Computational Geometry • Lecture
Well-Separated Pair Decompositions

Tamara Mchedlidze · Darren Strash
18.1.2016
Motivation: Spanners

Task:
A set of cities shall be connected by a new road network.
Motivation: Spanners

Task:
A set of cities shall be connected by a new road network.

Idea 1: Euclidean minimum spanning tree
Motivation: Spanners

Task:
A set of cities shall be connected by a new road network.

But for no pair $(x, y)$ the path length in the road network should be much larger than the distance $|xy|$.

Idea 1: Euclidean minimum spanning tree
Motivation: Spanners

Task:
A set of cities shall be connected by a new road network.

But for no pair \((x, y)\) the path length in the road network should be much larger than the distance \(\|xy\|\).

Idea 1: Euclidean minimum spanning tree
Idea 2: complete graph
Motivation: Spanners

**Task:**
A set of cities shall be connected by a new road network.

But for no pair \((x, y)\) the path length in the road network should be much larger than the distance \(|xy|\).

Construction costs must remain reasonable, e.g., only \(O(n)\) edges.

**Idea 1:** Euclidean minimum spanning tree
**Idea 2:** complete graph
Motivation: Spanners

Task:
A set of cities shall be connected by a new road network.

But for no pair \((x, y)\) the path length in the road network should be much larger than the distance \(||xy||\).

Construction costs must remain reasonable, e.g., only \(O(n)\) edges.

Idea 1: Euclidean minimum spanning tree
Idea 2: complete graph
Idea 3: sparse \(t\)-spanner
Well-Separated Pairs

**Def:** A pair of disjoint point sets $A$ and $B$ in $\mathbb{R}^d$ is called *$s$-well separated* for some $s > 0$, if $A$ and $B$ can each be covered by a ball of radius $r$ whose distance is at least $sr$. 

![Diagram of well-separated pairs](image)
Well-Separated Pairs

**Def:** A pair of disjoint point sets $A$ and $B$ in $\mathbb{R}^d$ is called \textbf{s-well separated} for some $s > 0$, if $A$ and $B$ can each be covered by a ball of radius $r$ whose distance is at least $sr$. 

\[ A' \quad A \quad \geq sr \quad B' \quad B \]
**Well-Separated Pairs**

**Def:** A pair of disjoint point sets \( A \) and \( B \) in \( \mathbb{R}^d \) is called \textit{s-well separated} for some \( s > 0 \), if \( A \) and \( B \) can each be covered by a ball of radius \( r \) whose distance is at least \( sr \).

\[
\begin{align*}
A' &\geq sr' \\
A &\geq sr \\
B' &\geq r \\
B &\geq r
\end{align*}
\]

**Obs:**
- \( s \)-well separated \( \Rightarrow \) \( s' \)-well separated for all \( s' \leq s \)
- singletons \( \{a\} \) and \( \{b\} \) are \( s \)-well separated for all \( s > 0 \)
Well-Separated Pair Decomposition (WSPD)

For well-separated pair \(\{A, B\}\) we know that the distance for all point pairs in \(A \otimes B = \{\{a, b\} \mid a \in A, b \in B, a \neq b\}\) is similar.

**Goal:** \(o(n^2)\)-sized data structure that approximates the distances of all \(\binom{n}{2}\) pairs of points in a set \(P = \{p_1, \ldots, p_n\}\).
Well-Separated Pair Decomposition (WSPD)

For well-separated pair \( \{A, B\} \) we know that the distance for all point pairs in \( A \otimes B = \{\{a, b\} \mid a \in A, b \in B, a \neq b\} \) is similar.

**Goal:** \( o(n^2) \)-sized data structure that approximates the distances of all \( \binom{n}{2} \) pairs of points in a set \( P = \{p_1, \ldots, p_n\} \).

**Def:** For a point set \( P \) and some \( s > 0 \) an **s-well separated pair decomposition** (\( s \)-WSPD) is a set of pairs \( \{\{A_1, B_1\}, \ldots, \{A_m, B_m\}\} \) with

- \( A_i, B_i \subset P \) for all \( i \)
- \( A_i \cap B_i = \emptyset \) for all \( i \)
- \( \bigcup_{i=1}^{m} A_i \otimes B_i = P \otimes P \)
- \( \{A_i, B_i\} \) s-well separated for all \( i \)
Example

28 point pairs
Example

28 point pairs

12 $s$-well separated pairs
Example

28 point pairs

12 $s$-well separated pairs

WSPD of size $O(n^2)$ is trivial. Can we do it in $O(n)$?
Recall: Quadtrees

**Def:** A quadtree $\mathcal{T}(P)$ for a point set $P$ is a rooted tree, where each internal node has four children. Each node corresponds to a square, and the squares of the leaves form a partition of the root square.
Recall: Quadtrees

**Def:** A **quadtree** $\mathcal{T}(P)$ for a point set $P$ is a rooted tree, where each internal node has four children. Each node corresponds to a square, and the squares of the leaves form a partition of the root square.

