We call this tree $T(2)$ implying that it is a tree in the two-dimensional skew-grid. We define two copies $T(2;x,y)$ and $T(2;x,z)$ of $T(2)$ for the dimension pairs (x,y) and (x,z), which we call (x,y)-tree and (x,z)-tree, respectively. The (x,y)-tree covers the range $0 \le y \le x$ and the (x,z)-tree covers the range $0 \le z \le x$.

Given a grid point $p = (i,j,k) \in \mathbf{Q}(3)$, we denote p as (x,y)-horizontal (resp. (x,y)-diagonal) if the edge between (i,j) and its parent in the (x,y)-tree is horizontal (resp. diagonal). Similarly, p is called (x,z)-horizontal (resp. (x,z)-diagonal) if the edge between (i,k) and its parent in the (x,z)-tree is horizontal (resp. diagonal).

The following case distinction defines the parent of $p = (i,j,k)$ in $T(3)$:

1. if p is (x,y)-horizontal and (x,z)-horizontal, its parent is $(i-1,j,k)$;
2. if p is (x,y)-diagonal and (x,z)-horizontal, its parent is $(i-1,j-1,k)$;
3. if p is (x,y)-diagonal and (x,z)-diagonal, its parent is $(i-1,j-1,k-1)$.

There is one case missing, namely if (i,j,k) is (x,y)-horizontal and (x,z)-diagonal. Our key observation is that this case cannot occur. By the definition of $\mathbf{Q}(3)$, we have $k \le j$. And by Lemma 3.6 there is never a diagonal edge below a horizontal one in an edge column of $T(2)$. Now if (i,j,k) is (x,y)-horizontal it must also be (x,z)-horizontal.

Therefore, we have defined a graph $T(3)$ in the grid $\mathbf{Q}(3)$, which uses only edges that are parallel to the vectors $(1,0,0)$, $(1,1,0)$, or $(1,1,1)$. Analogously, we can confirm that every node has at least one child. The following lemma follows from the definition of $T(3)$.

Lemma 4.1. *For every $p = (i,j,k) \in \mathbf{Q}(3)$, there is a unique path $\mathbf{p}$ from the origin o to p in $T(3)$. Thus, $T(3)$ is a tree rooted at o. The projection of $\mathbf{p}$ to the (x,y)-plane (resp. (x,z)-plane) coincides with the path from o to (i,j) (resp. (i,k)) in the (x,y)-tree (resp. (x,z)-tree).*

The next lemma is a consequence of Lemma 4.1 and Theorem 3.8:

Lemma 4.2. *For any plane $x = a$ where $0 \le a \le n$, let (a,b,c) and (a,b',c') be its intersection points with $\overline{op}$ and $\mathrm{dig}(op)$, respectively. Then, $|b-b'| < \log n$ and $|c-c'| < \log n$.*

We use the inverse map from the skew grid $\mathbf{Q}(3)$ to the three-dimensional orthogonal grid; this maps $T(3)$ to an orthogonal tree $\mathrm{DT}(3)$.

Proposition 4.3. *The L_1 distance from any point on the digital ray in $\mathrm{DT}(3)$ to the corresponding Euclidean line is at most $4\log n$ if the absolute value of each coordinate value of the point is bounded by n. Consequently, the L_1-Hausdorff distance between a line segment and the corresponding digital ray is a most $4\log n$.*

Proof. Let us examine how the distance changes during the inverse map. The vectors $(0, 1, 0)$ and $(0, 0, 1)$ are mapped to $(-1, 1, 0)$ and $(0, -1, 1)$, respectively. Thus, a vector $(0, s, t)$ is mapped to $(-s, s - t, t)$ and $| - s| + |s - t| + |t| \leq 2|s| + 2|t|$. Thus, for $|s| \leq n$ and $|t| \leq n$ we can apply Lemma 4.2 which yields the proposition. □

For the general d-dimensional grid, we have the following theorem:

Theorem 4.4. *Given a d-dimensional grid with n^d grid points in the orthogonal topology, we can define a spanning tree $T(d)$ such that the L_1-Hausdorff distance between the line segment $\overline{op}$ and the digital ray $\mathrm{dig}(op)$ is less than $2(d-1)\log n$ if the absolute value of each coordinate value of p is bounded by n.*

