Locality-Sensitive Orderings & Their Applications

Timothy Chan, Sariel Har-Peled, Mitchell Jones

September 11, 2018

University of Illinois at Urbana-Champaign

Orderings: Motivation

- Computing orderings:
  - Travelling salesman problem (hard)
  - Degeneracy of a graph (easy)

Orderings: Motivation

- Computing orderings:
  - Travelling salesman problem (hard)
  - Degeneracy of a graph (easy)
- Orderings are 1D embeddings
- Embedding into simpler structures

Orderings: Motivation

- Computing orderings:
  - Travelling salesman problem (hard)
  - Degeneracy of a graph (easy)
- Orderings are 1D embeddings
- Embedding into simpler structures
- E.g. $O(\log n)$-apx for k-median clustering: Metric space $\rightarrow$ tree

Orderings: Motivation

- Computing orderings:
  - Travelling salesman problem (hard)
  - Degeneracy of a graph (easy)
- Orderings are 1D embeddings
- Embedding into simpler structures
- E.g. $O(\log n)$-apx for k-median clustering: Metric space $\rightarrow$ tree
- Tradeoff between soln. quality and simpler algorithms

Orderings: Motivation

- Computing orderings:
  - Travelling salesman problem (hard)
  - Degeneracy of a graph (easy)
- Orderings are 1D embeddings
- Embedding into simpler structures
- E.g. $O(\log n)$-apx for k-median clustering: Metric space $\rightarrow$ tree
- Tradeoff between soln. quality and simpler algorithms
- In this talk: Orderings of points with special properties

Main Theorem

Main Theorem

For $\varepsilon \in (0,1)$, there is a set Π of size $O((1/\varepsilon^d)\log(1/\varepsilon))$ such that $\forall p, q \in [0,1)^d$, there exists $\sigma \in \Pi$ with:

Points between p and q in σ are distance at most $\varepsilon\|p - q\|$ from p or q.

Main Theorem

Main Theorem

For $\varepsilon \in (0,1)$, there is a set Π of size $O((1/\varepsilon^d)\log(1/\varepsilon))$ such that $\forall p, q \in [0,1)^d$, there exists $\sigma \in \Pi$ with:

Points between p and q in σ are distance at most $\varepsilon\|p - q\|$ from p or q.

Some applications

- New: $(1+\varepsilon)$-apx bichromatic closest pair

Main Theorem

Main Theorem

For $\varepsilon \in (0,1)$, there is a set Π of size $O((1/\varepsilon^d)\log(1/\varepsilon))$ such that $\forall p, q \in [0,1)^d$, there exists $\sigma \in \Pi$ with:

Points between p and q in σ are distance at most $\varepsilon\|p-q\|$ from p or q.

Some applications

- New: $(1+\varepsilon)$-apx bichromatic closest pair
- Simpler: dynamic $(1+\varepsilon)$-spanners

Main Theorem

Main Theorem

For $\varepsilon \in (0,1)$, there is a set Π of size $O((1/\varepsilon^d)\log(1/\varepsilon))$ such that $\forall p, q \in [0,1)^d$, there exists $\sigma \in \Pi$ with:

Points between p and q in σ are distance at most $\varepsilon\|p-q\|$ from p or q.

Some applications

- New: $(1+\varepsilon)$-apx bichromatic closest pair
- Simpler: dynamic $(1+\varepsilon)$-spanners
- New: dynamic k-vertex-fault-tolerant $(1+\varepsilon)$-spanners

Main Theorem

Main Theorem

For $\varepsilon \in (0,1)$, there is a set Π of size $O((1/\varepsilon^d)\log(1/\varepsilon))$ such that $\forall p, q \in [0,1)^d$, there exists $\sigma \in \Pi$ with:

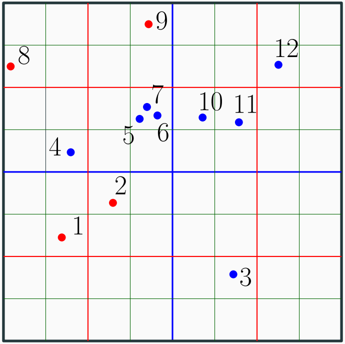
Points between p and q in σ are distance at most $\varepsilon\|p-q\|$ from p or q.

Some applications

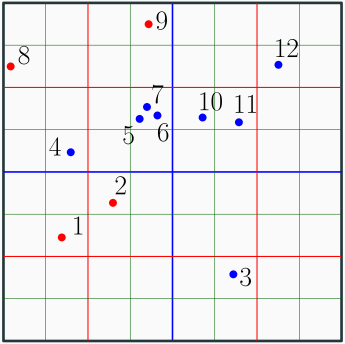
- New: $(1+\varepsilon)$-apx bichromatic closest pair
- Simpler: dynamic $(1+\varepsilon)$-spanners
- New: dynamic k-vertex-fault-tolerant $(1+\varepsilon)$-spanners
- ...

Warmup: Constant factor approximation for bichromatic closest pair

Bichromatic closest pair



Bichromatic closest pair

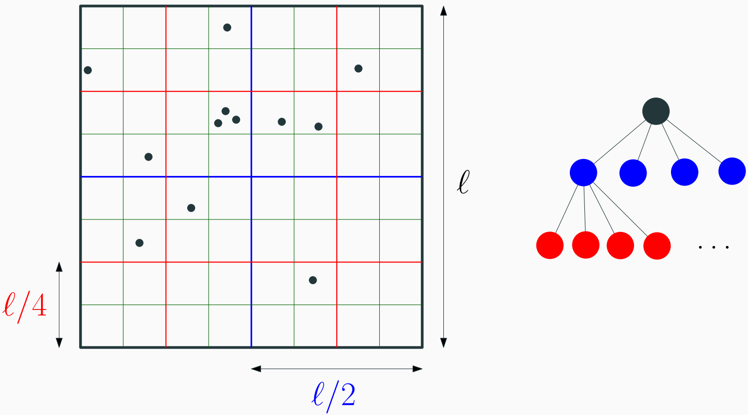


Problem (c-approximation)

Maintain a pair (r', b') s.t. $\|r' - b'\| \leq c \cdot \min_{(r,b)} \|r - b\|$.