**Lemma 1:** The height of $\mathcal{T}(P)$ is at most $\log(s/c) + 3/2$, where $c$ is the smallest distance in $P$ and $s$ is the side length of the root square $Q$.

**Thm 1:** A quadtree $\mathcal{T}(P)$ on $n$ points with height $h$ has $O(hn)$ nodes and can be constructed in $O(hn)$ time.
Compressed Quadtrees

**Def:** A *compressed* quadtree is a quadtree, in which each path of non-separating inner nodes is contracted into a single edge. Each such edge has a label to reconstruct the path structure.
Def: A **compressed** quadtree is a quadtree, in which each path of non-separating inner nodes is contracted into a single edge. Each such edge has a label to reconstruct the path structure.
**Def:** A **compressed** quadtree is a quadtree, in which each path of non-separating inner nodes is contracted into a single edge. Each such edge has a label to reconstruct the path structure.

- **quadtree**
- **compressed quadtree**
Properties of Compressed Quadtrees

Obs:  
- inner nodes split their point set into $\geq 2$ non-empty parts $\Rightarrow$ max. $n - 1$ inner nodes
- depth can be $d = n$, so the algorithm to construct quadtrees takes $O(n^2)$ time
Properties of Compressed Quadtrees

**Obs:**
- Inner nodes split their point set into $\geq 2$ non-empty parts $\Rightarrow$ max. $n - 1$ inner nodes
- Depth can be $d = n$, so the algorithm to construct quadtrees takes $O(n^2)$ time

**Thm 2:** A compressed quadtree for $n$ points in $\mathbb{R}^d$ with a fixed dimension $d$ can be constructed in $O(n \log n)$ time.

E.g. skip-quadtree [Eppstein et al. 2005] (without proof)
Packing Lemma

**Lemma 2:** Let $K$ be a ball with radius $r$ in $\mathbb{R}^d$ and let $X$ be a set of pairwise disjoint quadtree cells with side length $\geq x$ that intersect $K$. Then it holds

$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$
Packing Lemma

**Lemma 2:** Let $K$ be a ball with radius $r$ in $\mathbb{R}^d$ and let $X$ be a set of pairwise disjoint quadtree cells with side length $\geq x$ that intersect $K$. Then it holds

$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$ 

**Proof:**
Packing Lemma

**Lemma 2:** Let $K$ be a ball with radius $r$ in $\mathbb{R}^d$ and let $X$ be a set of pairwise disjoint quadtree cells with side length $\geq x$ that intersect $K$. Then it holds

$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$ 

**Proof:**
Packing Lemma

**Lemma 2:** Let $K$ be a ball with radius $r$ in $\mathbb{R}^d$ and let $X$ be a set of pairwise disjoint quadtree cells with side length $\geq x$ that intersect $K$. Then it holds

$$|X| \leq (1 + \left\lceil \frac{2r}{x} \right\rceil)^d.$$

**Proof:**

```

```

Tamara Mchedlidze · Darren Strash
Representatives and Level

Def: For each node $u$ of a quadtree $\mathcal{T}(P)$ for point set $P$ let $P_u = Q_u \cap P$ be the set of points in the corresponding square $Q_u$. In each leaf $u$ define the representative

$$\text{rep}(u) = \begin{cases} p & \text{falls } P_u = \{p\} \text{ (} u \text{ is leaf)} \\ \emptyset & \text{otherwise.} \end{cases}$$

For an inner node $v$ assign $\text{rep}(v) = \text{rep}(u)$ for a non-empty child $u$ of $v$. 
Representatives and Level

**Def:** For each node $u$ of a quadtree $\mathcal{T}(P)$ for point set $P$ let $P_u = Q_u \cap P$ be the set of points in the corresponding square $Q_u$. In each leaf $u$ define the representative

$$\text{rep}(u) = \begin{cases} p \text{ falls } P_u = \{p\} \text{ (u is leaf)} \\ \emptyset \text{ otherwise.} \end{cases}$$

For an inner node $v$ assign $\text{rep}(v) = \text{rep}(u)$ for a non-empty child $u$ of $v$.

