Near Optimal Locality Sensitive Orderings in Euclidean Space

Zhimeng Gao
Department of Computer Science and Engineering,
Hong Kong University of Science and Technology, Kowloon, Hong Kong

Sariel Har-Peled
Department of Computer Science, University of Illinois, Urbana, IL, USA

Abstract

For a parameter $\varepsilon \in (0,1)$, a set of ε-locality-sensitive orderings (LSOs) has the property that for any two points, $p, q \in [0,1)^d$, there exist an order in the set such that all the points between p and q (in the order) are ε-close to either p or q. Since the original construction of LSOs can not be (significantly) improved, we present a construction of modified LSOs, that yields a smaller set, while preserving their usefulness. Specifically, the resulting set of LSOs has size $M = O(\mathcal{E}^{d-1} \log \mathcal{E})$, where $\mathcal{E} = 1/\varepsilon$. This improves over previous work by a factor of $\mathcal{E}$, and is optimal up to a factor of $\log \mathcal{E}$.

As a consequence we get a flotilla of improved dynamic geometric algorithms, such as maintaining bichromatic closest pair, and spanners, among others. In particular, for geometric dynamic spanners the new result matches (up to the aforementioned $\log \mathcal{E}$ factor) the lower bound, Specifically, this is a near-optimal simple dynamic data-structure for maintaining spanners under insertions and deletions.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Orderings, approximation

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.60

Related Version *Full Version*: `https://arxiv.org/abs/2310.12792` [9]

Funding *Sariel Har-Peled*: Work on this paper was partially supported by a NSF AF award CCF-1907400.

1 Introduction

Given a total linear order σ of $\mathcal{H} = [0,1)^d$, and any two points $p, q \in \mathcal{H}$, for $p \prec_\sigma q$, consider the interval of all points between p and q according to σ:

$$\sigma(p,q) = \{u \in \mathcal{H} \mid p \prec_\sigma u \prec_\sigma q\}. \tag{1.1}$$

Chan et al. [6] showed that there is a "small" set Π of orderings, such that for all points $p, q \in \mathcal{H}$, there is an order $\sigma \in \Pi$ that is ε-local, where $\varepsilon \in (0,1)$ is a fixed parameter.

▶ **Definition 1.** *For a pair of points $p, q \in \mathcal{H} = [0,1)^d$, an order σ over the points of $\mathcal{H}$ is* **ε-local** *if*

$$\sigma(p,q) \quad \subseteq \quad \mathbb{b}(p, \varepsilon\ell) \cup \mathbb{b}(q, \varepsilon\ell), \qquad \text{where} \qquad \ell = \|pq\|,$$

where $\mathbb{b}(p,r)$ denotes the **ball** *of radius r centered at p, see Figure 1.1 (i).*

Namely, all the points between p and q in σ are in the vicinities of p and q in $\mathcal{H}$. Informally, there is an order in Π that respects the localities of p and q. Chan et al., inspired by locality sensitive hashing, refereed to σ as being a *locality sensitive ordering* (LSO).

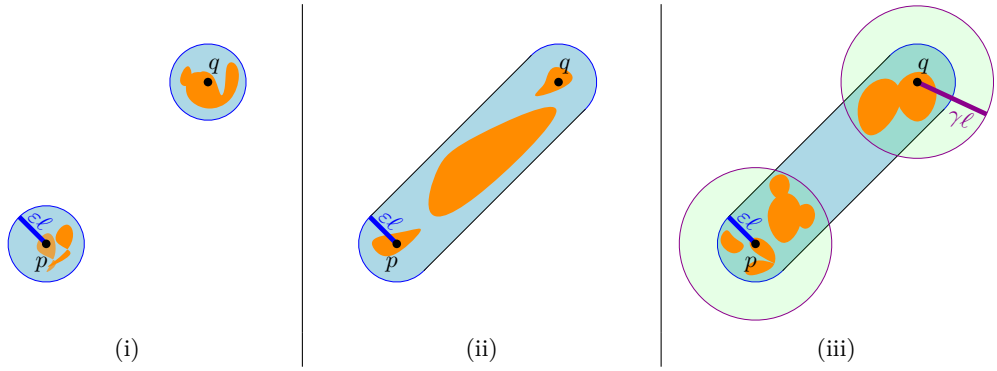

Figure 1.1 The three types of locality: (i) ε-locality. (ii) ε-hippodrome locality, and (iii) (ε,γ)-locality, which is the hippodrome locality together with an additional gap requirement. (The orange region is the set of points in between p and q in the ordering.)

Surprisingly, Chan et al. showed that one can compute such a "universal" set of orderings Π, of size $O(\mathcal{E}^d \log \mathcal{E})$, where $\mathcal{E} = 1/\varepsilon$. This set of orderings can be easily computed, and computing the order of two points according to a specific order in the set can be done quickly. Having such a set of orderings, reduces some dynamic geometric problems to the maintenance of a "few" sorted lists of points under insertions and deletions. Thus, certain problems in d dimensions, are reduced to a collection of problems in one dimension. This can be interpreted as a somewhat strange dimension reduction result, (multi)embedding $[0,1)^d$ in the real line, such that for any two points in the hypercube, there is at least one embedding that has certain desired geometric property.

For example, using these orderings one can easily maintain a spanner under insertion/deletion operations with logarithmic time per operation (ignoring terms depending on ε). Specifically, one maintains (under updates) sorted lists of the point set, according to each of the orderings in Π. Here, an edge in the spanner is present only if the two points are adjacent in one of these ordered lists. Note, that for each of these one dimensional instances, one maintains a dynamic ordered map of the points (e.g., balanced binary search tree).

The challenge. It is natural to try and reduce the number of orderings further, improving (in hopefully a black-box fashion) the results that use locality-sensitive orderings. Unfortunately, it is not hard (see Lemma 26) to show that $\Omega(\mathcal{E}^d)$ is a lower bound on the number of orderings under the ε-local condition, implying that the construction of Chan et al. is essentially optimal (ignoring the annoying $\log \mathcal{E}$ factor).

So to reduce the number of orderings further, we have to relax the requirement somehow. A natural first step is to enlarge the vicinity to p and q to be an "ε-hippodrome" region. Specifically, we require that

$$\sigma(p,q) \quad \subseteq \quad \hbar(pq, \varepsilon\ell) = \mathcal{CH}\big(\mathbb{b}(p,\varepsilon\ell) \cup \mathbb{b}(q,\varepsilon\ell)\big),$$

see Figure 1.1 (ii), where $\hbar(pq,r)$ is the *r-hippodrome* of pq. This condition is sufficient for some applications, but others require a stronger property of having a "gap" in the ordering.

▶ **Definition 2.** *For $\gamma \in (0,1/4)$, with $\gamma \geq \varepsilon$ (e.g., $\gamma = 1/4$), consider the requirement that*

$$\sigma(p,q) \quad \subseteq \quad \hbar(pq,\varepsilon\ell) \cap \Big(\mathbb{b}(p,\gamma\ell) \cup \mathbb{b}(q,\gamma\ell)\Big), \qquad \textit{where} \qquad \ell = \|pq\|,$$

see Figure 1.1 (iii). An order σ that has the last property is an **(ε,γ)-local** *for p and q.*

Computation model. The model of computation used here is a unit-cost real RAM, supporting standard arithmetic operations and comparisons (but no floor function), augmented with standard bitwise-logical operations (bitwise-exclusive-or and bitwise-and), which are widely available as assembly commands on all modern general purpose CPUs, and programming languages. This computational model is reasonable, and is used (for example) in working with compressed quadtrees efficiently [10].

