Locality-Sensitive Orderings & Their Applications

Timothy Chan, Sariel Har-Peled, Mitchell Jones
August 2019

University of Illinois at Urbana-Champaign
Low dimension proximity problems: $d = O(1)$

**Goal**: Dynamic data structures which maintain/return a $(1 + \varepsilon)$-approximation
In this talk

- **Quadtrees**: Basic data structure in computational geometry
- **Many orderings** of points in \( \mathbb{R}^d \) (\( \mathbb{Z} \)-order)
- **Two new tricks** to the mix
  - \( \implies \) Simpler data structures for many proximity problems
    (plus some new results)
New technique: Locality-sensitive orderings
New technique: Locality-sensitive orderings
New technique: Locality-sensitive orderings

Definition: Locality-Sensitive Orderings

Let \( \varepsilon \in (0, 1) \). A collection of orderings \( \Pi \) over \([0, 1]^d\) s.t. for all \( p, q \in [0, 1]^d \), exists \( \sigma \in \Pi \) where:

\[
\forall p \prec_{\sigma} z \prec_{\sigma} q : \min(||z - p||, ||z - q||) \leq \varepsilon ||p - q||.
\]
**Definition: Locality-Sensitive Orderings**

Let $\varepsilon \in (0, 1)$. A collection of orderings $\Pi$ over $[0, 1)^d$ s.t. for all $p, q \in [0, 1)^d$, exists $\sigma \in \Pi$ where:

$$\forall p \prec_\sigma z \prec_\sigma q : \min(\|z - p\|, \|z - q\|) \leq \varepsilon \|p - q\|.$$

**Theorem**

There are locality-sensitive orderings of size $O((1/\varepsilon^d) \log(1/\varepsilon))$. 
Main applications

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

- New: \((1 + \varepsilon)\text{-bichromatic closest pair}\)
- Simpler: Dynamic \((1 + \varepsilon)\text{-spanners}\)
Main applications

- **New:** (1 + $\varepsilon$)-bichromatic closest pair
- **Simpler:** Dynamic (1 + $\varepsilon$)-spanners
- **New:** *Dynamic* $k$-vertex-fault-tolerant (1 + $\varepsilon$)-spanners
Main applications

- New: $(1 + \varepsilon)$-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

Problem ($c$-approximation)

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

8/41
Problem \((c\text{-approximation})\)

Maintain a pair \((r', b')\) s.t. \(\|r' - b'\| \leq c \cdot \min_{(r,b)} \|r - b\|\).
Quadtrees
Quadtrees
Quadtrees
Quadtrees: \( \mathcal{Z} \)-order

DFS of a quadtree \( \implies \) ordering of points (\( \mathcal{Z} \)-order)
Quadtrees: \(\subseteq\)-order

DFS of a quadtree \(\Rightarrow\) ordering of points (\(\supseteq\)-order)
Ordering of points

Hope: points close together $\approx$ nearby in ordering
Ordering of points

Hope: points close together $\approx$ nearby in ordering
Computing the \( \mathcal{Z} \)-order

- Let \( p = (x, y) \in [2^w] \times [2^w] \)
- \( x = x_wx_{w-1} \ldots x_1 \)
- \( y = y_wy_{w-1} \ldots y_1 \)

\[ \text{Position of } p \text{ in } \mathcal{Z}-\text{order} = \text{shuffle}(p) \]

\[ 0010 = 2 \]
Computing the \( \mathcal{Z} \)-order

- Let \( p = (x, y) \in [2^w] \times [2^w] \)
- \( x = x_wx_{w-1}\ldots x_1 \)
- \( y = y_wy_{w-1}\ldots y_1 \)
- \( \text{shuffle}(p) = y_wx_wy_{w-1}x_{w-1}\ldots y_1x_1 \)
- Position of \( p \) in \( \mathcal{Z} \)-order = \( \text{shuffle}(p) \)
Computing the $\mathcal{Z}$-order

- Let $p = (x, y) \in [2^w] \times [2^w]$
- $x = x_wx_{w-1}\ldots x_1$
- $y = y_wy_{w-1}\ldots y_1$
- shuffle$(p) = y_wx_wy_{w-1}x_{w-1}\ldots y_1x_1$
- Position of $p$ in $\mathcal{Z}$-order = shuffle$(p)$

**Lemma**

shuffle$(p)$ and shuffle$(q)$ can be compared with $O(1)$ bitwise-and/xor 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
Solving the problem in 1D: A solution?