**Def:** For each node $u$ of a quadtree $\mathcal{T}(P)$ let $\text{level}(u)$ be the level of $u$ in the corresponding *uncompressed* quadtree. We have $\text{level}(u) \leq \text{level}(v)$ iff $\text{area}(Q_u) \geq \text{area}(Q_v)$. 
Representatives and Level

**Def:** For each node $u$ of a quadtree $\mathcal{T}(P)$ for point set $P$ let $P_u = Q_u \cap P$ be the set of points in the corresponding square. Define the representative $\text{rep}(u) = \{ p \mid p \in P_u \}$ ($u$ is leaf) $\emptyset$ otherwise.

For an inner node $v$ assign $\text{rep}(v) = \text{rep}(u)$ for a non-empty child $u$ of $v$.

**Def:** For each node $u$ of a quadtree $\mathcal{T}(P)$ let $\text{level}(u)$ be the level of $u$ in the corresponding *uncompressed* quadtree. We have $\text{level}(u) \leq \text{level}(v)$ iff $\text{area}(Q_u) \geq \text{area}(Q_v)$. 
Constructing a WSPD

\[ \text{wsPairs}(u, v, \mathcal{T}, s) \]

**Input:** quadtree nodes \( u, v \), quadtree \( \mathcal{T} \), \( s > 0 \)

**Output:** WSPD for \( P_u \otimes P_v \)

if \( \text{rep}(u) = \emptyset \) or \( \text{rep}(v) = \emptyset \) or leaf \( u = v \) then return \( \emptyset \)

else if \( P_u \) and \( P_v \) \( s \)-well separated then return \( \{ \{ u, v \} \} \)

else

if \( \text{level}(u) > \text{level}(v) \) then swap \( u \) and \( v \)

\((u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } \mathcal{T}\)

return \( \bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, \mathcal{T}, s) \)
Constructing a WSPD

\[
\text{wsPairs}(u, v, \mathcal{T}, s)
\]

**Input:** quadtree nodes \(u, v\), quadtree \(\mathcal{T}\), \(s > 0\)

**Output:** WSPD for \(P_u \otimes P_v\)

if \(\text{rep}(u) = \emptyset\) or \(\text{rep}(v) = \emptyset\) or leaf \(u = v\) then return \(\emptyset\)

else if \(P_u\) and \(P_v\) \(s\)-well separated then return \(\{\{u, v\}\}\)

else

if \(\text{level}(u) > \text{level}(v)\) then swap \(u\) and \(v\)

\((u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } \mathcal{T}\)

return \(\bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, \mathcal{T}, s)\)
Constructing a WSPD

wsPairs(u, v, T, s)

Input: quadtree nodes u, v, quadtree T, s > 0
Output: WSPD for \( P_u \otimes P_v \)

if rep(u) = ∅ or rep(v) = ∅ or leaf u = v then return ∅
else if \( P_u \) and \( P_v \) s-well separated then return \{\{u, v\}\}
else
  if level(u) > level(v) then swap u and v
  \((u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } T\)
  return \(\bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, T, s)\)
Constructing a WSPD

wsPairs(\(u, v, \mathcal{T}, s\))

- **Input**: quadtree nodes \(u, v\), quadtree \(\mathcal{T}\), \(s > 0\)
- **Output**: WSPD for \(P_u \otimes P_v\)

if rep(\(u\)) = \(\emptyset\) or rep(\(v\)) = \(\emptyset\) or leaf \(u = v\) then return \(\emptyset\)
else if \(P_u\) and \(P_v\) \(s\)-well separated then return \(\{\{u, v\}\}\)
else
    if level(\(u\)) > level(\(v\)) then swap \(u\) and \(v\)
    \((u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } \mathcal{T}\)
    return \(\bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, \mathcal{T}, s)\)
Constructing a WSPD

\text{wsPairs}(u, v, \mathcal{T}, s)

\textbf{Input}: quadtree nodes \(u, v\), quadtree \(\mathcal{T}\), \(s > 0\)

\textbf{Output}: WSPD for \(P_u \otimes P_v\)

\begin{align*}
\text{if } \text{rep}(u) = \emptyset \text{ or rep}(v) = \emptyset \text{ or leaf } u = v & \text{ then return } \emptyset \\
\text{else if } P_u \text{ and } P_v \text{ s-well separated} & \text{ then return } \{\{u, v\}\}
\end{align*}

\text{else}

\begin{align*}
\text{if level}(u) > \text{level}(v) & \text{ then swap } u \text{ and } v \\
(u_1, \ldots, u_m) & \leftarrow \text{children of } u \text{ in } \mathcal{T} \\
\text{return } \bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, \mathcal{T}, s)
\end{align*}
Constructing a WSPD

wsPairs\((u, v, T, s)\)