Previous related work. LSOs were introduced by Chan et al. [6]. Variations of LSOs for doubling metrics, and other metric spaces, were studied by Filtser and Le [8] (Triangle LSO, Left-sided LSO), and Filtser [7] (Triangle LSO, Rooted LSO). Since these works study LSO in the non-euclidean settings, they are not directly related to our work here (and they yield, naturally, worse bounds). LSOs were used to construct reliable spanners by Buchin et al. [2, 3] in a surprisingly simple plug & play fashion. There is some work on quadtrees for hyperbolic space by Kisfaludi-Bak and van Wordragen [12], which potentially can lead for LSOs for such spaces.

Our results

In this work, we show that there is a universal set of LSOs of $[0,1)^d$ that is of size $O(\mathcal{E}^{d-1}\log\mathcal{E})$, such that these orderings have the somewhat weaker (ε,γ)-locality property, where $\gamma \geq \varepsilon$ is a large fixed constant. This improves the result of Chan et al. by a factor of $1/\varepsilon$. Furthermore, for many of the applications using LSOs, one can replace the LSOs by the new set of "weaker" orderings, thus strictly improving their dependence on ε. In particular, for spanners the new construction enables maintaining dynamic spanners. Notably, up to $\log\mathcal{E}$ factor, the new construction matches the known lower bound on the number of edges in a spanner, in the *static* case, thus implying that the new orderings are almost optimal, and no approach based on LSOs can do (significantly) better in the Euclidean case.

Applications. Since we have to slightly weaken the definition of LSOs to get the improved size, we have to rederive the proofs of correctness for some of the applications. LSOs are especially useful in applications where one wants to solve some geometric problem under insertions and deletions of points. Since LSOs decouple the geometric problem from the data-structure problem (i.e., all one needs to do is to maintain sorted lists of points under insertions and deletions), this results in surprisingly simple data-structures that supports updates:

1. Locality graph. In Section 4.1 we outline the basic approach – one maintains a graph over the point set, where two points are connected by an edge, if they are adjacent in one of the LSOs. The locality graph can be updated in $O(\mathcal{E}^{d-1}\log^2\mathcal{E}\cdot\log n)$ time, and has the property that every point (i.e., vertex) has degree at most $O(\mathcal{E}^{d-1}\log\mathcal{E})$ in the graph.
2. Bichromatic closest pair. In Section 4.2, we show how to use LSOs to dynamically maintain the bichromatic closest pair, as the shortest bichromatic edge in the locality graph is the desired approximation. This also readily leads to a data-structure that supports $(1+\varepsilon)$-approximate nearest-neighbor queries. We do not present the details for this here, as this result is, relatively speaking, less interesting.
3. Maintaining $(1+\varepsilon)$-spanners. It turns out that the above locality graph is a $(1+\varepsilon)$-spanner. Since we can update the locality graph quickly, as detailed above, this readily leads to improved dynamic data-structures for maintaining spanners. This is described in Section 4.3. The spanner construction uses the gap property mentioned above.

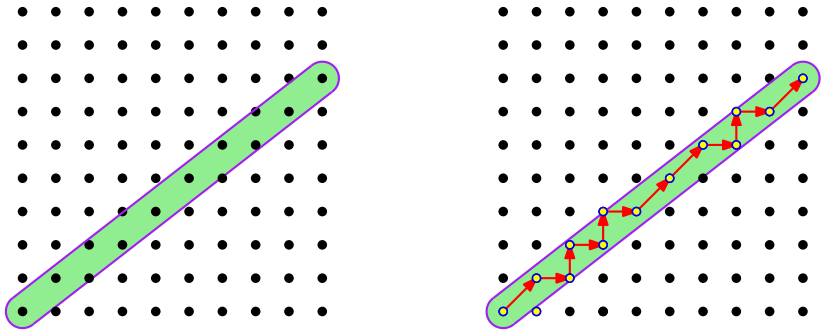

Figure 1.2 A path inside along a hippodrome can cover many long edges.

All these results improve by a factor of $\mathcal{E}$ over the previous results of Chan et al.. The new LSOs also improves, by a factor of $\mathcal{E}$, in a plug and play fashion other results, but the improvement is somewhat less interesting in these cases. This includes reliable spanners [2], dynamic approximate minimum spanning trees [6], and vertex/edge fault-tolerant spanners [6].

Sketch of the old construction. We describe shortly the construction of LSOs by Chan et al. [6], see Section 2.1 for more details. Chat et al. reduced the problem of computing LSOs to the following problem – consider the integer grid $B = [\![4\mathcal{E}]\!]^d$, where $\mathcal{E} = 1/\varepsilon$ and $[\![n]\!] = \{1, \ldots, n\}$. An edge of G is *long* if it corresponds to a pair of points of B that are in distance at least (say) $2/\varepsilon$ from each other. The long edges corresponds to well-separated pairs of points, and these are the ones we must "serve". (Understanding why we concentrate on the long edges requires a bit deeper dive into the details, see Section 2.1.) To this end, consider the complete graph G over B. Constructing LSOs is then achieved by constructing LSOs for B. A single ordering (i.e., LSO) for B is simply a Hamiltonian path of G, and the task at hand is to compute a small number of Hamiltonian paths of G that visits all the long edges. Specifically, a Hamiltonian path (in a graph) induces an ordering of the vertices, thus Hamiltonian paths serves as a way to encode the desired orderings. Having a long edge in such a path, ensures this pair is consecutive in the associated ordering.

Walecki [1] showed a decomposition of the clique into Hamiltonian paths, and this leads to the desired LSOs of B. The problem with this approach is that it requires $\Omega(\mathcal{E}^d)$ paths/LSOs. There is an additional blowup by a factor of $O(\log \mathcal{E})$ in the number of LSOs, when converting the LSOs of B to LSOs of $[0, 1)^d$.

Sketch of new approach. Not all hope is lost however – the previous construction effectively ignores the geometry. In addition, one can weaken the requirement, so that the new LSOs for B have the ε-hippodrome locality condition, see Figure 1.1 (ii). So consider two points of B that are far from each other, and the hippodrome induced by the segment connecting them, see Figure 1.2. We now order the points inside this region in their order along the direction induced by the two points. Clearly, this path will have all the long pairs inside the hippodrome covered (by the transitive closure of this directed path). The question is how to find such paths, and glue them together into global orderings that cover all such long edges.

To this end, we cover the set of directions by an ε-packing of size $O(\mathcal{E}^{d-1})$ (this is where the savings come from!). For each direction, we show how to construct a constant number of paths that cover all the long edges that their direction is ε-close to the current direction. To this end, we compute a packing of the projected points, and then use coloring to decompose

the packing into $O(1)$ packings, such that the points picked in each set, form independent paths that do not interact with each other. We then take each such set of paths, and concatenate these paths together to get one ordering. Overall, we get $O(\mathcal{E}^{d-1})$ orderings of B, that cover all the long edges, as desired. Plugging this into the machinery of Chan et al. then yields the desired LSOs of $[0,1)^d$.

