CMSC 754: Lecture 17
Applications of WSPDs

Reading: This material is not covered in our text. The WSPD utility lemma is from M. Smid, "The well-separated pair decomposition and its applications," (2005).

Review: Recall that given a parameter $s > 0$, we say that two sets of A and B are *s-well separated* if the sets can be enclosed within two spheres of radius r such that the closest distance between these spheres is at least sr. Given a point set P and separation factor $s > 0$, recall that an *s-well separated pair decomposition* (s-WSPD) is a collection of pairs of subsets of P $\{\{A_1, B_1\}, \{A_2, B_2\}, \ldots, \{A_m, B_m\}\}$ such that

(1) $A_i, B_i \subseteq P$, for $1 \le i \le m$
(2) $A_i \cap B_i = \emptyset$, for $1 \le i \le m$
(3) $\bigcup_{i=1}^{n} A_i \otimes B_i = P \otimes P$
(4) A_i and B_i are s-well separated, for $1 \le i \le m$,

where $A \otimes B$ denotes the set of all unordered pairs from A and B.

Last time we showed that, given $s \ge 1$, there exists an s-WSPD of size $O(s^d n)$, which can be constructed in time $O(n \log n + s^d n)$. (The algorithm works for any $s > 0$, and the s^d term is more accurately stated as $\max(1, s)^d$.)

Recall that the WSPD is represented as a set of unordered pairs of nodes of a compressed quadtree decomposition of P. It is possible to associate each nonempty node u of the compressed quadtree with a *representative point*, denoted $\mathrm{rep}(u)$, chosen from its descendants. We will make use of this fact in some of our constructions below.

Today we discuss a number of applications of WSPDs. Many of the applications will make use of the following handy technical lemma (see Fig. 1).

Lemma: (WSPD Utility Lemma) If the pair $\{P_u, P_v\}$ is s-well separated and $x, x' \in P_u$ and $y, y' \in P_v$ then:

(i) $\|x - x'\| \le \frac{2}{s} \cdot \|x - y\|$
(ii) $\|x' - y'\| \le \left(1 + \frac{4}{s}\right) \|x - y\|$

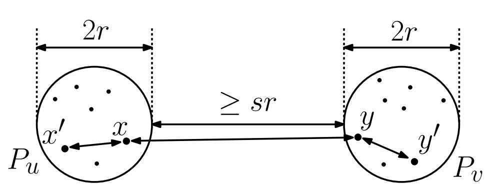

Fig. 1: WSPD Utility Lemma.

Proof: Since the pair is s-well separated, we can enclose each of P_u and P_v in a ball of radius r such that the minimum separation between these two balls is at least sr. It follows that $\max(\|x - x'\|, \|y - y'\|) \leq 2r$, and any pair from $\{x, x'\} \times \{y, y'\}$ is separated by a distance of at least sr. Thus, we have

$$\|x - x'\| \;\leq\; 2r \;=\; \frac{2r}{sr} sr \;\leq\; \frac{2r}{sr}\|x - y\| \;=\; \frac{2}{s}\|x - y\|,$$

which proves (i). Also, through an application of the triangle inequality ($\|a - c\| \leq \|a - b\| + \|b - c\|$) and the fact that $2r \leq \frac{2}{s}\|x - y\|$ we have

$$\begin{aligned}\|x' - y'\| \;&\leq\; \|x' - x\| + \|x - y\| + \|y - y'\| \;\leq\; 2r + \|x - y\| + 2r \\ &\leq\; \frac{2}{s}\|x - y\| + \|x - y\| + \frac{2}{s}\|x - y\| \;=\; \left(1 + \frac{4}{s}\right)\|x - y\|,\end{aligned}$$

which proves (ii).

Approximating the Diameter: The *diameter* of a point set is defined to be the maximum distance between any pair of points of the set. (For example, the points x and y in Fig. 2(a) define the diameter.)

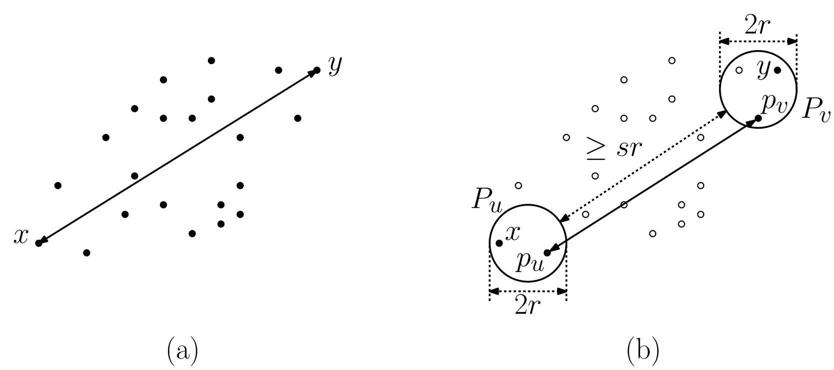

Fig. 2: Approximating the diameter.