Proof. Let $x_1, x_2, \ldots, x_d$ be the coordinates of the d-dimensional space and define $\mathbf{Q}(d)$ by $0 \leq x_d \leq x_{d-1} \leq \ldots \leq x_1$. As before we define copies $T(2; x_1, x_i)$ of $T(2)$ for the dimension pairs (x_1, x_i), where $i = 2, 3 \ldots, d$. Now, let us consider a grid point $p = (p_1, p_2, \ldots, p_d) \in \mathbf{Q}(d)$ and define its parent in $T(d)$. From Lemma 3.6, there exists an integer $2 \leq i \leq d + 1$ such that (p_1, p_j) is diagonal in $T(2; x_1, x_j)$ for $j < i$ and horizontal for $j \geq i$. Note that all edges (p_1, p_j) are horizontal (resp. diagonal) if $i = 2$ (resp. $i = d + 1$). We connect p by an edge with the vector $(1, 1, \ldots, 1, 0, \ldots, 0, 0)$ to its parent, where the vector has $(i - 1)$ unit entries and $(d - i + 1)$ zero entries. This yields a spanning tree of the grid points of $\mathbf{Q}(d)$. The remaining analysis is analogous to the three-dimensional case. □

5 Digital mountain approximation

Consider a $[0, 1]$-valued function f on $\mathbf{P}$, which we call a pixel image function. In computer vision, it is important to find an approximation of a given pixel grid image (represented by a function) by using another function with a nice property. The problem comes as a natural variant of the least-squares method and is formulated as follows: Let us fix a family $\mathcal{O}$ of pixel image functions with some nice property. Given a pixel image function f, we would like to find $\phi \in \mathcal{O}$ minimizing the L_2 distance $|f - \phi|_2 = [\sum_{p \in \mathbf{P}} (f(p) - \phi(p))^2]^{1/2}$.

Picture retouching is a typical process on a pixel image: The user clips a part of a digital picture and retouches it; for example, to remove noise, waves, scars and/or stains in a picture. A useful operation in picture retouching is as follows: Given a *peak* position o (as user's input or automatically), reform the clipped part into a distribution peaked at o and gradually fading out to the boundary. This can be formulated as the following particular function approximation problem: Given a function f defined on $\mathbf{P}$, its *level set* at a height t is $L(f, t) = \{p \in \mathbf{P} : f(p) \geq t\}$. The boundary of a level set is often called a contour. We call f a *mountain* function with the peak position $o \in \mathbf{P}$ if each of its level sets is a digital star-shaped region centered at o (thus, each contour is a digital star-shaped polygon).

The optimal mountain approximation problem is as follows: Given a real-valued function f defined on $\mathbf{P}$, we would like to find a digital mountain function

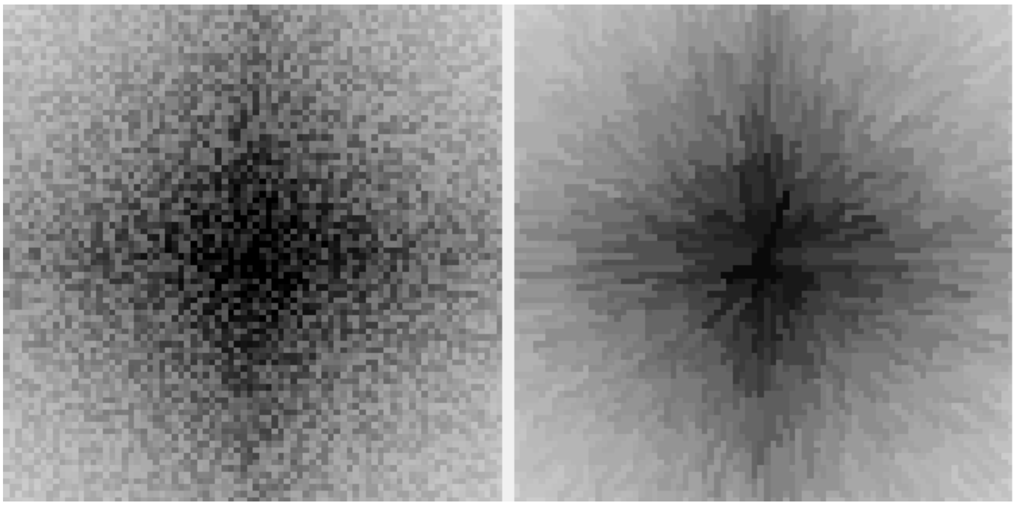


Figure 7: Mountain approximation: the values of the pixel image functions f (left) and ϕ (right) are represented by gray levels.

ϕ minimizing the L_2 distance $|f-\phi|_2 = [\sum_{p\in\mathbf{P}}(f(p)-\phi(p))^2]^{1/2}$. Geometrically, the problem can be regarded as transforming a terrain represented by f to a mountain. Figure 7 illustrates how the mountain approximation works in our implementation.