We are hiding some of the technical details under the hood – for example, we are dealing with grid cells, and not grid points, etc. See Section 3 for details of the construction. The final result is stated in Theorem 19.

Orderings with a gap. As mentioned above, a somewhat stronger property of having a gap in the LSOs is needed, so that one can construct spanners using them. The idea is to build LSOs for B, by first treating it as a crude grid, using the previous construction of Chan et al. at the top grid, and then combine it with the new construction for the bottom part of the grid. The details are somewhat technical, see Section 3.3 for details.

Why does the size of LSOs matter? LSOs seems to be algorithmically equivalent to other geometric tools, such as well-separated pairs decomposition [4]. Thus, they seem to be a fundamental property of the underlying space. It is thus natural to quantify exactly what the minimum size of LSOs needed for $\mathbb{R}^d$, as a function of d and ε. Our work thus can be interpreted as (almost) settling this question.

Paper organization

We describe the old construction of Chan et al. in Section 2.1. We present some necessary standard results on nets/packings in Section 2.2. In Section 3.1 we describing how to construct the needed packings on projected points, and how to decompose such a packing into a packing of larger radius. At last, we are ready. We present, in Section 3.2, the construction of the orderings with the hippodrome property, and, in Section 3.3, we present the modified construction with the gap property.

The applications are presented in Section 4. In Section 4.1 we show how the orderings give rise to a locality graph that captures all the information needed for the applications. In particular, this graph can be maintained efficiently under insertions and deletions. In Section 4.2, we present the improved dynamic data-structure for dynamic bichromatic closest pair. In Section 4.3, we present the improved result for spanners. We also provide a proof of correctness showing that the new orderings are sufficient to guarantee that the locality graph is a spanner. We also point out, in Section 5, that any LSOs that their locality graph is a spanner, must have $\Omega(\mathcal{E}^{d-1})$ orderings, thus implying that our construction of LSOs is tight up to a factor of $\log \mathcal{E}$.

2 Preliminaries

2.1 Locality sensitive orderings

For a set X, consider a total order (or *ordering*) $\prec$ on the elements of X. Two elements $x, y \in X$ are ***adjacent*** if there is no element $z \in X$, such that $x \prec z \prec y$ or $y \prec z \prec x$. Our purpose is to define a small number of orderings on $[0,1)^d$ so that certain geometric tasks can be carried out using these orderings.

In the following, we fix a parameter $\varepsilon = 1/2^\lambda$, for some integer $\lambda > 0$, and use

$$\mathcal{E} = 1/\varepsilon = 2^\lambda \qquad \text{and} \qquad \lambda = \log_2 \mathcal{E}. \tag{2.1}$$

Our starting point is the result of Chan et al. [6]. Since we use the same framework, we quickly sketch the above construction (reproducing some relevant definitions).

▶ **Definition 3.** *Let $\mathcal{C} \subseteq \mathbb{R}^d$ be an axis-parallel cube with side length ζ. For an integer $t > 1$, partitioning $\mathcal{C}$ uniformly into a $t \times t \times \cdots \times t$ subcubes, forms a* ***t*-grid** *$\mathcal{G}(\mathcal{C}, t)$. The diameter of a cube $\square$ is $\operatorname{diam}(\square) = \operatorname{sidelength}(\square)\sqrt{d}$, and let $\mathsf{c}(\square)$ denote its* **center**. *Let $\mathbb{b}(\square)$ be the smallest ball enclosing $\square$ – that is, $\mathbb{b}(\square) = \mathbb{b}(\mathsf{c}(\square), diam(\square)/2)$*

▶ **Definition 4.** *An* **ε-quadtree** *$\mathcal{T}_\varepsilon$ is a quadtree-like structure, built on a cube with side length ζ, where each cell is partitioned into an $\mathcal{E}$-grid. The construction then continues recursively into each grid cell of interest. As such, a node in this tree has up to $\mathcal{E}^d$ children, and a node at level $i \geq 0$ has an associated cube of side length $\zeta\varepsilon^i$. When $\mathcal{E} = 2$, this is a regular quadtree.*

By fixing the root of the ε-quadtree to be $[0, 2^i)^d$, for $i = 1, \ldots, \lambda$, one gets the following.

▶ **Lemma 5.** *Let $\mathcal{T}$ be a regular quadtree over $[0, 2)^d$. There are ε-quadtrees $\mathcal{T}_\varepsilon^1, \ldots, \mathcal{T}_\varepsilon^\lambda$, such that the collection of cells at each level in $\mathcal{T}$ is contained in exactly one of these ε-quadtrees, where $\lambda = \log_2 \mathcal{E}$, see Eq. (2.1).*

The problem with using a quadtree, for our purposes, is the alignment of the boundaries of their cells. Intuitively, two points that are close together might be separated by quadtree boundary, that belongs to cells that are dramatically bigger than this distance. Fortunately, this problem can be overcome by shifting (the point set, or the quadtree) $d + 1$ times.

▶ **Lemma 6** ([5], Lemma 3.3)**.** *Consider any two points $p, q \in [0, 1)^d$, and let $\mathcal{T}$ be the infinite quadtree of $[0, 2)^d$. For $D = 2\lceil d/2 \rceil$ and $i = 0, \ldots, D$, let $\nu_i = (i/(D + 1), \ldots, i/(D + 1))$. Then there exists an $i \in \{0, \ldots, D\}$, such that $p + \nu_i$ and $q + \nu_i$ are contained in a cell of $\mathcal{T}$ with side length $\leq 2(D + 1)\,\|pq\|$.*

The final tool needed is a small set of orderings, such that any two subcells, in a t-grid, are adjacent in one of the orderings.

▶ **Lemma 7** ([6])**.** *Let S be set of t^d subcells of a t-grid $\mathcal{G}(\mathcal{C}, t)$. There is a set Π of $O(t^d)$ orderings of S, such that each pair of elements of S is adjacent in at least one of the orderings.*

2.1.1 From order on a grid, to orderings of the unit cube

Chan et al. [6] showed how to convert an order of a t-grid into an ordering of $[0, 1)^d$. We now describe this construction in detail. Let $\mathcal{F}$ be the set of λ ε-quadtrees of Lemma 5, and let $\mathcal{S} \subseteq [0, 1)^d$ be the set of at most $d + 1$ shifts of Lemma 6. Let $\mathfrak{O}$ be the set of orderings of the t-grid $\mathcal{G}(\mathcal{C}, \mathcal{E})$, for some cube $\mathcal{C}$, as constructed by Lemma 7.

Next, consider an ε-quadtree $\mathcal{T}_\varepsilon \in \mathcal{F}$, a shift $\nu \in \mathcal{S}$, and an ordering $\sigma \in \mathfrak{O}$. This immediately induces an ordering over the points of $[0, 1)^d$. Indeed, consider two points $p, q \in [0, 1)^d$, and consider their shifted image $p' = \nu + p$ and $q' = \nu + q$. There is a unique least common ancestor (LCA) node $u \in \mathcal{T}_\varepsilon$ to the leafs where p' and q' would be stored in the quadtree. The node u has $\mathcal{E}^d$ children, with p' and q' belonging to two different children, say u_p and u_q. If σ has u_p before u_q, then we consider $p \prec_\sigma q$, and otherwise $p \succ_\sigma q$. Using standard bit operations, and the ordering σ listed explicitly (which requires $O(\mathcal{E}^d)$ space), this can be done in $O(\lambda) = O(\log \frac{1}{\varepsilon})$ time [6]. (The quadtree $\mathcal{T}_\varepsilon$ is used implicitly, so it is not computed explicitly.) Overall, the number of orderings this yield, is $\mathcal{O}(|\mathfrak{O}|\,|\mathcal{S}|\,\lambda)$. Given two points comparing their order according to a specific given order takes $O(\lambda)$ time. The resulting set of orderings of $[0, 1)^d$ is of size $O(\mathcal{E}^d \log \mathcal{E})$. This implies the following theorem, see [6] for more details.