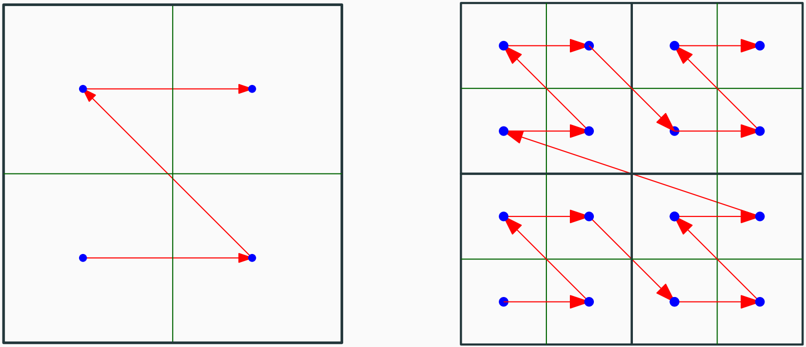
Preliminaries: Quadtrees

"Hierarchy of grids"



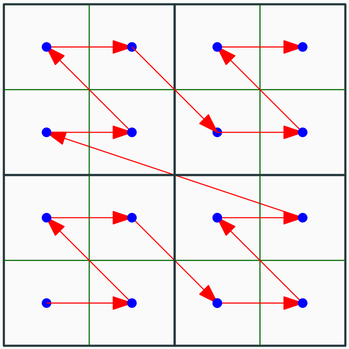
Preliminaries: $\mathbb{Z}$-order

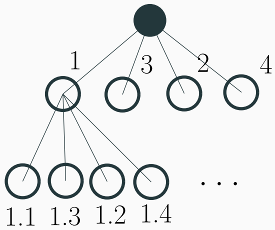
Mapping points into 1D



Preliminaries: Quadtrees and $\mathbb{Z}$-order

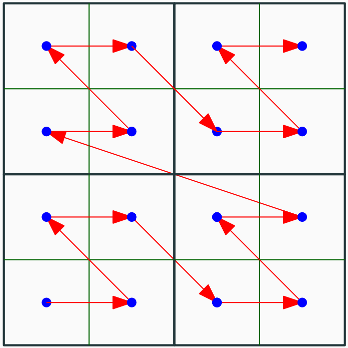
- DFS of a quadtree produces a $\mathbb{Z}$-order

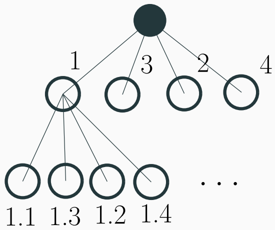




Preliminaries: Quadtrees and $\mathcal{Z}$-order

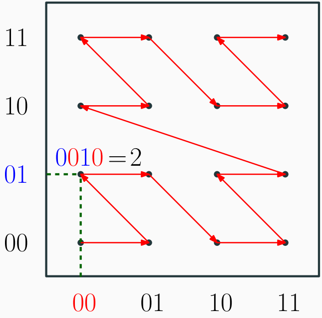
- DFS of a quadtree produces a $\mathcal{Z}$-order
- Only need to specify an order on 4 cells (or 2^d for higher dimensions)





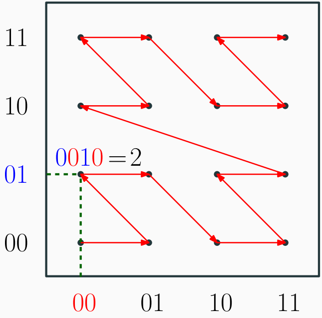
Preliminaries: Computing the $\mathbb{Z}$-order

- Let $p = (x, y) \in [2^w] \times [2^w]$
- $x = x_w x_{w-1} \dots x_1$
- $y = y_w y_{w-1} \dots y_1$



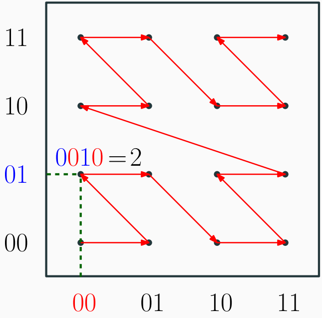
Preliminaries: Computing the $\mathbb{Z}$-order

- Let $p = (x, y) \in [2^w] \times [2^w]$
- $x = x_w x_{w-1} \dots x_1$
- $y = y_w y_{w-1} \dots y_1$
- $\text{shuffle}(p) = y_w x_w y_{w-1} x_{w-1} \dots y_1 x_1$
- Position of p in $\mathbb{Z}$-order = shuffle(p)



Preliminaries: Computing the $\mathbb{Z}$-order

- Let $p = (x, y) \in [2^w] \times [2^w]$
- $x = x_w x_{w-1} \ldots x_1$
- $y = y_w y_{w-1} \ldots y_1$
- $\text{shuffle}(p) = y_w x_w y_{w-1} x_{w-1} \ldots y_1 x_1$
- Position of p in $\mathbb{Z}$-order $= \text{shuffle}(p)$

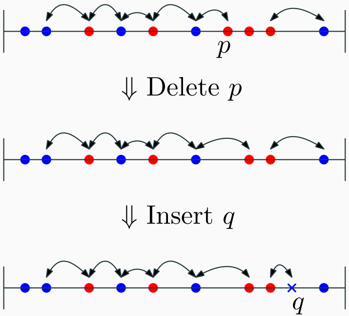


Lemma

shuffle(p) and shuffle(q) can be compared in $O(1)$ and/exclusive-or operations.

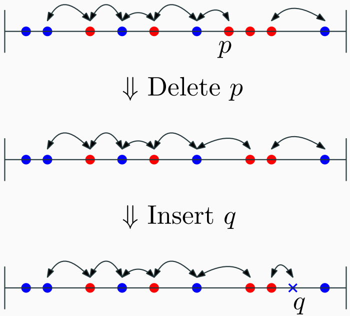
Solving the problem in 1D: A solution?