The following two results given by Chen et al. [3] are our basic tools to compute the optimal digital mountain approximation:

Proposition 5.1 (Chen et al. [3])**.** *Let $\mathcal{R}$ be a family closed under intersection and union of regions and let $R(f,t)$ be the region $R \in \mathcal{R}$ maximizing $\sum_{p\in R}(f(p)-t)$ for a given real value t. If there is more than one such region, there are a maximum and a minimum (in terms of set inclusion) among those regions. We denote them $R_{\max}(f,t)$ and $R_{\min}(f,t)$, respectively.*

We call t a critical height if $R_{\max}(f,t) \neq R_{\min}(f,t)$. Chen et al. [3] showed that it suffices to compute $R(f,t)$ for each critical height t in order to compute ϕ.

Theorem 5.2 (Chen et al. [3])**.** *Let $\mathcal{R}$ be a region family closed under intersection and union of regions and let $\mathcal{F}$ be a family of pixel image functions whose level sets are regions in $\mathcal{R}$. Then for the function $\phi \in \mathcal{F}$ that minimizes the L_2-distance from f the level set $L(\phi,t) = R_{\max}(f,t)$. Moreover, if $\phi(p) = t$ for a pixel $p \in \mathbf{P}$ then $p \in R_{\max}(f,t) \setminus R_{\min}(f,t)$.*

Let us consider the family $\mathcal{S}$ of digital star-shaped regions. For each vertex $v \in V$ of the tree $DT(2)$, we give a parametric weight $w(v,t) = f(v) - t$, where $f(v)$ is the value of the input function f at the pixel corresponding to v. $R(f,t)$ must be a rooted subtree of $\mathrm{DT}(2)$ maximizing the sum of the parametric weights of the vertices. For a given t, it is quite easy to compute $R(f,t)$: We traverse $\mathrm{DT}(2)$ in a bottom-up fashion starting from the leaves and remove each vertex v and the subtree rooted at v if the sum of the parametric weights of v in the subtree (ignoring removed vertices so far) is negative. The final subtree obtained by the algorithm gives $R_{\max}(f, t)$. If we replace “negative” by “non-positive” in the above procedure, we obtain $R_{\min}(f, t)$. Clearly, this can be done in linear time in terms of the tree size.

Now, we can apply a so-called *hand probing* operation: Given two heights $t_1 < t_2$ where $R_1 = R_{\max}(f, t_1) \neq R_2 = R_{\max}(f, t_2)$, we find a height t_3 with $t_1 < t_3 < t_2$ such that R_1 and R_2 have the same parametric weight at t_3 and compute $R_3 = R_{\max}(f, t_3)$. This operation can be done in linear time in terms of the tree size. If the height t_3 is a critical height, we define $\phi(p) = t_3$ for all $p \in R_{\max}(f, t_3) \setminus R_{\min}(f, t_3)$. We recursively process the height intervals (t_1, t_3) and (t_3, t_2) and thus find all critical heights in $O(h)$ hand-probing operations, where h is the number of different heights in the input data. Note that h is bounded by $h \leq \min\{N, \Gamma\}$, where Γ is the number of gray levels, for example, $\Gamma = 256$, and N is the size of the tree DT(2). So the total time complexity is $O(Nh)$.

We can speed up the algorithm to run in $O(N \log h)$ time using the methods of Chen et al. [3]. Their idea is based on using a contracted tree that, for a query with heights $t_1 < t_2$, represents only the pixels in the set $R_1 \setminus R_2$ instead of the full tree DT(2). Thus each full level of the binary search tree can be handled in $O(N)$ time. Obviously, the number of levels of the search tree is $O(\log h)$ and thus the optimal mountain approximation can be computed in $O(N \log h)$ time. Note that if the peak position o is not specified by the user we need to test all candidate positions to find the best one.

We remark that the result can be easily extended to the d-dimensional case that is an analogue of the pyramid construction problem considered in [3]. Note that this algorithm can also be extended to the non-monotonic rays defined in Section 3.4. We can also control the curvature of the contours by using the method of Wu and Chen [19], where we consider a directed acyclic graph obtained by adding artificial edges to $DT(2)$, although we need a minimum-cost-flow algorithm for solving that version.

6 Concluding remarks

Although our $O(\log n)$ bound for the distance is asymptotically optimal, we may improve the constant factor: The lower bound factor in discrepancy theory is merely 0.06 [11]. An obviously important problem is to investigate the definition of consistent digital line segments for all pairs of grid points or, as a first step, for digital rays with multiple origins. As shown in the introduction, if the set of digital line segments satisfies the axioms, the distance bound seems to become $\Omega(n)$; it is an interesting question to prove or disprove this.