▶ **Theorem 8** ([6]). *There is a set Π^+ of $\mathcal{O}(\mathcal{E}^d \log \mathcal{E})$ orderings of $[0,1)^d$, such that for any two points $p, q \in [0,1)^d$ there is an ordering $\sigma \in \Pi^+$ defined over $[0,1)^d$, such that for any point u, with $p \prec_\sigma u \prec_\sigma q$, we have that either $\|pu\| \leq \varepsilon \|pq\|$ or $\|qu\| \leq \varepsilon \|pq\|$.*

2.2 Coverings and packings

▶ **Definition 9.** *A unit length vector $v \in \mathbb{R}^d$ is a* **direction**. *The set of all directions form the* **unit sphere** $\mathbb{S}$ *centered at the origin.*

▶ **Definition 10.** *Consider a set $P \subseteq \mathbb{R}^d$, and a parameter r. A set $N \subseteq P$ is an* ***r*-covering** *of P, if for all $p \in P$, there exists a point $p' \in N$, such that $\|pp'\| \leq r$. A point set N is* ***r*-separated** *if for any two distinct points $p, q \in N$, we have $\|pq\| > r$.*

A subset $N \subseteq P$ that is both r-separated and an r-covering of P is an ***r*-packing**[1] *of P.*

For a finite set P in $\mathbb{R}^d$, an r-packing can be computed in linear time [11] (for constant d). We need such a packing of the unit sphere.

▶ **Lemma 11.** *For any $r \in (0,1)$, there is an r-packing N of the unit sphere $\mathbb{S}$ of size $O(1/r^{d-1})$, and it can be computed in $O(1/r^{d-1})$ time.*

See [9] for the proof.

▶ Remark 12. A standard packing argument implies that any ε-packing of the unit sphere in $\mathbb{R}^d$ must be of size $\Theta(\mathcal{E}^{d-1})$.

3 The new construction of LSOs

3.1 Projections

▶ **Definition 13.** *For a direction v, the* **projection** *by v, of a point $p \in \mathbb{R}^d$, denoted by $\downarrow_v(p) = p - v\langle p, v\rangle$, is the projection of p onto the hyperplane passing through the origin and perpendicular to v, where $\langle p, v\rangle$ denotes the dot-product of p and v. For two points $p, q \in \mathbb{R}^d$, let $\mathsf{d}_v(p,q) = \|\downarrow_v(p) - \downarrow_v(q)\|$ be the distance between the two projected points.*

▶ **Lemma 14.** *Let $R > \tau > 0$ be parameters. One can compute, a set $\mathcal{D}$ of directions, such that for any two points $p, q \in \mathbb{R}^d$, such that $\|pq\| \leq R$, we have that there is a direction $v \in \mathcal{D}$, such that $\mathsf{d}_v(p,q) \leq \tau$. The time to compute $\mathcal{D}$ is $O(|\mathcal{D}|) = O\big((R/\tau)^{d-1}\big)$.*

Proof. Let $\mathcal{D}$ be an ξ-packing of the unit sphere $\mathbb{S}$, for $\xi = \min(\tau/R, 1/4)$, computed by Lemma 11. As for correctness, consider the line ℓ connecting p to q. Let u be the direction vector of ℓ, and let $v \in \mathcal{D}$ be the closest direction to u. By construction, $\|uv\| \leq \xi$. Let g be the line passing through p in the direction of v, and let q' be the projection of q to this line, see Figure 3.1.

Clearly, $\mathsf{d}_v(p,q) = \|qq'\| = \|pq\| \sin \angle qpq' \leq R \sin \xi \leq R\xi \leq R\frac{\tau}{R} = \tau$, since $\|pq\| \leq R$, $\sin x \leq x$ for x positive, $\xi > 0$, and $\xi \leq \tau/R$. ◀

▶ **Lemma 15.** *Let $N \subseteq \mathbb{R}^d$ be an r-packing of some set in $\mathbb{R}^d$, for some real number r. For a parameter $R > r$, the set N can be partitioned, into $m = O\big((R/r)^d\big)$ disjoint sets $N_1, \ldots, N_m$, such that, for all i, N_i is R-separated. The partition can be computed in $O(nm)$ time, where $n = |N|$.*

[1] Confusingly r-packings are also known as r-nets, but as the r-net is an overloaded concept, we use the alternative, hopefully less-confusing, term.

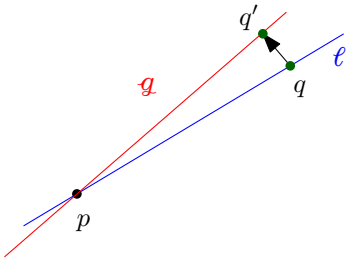

Figure 3.1 Illustration for the proof of Lemma 14.

Proof. The algorithm computes in each iteration an R-packing of N, removes its points from N, and repeat the process till N is exhausted. The bound on the number of resulting sets follows by an easy packing argument. Indeed, if a point $p \in N$ is removed in the ith iteration, then it is in distance $\leq R$ from all the previous point sets extracted (as otherwise, the earlier computed point-sets would not have been R-packings). However, the original point set N is r-separated, which implies that this can happen at most $O\big((R/r)^d\big)$ times. Since computing a net can be done in linear time, the claim follows. ◀

3.2 Constructing the ordering

▶ **Lemma 16.** *Consider a t-grid $\mathcal{G} = \mathcal{G}(\mathcal{C}, t)$ of an axis-parallel cube $\mathcal{C} \subseteq \mathbb{R}^d$, where t is a fixed positive integer. Then, one can compute a set $\mathfrak{O} = \mathfrak{O}(t, d)$ of $O(t^{d-1})$ orderings of the cells of $\mathcal{G}$, such that for any two cells $\square_1, \square_2 \in \mathcal{G}$, there exists an ordering $\sigma \in \mathfrak{O}$, where for all $\square_1 \prec_\sigma \square_3 \preccurlyeq_\sigma \square_4 \prec_\sigma \square_2$, we have:*

1. $\|\mathsf{c}(\square_3)\mathsf{c}(\square_4)\| \leq \|\mathsf{c}(\square_1)\mathsf{c}(\square_2)\|$,
2. *$\square_3$ and $\square_4$ each intersects the segment $\mathsf{c}(\square_1)\mathsf{c}(\square_2)$.*

Here $\mathsf{c}(\square)$ denotes the center point of $\square$.