The diameter can be computed exactly by brute force in $O(n^2)$ time. For points in the plane, it is possible to compute the diameter[1] in $O(n \log n)$ time. Generalizing this method to higher dimensions results in an $O(n^2)$ running time, which is no better than brute force search.

Using the WSPD construction, we can easily compute an ε-approximation to the diameter of a point set P in linear time. Given ε, we let $s = 4/\varepsilon$ and construct an s-WSPD. As mentioned above, each pair (P_u, P_v) in our WSPD construction consists of the points descended from two nodes, u and v, in a compressed quadtree. Let $p_u = \mathrm{rep}(u)$ and $p_v = \mathrm{rep}(v)$ denote the representative points associated with u and v, respectively. For every well separated pair $\{P_u, P_v\}$, we compute the distance $\|p_u - p_v\|$ between their representative, and output the pair achieving the largest such distance.

To prove correctness, let x and y be the points of P that realize the diameter. Let $\{P_u, P_v\}$ be the well separated pair containing these points, and let p_u and p_v denote their respective

[1] This is nontrivial, but is not much harder than a homework exercise. In particular, observe that the diameter points must lie on the convex hull. After computing the hull, it is possible to perform a rotating sweep that finds the diameter.

representatives. By the WSPD Utility Lemma we have

$$\|x-y\| \;\le\; \left(1+\frac{4}{s}\right)\|p_u-p_v\| \;=\; (1+\varepsilon)\|p_u-p_v\|.$$

Since $\{x,y\}$ is the diametrical pair, we have

$$\frac{\|x-y\|}{1+\varepsilon} \;\le\; \|p_u-p_v\| \;\le\; \|x-y\|,$$

which implies that the output pair $\{p_u,p_v\}$ is an ε-approximation to the diameter. The running time is dominated by the size of the WSPD, which is $O(s^d n) = O(n/\varepsilon^d)$. If we treat ε as a constant, this is $O(n)$.

Closest Pair (Exact!): The same sort of approach could be used to produce an ε-approximation to the closest pair as well, but surprisingly, there is a much better solution. If we were to generalize the above algorithm, we would first compute an s-WSPD for an appropriate value of s, and for each well separated pair $\{P_u,P_v\}$ we would compute the distance $\|p_u-p_v\|$, where $p_u=\text{rep}(u)$ and $p_v=\text{rep}(v)$, and return the smallest such distance. As before, we would like to argue that (assuming s is chosen properly) this will yield an approximation to the closest pair. It is rather surprising to note that, if s is chosen carefully, this approach yields the *exact* closest pair, not just an approximation.

To see why, consider a point set P, let x and y be the closest pair of points and let p_u and p_v be the representatives from their associated well separated pair. If it were the case that $x=p_u$ and $y=p_v$, then the representative-based distance would be exact. Suppose therefore that either $x\ne p_u$ or $y\ne p_v$. But wait! If the separation factor is high enough, this would imply that either $\|x-p_u\|<\|x-y\|$ or $\|y-p_v\|<\|x-y\|$, either of which contradicts the fact that x and y are the closest pair.

To make this more formal, let us assume that $\{x,y\}$ is the closest pair and that $s>2$. We know that P_u and P_v lie within balls of radius r that are separated by a distance of at least $sr>2r$. If $p_u\ne x$, then we have

$$\|p_u-x\| \;\le\; 2r \;<\; sr \;\le\; \|x-y\|,$$

yielding a contradiction. Therefore $p_u=\text{rep}(u)=x$. By a symmetrical argument $p_v=\text{rep}(v)=y$. Since the representative was chosen arbitrarily, it follows that the $P_u=\{x\}$ and $P_v=\{y\}$. Therefore, the closest representatives are in fact, the *exact* closest pair.

Since s can be chosen to be arbitrarily close to 2, the running time is $O(n\log n+2^d n)=O(n\log n)$, since we assume that d is a constant. Although this is not a real improvement over our existing closest-pair algorithm, it is interesting to note that there is yet another way to solve this problem.