**Input**: quadtree nodes \(u, v\), quadtree \(T\), \(s > 0\)

**Output**: WSPD for \(P_u \otimes P_v\)

if \(\text{rep}(u) = \emptyset\) or \(\text{rep}(v) = \emptyset\) or leaf \(u = v\) then return \(\emptyset\)
else if \(P_u\) and \(P_v\) \(s\)-well separated then return \({\{u, v\}}\)
else

  if level\((u)\) > level\((v)\) then swap \(u\) and \(v\)

  \((u_1, \ldots, u_m) \leftarrow \text{children of } u\ \text{in } T\)

  return \(\bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, T, s)\)
Constructing a WSPD

wsPairs(u, v, T, s)

Input: quadtree nodes u, v, quadtree T, s > 0
Output: WSPD for \( P_u \otimes P_v \)

if \( \text{rep}(u) = \emptyset \) or \( \text{rep}(v) = \emptyset \) or leaf \( u = v \) then return \( \emptyset \)
else if \( P_u \) and \( P_v \) s-well separated then return \( \{\{u, v\}\} \)
else
  if level(u) > level(v) then swap u and v
  \((u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } T\)
  return \( \bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, T, s)\)
Constructing a WSPD

\( \text{wsPairs}(u, v, \mathcal{T}, s) \)

**Input:** quadtree nodes \( u, v \), quadtree \( \mathcal{T} \), \( s > 0 \)

**Output:** WSPD for \( P_u \otimes P_v \)

if \( \text{rep}(u) = \emptyset \) or \( \text{rep}(v) = \emptyset \) or leaf \( u = v \) then return \( \emptyset \)
else if \( P_u \) and \( P_v \) \( s \)-well separated then return \( \{ \{u, v\}\} \)
else
  if \( \text{level}(u) > \text{level}(v) \) then swap \( u \) and \( v \)
  \((u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } \mathcal{T} \)
  return \( \bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, \mathcal{T}, s) \)
Constructing a WSPD

wsPairs($u, v, \mathcal{T}, s$)

**Input:** quadtree nodes $u,v$, quadtree $\mathcal{T}$, $s > 0$

**Output:** WSPD for $P_u \otimes P_v$

- if $\text{rep}(u) = \emptyset$ or $\text{rep}(v) = \emptyset$ or leaf $u = v$ then return $\emptyset$
- else if $P_u$ and $P_v$ $s$-well separated then return $\{\{u, v\}\}$
- else
  - if $\text{level}(u) > \text{level}(v)$ then swap $u$ and $v$
  - $(u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } \mathcal{T}$
  - return $\bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, \mathcal{T}, s)$

Circles around $Q_u$ and $Q_v$ (or radius 0 for point in a leaf) increase smaller circle and check if distance $\geq sr$.
Constructing a WSPD

wsPairs\((u, v, T, s)\)

**Input:** quadtree nodes \(u, v\), quadtree \(T\), \(s \geq 0\)

**Output:** WSPD for \(P_u \otimes P_v\)

if \(\text{rep}(u) = \emptyset\) or \(\text{rep}(v) = \emptyset\) or leaf \(u = v\) then return \(\emptyset\)

everse if \(P_u\) and \(P_v\) \(s\)-well separated then return \(\{\{u, v\}\}\)

else

\[
\begin{align*}
&\text{if level}(u) > \text{level}(v) \text{ then swap } u \text{ and } v \\
&(u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } T \\
&\text{return } \bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, T, s)
\end{align*}
\]

\[
\{\{b, c\}, \{d\}\}
\]

circles around \(Q_u\) and \(Q_v\) (or radius 0 for point in a leaf)
increase smaller circle and check if distance \(\geq sr\)
Constructing a WSPD

wsPairs(u, v, T, s)

**Input:** quadtree nodes u, v, quadtree T, s > 0

**Output:** WSPD for $P_u \otimes P_v$

if $\text{rep}(u) = \emptyset$ or $\text{rep}(v) = \emptyset$ or leaf $u = v$ then return $\emptyset$

else if $P_u$ and $P_v$ s-well separated then return $\{\{u, v\}\}$

else

    if $\text{level}(u) > \text{level}(v)$ then swap $u$ and $v$

    $(u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } T$

    return $\bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, T, s)$

The diagram shows a quadtree with nodes labeled a, b, c, d, and e. The tree structure and the resulting WSPD are depicted, with the output for $\text{wsPairs}(u_0, v, T, s)$ being $\{\{a\}, \{d\}\}$.
Constructing a WSPD

wsPairs($u, v, T, s$)