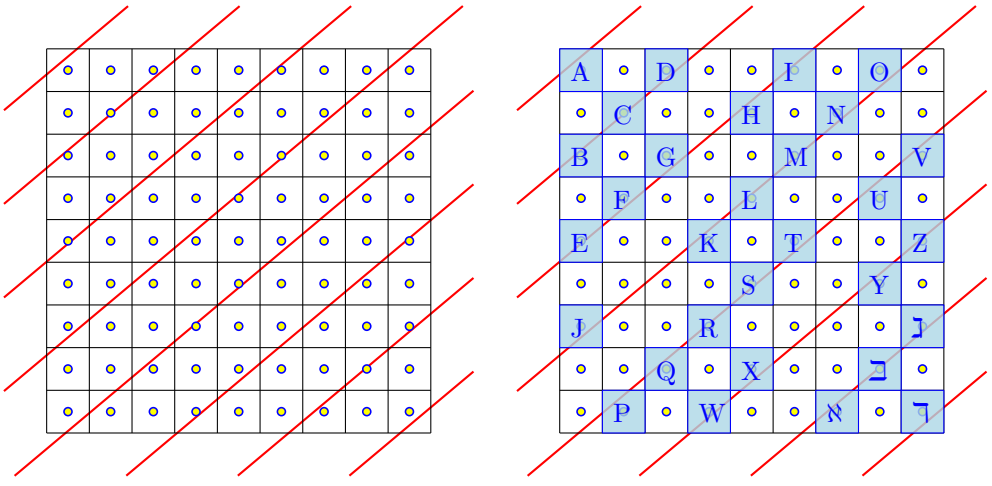

Figure 3.2 Illustration of one of the orderings of Lemma 16.

Proof. Let ζ be the sidelength of the individual cells of $\mathcal{G}$. Thus $\operatorname{sidelen}(\mathcal{C}) = t\zeta$. Let $P = \{\mathsf{c}(\square) \mid \square \in \mathcal{G}(\mathcal{C}, t)\}$, $R = \operatorname{diam}(\mathcal{C}) = t\sqrt{d}\zeta$. and let $\tau = \zeta/(8d)$. Let $\mathcal{D}$ be the set of directions as computed by the algorithm of Lemma 14 for $\tau/2$ and R. Observe that $R/(\tau/2) = O(t)$, and $|\mathcal{D}| = O(t^{d-1})$.

For each direction $v \in \mathcal{D}$, consider the projected cube $\mathcal{C}_v = \downarrow_v(\mathcal{C})$, the projected point set $P_v = \downarrow_v(P)$, and let N_v be a $\tau/2$-packing of P_v (which is a point set lying on the hyperplane perpendicular to v passing through the origin). Using Lemma 15, split N_v into ψ-separated sets $N_v^1, \ldots, N_v^m$, where $\psi = 2\sqrt{d}\zeta$, and $m = O\big((\psi/\tau)^{d-1}\big) = O(1)$ sets. Observe that for any $\square \in \mathcal{G}$, its projection $\downarrow_v(\square)$ has diameter $\leq \sqrt{d}\zeta < \psi$. Namely, no projected cell can be stabbed by two points belonging to the same ψ-separated set N_v^i. The algorithm computes an ordering for each set N_v^i. The basic idea is to start with an empty ordering, and then for each point $p \in N_v^i$, the algorithm would concatenate (say, to the end of the ordering computed so far) some of the projected cells that p stabs (in their order along the line induced by p). All the remaining unassigned cells would be concatenated to the resulting order in the end, in an arbitrary fashion.

We now describe this in more detail, for a fixed v and i. This fix a set N_v^i. For $j = 1, \ldots, \big|N_v^i\big|$, let p_j be a point of N_v^i not handled yet. The algorithm computes the oriented line ℓ_j passing through p_j in the direction of v. Let Z_j be the set of all the cells of $\mathcal{G}$ that ℓ_j stabs, and their center is in distance at most τ from ℓ_j. All such cells intersect ℓ in a "long" interval of length $\geq \zeta - 2\tau \geq (3/4)\zeta \geq 6\tau$ (that is, all the cells of Z_j have their centers "almost" on ℓ_j). Thus, the distance between the centers of the cells of Z_j in the original space, or the distance between their corresponding projections on ℓ_j are the same up to $\pm 2\tau$. In particular, we sort the cells of Z_j according to the order of their projected centers along ℓ_j, and append them (in this sorted order) to the computed order. This process is illustrated in Figure 3.2. Note, that in the end of this process, most grid cells of $\mathcal{G}$ are not included in the computed order. We append all these yet unordered cells in an arbitrary fashion to the end of the ordering. Let $\sigma(v, i)$ denote the resulting ordering of the cells of $\mathcal{G}$.

Finally, let

$$\mathfrak{O}(t, d) = \{\sigma(v, i) \mid v \in \mathcal{D}, i = 1, \ldots, m\}$$

denote the resulting set of orderings.

As for correctness, consider any two cells $\square_1, \square_2 \in \mathcal{G}$. By construction there is a direction $v \in \mathcal{D}$, such that the projected centers $\mathsf{c}_1 = \downarrow_v(\mathsf{c}(\square_1))$ and $\mathsf{c}_2 = \downarrow_v(\mathsf{c}(\square_2))$ are in distance at most $\tau/2$ from each other. In particular, let p be the closest point to either of them in the $\tau/2$-net N_v. Clearly, p is in distance at most τ from c_1 and from c_2. In particular, let i be the index, such that $p \in N_v^i$, and observe that when the algorithm handles p, say in the jth inner iteration (so $p = p_j$), it would include both $\square_1$ and $\square_2$ in the ordering (as they both belong to the set Z_j), and all the cells included in the ordering in between these two cells are in Z_j and lie in between these two cells along ℓ_j. This readily implies properties (i) and (ii). ◀

3.3 An ordering with a gap

For a segment s and a constant c, let $c\mathsf{s}$ denote the scaling of s by a factor of c around its center. We require the following additional property: If any two cells in a grid are far enough from each other, then there is an ordering, such that between the two cells in the ordering there is a "gap" which is roughly the distance between the two cells. This is essentially a weakened version of the property that the original locality sensitive orderings had.

For two sets $X, Y \subseteq \mathbb{R}^d$, let $\mathsf{d}(X, Y) = \min_{x \in X, y \in Y} \|xy\|$.

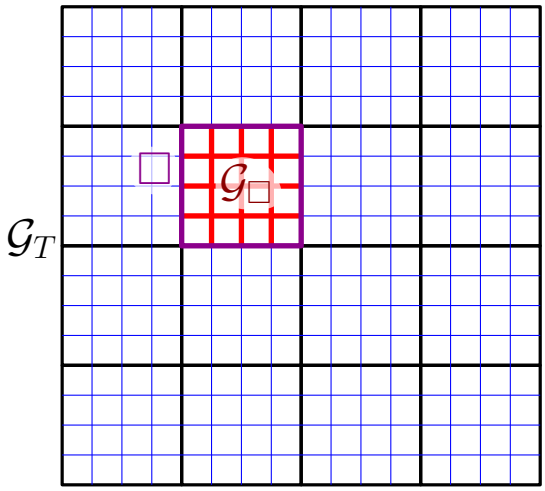

Figure 3.3

▶ **Lemma 17.** *Consider a t-grid $\mathcal{G} = \mathcal{G}(\mathcal{C}, t)$ of an axis-parallel cube $\mathcal{C} \subseteq \mathbb{R}^d$ with the cells having sidelength ζ, and a parameter $\alpha \in [\![t]\!] = \{1, \ldots, t\}$, where t is some fixed positive integer that is a power of two. Then, one can compute a set $\mathfrak{O} = \mathfrak{O}(t, \alpha, d)$ of $O(t^{2d-1}/\alpha^d)$ orderings of the cells of $\mathcal{G}$, such that for any two cells $\square_1, \square_2 \in \mathcal{G}$, with $\mathsf{d}(\square_1, \square_2) \geq \alpha\zeta$, there exists an ordering $\sigma \in \mathfrak{O}$, such that for all $\square \in \mathcal{G}$, with $\square_1 \prec_\sigma \square \prec_\sigma \square_2$, we have:*