- Map the point set to 1D



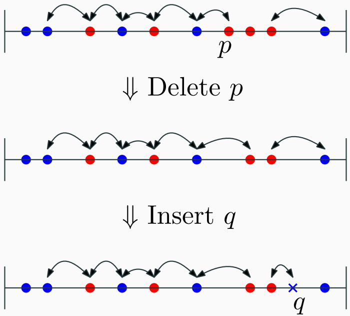
Solving the problem in 1D: A solution?

- Map the point set to 1D
- Maintain sorted order in binary tree



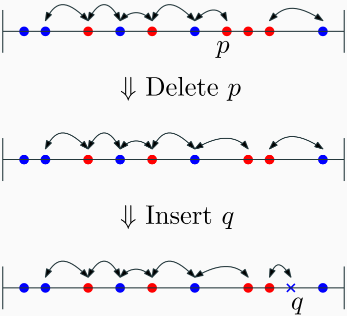
Solving the problem in 1D: A solution?

- Map the point set to 1D
- Maintain sorted order in binary tree
- Maintain min-heap of consecutive red/blue pairs



Solving the problem in 1D: A solution?

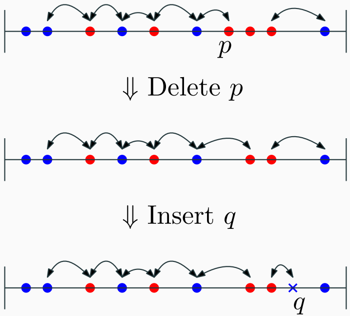
- Map the point set to 1D
- Maintain sorted order in binary tree
- Maintain min-heap of consecutive red/blue pairs
- Updates change $O(1)$ consecutive pairs



Solving the problem in 1D: A solution?

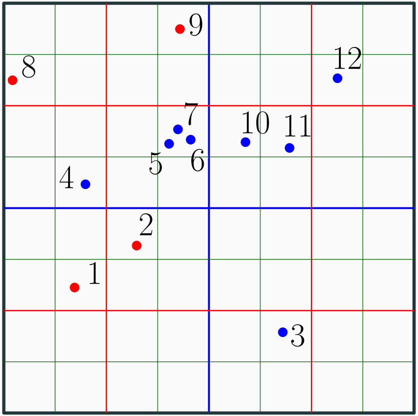
- Map the point set to 1D
- Maintain sorted order in binary tree
- Maintain min-heap of consecutive red/blue pairs
- Updates change $O(1)$ consecutive pairs

$\implies$ Update time $O_d(\log n)$



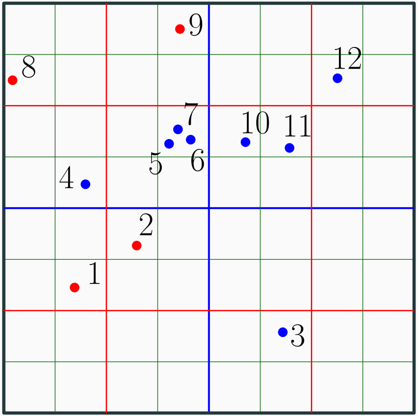
Not quite a solution

- Points nearby in $\mathbb{R}^d$ $\not\Rightarrow$ nearby in $\mathbb{Z}$-order



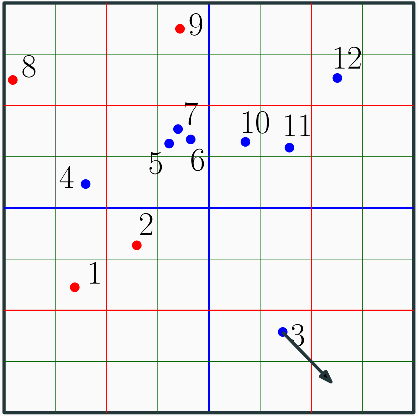
Not quite a solution

- Points nearby in $\mathbb{R}^d \not\Rightarrow$ nearby in $\mathbb{Z}$-order
- Idea: Shift the point set



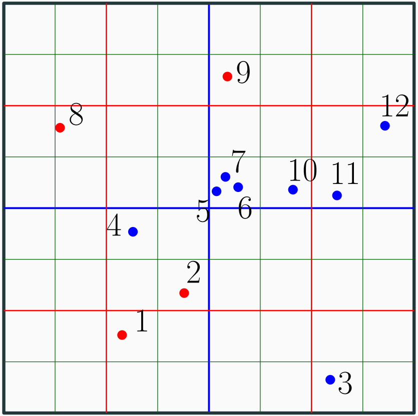
Not quite a solution

- Points nearby in $\mathbb{R}^d$ $\not\Rightarrow$ nearby in $\mathcal{Z}$-order
- Idea: Shift the point set



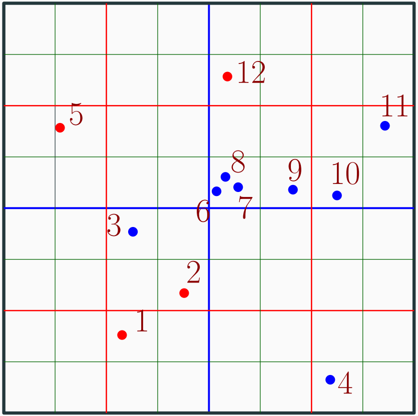
Not quite a solution

- Points nearby in $\mathbb{R}^d \not\Rightarrow$ nearby in $\mathcal{Z}$-order
- Idea: Shift the point set



Not quite a solution

- Points nearby in $\mathbb{R}^d \not\Rightarrow$ nearby in $\mathbb{Z}$-order
- Idea: Shift the point set



Preliminaries: Shifting

Lemma [Chan '98]

For $i = 0, \ldots, d$, let $v_i = (i/(d+1), \ldots, i/(d+1))$.

Let $p, q \in [0,1)^d$ and $\mathcal{T}$ be a quadtree over $[0,2)^d$.