7 Acknowledgments

Part of this work was done while Martin Nöllenburg visited Tohoku University in February 2007 supported by MITI Grant on Fundmental Research (B). The authors thank János Pach for his stimulating discussion on this topic, especially on the problem of the lower distance bound. The authors also thank Alexander Wolff who gave several suggestions on a preliminary version of this paper.

References

[1] T. Asano, D. Z. Chen, N. Katoh, and T. Tokuyama. Efficient algorithms for optimization-based image segmentation. *Internat. J. Comput. Geom. Appl.*, 11(2):145–166, 2001.

[2] Cgal, Computational Geometry Algorithms Library. http://www.cgal.org.

[3] D. Z. Chen, J. Chun, N. Katoh, and T. Tokuyama. Efficient algorithms for approximating a multi-dimensional voxel terrain by a unimodal terrain. In *Proc. 10th Ann. Internat. Conf. Computing and Combinatorics (COCOON'04)*, volume 3106 of *Lecture Notes in Comput. Sci.*, pages 238–248. Springer-Verlag, 2004.

[4] J. Chun, K. Sadakane, and T. Tokuyama. Efficient algorithms for constructing a pyramid from a terrain. In *Proc. Japanese Conf. Discrete and Comput. Geom. (JCDCG'02)*, volume 2866 of *Lecture Notes in Comput. Sci.*, pages 108–117. Springer-Verlag, 2003.

[5] P. Erdős. Problems and results on diophantine approximation. *Compos. Math.*, 16:52–65, 1964.

[6] J. E. Goodman, R. Pollack, and B. Sturmfels. Coordinate representation of order types requires exponential storage. In *Proc. 21st Ann. ACM Symp. Theory Comput. (STOC'89)*, pages 405–410. ACM, 1989.

[7] M. T. Goodrich, L. J. Guibas, J. Hershberger, and P. J. Tanenbaum. Snap rounding line segments efficiently in two and three dimensions. In *Proc. 13th Ann. ACM Symp. Comput. Geom. (SoCG'97)*, pages 284–293, 1997.

[8] D. H. Greene and F. F. Yao. Finite-resolution computational geometry. In *Proc. 27th Ann. IEEE Symp. Foundations Comput. Sci. (FOCS'86)*, pages 143–152. IEEE, 1986.

[9] R. Klette and A. Rosenfeld. Digital straightness—a review. *Discrete Appl. Math.*, 139(1–3):197–230, 2004.

[10] J. Matoušek. *Geometric Discrepancy: An Illustrated Guide.* Springer-Verlag, 1999.

[11] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM, 1992.

[12] J. Pach, R. Pollack, and J. Spencer. Graph distance and Euclidean distance on the grid. In R. Bodendiek and R. Henn, editors, *Topics in Graph Theory and Combinatorics*, pages 555–559. Physica-Verlag, 1990.

[13] W. M. Schmidt. Irregularities of distribution, VII. *Acta Arithmetica*, 21:45–50, 1972.

[14] W. M. Schmidt. *Lectures on Irregularities of Distribution.* Tata Inst. Fund. Res. Bombay, 1977.

[15] O. Schwarzkopf. The extensible drawing editor ipe. In *Proc. 11th Ann. ACM Symp. Comput. Geom. (SoCG'95)*, pages C10–C11, 1995.

[16] K. Sugihara. Robust geometric computation based on topological consistency. In V. N. Alexandrov, J. Dongarra, B. A. Juliano, R. S. Renner, and C. J. K. Tan, editors, *Proc. Internat. Conf. Computational Science, Part 1 (ICCS'01)*, volume 2073 of *Lecture Notes in Comput. Sci.*, pages 12–26. Springer-Verlag, 2001.

[17] J. van der Corput. Verteilungsfunktionen I & II. *Nederl. Akad. Wetensch. Proc.*, 38:813–820, 1058–1066, 1935.

[18] X. Wu. Efficient algorithms for the optimal-ratio region detection problems in discrete geometry with applications. In *Proc. 17th Internat. Symp. Algorithms and Computation (ISAAC'06)*, volume 4288 of *Lecture Notes in Comput. Sci.*, pages 289–299. Springer-Verlag, 2006.

[19] X. Wu and D. Z. Chen. Optimal net surface problems with applications. In *Proc. 29th Internat. Coll. Automata, Languages and Programming (ICALP'02)*, volume 2380 of *Lecture Notes in Comput. Sci.*, pages 1029–1042. Springer-Verlag, 2002.