**Input:** quadtree nodes $u, v$, quadtree $T$, $s > 0$

**Output:** WSPD for $P_u \otimes P_v$

if $\text{rep}(u) = \emptyset$ or $\text{rep}(v) = \emptyset$ or leaf $u = v$ then return $\emptyset$
else if $P_u$ and $P_v$ s-well separated then return $\{ \{u, v\}\}$
else

if $\text{level}(u) > \text{level}(v)$ then swap $u$ and $v$

$(u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } T$

return $\bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, T, s)$

---

```
{{{b, c}, {d}}}  
{{{a}, {d}}}     
{{{b, c}, {e}}}  
{{{d}, {e}}}     
{{{a}, {b}}}     
{{{a}, {c}}}     
{{{b}, {c}}}     
{{{a}, {e}}}     
```
Constructing a WSPD

wsPairs(\(u, v, T, s\))

**Input:** quadtree nodes \(u, v\), quadtree \(T\), \(s > 0\)

**Output:** WSPD for \(P_u \otimes P_v\)

if \(\text{rep}(u) = \emptyset\) or \(\text{rep}(v) = \emptyset\) or leaf \(u = v\) then return \(\emptyset\)

else if \(P_u\) and \(P_v\) \(s\)-well separated then return \(\{\{u, v\}\}\)

else

- if \(\text{level}(u) > \text{level}(v)\) then swap \(u\) and \(v\)

- \((u_1, \ldots, u_m) \leftarrow \text{children of } u \text{ in } T\)

- return \(\bigcup_{i=1}^{m} \text{wsPairs}(u_i, v, T, s)\)

- initial call \(\text{wsPairs}(u_0, u_0, T, s)\)
- avoid duplicates \(\text{wsPairs}(u_i, u_j, T, s)\) and \(\text{wsPairs}(u_j, u_i, T, s)\)
- leaf pairs are always \(s\)-well separated, so algorithm terminates
- output are pairs of quadtree nodes

**How?**

**Space use?**
Constructing a WSPD

wsPairs(u, v, T, s)

**Input:** quadtree nodes u, v, quadtree T, s > 0

**Output:** WSPD for $P_u \otimes P_v$

if rep(u) = ∅ or rep(v) = ∅ or leaf $u = v$ then return ∅
else if $P_u$ and $P_v$ s-well separated then return {{u, v}}
else
    if level(u) > level(v) then swap u and v
    $(u_1, \ldots, u_m) \leftarrow$ children of u in T
    return $\bigcup_{i=1}^{m}$ wsPairs($u_i, v, T, s$)

- initial call wsPairs($u_0, u_0, T, s$)
- avoid duplicates wsPairs($u_i, u_j, T, s$) and wsPairs($u_j, u_i, T, s$) How?
- leaf pairs are always s-well separated, so algorithm terminates
- output are pairs of quadtree nodes Space use?

**Question:** How many pairs does the algorithm create?
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$. 
Analysis of WSPD Construction

Thm 3: Given a point set \( P \) in \( \mathbb{R}^d \) and \( s \geq 1 \) we can construct an \( s \)-WSPD with \( O(s^d n) \) pairs in time \( O(n \log n + s^d n) \).

Sketch of proof:

- simplifying assumption: no quadtree compression required
  \( \Rightarrow \) in \( \text{wsPairs}(u, v, T, s) \) sizes of \( u \) and \( v \) differ by at most factor 2
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$, we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

- **simplifying assumption:** no quadtrees required
  - $\Rightarrow$ in $\text{wsPairs}(u, v, T, s)$ sizes of $u$ and $v$ differ by at most factor 2
- **goal:** count calls to $\text{wsPairs}$
  - call is **trivial** if it produces no further recursive calls
  - each trivial call produces at most one ws pair
  - each non-trivial call produces $\leq 2^d$ trivial calls and thus $\leq 2^d$ ws pairs
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

- simplifying assumption: no quadtree compression required
  $\Rightarrow$ in $\text{wsPairs}(u, v, T, s)$ sizes of $u$ and $v$ differ by at most factor 2

- **goal:** count calls to $\text{wsPairs}$
  - call is **trivial** if it produces no further recursive calls
  - each trivial call produces at most one ws pair
  - each non-trivial call produces $\leq 2^d$ trivial calls and thus $\leq 2^d$ ws pairs

- let’s count non-trivial calls and charge cost to the smaller of the two cells
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**
- **simplifying assumption:** no quadtree compression required
  \[ \Rightarrow \text{in wsPairs}(u, v, T, s) \text{ sizes of } u \text{ and } v \text{ differ by at most factor 2} \]
- **goal:** count calls to wsPairs
  - call is **trivial** if it produces no further recursive calls
  - each trivial call produces at most one ws pair
  - each non-trivial call produces $\leq 2^d$ trivial calls and thus $\leq 2^d$ ws pairs
- let’s count non-trivial calls and charge cost to the smaller of the two cells