- Map the point set to 1D
- Maintain sorted order
- Maintain consecutive red/blue pairs with min-heap

Insert $q$

Delete $p$
Solving the problem in 1D: A solution?

- Map the point set to 1D
- Maintain sorted order
- Maintain consecutive red/blue pairs with min-heap
- Updates change $O(1)$ consecutive pairs
Solving the problem in 1D: A solution?

- Map the point set to 1D
- Maintain sorted order
- Maintain consecutive red/blue pairs with min-heap
- Updates change $O(1)$ consecutive pairs
  ⟹ Update time $O(\log n)$
Not quite a solution

- Points nearby in $\mathbb{R}^d \implies$ nearby in $\mathbb{Z}$-order
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 $\mathbb{Z}$-order
- Idea: Shift the point set
Points nearby in $\mathbb{R}^d$ $\implies$ nearby in $\mathbb{Z}$-order

Idea: Shift the point set
Points nearby in $\mathbb{R}^d \implies$ nearby in $\mathbb{Z}$-order

Idea: Shift the point set
Shifting

**Lemma** [Chan, 1998]

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

For any $p, q \in [0, 1]^d$, exists $i \in \{0, \ldots, d\}$ and quadtree cell $\square$:

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: $P_0, \ldots, P_d$

  DS for $P_0$  DS for $P_1$  ...  DS for $P_d$
A correct solution

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

$$DS \text{ for } P_0 \quad DS \text{ for } P_1 \quad \ldots \quad DS \text{ for } P_d$$

$\Rightarrow O_d(\log n)$ update time
A correct solution

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

  \[
  \text{DS for } P_0 \quad \text{DS for } P_1 \quad \ldots \quad \text{DS for } P_d
  \]

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

- Claim: $O_d(1)$ approximation
Correctness (cont.)
\[
sidelength(\Box) \leq 2(d + 1)\|r - b\|
\]
$\text{sidelength}(\square) \leq 2(d + 1)\| r - b \|$
Correctness (cont.)

\[
sidelength(\square) \leq 2(d + 1)\|r - b\|
\]
Correctness (cont.)

\[
\text{sidelength}(\Box) \leq 2(d + 1)\|r - b\|
\]

\[
\|r' - b'\| \leq \text{diam}(\Box) \leq \sqrt{d} \cdot \text{sidelength}(\Box) = O_d(1)\|r - b\|
\]
The challenge:

$(1 + \varepsilon)$-approximate bichromatic closest pair
Key idea I: Reducing the approximation factor

- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
Key idea I: Reducing the approximation factor

> Assume \( \varepsilon = 2^{-E} \) for \( E \in \mathbb{N} \)

> **Idea:** Pack many 
  “\( \varepsilon \)-quadtrees” into a regular quadtree
Key idea I: Reducing the approximation factor

- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
- **Idea:** Pack many “$\varepsilon$-quadtrees” into a regular quadtree
- $\varepsilon$-quadtrees have $1/\varepsilon^d$ children
Key idea I: Reducing the approximation factor

- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
- **Idea**: Pack many “$\varepsilon$-quadtrees” into a regular quadtree
- $\varepsilon$-quadtrees have $1/\varepsilon^d$ children
Key idea I: Reducing the approximation factor

- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
- Idea: Pack many “$\varepsilon$-quadtrees” into a regular quadtree
  - $\varepsilon$-quadtrees have $1/\varepsilon^d$ children
  - Can partition a regular quadtree into $\log(1/\varepsilon)$ $\varepsilon$-quadtrees

$\varepsilon = 2^{-3}$

[Diagram showing a regular quadtree with levels labeled $1/2$, $1/4$, $1/8$, $1/16$, ...]
Key idea I: Reducing the approximation factor