There exists $i \in \{0, \ldots, d\}$ and $\square \in \mathcal{T}$:

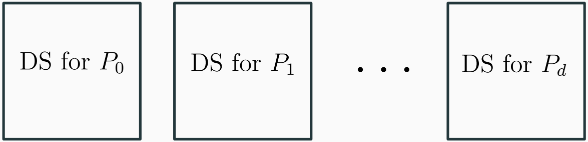
1. $p + v_i, q + v_i \in \square$
2. $(d+1)\|p - q\| < \text{sidelength}(\square) \leq 2(d+1)\|p - q\|$.

A correct solution

- Shift point set $d+1$ times: $P_0, \ldots, P_d$

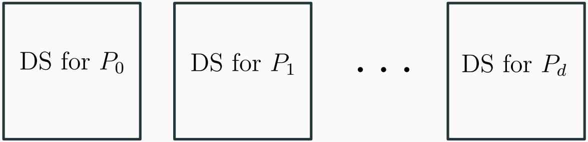
A correct solution

- Shift point set $d+1$ times: $P_0, \ldots, P_d$



A correct solution

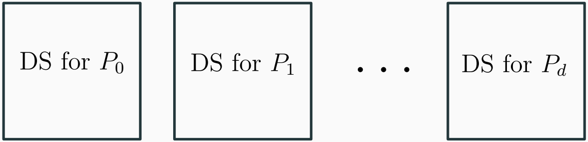
- Shift point set $d+1$ times: $P_0, \ldots, P_d$



$\implies$ $O_d(\log n)$ update time

A correct solution

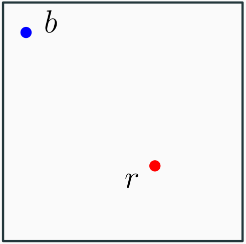
- Shift point set $d + 1$ times: $P_0, \ldots, P_d$



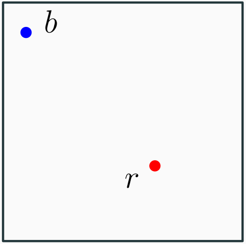
$\implies$ $O_d(\log n)$ update time

- Claim: $O_d(1)$ approximation

Correctness (cont.)

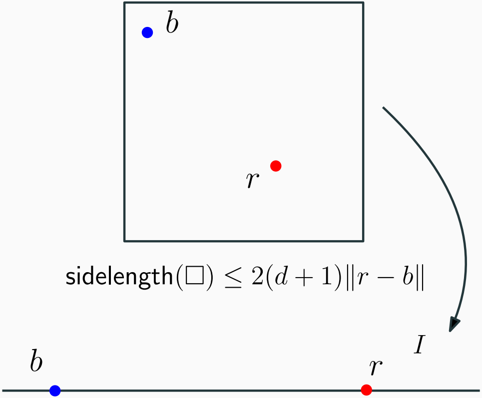


Correctness (cont.)



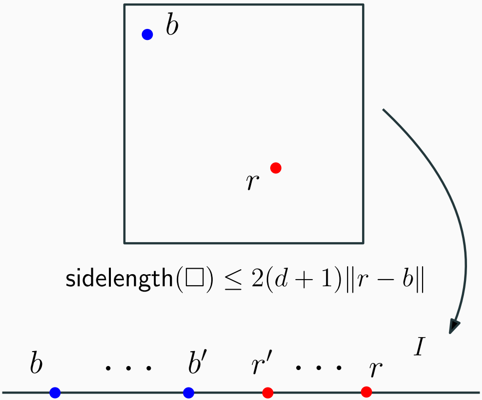
sidelength($\Box$) $\leq 2(d+1)\|r-b\|$

Correctness (cont.)

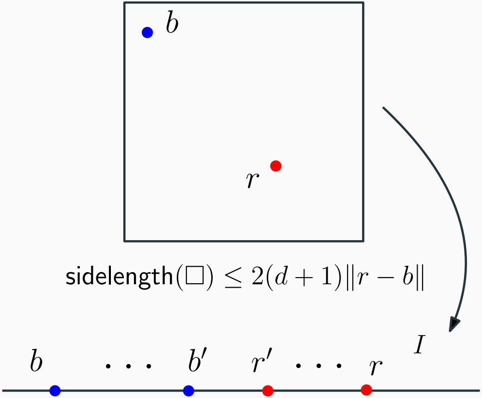


sidelength(□) $\leq 2(d+1)\|r-b\|$

Correctness (cont.)



Correctness (cont.)



sidelength(□) $\leq 2(d+1)\|r-b\|$

$\|r'-b'\| \leq$ diam(□) $\leq \sqrt{d} \cdot$ sidelength(□) $= O_d(1)\|r-b\|$

$(1+\varepsilon)$-approximate bichromatic closest pair

Reducing the approximation factor

- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$

Reducing the approximation factor

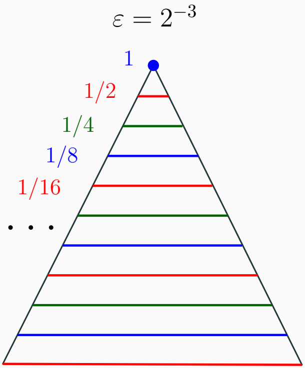
- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
- Idea: Pack many "ε-quadtrees" into a regular quadtree

Reducing the approximation factor

- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
- Idea: Pack many "ε-quadtrees" into a regular quadtree
- ε-quadtrees have $1/\varepsilon^d$ children

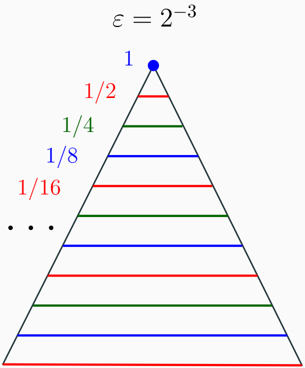
Reducing the approximation factor

- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
- Idea: Pack many "ε-quadtrees" into a regular quadtree
- ε-quadtrees have $1/\varepsilon^d$ children



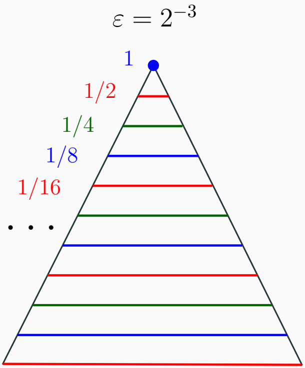
Reducing the approximation factor

- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
- Idea: Pack many "ε-quadtrees" into a regular quadtree
- ε-quadtrees have $1/\varepsilon^d$ children
- "Partitions" a regular quadtree into $\lg(1/\varepsilon)$ ε-quadtrees