**goal:** each quadtree node has cost $O(s^d)$
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**
- simplifying assumption: no quadtree compression required
  $\Rightarrow$ in $wsPairs(u, v, T, s)$ sizes of $u$ and $v$ differ by at most factor 2
- **goal:** count calls to $wsPairs$
  - call is **trivial** if it produces no further recursive calls
  - each trivial call produces at most one $ws$ pair
  - each non-trivial call produces $\leq 2^d$ trivial calls and thus $\leq 2^d$ $ws$ pairs
- let’s count non-trivial calls and charge cost to the smaller of the two cells
- call non-trivial $\Rightarrow u$ and $v$ not $ws$, $u \geq v$
Analysis of WSPD Construction

**Thm 3:** Given a point set \( P \) in \( \mathbb{R}^d \) and \( s \geq 1 \) we can construct an \( s \)-WSPD with \( O(s^d n) \) pairs in time \( O(n \log n + s^d n) \).

**Sketch of proof:**

- **simplifying assumption:** no quadtree compression required
  \[ \Rightarrow \text{in wsPairs}(u, v, T, s) \text{ sizes of } u \text{ and } v \text{ differ by at most factor 2} \]
- **goal:** count calls to wsPairs
  - call is **trivial** if it produces no further recursive calls
  - each trivial call produces at most one ws pair
  - each non-trivial call produces \( \leq 2^d \) trivial calls and thus \( \leq 2^d \) ws pairs
- let’s count non-trivial calls and charge cost to the smaller of the two cells
- call non-trivial \( \Rightarrow u \text{ and } v \text{ not ws, } u \geq v \)
- let \( x \) be side length of \( v \) and \( r_v = x \sqrt{d}/2 \) the radius of the enclosing ball
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

- **simplifying assumption:** no quadtree compression required

  $\Rightarrow$ in $\text{wsPairs}(u, v, T, s)$ sizes of $u$ and $v$ differ by at most factor 2

- **goal:** count calls to $\text{wsPairs}$
  - call is **trivial** if it produces no further recursive calls
  - each trivial call produces at most one ws pair
  - each non-trivial call produces $\leq 2^d$ trivial calls and thus $\leq 2^d$ ws pairs

- let’s count non-trivial calls and charge cost to the smaller of the two cells

- call non-trivial $\Rightarrow u$ and $v$ not ws, $u \geq v$

- let $x$ be side length of $v$ and $r_v = x\sqrt{d}/2$ the radius of the enclosing ball

- side length of $u$ is $x$ or $2x$ and $r_u \leq 2r_v$
Well-Separated Pair Decompositions

Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

- **simplifying assumption:** no quadtree compression required
  $\Rightarrow$ in $\text{wsPairs}(u, v, T, s)$ sizes of $u$ and $v$ differ by at most factor 2

- **goal:** count calls to $\text{wsPairs}$
  - call is **trivial** if it produces no further recursive calls
  - each trivial call produces at most one $\text{ws}$ pair
  - each non-trivial call produces $\leq 2^d$ trivial calls and thus $\leq 2^d$ $\text{ws}$ pairs

- let’s count non-trivial calls and charge cost to the smaller of the two cells

- call non-trivial $\Rightarrow u$ and $v$ not $\text{ws}$, $u \geq v$

- let $x$ be side length of $v$ and $r_v = x\sqrt{d}/2$ the radius of the enclosing ball

- side length of $u$ is $x$ or $2x$ and $r_u \leq 2r_v$

- $u, v$ not $\text{ws}$ $\Rightarrow$ ball distance $\leq s \max\{r_u, r_v\} \leq 2s r_v = sx\sqrt{d}$
Thm 3: Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

Sketch of proof:
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

- Ball centers have distance
  - $\leq r_v + r_u + sx\sqrt{d}$
  - $\leq (3/2 + s)x\sqrt{d}$
  - $\leq 3sx\sqrt{d} =: R_v$
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

- Ball centers have distance
  \[
  \leq r_v + r_u + sx\sqrt{d}
  \leq (3/2 + s)x\sqrt{d}
  \leq 3sx\sqrt{d} =: R_v
  \]

- All cells charging cost to $v$ have size $x$ or $2x$ and intersect $K_v$; let $C$ be their number and apply Lemma 2 (see board)
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

Recall Lemma 2:
Given ball $K$ with radius $r$ in $\mathbb{R}^d$ and set $X$ of pairwise disjoint quadtree cells with side length $\geq x$ that intersect $K$. Then
$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

Recall Lemma 2:

Given ball $K$ with radius $r$ in $\mathbb{R}^d$ and set $X$ of pairwise disjoint quadtree cells with side length $\geq x$ that intersect $K$. Then