1. *$\square$ intersects the segment $\mathsf{c}(\square_1)\mathsf{c}(\square_2)$.*
2. *$\mathsf{d}(\square, \square_1) \leq (\alpha/4)\zeta$ or $\mathsf{d}(\square, \square_2) \leq (\alpha/4)\zeta$.*

Proof. Assume $t = 2^i$ for some integer i. And let j be the largest integer, such that $\beta = 2^j < \lceil \alpha/8d \rceil$. We divide $\mathcal{C}$ into a grid of "large" cells $\mathcal{G}_T = \mathcal{G}(\mathcal{C}, t/\beta)$. Observe that every cell of $\mathcal{G}_T$ has diameter

$$\beta\sqrt{d}\zeta < \lceil \alpha/8d \rceil \sqrt{d}\zeta < (\alpha/4)\zeta,$$

and $N = |\mathcal{G}_T| = \Theta\big((t/\beta)^d\big) = \Theta\big((t/\alpha)^d\big)$. We partition every cell $\square$ of $\mathcal{G}_T$ into a secondary grid $\mathcal{G}_\square = \mathcal{G}(\square, \beta)$, see Figure 3.3. After applying this to all the grid cells of $\mathcal{G}_T$, this results in the original grid $\mathcal{G}$. Let $\mathfrak{O}_T$ be the set of orderings, of Lemma 7, of $\mathcal{G}_T$ that has the property that every two cells of $\mathcal{G}_T$ are adjacent in one of the orderings, where $|\mathfrak{O}_T| = O(N)$. Let $\mathfrak{O}_B$ be the set of orderings for the original grid $\mathcal{G}$ from Lemma 16. Consider an order $\sigma_T \in \mathcal{G}_T$ and an order $\sigma_B \in \mathfrak{O}_B$. They induce an ordering of the cells of $\mathcal{G}$ as follows – the top ordering σ_T orders the cells of $\mathcal{G}$ into secondary blocks, where each secondary block is all the cells of $\mathcal{G}$ that lie in a single cell of $\mathcal{G}_T$. Every such block is then sorted using the order σ_B. this defines a new set of orders $\mathfrak{O}$ over $\mathcal{G}$ of size $O(|\mathfrak{O}_T| \cdot |\mathfrak{O}_B|) = O(t^{2d-1}/\alpha^d)$.

Consider two cells $\square_1, \square_2 \in \mathcal{G}$, such that $\mathsf{d}(\square_1, \square_2) \geq \alpha\zeta$. Let $\square_1', \square_2' \in \mathcal{G}_T$ be the two cells containing $\square_1$ and $\square_2$, respectively. As $\square_1$ and $\square_2$ are "far" from each other, the cells $\square_1'$ and $\square_2'$ are distinct, and not adjacent in $\mathcal{G}_T$. Let σ_T be the ordering having $\square_1'$ and $\square_2'$ adjacent in the ordering. By Lemma 16, there is an order $\sigma_B \in \mathfrak{O}_B$, such that all the grid cells in between $\square_1$ and $\square_2$ in the ordering stab the segment connecting their centers. The combined ordering (σ_T, σ_B) has between $\square_1$ and $\square_2$ only cells that intersect the segment connecting their centers, that lie inside $\square_1'$ or $\square_2'$, as desired. ◀

▶ Remark 18. The number of orderings in Lemma 17 seems excessive, but it is sufficient for our application, as we are going to use $\alpha = \Omega(t)$, where the resulting total number of orderings is $O(t^{2d-1}/\alpha^d) = O(t^{d-1})$.

3.4 Result

▶ **Theorem 19.** *Let $\varepsilon, \gamma \in (0, 1/2]$ be fixed constants, such that $\gamma \geq \varepsilon$. Then, there is a set Π of $m = O\big((\mathcal{E}^{d-1} \log \mathcal{E})/\gamma^d\big)$ orderings of $[0,1)^d$, such that for any two points $p, q \in [0,1)^d$, there is an ordering $\sigma \in \Pi$ defined over $[0,1)^d$, such that*

1. *for all points u, with $p \prec_\sigma u \prec_\sigma q$, we have $\mathsf{d}(u, pq) \leq \varepsilon \|pq\|$ and $\mathsf{d}(u, \{p, q\}) \leq \gamma \|pq\|$,*
2. *for all points u, v with $p \prec_\sigma u \prec_\sigma v \prec_\sigma q$, we have $\|uv\| \leq (1+\varepsilon)\|pq\|$.*

Namely, Π is a set of m orderings that satisfy the (ε, γ)-locality property, see Definition 2.

Proof. Let $\varepsilon' = 1/2^\lambda$, for the minimum λ such that $\varepsilon' < \varepsilon/(4d^2)$. Let

$$\mathcal{E} = 1/\varepsilon', \qquad \text{and} \qquad \alpha = \max\big(1, \lfloor \gamma\mathcal{E}/(2d^2) \rfloor\big)$$

Let $\mathfrak{O} = \mathfrak{O}(\mathcal{E}, \alpha, d)$ be the set of orderings of $\mathcal{G}(\mathcal{C}, \mathcal{E})$ as defined by Lemma 17. As descried in Section 2.1.1, the set $\mathfrak{O}$ induces a set $\Pi = \Pi(\varepsilon', \alpha)$ of orderings of $[0,1)^d$, where

$$|\Pi| = O\Big(\frac{\mathcal{E}^{2d-1}}{\alpha^d}\lambda\Big) = O\Big(\frac{\mathcal{E}^{d-1}}{\gamma^d}\lambda\Big) = O\Big(\frac{1}{\varepsilon^{d-1}\gamma^d}\log\frac{1}{\varepsilon}\Big).$$

So consider any two points $p, q \in [0,1)^d$. There is a shift $\nu \in \mathcal{S}$, and an ε'-quadtree $\mathcal{T}_\varepsilon \in \mathcal{F}$, such that the LCA of p and q in $\mathcal{T}_\varepsilon$ is a node w, such that $\mathrm{sidelen}(\square_w) \leq (d+1)\|pq\|$, see Lemma 6, where $\mathrm{sidelen}(\square) = \mathrm{diam}(\square)/\sqrt{d}$ is the *sidelength* of $\square$. In particular, let $\zeta = \mathrm{sidelen}(\square_w)/\mathcal{E}$ be the sidelength of the subcells of the grid of $\mathcal{T}_\varepsilon$ for $\square_w$. Observe that $\|pq\| \geq \mathcal{E}\zeta/(d+1)$. Let $\square_1$ and $\square_2$ be the cells of the two children of w that contains p and q, respectively. In particular, we have that

$$\mathsf{d}(\square_1, \square_2) \geq \frac{\mathcal{E}\zeta}{d+1} - 2\sqrt{d}\zeta \geq \alpha\zeta.$$