Reducing the approximation factor

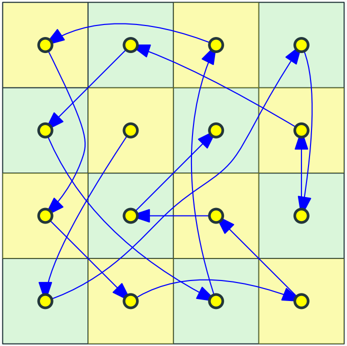
- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
- Idea: Pack many "ε-quadtrees" into a regular quadtree
- ε-quadtrees have $1/\varepsilon^d$ children
- "Partitions" a regular quadtree into $\lg(1/\varepsilon)$ ε-quadtrees
- Call them $\mathcal{T}_\varepsilon^1, \ldots, \mathcal{T}_\varepsilon^E$



$O(1)$ problems

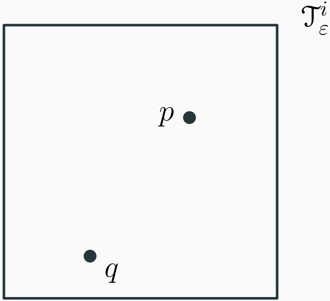
Extend $\mathcal{Z}$-order to ε-quadtrees by ordering $1/\varepsilon^{d}$ child cells

10	6	9	3
7	1	16	5
11	15	14	4
2	12	8	13



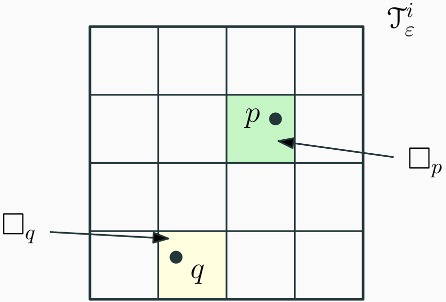
What $\mathcal{Z}$-order should we pick?

$O(1)$ problems (cont.)



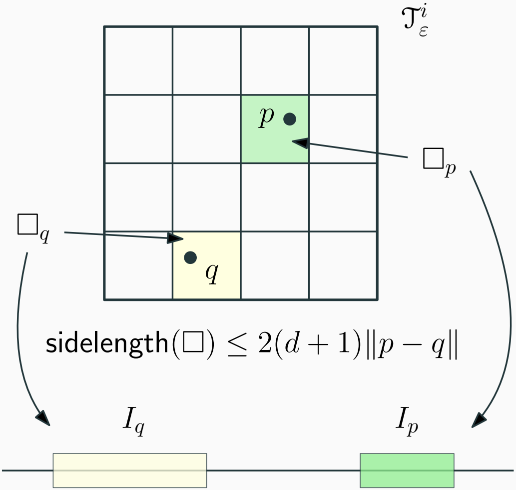
sidelength$(\square) \leq 2(d+1)\|p-q\|$

$O(1)$ problems (cont.)



$$\text{sidelength}(\square) \leq 2(d+1)\|p-q\|$$

$O(1)$ problems (cont.)



sidelength$(\square) \leq 2(d+1)\|p-q\|$

Ordering quadtree cells

Idea

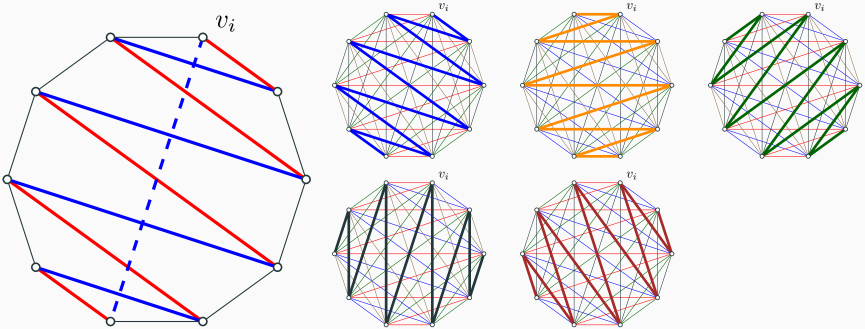
Pick a set $\mathfrak{O}$ of orderings of the $1/\varepsilon^d$ cells such that:

For any $\square_1, \square_2$, there is an ordering $\sigma \in \mathfrak{O}$ with $\square_1$ adjacent to $\square_2$

A necessary subproblem

Lemma [Alspach '08]

For n elements $\{0, \ldots, n-1\}$, there is a set $\mathfrak{O}$ of $\lceil n/2 \rceil$ orderings of the elements, such that, for all $i, j \in \{0, \ldots, n-1\}$, there exist an ordering $\sigma \in \mathfrak{O}$ in which i and j are adjacent.



Ordering quadtree cells

Corollary

There is a set $\mathfrak{O}(1/\varepsilon)$ of $O(1/\varepsilon^d)$ orderings, such that for any $\square_1, \square_2$, there exists an order $\sigma \in \mathfrak{O}(1/\varepsilon)$ where $\square_1$ and $\square_2$ are adjacent in σ.

What we have so far

- $d + 1$ shifted point sets

What we have so far

- $d+1$ shifted point sets
- $\lg(1/\varepsilon)$ ε-quadtrees

What we have so far

- $d + 1$ shifted point sets
- $\lg(1/\varepsilon)$ ε-quadtrees
- $O(1/\varepsilon^d)$ orderings

What we have so far

- $d + 1$ shifted point sets
- $\lg(1/\varepsilon)$ ε-quadtrees
- $O(1/\varepsilon^d)$ orderings

$\implies O_d\big((1/\varepsilon^d)\log(1/\varepsilon)\big)$ different orderings of P

What we have so far

- $d + 1$ shifted point sets
- $\lg(1/\varepsilon)$ ε-quadtrees
- $O(1/\varepsilon^d)$ orderings

  $\implies O_d\big((1/\varepsilon^d)\log(1/\varepsilon)\big)$ different orderings of P
- Let Π denote these set of orderings

What we have so far

Main Theorem

For $\varepsilon \in (0, 1)$, there is a set Π of size $O((1/\varepsilon^d) \log(1/\varepsilon))$ such that $\forall p, q \in [0, 1)^d$, there exists $\sigma \in \Pi$ with:

Points between p and q in σ are distance at most $\varepsilon\|p - q\|$ from p or q.