$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$

- all cells charging cost to $v$ have size $x$ or $2x$ and intersect $K_v$; let $C$ be their number and apply Lemma 2 (see board)
- yields $C = O(s^d)$
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

- have $O(n)$ nodes in $\mathcal{T}$
- each causes $O(s^d)$ non-trivial calls
- each non-trivial call produces $O(2^d)$ ws-pairs

**Recall Lemma 2:** Given ball $K$ with radius $r$ in $\mathbb{R}^d$ and set $X$ of pairwise disjoint quadtree cells with side length $\geq x$ that intersect $K$. Then

$$|X| \leq (1 + \lceil 2r/x \rceil)^d.$$  

- all cells charging cost to $v$ have size $x$ or $2x$ and intersect $K_v$; let $C$ be their number and apply Lemma 2 (see board)
- yields $C = O(s^d)$
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Sketch of proof:**

- have $O(n)$ nodes in $\mathcal{T}$
- each causes $O(s^d)$ non-trivial calls
- each non-trivial call produces $O(2^d)$ ws-pairs
- in total $O(s^d n)$ ws-pairs
- time: $O(n \log n)$ for quadtree and $O(s^d n)$ for the $s$-WSPD

**Recall Lemma 2:**

Given ball $K$ with radius $r$ in $\mathbb{R}^d$ and set $X$ of pairwise disjoint quadtree cells with side length $\geq x$ that intersect $K$. Then $|X| \leq (1 + \lceil 2r/x \rceil)^d$.

- all cells charging cost to $v$ have size $x$ or $2x$ and intersect $K_v$; let $C$ be their number and apply Lemma 2 (see board)
- yields $C = O(s^d)$
Analysis of WSPD Construction

**Thm 3:** Given a point set $P$ in $\mathbb{R}^d$ and $s \geq 1$ we can construct an $s$-WSPD with $O(s^d n)$ pairs in time $O(n \log n + s^d n)$.

**Obs:** each point pair $\{u, v\}$ is represented by exactly one ws-pair $\{A_i, B_i\}$ in this WSPD
For a set $P$ of $n$ points in $\mathbb{R}^d$ the **Euclidean graph** $\mathcal{E}G(P) = (P, (\frac{P}{2}))$ is the complete weighted graph, whose edge weights correspond to the Euclidean distances of the edges’ endpoints.
$t$-Spanner

For a set $P$ of $n$ points in $\mathbb{R}^d$ the **Euclidean graph**
$\mathcal{E}G(P) = (P, (\binom{P}{2}))$ is the complete weighted graph, whose edge weights correspond to the Euclidean distances of the edges’ endpoints.
**$t$-Spanner**

For a set $P$ of $n$ points in $\mathbb{R}^d$ the **Euclidean graph** $\mathcal{E}G(P) = (P, (P^2))$ is the complete weighted graph, whose edge weights correspond to the Euclidean distances of the edges’ endpoints.

Since $\mathcal{E}G(P)$ has $\Theta(n^2)$ edges, one is often interested in a sparse graphs with $O(n)$ edges, whose shortest paths approximate the edge weights in $\mathcal{E}G(P)$.

![Graph Diagram](image_url)
For a set $P$ of $n$ points in $\mathbb{R}^d$ the Euclidean graph $\mathcal{E}(P) = (P, (P, 2))$ is the complete weighted graph, whose edge weights correspond to the Euclidean distances of the edges’ endpoints.

Since $\mathcal{E}(P)$ has $\Theta(n^2)$ edges, one is often interested in a sparse graphs with $O(n)$ edges, whose shortest paths approximate the edge weights in $\mathcal{E}(P)$.

**Def:** A weighted graph $G$ with vertex set $P$ is called **$t$-spanner** for $P$ and a stretch factor $t \geq 1$, if for all pairs $x, y \in P$ it holds

$$||xy|| \leq \delta_G(x, y) \leq t \cdot ||xy||,$$

where $\delta_G(x, y) =$ length of shortest $x$-$y$-path in $G$. 

**t-Spanner**

For a set $P$ of $n$ points in $\mathbb{R}^d$ the Euclidean graph $\mathcal{E}(P) = (P, (P, 2))$ is the complete weighted graph, whose edge weights correspond to the Euclidean distances of the edges’ endpoints.
WSPD und $t$-Spanner

**Def:** For $n$ points $P$ in $\mathbb{R}^d$ and a WSPD $W$ of $P$ define the graph $G = (P, E)$, where

$$E = \{ \{x, y\} \mid \exists \{u, v\} \in W \text{ with } \text{rep}(u) = x, \text{rep}(v) = y \}.$$  

**Recall:** For each node $u$ of a quadtree $\mathcal{T}(P)$ for point set $P$ let $P_u = Q_u \cap P$ be the set of points in the corresponding square $Q_u$. In each leaf $u$ define the representative

$$\text{rep}(u) = \begin{cases} p & \text{falls } P_u = \{p\} \text{ (}u\text{ is leaf)} \\ \emptyset & \text{otherwise.} \end{cases}$$

For inner node $v$ assign $\text{rep}(v) = \text{rep}(u)$ for non-empty child $u$ of $v$. 