- Assume $\varepsilon = 2^{-E}$ for $E \in \mathbb{N}$
- **Idea**: Pack many “$\varepsilon$-quadtrees” into a regular quadtree
- $\varepsilon$-quadtrees have $1/\varepsilon^d$ children
- Can **partition** a regular quadtree into $\log(1/\varepsilon)$ $\varepsilon$-quadtrees
- Call them $\mathcal{Q}^1_\varepsilon, \ldots, \mathcal{Q}^E_\varepsilon$
Extend \( \mathbb{Z} \)-order to \( \varepsilon \)-quadtrees by ordering \( 1/\varepsilon^d \) child cells

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>6</th>
<th>9</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>16</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>14</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>8</td>
<td>13</td>
<td></td>
</tr>
</tbody>
</table>

Which order to pick?
O(1) problems

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

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

\[
sidelength(□) \leq 2(d + 1)\| p - q \|
\]
Problem

Find a family $\mathcal{O}$ of orderings of the $1/\varepsilon^d$ cells s.t.:

For any $\Box_1, \Box_2$, there is an ordering $\sigma \in \mathcal{O}$ with $\Box_1$ adjacent to $\Box_2$. 
**Lemma [Alspach, 2008]**

For $[n] = \{1, \ldots, n\}$, there are $\lceil n/2 \rceil$ orderings $\mathcal{O}$ of $[n]$ such that for all $i, j \in [n]$, $\exists \sigma \in \mathcal{O}$ where $i$ and $j$ are adjacent in $\sigma$. 
Corollary

There is a set $\mathcal{O}(\varepsilon)$ of $O(1/\varepsilon^d)$ orderings such that for any $\Box_1, \Box_2$, there is an order $\sigma \in \mathcal{O}(\varepsilon)$ where $\Box_1$ and $\Box_2$ are adjacent.
What we have so far

- $d + 1$ shifted point sets $\equiv d + 1$ quad trees
What we have so far

- $d + 1$ shifted point sets $\equiv d + 1$ quadtrees
- Each quadtree has $\log(1/\varepsilon) \varepsilon$-quadtrees
What we have so far

- $d + 1$ shifted point sets $\equiv d + 1$ quadtrees
- Each quadtree has $\log(1/\epsilon)$ $\epsilon$-quadtrees
- Each $\epsilon$-quadtree has $O(1/\epsilon^d)$ orderings
What we have so far

- \( d + 1 \) shifted point sets \( \equiv d + 1 \) quadtrees
- Each quadtree has \( \log(1/\varepsilon) \) \( \varepsilon \)-quadtrees
- Each \( \varepsilon \)-quadtree has \( O(1/\varepsilon^d) \) orderings
  \[ \Rightarrow O_d\left((1/\varepsilon^d) \log(1/\varepsilon)\right) \text{ different orderings of } P \]
What we have so far

- $d + 1$ shifted point sets $\equiv d + 1$ quadtrees
- Each quadtree has $\lg(1/\varepsilon)$ $\varepsilon$-quadtrees
- Each $\varepsilon$-quadtree has $O(1/\varepsilon^d)$ orderings
  $\implies O_d((1/\varepsilon^d) \log(1/\varepsilon))$ different orderings of $P$
- $\Pi$ is this family of locality-sensitive orderings
What we have so far

- $d + 1$ shifted point sets $\equiv d + 1$ quadtrees
- Each quadtree has $\lg(1/\varepsilon)$ $\varepsilon$-quadtrees
- Each $\varepsilon$-quadtree has $O(1/\varepsilon^d)$ orderings
  $\Rightarrow O_d\left((1/\varepsilon^d) \log(1/\varepsilon)\right)$ different orderings of $P$
- $\Pi$ is this family of locality-sensitive orderings
- For $\sigma \in \Pi$, can decide $p \prec_\sigma q$ with $O(\log(1/\varepsilon))$ bitwise-logical operations.
The solution

- Maintain the 1D data structure for all orderings $\Pi$
The solution

- Maintain the 1D data structure for all orderings $\Pi$
- $|\Pi| = O\left(\frac{1}{\varepsilon^d} \log\frac{1}{\varepsilon}\right)$
The solution