What we have so far

Main Theorem

For $\varepsilon \in (0,1)$, there is a set Π of size $O((1/\varepsilon^d)\log(1/\varepsilon))$ such that $\forall p, q \in [0,1)^d$, there exists $\sigma \in \Pi$ with:

Points between p and q in σ are distance at most $\varepsilon\|p-q\|$ from p or q.

Lemma

Let $p, q \in [0,1)^d$ and $\sigma \in \Pi$. Can decide if $p \prec_\sigma q$ using $O_d(\log(1/\varepsilon))$ bitwise-logical operations.

The solution

- Maintain the 1D data structure for all orderings Π

The solution

- Maintain the 1D data structure for all orderings Π
- $|\Pi| = O((1/\varepsilon^d)\log(1/\varepsilon))$

The solution

- Maintain the 1D data structure for all orderings Π
- $|\Pi| = O((1/\varepsilon^d)\log(1/\varepsilon))$
- Update time: $O(|\Pi| \cdot \log(n) \cdot \log(1/\varepsilon)) =$
  $O_d((1/\varepsilon^d)\log(n)\log^2(1/\varepsilon))$

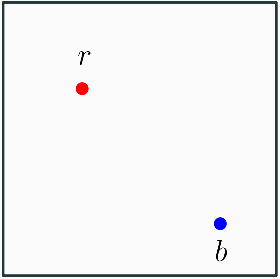
The solution

- Maintain the 1D data structure for all orderings Π
- $|\Pi| = O((1/\varepsilon^d)\log(1/\varepsilon))$
- Update time: $O(|\Pi| \cdot \log(n) \cdot \log(1/\varepsilon)) = O_d((1/\varepsilon^d)\log(n)\log^2(1/\varepsilon))$
- Space: $O(|\Pi| \cdot n) = O_d((n/\varepsilon^d)\log(1/\varepsilon))$

The solution

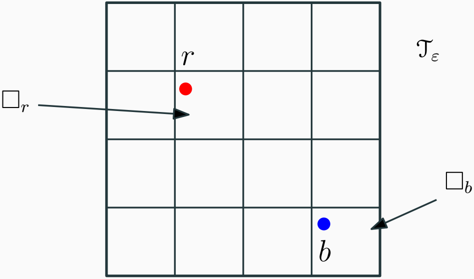
- Maintain the 1D data structure for all orderings Π
- $|\Pi| = O((1/\varepsilon^d)\log(1/\varepsilon))$
- Update time: $O(|\Pi| \cdot \log(n) \cdot \log(1/\varepsilon)) =$ $O_d((1/\varepsilon^d)\log(n)\log^2(1/\varepsilon))$
- Space: $O(|\Pi| \cdot n) = O_d((n/\varepsilon^d)\log(1/\varepsilon))$
- Claim: Maintains r', b' with $\|r' - b'\| \leq (1+\varepsilon)\|r - b\|$

Correctness



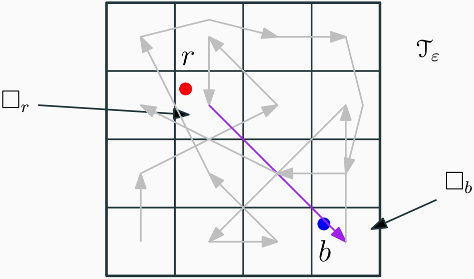
sidelength($\square$) $\leq 2(d+1)\|r-b\|$

Correctness



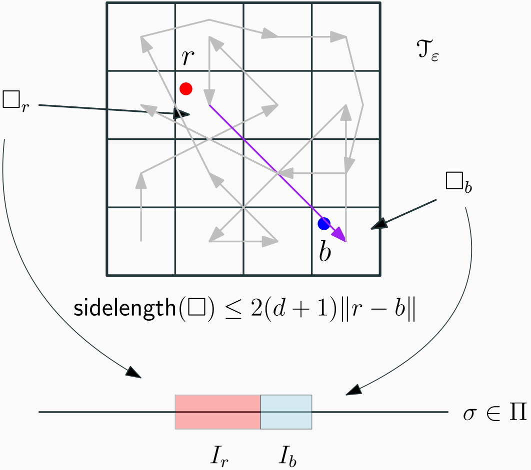
sidelength$(\square) \leq 2(d+1)\|r-b\|$

Correctness

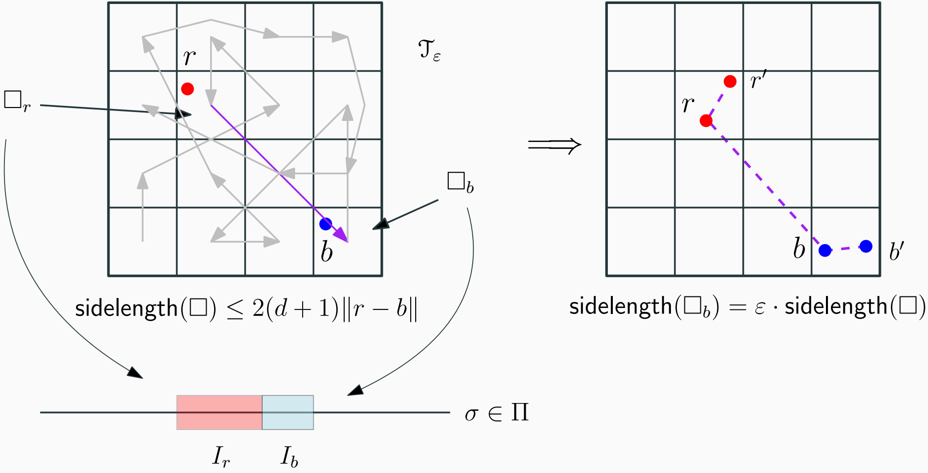


sidelength$(\square) \leq 2(d+1)\|r-b\|$

Correctness



Correctness



A simple data structure for dynamic $(1+\varepsilon)$-spanners

Spanners

Definition
For a set n of P points in $\mathbb{R}^d$ and $t \geq 1$, a t-spanner of P is a graph $G = (P, E)$ such that for all $p, q \in P$,

$$\|p - q\| \leq \mathrm{dist}_G(p, q) \leq t\|p - q\|.$$

Construction

- For each $\sigma \in \Pi$, connect the n consecutive points with $n-1$ edges

Construction

- For each $\sigma \in \Pi$, connect the n consecutive points with $n-1$ edges
- $(n-1)|\Pi| = O_d((n/\varepsilon^d)\log(1/\varepsilon))$ edges