Low-Stretch Spanners: Recall that a set P of n points in $\mathbb{R}^d$ defines a complete weighted graph, called the *Euclidean graph*, in which each point is a vertex, and every pair of vertices is connected by an edge whose weight is the Euclidean distance between these points. This graph is *dense*, meaning that it has $\Theta(n^2)$ edges. Intuitively, a spanner is a *sparse graph* (having only

$O(n)$ edges) in which shortest paths are not significantly longer than the Euclidean distance between points. Such a graph is called a (Euclidean) *spanner*.

More formally, suppose that we are given a set P in $\mathbb{R}^d$ and a parameter $t \geq 1$, called the *stretch factor*. A *t-spanner* is a weighted graph G whose vertex set is P and, given any pair of points $x, y \in P$ we have

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

where $\delta_G(x, y)$ denotes the length of the shortest path between x and y in G.

In an earlier lecture, we showed that the Delaunay triangulation of P is an $O(1)$-spanner. This was only really useful in the plane, since in dimension 3 and higher, the Delaunay triangulation can have a quadratic number of edges. Here we consider the question of how to produce a spanner in any space of constant dimension that achieves any desired stretch factor $t > 1$. There are many different ways of building spanners. Here we will discuss a straightforward method based on a WSPD of the point set.

WSPD-based Spanner Construction: Given the point set P and a (constant) stretch factor t, the idea is to build an s-WSPD for P, where s is an appropriately chosen separation factor (which will depend on t). We will then create one edge in the spanner from each well-separated pair.

Given t, we set $s = 4(t + 1)/(t - 1)$. (Later we will justify the mysterious choice.) For each well-separated pair $\{P_u, P_v\}$ associated with the nodes u and v of the quadtree, let $p_u = \text{rep}(u)$ and let $p_v = \text{rep}(v)$. Add the undirected edge $\{p_u, p_v\}$ to our graph. Let G be the resulting undirected weighted graph (see Fig. 3). G will be the desired spanner. Clearly the number of edges of G is equal to the number of well-separated pairs, which is $O(s^d n) = O(n)$, and it can be built in the same $O(n \log n + s^d n) = O(n \log n)$ running time as the WSPD construction.

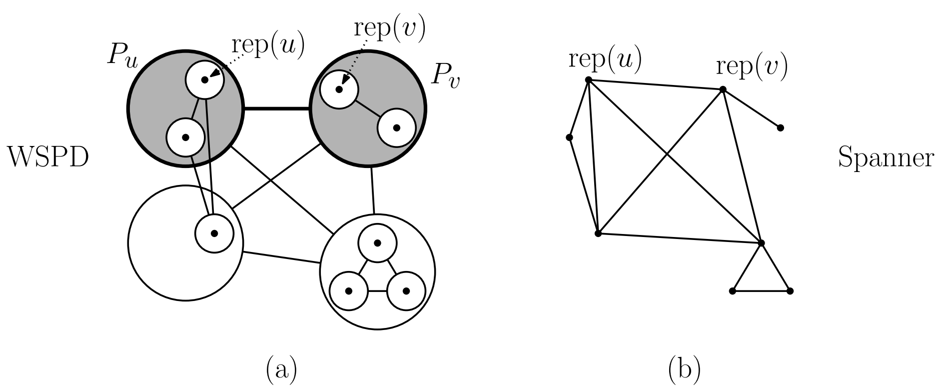

Fig. 3: A WSPD and its associated spanner.

Correctness: To establish the correctness of our spanner construction algorithm, it suffices to show that for all pairs $x, y \in P$, we have

$$\|x - y\| \;\leq\; \delta_G(x, y) \;\leq\; t \cdot \|x - y\|.$$

Clearly, the first inequality holds trivially, because (by the triangle inequality) no path in any graph can be shorter than the distance between the two points. To prove the second inequality, we apply an induction based on the number of edges of the shortest path in the spanner.

For the basis case, observe that, if x and y are joined by an edge in G, then clearly $\delta_G(x, y) = \|x - y\| \le t \cdot \|x - y\|$ for all $t \ge 1$.