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

\[
\text{sidelength}(\Box) \leq 2(d + 1)\|r - b\|
\]
Correctness

$\text{sidelength}(\square) \leq 2(d + 1)\|r - b\|$
Correctness

\[ \text{sidelength}(□) \leq 2(d + 1)\| r - b \| \]
Correctness

$\text{sidelength}(\Box) \leq 2(d + 1)\|r - b\|$
Correctness

\[ \text{sidelength}(\square) \leq 2(d + 1)\|r - b\| \]

\[ \text{sidelength}(\square_b) = \varepsilon \cdot \text{sidelength}(\square) \]

\[ \sigma \in \Pi \]

\[ I_r \quad I_b \]
Our result

Can maintain the \((1 + \varepsilon)\)-approximate bichromatic closest pair dynamically with:

1. \(O(\log n \log^2 (1/\varepsilon)/\varepsilon^d)\) update time
2. \(O(n \log (1/\varepsilon)/\varepsilon^d)\) space
The result

**Main Theorem**

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

Points between \( p \) and \( q \) in \( \sigma \) are distance at most \( \varepsilon \| p - q \| \) from \( p \) or \( q \).
A simple data structure for dynamic \((1 + \varepsilon)\)-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 $p, q \in P$,

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

Problem

Maintain a $(1 + \varepsilon)$-spanner of $P$ dynamically.
## Previous work & result

<table>
<thead>
<tr>
<th>reference</th>
<th>insertion time</th>
<th>deletion time</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Roditty, 2012]</td>
<td>$O(\log n)$</td>
<td>$O(n^{1/3} \log^{O(1)} n)$</td>
</tr>
<tr>
<td>[Gottlieb and Roditty, 2008a]</td>
<td>$O(\log^2 n)$</td>
<td>$O(\log^3 n)$</td>
</tr>
<tr>
<td>[Gottlieb and Roditty, 2008b]</td>
<td>$O(\log n)$</td>
<td>$O(\log n)$</td>
</tr>
</tbody>
</table>

## Our result

Can dynamically maintain a $(1 + \varepsilon)$-spanner of $P$ with:

1. $O(n \log(1/\varepsilon)/\varepsilon^d)$ edges
2. $O(\log(1/\varepsilon)/\varepsilon^d)$ maximum degree
3. $O(\log n \log^2(1/\varepsilon)/\varepsilon^d)$ update time
For each $\sigma \in \Pi$, add edges between consecutive points.
For each $\sigma \in \Pi$, add edges between consecutive points

$(n - 1)|\Pi| = O_d((n/\varepsilon^d) \log(1/\varepsilon))$ edges
Construction

- For each $\sigma \in \Pi$, add edges between consecutive points
- $(n - 1)|\Pi| = O_d((n/\varepsilon^d) \log(1/\varepsilon))$ edges
- Maximum degree $\leq 2|\Pi| = O_d((1/\varepsilon^d) \log(1/\varepsilon))$
Construction

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

$(n - 1)|\Pi| = O_d((n/\varepsilon^d) \log(1/\varepsilon))$ edges

Maximum degree $\leq 2|\Pi| = O_d((1/\varepsilon^d) \log(1/\varepsilon))$

Update time $O_d((1/\varepsilon^d) \log(n) \log^2(1/\varepsilon))$

Claim: $G$ is a $(1 + \varepsilon)$-spanner
Proof idea

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

- \( G \) is a \((1 + c \varepsilon)\)-spanner for const. \( c \)

- Readjust \( \varepsilon \) by \( c \)

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

- \( \text{sidelength}(\square_q) = \varepsilon \cdot \text{sidelength}(\square) \)
Proof idea

- Proof 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. \( c_d \)

sidelength(\(\Box\)) \(\leq 2(d + 1)\|p - q\|\)
sidelength(\(\Box_q\)) = \(\varepsilon \cdot \text{sidelength}(\Box)\)
Proof idea

- Proof 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 \( \varepsilon \) by \( c_d \)

sidelength(\( \square \)) \leq 2(d + 1)\|p - q\|
sidelength(\( \square_q \)) = \varepsilon \cdot \text{sidelength}(\( \square \))
Static & dynamic vertex-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 \( P \setminus P' \).