Construction

- For each $\sigma \in \Pi$, connect the n consecutive points with $n-1$ edges
- $(n-1)|\Pi| = O_d((n/\varepsilon^d)\log(1/\varepsilon))$ edges
- Maximum degree $O_d((1/\varepsilon^d)\log(1/\varepsilon))$

Construction

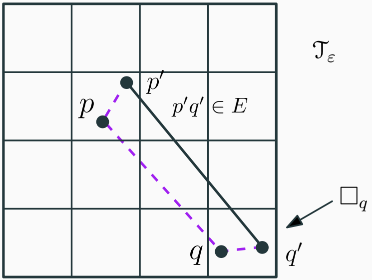
- For each $\sigma \in \Pi$, connect the n consecutive points with $n-1$ edges
- $(n-1)|\Pi| = O_d((n/\varepsilon^d)\log(1/\varepsilon))$ edges
- Maximum degree $O_d((1/\varepsilon^d)\log(1/\varepsilon))$
- Update time $O_d((1/\varepsilon^d)\log(n)\log^2(1/\varepsilon))$

Construction

- For each $\sigma \in \Pi$, connect the n consecutive points with $n-1$ edges
- $(n-1)|\Pi| = O_d((n/\varepsilon^d)\log(1/\varepsilon))$ edges
- Maximum degree $O_d((1/\varepsilon^d)\log(1/\varepsilon))$
- Update time $O_d((1/\varepsilon^d)\log(n)\log^2(1/\varepsilon))$
- Claim: G is a $(1+\varepsilon)$-spanner

Proof idea

- Prove by induction on length of pairs:
  $\text{dist}_G(p, q) \leq (1 + \varepsilon)\|p - q\|$

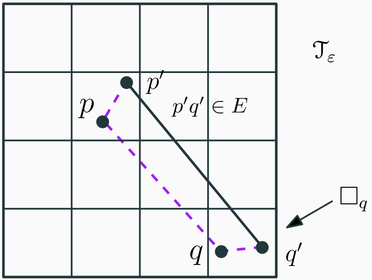


$$\text{sidelength}(\square) \leq 2(d+1)\|p - q\|$$

$$\text{sidelength}(\square_q) = \varepsilon \cdot \text{sidelength}(\square)$$

Proof idea

- Prove by induction on length of pairs:
  $\mathrm{dist}_G(p, q) \leq (1+\varepsilon)\|p-q\|$
- G is a $(1 + c_d\varepsilon)$-spanner for const. c_d

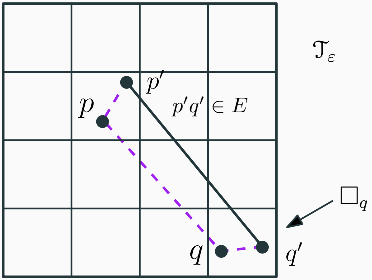


sidelength$(\square) \leq 2(d+1)\|p-q\|$

sidelength$(\square_q) = \varepsilon \cdot$ sidelength$(\square)$

Proof idea

- Prove by induction on length of pairs: $\text{dist}_G(p, q) \leq (1 + \varepsilon)\|p - q\|$
- G is a $(1 + c_d\varepsilon)$-spanner for const. c_d
- Readjust ε by c_d



sidelength$(\square) \leq 2(d+1)\|p - q\|$

sidelength$(\square_q) = \varepsilon \cdot$ sidelength$(\square)$

Static & dynamic vertex-fault-tolerant spanners

Fault-tolerant spanners

Definition

For a set of n points P in $\mathbb{R}^d$ and $t \geq 1$, a k-vertex-fault-tolerant t-spanner of P is a graph $G = (P, E)$ such that

1. G is a t-spanner, and
2. For any $P' \subseteq P$, $|P'| \leq k$, $G \setminus P'$ is a t-spanner for $P \setminus P'$.

Construction

- For each $\sigma \in \Pi$ and each $p \in P$, connect p to it's $k + 1$ predecessors and successors

Construction

- For each $\sigma \in \Pi$ and each $p \in P$, connect p to it's $k+1$ predecessors and successors
- $O(kn|\Pi|) = O_d((kn/\varepsilon^d)\log(1/\varepsilon))$ edges

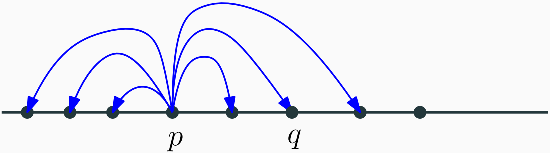
Construction

- For each $\sigma \in \Pi$ and each $p \in P$, connect p to it's $k+1$ predecessors and successors
- $O(kn|\Pi|) = O_d((kn/\varepsilon^d)\log(1/\varepsilon))$ edges
- Maximum degree $O_d((k/\varepsilon^d)\log(1/\varepsilon))$