Now, Lemma 17 guarantees the existence of an ordering $\sigma \in \mathfrak{O}(\alpha, \mathcal{E}, d)$ that stabs $\square_1$ and $\square_2$, and all the cells in between in this ordering are "$(\alpha/4)$-close" to either $\square_1$ or $\square_2$. In particular, the ordering of $[0,1)^d$ induced by $\nu, \mathcal{T}_\varepsilon$ and σ has the desired property. ◀

▶ Remark 20. Computing the orderings of Theorem 19 can be done in $\mathcal{E}^{O(d)}$ time, and since this is a preprocessing stage, and conceptually we consider ε to be a constant, we ignore this preprocessing time in our statement. This results in $m = O\big((\mathcal{E}^{d-1}\log\mathcal{E})/\gamma^d\big)$ orderings, each order requires $O(\mathcal{E}^d)$ space to store it. In the following, γ is a fixed "large" constant (e.g., $\gamma = 1/8$), so the overall space to store these orderings is $O(\mathcal{E}^{2d-1}\log\mathcal{E})$. Given an order σ and two points, comparing the two points according to σ requires answering a single LCA query on a quadtree, see Section 2.1.1. With the appropriate bit operations such a query can be carried out in $O(\log\mathcal{E})$ time [6].

4 Applications

4.1 Locality graph

Given an ordering σ of Theorem 19, one can maintain a sorted list of n points in $[0,1)^d$ in $O(\log n \log\mathcal{E})$ time per insertion/deletion. Indeed, using any standard balanced binary search tree storing n elements, requires $O(\log n)$ time per operation, and each such operation performs $O(\log n)$ comparisons. Each comparison takes $O(\log\mathcal{E})$ time to perform, see Remark 20. In particular, a natural approach is to maintain a dynamic graph, over the (evolving) point set P, where two points are connected by an edge $\iff$ the two points are adjacent in one of the orderings of Π provided by Theorem 19. Note that any insertion/deletion would cause $O(\mathcal{E}^{d-1}\log\mathcal{E})$ edges to be inserted/deleted to this locality graph of P.

▶ **Definition 21.** *For parameters $\varepsilon, \gamma \in (0,1)$, and a set of points $P \subseteq [0,1)^d$, let $\mathcal{L} = \mathcal{L}(P, \varepsilon, \gamma)$ be the* **locality graph***, described above, for the set of orderings Π computed by Theorem 19.*

▶ **Lemma 22.** *For $\varepsilon \in (0,1)$ and $\gamma \in (1/32, 1)$ (with $\gamma \geq \varepsilon$), the locality graph $\mathcal{L} = \mathcal{L}(P, \varepsilon, \gamma)$ defined for a set P of n points in $[0,1)^d$, has $O(\Psi n)$ edges, where $\Psi = O(\mathcal{E}^{d-1} \log \mathcal{E})$. An insertion/deletion into P can be performed in $O\big((\Psi \log \mathcal{E}) \log n\big)$ time, and involve the deletion/insertion of at most Ψ edges of $\mathcal{L}$.*

4.2 Bichromatic closest pair

▶ **Theorem 23.** *Given two sets of points, R and B, both subsets of $[0,1)^d$, and a parameter $\varepsilon \in (0, 1/2)$, one can maintain a $(1+\varepsilon)$-approximation to the closest bichromatic pair $R \times B$. Each insertion/deletion takes $O(\mathcal{E}^{d-1} \log^2 \mathcal{E} \cdot \log n)$ time per operation, where n is the maximum size of $|R| + |B|$ over time, and $\mathcal{E} = 1/\varepsilon$. This data-structure use $O(\Psi n)$ space, where $\Psi = O(\mathcal{E}^{d-1} \log \mathcal{E})$. At all times, it maintains a pair of points $r \in R$ and $b \in B$, such that $\|rb\| \leq (1+\varepsilon)\mathsf{d}(R, B)$, where $\mathsf{d}(R, B) = \min_{b \in B, r \in R} \|br\|$.*

See [9] for the proof.

4.3 Dynamic spanners

For a set of points P, we show next, that the locality graph $\mathcal{L} = \mathcal{L}(P, \varepsilon/32, 1/8)$ is an $(1+\varepsilon)$-spanner. Formally, for two points $x, y \in P$, such that xy is an edge of $\mathcal{L}$, its weight is $\|xy\|$. For any two points $x, y \in P$, let $\mathscr{g}(x, y)$ denote the length of their shortest path in $\mathcal{L}$. The graph $\mathcal{L}$ is ***$(1+\varepsilon)$-spanner***, if for all $x, y \in P$, we have $\mathscr{g}(x, y) \leq (1+\varepsilon)\|xy\|$. In the following, for $X, Y \subseteq P$, let $\mathscr{g}(X, Y) = \min_{x \in X, y \in Y} \mathscr{g}(x, y)$.

▶ **Theorem 24.** *Let $P \subseteq [0,1)^d$ be a set of n points, and let $\varepsilon \in (0,1)$ be a parameter. The locality graph $\mathcal{L} = \mathcal{L}(P, \varepsilon/32, 1/8)$ is an $(1+\varepsilon)$-spanner of P. The graph $\mathcal{L}$ has $O(n\mathcal{E}^{d-1} \log \mathcal{E})$ edges. Updating $\mathcal{L}$, after an insertion or a deletion of a point, takes $O\big((\mathcal{E}^{d-1} \log^2 \mathcal{E}) \log n\big)$ time. Each such operation causes at most $O(\mathcal{E}^{d-1} \log \mathcal{E})$ edges to be removed from or inserted into $\mathcal{L}$, and this also bounds the maximum degree of a vertex of $\mathcal{L}$.*

Proof. Let Π be the set of orderings used to construct $\mathcal{L}$ – specifically, it is the one provided by Theorem 19 so that the orderings of Π are $(\varepsilon/32, 1/8)$-local.

We prove the spanner property by induction on the distance of pairs of points of P. It is easy to verify, arguing as in Theorem 23, that the closest pair of points of P are connected by an edge in the locality graph $\mathcal{L}$, which establish the base of the induction.

Fix a pair $p, q \in P$ and assume by the induction hypothesis that for all pairs $x, y \in P$, such that $\|xy\| < \|pq\|$, we have that $\mathscr{g}(x, y) \leq (1+\varepsilon)\|xy\|$. Let $\sigma \in \Pi$ be the order in Π that is $(\varepsilon/32, 1/8)$-local for p and q,

Let $L = \|pq\|$. The set $P' = P \cap \sigma[p, q]$ is contained in the union of the two sets

$$C = \mathscr{b}(p, \gamma L) \cap \mathscr{h}\big(pq, \xi L\big) \qquad \text{and} \qquad D = \mathscr{b}(q, \gamma L) \cap \mathscr{h}\big(pq, \xi L\big),$$

where $\gamma = 1/8$ and $\xi = \varepsilon/32$, see Figure 4.1 and Figure 1.1 (iii). We consider P' to be sorted by σ. There must be two consecutive points $u, v \in P'$, such that $u \in C$ and $v \in D$. By construction, uv is an edge of $\mathcal{L}$. By the gap property, we have that $\|uv\| \geq \mathsf{d}(C, D) = (1 - 2\gamma)\|pq\| \geq (3/4)L$.