**Problem**

For a static point set \( P \), efficiently construct a "small" \( k \)-VFT \((1 + \varepsilon)\)-spanner.
## Previous work & result

<table>
<thead>
<tr>
<th>reference</th>
<th># edges</th>
<th>degree</th>
<th>running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Levcopoulos et al., 1998]</td>
<td>$2^{O(k)}n$</td>
<td>$2^{O(k)}$</td>
<td>$O(n \log n + 2^{O(k)}n)$</td>
</tr>
<tr>
<td></td>
<td>$O(k^2n)$</td>
<td>unbounded</td>
<td>$O(n \log n + k^2 n)$</td>
</tr>
<tr>
<td></td>
<td>$O(kn \log n)$</td>
<td>unbounded</td>
<td>$O(kn \log n)$</td>
</tr>
<tr>
<td>[Lukovszki, 1999]</td>
<td>$O(kn)$</td>
<td>$O(k^2)$</td>
<td>$O(n \log^{d-1}n + kn \log \log n)$</td>
</tr>
<tr>
<td>[Czumaj and Zhao, 2004]</td>
<td>$O(kn)$</td>
<td>$O(k)$</td>
<td>$O(kn \log^d n + k^2 n \log k)$</td>
</tr>
<tr>
<td>[Chan et al., 2015]</td>
<td>$O(k^2n)$</td>
<td>$O(k^2)$</td>
<td>$O(n \log n + k^2 n)$</td>
</tr>
<tr>
<td>[Kapoor and Li, 2013] &amp;</td>
<td>$O(kn)$</td>
<td>$O(k)$</td>
<td>$O(n \log n + kn)$</td>
</tr>
<tr>
<td>[Solomon, 2014]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Our result

A $k$-VFT $(1 + \varepsilon)$-spanner of $P$ with

1. $O(kn \log(1/\varepsilon)/\varepsilon^d)$ edges
2. $O(k \log(1/\varepsilon)/\varepsilon^d)$ maximum degree
3. $O((n \log n \log(1/\varepsilon) + kn) \log(1/\varepsilon)/\varepsilon^d)$ construction time
For each $\sigma \in \Pi$ and each $p \in P$, connect $p$ to its $k + 1$ predecessors and successors in $\sigma$
For each $\sigma \in \Pi$ and each $p \in P$, connect $p$ to its $k + 1$ predecessors and successors in $\sigma$.

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

- \( O(kn|\Pi|) = O_d((kn/\varepsilon^d) \log(1/\varepsilon)) \) edges
- Maximum degree = \( O(k|\Pi|) = O_d((k/\varepsilon^d) \log(1/\varepsilon)) \)
Sketch proof

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

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

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

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

$k = 2$
Sketch proof

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

$k = 2$
Sketch proof

- $G$ is 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 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 $P \setminus P'$
Any update changes $O(k)$ edges in $G$

$k = 2$
Any update changes $O(k)$ edges in $G$

$k = 2$
Any update changes $O(k)$ edges in $G$

$k = 2$
Any update changes $O(k)$ edges in $G$

$k = 2$

Update time $O_d\left((\log n \log(1/\varepsilon) + k)|\Pi|\right)$
**Our result**

A $k$-VFT $(1 + \varepsilon)$-spanner of $P$ with

1. $O(kn \log(1/\varepsilon)/\varepsilon^d)$ edges
2. $O(k \log(1/\varepsilon)/\varepsilon^d)$ maximum degree
3. $O\left((n \log n \log(1/\varepsilon) + kn) \log(1/\varepsilon)/\varepsilon^d\right)$ construction time

**New:** Can also maintain dynamically with update time

$$O \left((\log n \log(1/\varepsilon) + k) \log(1/\varepsilon)/\varepsilon^d\right) .$$
Conclusion
Main Theorem

For $\varepsilon \in (0, 1)$, there is a set $\Pi$ of size $O\left(\frac{1}{\varepsilon^d} \log\left(\frac{1}{\varepsilon}\right)\right)$ s.t. $
abla p, q \in [0, 1)^d$, $\exists \sigma \in \Pi$ with:

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

Remarks

- Extends to $\| \cdot \|_p$ norms
Main Theorem

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

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

Remarks

- Extends to $\| \cdot \|_p$ norms
- “Replacement” for well-separated pair decomposition
Main Theorem

For \( \varepsilon \in (0, 1) \), there is a set \( \Pi \) of size \( O\left(\frac{1}{\varepsilon^d} \log\left(\frac{1}{\varepsilon}\right)\right) \) s.t. \( \forall p, q \in [0, 1)^d \), \( \exists \sigma \in \Pi \) with:

Points between \( p \) and \( q \) in \( \sigma \) are distance at most \( \varepsilon \| p - q \| \) from \( p \) or \( q \).