WSPD und \( t \)-Spanner

**Def:** For \( n \) points \( P \) in \( \mathbb{R}^d \) and a WSPD \( W \) of \( P \) define the graph \( G = (P, E) \), where

\[
E = \{ \{x, y\} \mid \exists \{u, v\} \in W \text{ with } \text{rep}(u) = x, \text{rep}(v) = y \}.
\]
**WSPD und t-Spanner**

**Def:** For $n$ points $P$ in $\mathbb{R}^d$ and a WSPD $W$ of $P$ define the graph $G = (P, E)$, where

$$E = \{\{x, y\} \mid \exists \{u, v\} \in W \text{ with } \text{rep}(u) = x, \text{rep}(v) = y\}.$$ 

**Lemma 3:** If $W$ is a $s$-WSPD for a suitable $s = s(t) \geq 4$, then $G$ is a $t$-spanner for $P$ with $O(s^d n)$ edges.
**WSPD und $t$-Spanner**

**Def:** For $n$ points $P$ in $\mathbb{R}^d$ and a WSPD $W$ of $P$ define the graph $G = (P, E)$, where
\[ E = \{\{x, y\} \mid \exists \{u, v\} \in W \text{ with } \text{rep}(u) = x, \text{rep}(v) = y\}. \]

**Lemma 3:** If $W$ is a $s$-WSPD for a suitable $s = s(t) \geq 4$, then $G$ is a $t$-spanner for $P$ with $O(s^d n)$ edges.

**Proof:** (blackboard)
Well-Separated Pair Decompositions

**Def:** For \( n \) points \( P \) in \( \mathbb{R}^d \) and a WSPD \( W \) of \( P \) define the graph \( G = (P, E) \), where
\[
E = \{\{x, y\} \mid \exists\{u, v\} \in W \text{ with } \text{rep}(u) = x, \text{rep}(v) = y\}.
\]

**Lemma 3:** If \( W \) is a \( s \)-WSPD for a suitable \( s = s(t) \geq 4 \), then \( G \) is a \( t \)-spanner for \( P \) with \( O(s^d n) \) edges.

**Proof:** (blackboard)
Summary

**Thm 4:** For a set $P$ of $n$ points in $\mathbb{R}^d$ and some $\varepsilon \in (0, 1]$ we can compute an $(1 + \varepsilon)$-spanner for $P$ with $O(n/\varepsilon^d)$ edges in $O(n \log n + n/\varepsilon^d)$ time.
Summary

**Thm 4:** For a set $P$ of $n$ points in $\mathbb{R}^d$ and some $\varepsilon \in (0, 1]$ we can compute an $(1 + \varepsilon)$-spanner for $P$ with $O(n/\varepsilon^d)$ edges in $O(n \log n + n/\varepsilon^d)$ time.

**Proof:** For $t = (1 + \varepsilon)$ we have with $s = 4 \cdot \frac{t+1}{t-1}$ that

$$O(s^d n) = O \left( \left( 4 \cdot \frac{2 + \varepsilon}{\varepsilon} \right)^d n \right) \subseteq O \left( \left( \frac{12}{\varepsilon} \right)^d n \right) = O \left( \frac{n}{\varepsilon^d} \right)$$
Summary

**Thm 4:** For a set $P$ of $n$ points in $\mathbb{R}^d$ and some $\varepsilon \in (0, 1]$ we can compute an $(1 + \varepsilon)$-spanner for $P$ with $O(n/\varepsilon^d)$ edges in $O(n \log n + n/\varepsilon^d)$ time.

**Proof:** For $t = (1 + \varepsilon)$ we have with $s = 4 \cdot \frac{t+1}{t-1}$ that

$$O(s^d n) = O\left(\left(4 \cdot \frac{2 + \varepsilon}{\varepsilon}\right)^d n\right) \subseteq O\left(\left(\frac{12}{\varepsilon}\right)^d n\right) = O\left(\frac{n}{\varepsilon^d}\right)$$

**Diagram:**

\[ P \quad \text{compressed quadtree} \quad \text{WSPD} \quad (1 + \varepsilon)\text{-spanner} \]

\[ O(n \log n) \quad O(n/\varepsilon^d) \quad O(n/\varepsilon^d) \]