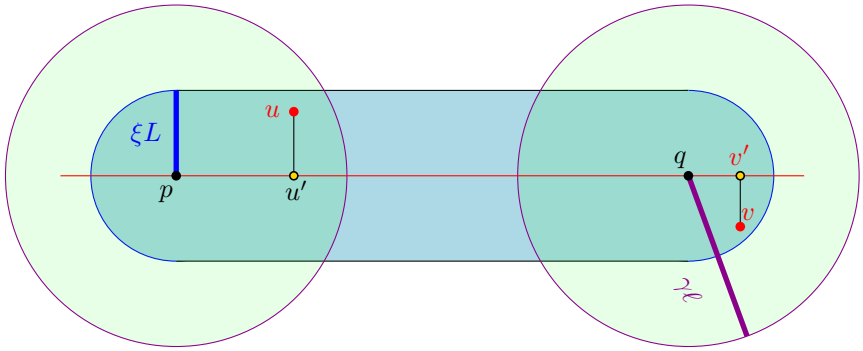

Figure 4.1 Illustration for the proof of Theorem 24.

Let u', v' be the projections of u, v, respectively, on the line spanned by pq. Observe that $\|uu'\| \leq \xi L$, and $\|vv'\| \leq \xi L$. As such, we have

$$\begin{aligned}
\not{g}(p,q) &\leq \not{g}(p,u) + \not{g}(u,v) + \not{g}(v,q) \leq (1+\varepsilon)\|pu\| + \|uv\| + (1+\varepsilon)\|vq\| \\
&= (1+\varepsilon)(\|pu\| + \|uv\| + \|vq\|) - \varepsilon\|uv\| \\
&\leq (1+\varepsilon)(\|pu'\| + \|u'v'\| + \|v'q\| + 4\xi L) - (3/4)\varepsilon L.
\end{aligned}$$

Observe that p, u', v', q are all on the same line (spanned by p and q), and are contained in the hippodrome $\hbar(pq, \xi L)$, the point u' is closer to p than to q, and similarly v' is closer to q than to p. It follows hat $\|pu'\| + \|u'v'\| + \|v'q\| \leq \|pq\| + 4\xi L$. Thus, resuming the above calculation, we have

$$\begin{aligned}
\not{g}(p,q) &\leq (1+\varepsilon)(\|pq\| + 8\xi L) - (3/4)\varepsilon L = (1+\varepsilon)(L + 8\xi L) - (3/4)\varepsilon L \\
&\leq (1+\varepsilon)L + 16\xi L - (3/4)\varepsilon L \\
&= (1+\varepsilon)L + (1/2 - 3/4)\varepsilon L \leq (1+\varepsilon)L.
\end{aligned}$$

◀

▶ Remark 25. Observe that the gap property is critical in making the proof of Theorem 24 work – we need a long "bridge" edge that the spanner uses, to charge the inductive error to.

5 Lower bounds on the number of orderings

5.1 Lower bound of number of LSOs with the ε-local property

▶ **Lemma 26.** *For any fixed integer d, and $\varepsilon \in (0,1)$, there is a set P of $\Theta(\mathcal{E}^d)$ points in $\mathbb{R}^d$, such that any set of Π of LSOs of P with the ε-local property, see Definition 1, must be of size $\Omega(\mathcal{E}^d)$, where $\mathcal{E} = 1/\varepsilon$.*

See [9] for the proof.

5.2 Lower bound for LSOs with the (ε, γ)-local property

▶ **Theorem 27.** *For any fixed integer d, and $\varepsilon \in (0,1)$, there exists a set of point P, of size $\Omega(\mathcal{E}^{d-1})$, such that any set Π of LSOs for P with (ε,γ)-local property must be of size $\Omega(\mathcal{E}^{d-1})$, for any $\gamma \geq \varepsilon$.*

See [9] for the proof.

5.3 Lower bound for LSOs that their locality graph is a spanner

Le et al. [13] showed that any ε-spanner in $\mathbb{R}^d$ of n points has $\Omega(n/\varepsilon^{d-1})$ edges. Each LSO gives rise to $n-1$ edges in the locality graph. As such, one needs at least $\Omega(1/\varepsilon^{d-1})$ LSOs to construct an ε-spanner using the locality graph.

▶ **Lemma 28.** *Any set of LSOs that their locality graph is an* $(1+\varepsilon)$*-spanner has size* $\Omega(1/\varepsilon^{d-1})$.

References

1 Brian Alspach. The wonderful Walecki construction. *Bull. Inst. Combin. Appl.*, 52:7–20, 2008.
2 Kevin Buchin, Sariel Har-Peled, and Dániel Oláh. A spanner for the day after. *Discrete Comput. Geom.*, 64(4):1167–1191, 2020. doi:10.1007/s00454-020-00228-6.
3 Kevin Buchin, Sariel Har-Peled, and Dániel Oláh. Sometimes reliable spanners of almost linear size. *J. Comput. Geom.*, 13(1):178–196, 2022. doi:10.20382/jocg.v13i1a6.
4 P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. Assoc. Comput. Mach.*, 42(1):67–90, 1995. doi:10.1145/200836.200853.
5 T. M. Chan. Approximate nearest neighbor queries revisited. *Discrete Comput. Geom.*, 20(3):359–373, 1998. doi:10.1007/PL00009390.
6 Timothy M. Chan, Sariel Har-Peled, and Mitchell Jones. On locality-sensitive orderings and their applications. *SIAM J. Comput.*, 49(3):583–600, 2020. doi:10.1137/19M1246493.
7 Arnold Filtser. Labeled nearest neighbor search and metric spanners via locality sensitive orderings. In Erin W. Chambers and Joachim Gudmundsson, editors, *Proc. 39th Int. Annu. Sympos. Comput. Geom.* (SoCG), volume 258 of *LIPIcs*, pages 33:1–33:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.SoCG.2023.33.
8 Arnold Filtser and Hung Le. Locality-sensitive orderings and applications to reliable spanners. In *Proc. 54rd Annu. ACM Sympos. Theory Comput.* (STOC), STOC 2022, pages 1066–1079, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3520042.
9 Zhimeng Gao and Sariel Har-Peled. Almost optimal locality sensitive orderings in euclidean space. *CoRR*, abs/2310.12792, 2023. doi:10.48550/arXiv.2310.12792.
10 S. Har-Peled. *Geometric Approximation Algorithms*, volume 173 of *Math. Surveys & Monographs*. Amer. Math. Soc., Boston, MA, USA, 2011. doi:10.1090/surv/173.
11 Sariel Har-Peled and Benjamin Raichel. Net and prune: A linear time algorithm for Euclidean distance problems. *J. Assoc. Comput. Mach.*, 62(6):44:1–44:35, December 2015. doi:10.1145/2831230.
12 Sándor Kisfaludi-Bak and Geert van Wordragen. A quadtree for hyperbolic space. *CoRR*, abs/2305.01356, 2023. doi:10.48550/arXiv.2305.01356.
13 Hung Le and Shay Solomon. Truly optimal euclidean spanners, 2021. arXiv:1904.12042.