Remarks

- Extends to \( \| \cdot \|_p \) norms
- "Replacement" for well-separated pair decomposition
- \( \approx \) locality-sensitive hashing (smaller family of orders, weaker guarantees)
Applications

1. **Approximate bichromatic closest pair**: Improved update time
   \[ \approx O(\log^3 n) \] [Eppstein, 1995] \rightarrow O(\log n)
1. **Approximate bichromatic closest pair**: Improved update time
   \[ \approx O(\log^3 n) \] [Eppstein, 1995] \[\rightarrow O(\log n)\]

2. **Dynamic spanners**: Simpler in Euclidean setting, matches best known
   [Gottlieb and Roditty, 2008b]
Applications

1. **Approximate bichromatic closest pair**: Improved update time
   \[ \approx O(\log^3 n) \ [\text{Eppstein, 1995}] \rightarrow O(\log n) \]

2. **Dynamic spanners**: Simpler in Euclidean setting, matches best known [Gottlieb and Roditty, 2008b]

3. **Static vertex-fault-tolerant spanners**: Simpler in Euclidean setting, matches best known [Kapoor and Li, 2013] and [Solomon, 2014]
Applications

1. **Approximate bichromatic closest pair**: Improved update time
   \[ \approx O(\log^3 n) \] [Eppstein, 1995] \(\rightarrow\) \(O(\log n)\)

2. **Dynamic spanners**: Simpler in Euclidean setting, matches best known [Gottlieb and Roditty, 2008b]

3. **Static vertex-fault-tolerant spanners**: Simpler in Euclidean setting, matches best known [Kapoor and Li, 2013] and [Solomon, 2014]

4. **Dynamic vertex-fault-tolerant spanners**: New
Applications

1. **Approximate bichromatic closest pair**: Improved update time
   \[ \approx O(\log^3 n) \] [Eppstein, 1995] \rightarrow O(\log n)

2. **Dynamic spanners**: Simpler in Euclidean setting, matches best known [Gottlieb and Roditty, 2008b]

3. **Static vertex-fault-tolerant spanners**: Simpler in Euclidean setting, matches best known [Kapoor and Li, 2013] and [Solomon, 2014]

4. **Dynamic vertex-fault-tolerant spanners**: New

5. **Approximate nearest neighbor**: Not new
Applications

1. **Approximate bichromatic closest pair**: Improved update time
   \[ \approx O(\log^3 n) \] [Eppstein, 1995] \[\rightarrow O(\log n)\]

2. **Dynamic spanners**: Simpler in Euclidean setting, matches best known [Gottlieb and Roditty, 2008b]

3. **Static vertex-fault-tolerant spanners**: Simpler in Euclidean setting, matches best known [Kapoor and Li, 2013] and [Solomon, 2014]

4. **Dynamic vertex-fault-tolerant spanners**: New

5. **Approximate nearest neighbor**: Not new

6. **Dynamic approximate MST**: Follows by dynamic spanners
Applications

1. **Approximate bichromatic closest pair**: Improved update time approximately $O(\log^3 n)$ [Eppstein, 1995] → $O(\log n)$

2. **Dynamic spanners**: Simpler in Euclidean setting, matches best known [Gottlieb and Roditty, 2008b]

3. **Static vertex-fault-tolerant spanners**: Simpler in Euclidean setting, matches best known [Kapoor and Li, 2013] and [Solomon, 2014]

4. **Dynamic vertex-fault-tolerant spanners**: New

5. **Approximate nearest neighbor**: Not new

6. **Dynamic approximate MST**: Follows by dynamic spanners

7. **Static robust $(1 + \varepsilon)$-spanners**: See [Buchin et al., 2018]