Update time

Any update changes $O(k)$ edges in G

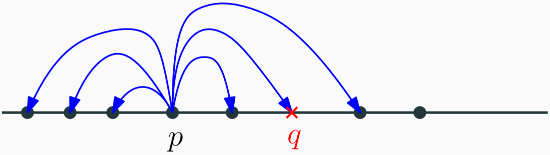
$k = 2$



Update time

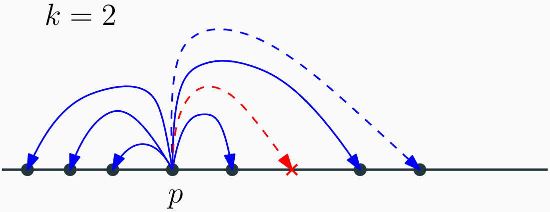
Any update changes $O(k)$ edges in G

$k = 2$



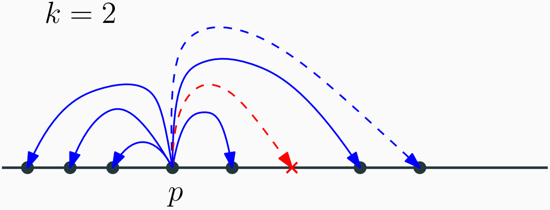
Update time

Any update changes $O(k)$ edges in G



Update time

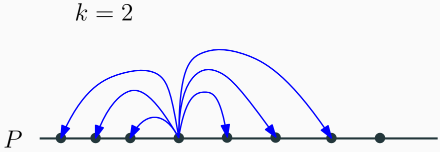
Any update changes $O(k)$ edges in G



Update time $O_d((\log n \log(1/\varepsilon) + k) \log(1/\varepsilon)/\varepsilon^d)$

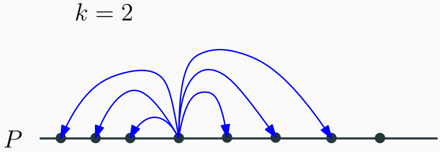
Sketch proof

- G is already a $(1+\varepsilon)$-spanner



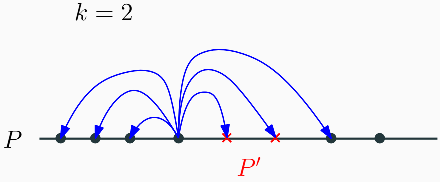
Sketch proof

- G is already a $(1+\varepsilon)$-spanner
- Consider $P' \subseteq P$, $|P'| \leq k$



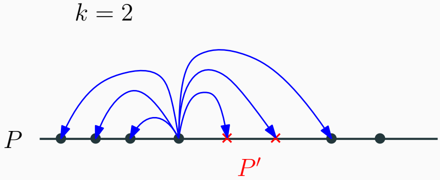
Sketch proof

- G is already a $(1+\varepsilon)$-spanner
- Consider $P' \subseteq P$, $|P'| \leq k$



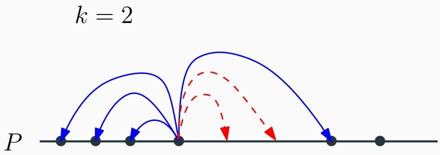
Sketch proof

- G is already a $(1+\varepsilon)$-spanner
- Consider $P' \subseteq P$, $|P'| \leq k$
- Let $\sigma \in \Pi$ with P' removed



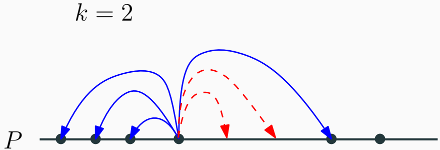
Sketch proof

- G is already a $(1+\varepsilon)$-spanner
- Consider $P' \subseteq P$, $|P'| \leq k$
- Let $\sigma \in \Pi$ with P' removed



Sketch proof

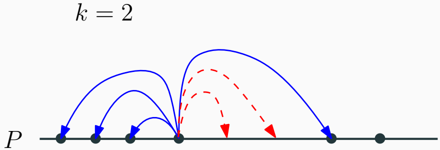
- G is already a $(1+\varepsilon)$-spanner
- Consider $P' \subseteq P$, $|P'| \leq k$
- Let $\sigma \in \Pi$ with P' removed
- Consecutive points in $P \setminus P'$ remain in $G \setminus P'$ (by construction)



Sketch proof

- G is already a $(1+\varepsilon)$-spanner
- Consider $P' \subseteq P$, $|P'| \leq k$
- Let $\sigma \in \Pi$ with P' removed
- Consecutive points in $P \setminus P'$ remain in $G \setminus P'$ (by construction)

  $\implies G \setminus P'$ is a $(1+\varepsilon)$-spanner for $P \setminus P'$



Conclusion

Main Theorem

Main Theorem

For $\varepsilon \in (0, 1)$, there is a set Π of size $O((1/\varepsilon^d)\log(1/\varepsilon))$ such that $\forall p, q \in [0, 1)^d$, there exists $\sigma \in \Pi$ with:

Points between p and q in σ are distance at most $\varepsilon\|p - q\|$ from p or q.

Applications

1. Approximate bichromatic closest pair (improved update time to $O(\log n)$)

Applications

1. Approximate bichromatic closest pair (improved update time to $O(\log n)$)
2. Dynamic spanners (simpler data structure)

Applications

1. Approximate bichromatic closest pair (improved update time to $O(\log n)$)
2. Dynamic spanners (simpler data structure)
3. Static vertex-fault-tolerant spanners (simple data structure)

Applications

1. Approximate bichromatic closest pair (improved update time to $O(\log n)$)
2. Dynamic spanners (simpler data structure)
3. Static vertex-fault-tolerant spanners (simple data structure)
4. Dynamic vertex-fault-tolerant spanners (previous work?)

Applications

1. Approximate bichromatic closest pair (improved update time to $O(\log n)$)
2. Dynamic spanners (simpler data structure)
3. Static vertex-fault-tolerant spanners (simple data structure)
4. Dynamic vertex-fault-tolerant spanners (previous work?)

Other applications:

1. Approximate nearest neighbor (not new)

Applications

1. Approximate bichromatic closest pair (improved update time to $O(\log n)$)
2. Dynamic spanners (simpler data structure)
3. Static vertex-fault-tolerant spanners (simple data structure)
4. Dynamic vertex-fault-tolerant spanners (previous work?)

Other applications:

1. Approximate nearest neighbor (not new)
2. Dynamic approximate MST (uses dynamic spanners)