If, on the other hand, there is no direct edge between x and y, we know that x and y must lie in some well-separated pair $\{P_u, P_v\}$ defined by the pair of nodes $\{u, v\}$ in the quadtree. let $p_u = \text{rep}(u)$ and $p_v = \text{rep}(v)$ be the respective representative points. (It might be that $p_u = x$ or $p_v = y$, but not both.) Let us consider the length of the path from x to p_u to p_v to y. Since the edge $\{p_u, p_v\}$ is in the graph, we have

$$\begin{aligned} \delta_G(x, y) &\le \delta_G(x, p_u) + \delta_G(p_u, p_v) + \delta_G(p_v, y) \\ &\le \delta_G(x, p_u) + \|p_u - p_v\| + \delta_G(p_v, y). \end{aligned}$$

(See Fig. 4.)

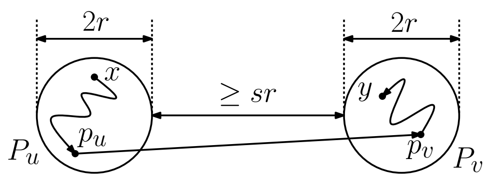

Fig. 4: Proof of the spanner bound.

The paths from x to p_u and p_v to y are subpaths of the full spanner path from x to y, and hence they use fewer edges. Thus, we may apply the induction hypothesis, which yields $\delta_G(x, p_u) \le t\|x - p_u\|$ and $\delta_G(p_v, y) \le t\|p_v - y\|$, yielding

$$\delta_G(x, y) \le t\,(\|x - p_u\| + \|p_v - y\|) + \|p_u - p_v\|. \tag{1}$$

By the WSPD Utility Lemma (with $\{x, p_u\}$ from one pair and $\{y, p_v\}$ from the other) we have

$$\max(\|x - p_u\|, \|p_v - y\|) \le \frac{2}{s} \cdot \|x - y\| \qquad \text{and} \qquad \|p_u - p_v\| \le \left(1 + \frac{4}{s}\right) \|x - y\|.$$

Combining these observations with Eq. (1) we obtain

$$\delta_G(x, y) \le t\left(2 \cdot \frac{2}{s} \cdot \|x - y\|\right) + \left(1 + \frac{4}{s}\right)\|x - y\| = \left(1 + \frac{4(t+1)}{s}\right)\|x - y\|.$$

To complete the proof, observe that it suffices to select s so that $1 + 4(t+1)/s \le t$. Towards this end, let us set

$$s = 4\left(\frac{t+1}{t-1}\right).$$

This is well defined for any $t > 1$. By substituting in this value of s, we have

$$\delta_G(x,y) \;\leq\; \left(1 + \frac{4(t+1)}{4(t+1)/(t-1)}\right)\|x-y\| \;=\; (1+(t-1))\|x-y\| \;=\; t\cdot\|x-y\|,$$

which completes the correctness proof.

Because we have one spanner edge for each well-separated pair, the number of edges in the spanner is $O(s^d n)$. Since spanners are most interesting for small stretch factors, let us assume that $t \leq 2$. If we express t as $t = 1+\varepsilon$ for $\varepsilon \leq 1$, we see that the size of the spanner is

$$O(s^d n) \;=\; O\left(\left(4\frac{(1+\varepsilon)+1}{(1+\varepsilon)-1}\right)^d n\right) \;\leq\; O\left(\left(\frac{12}{\varepsilon}\right)^d n\right) \;=\; O\left(\frac{n}{\varepsilon^d}\right).$$

In conclusion, we have the following theorem:

Theorem: Given a point set P in $\mathbb{R}^d$ and $\varepsilon > 0$, a $(1+\varepsilon)$-spanner for P containing $O(n/\varepsilon^d)$ edges can be computed in time $O(n\log n + n/\varepsilon^d)$.

Approximating the Euclidean MST: The Euclidean Minimum Spanning Tree (EMST) of a point set P is the minimum spanning tree of the complete Euclidean graph on P. In an earlier lecture, we showed that the EMST is a subgraph of the Delaunay triangulation of P. This provided an $O(n\log n)$ time algorithm in the plane. Unfortunately, the generalization to higher dimensions was not interesting because the worst-case number of edges in the Delaunay triangulation is quadratic in dimensions 3 and higher.

We will now that for any constant approximation factor ε, it is possible to compute an ε-approximation to the minimum spanning tree in any constant dimension d. Given a graph G with v vertices and e edges, it is well known that the MST of G can be computed in time $O(e + v\log v)$. It follows that we can compute the EMST of a set of points in any dimension by first constructing the Euclidean graph and then computing its MST, which takes $O(n^2)$ time. To compute the approximation to the EMST, we first construct a $(1+\varepsilon)$-spanner, call it G, and then compute and return the MST of G (see Fig. 5). This approach has an overall running time of $O(n\log n + s^d n)$.

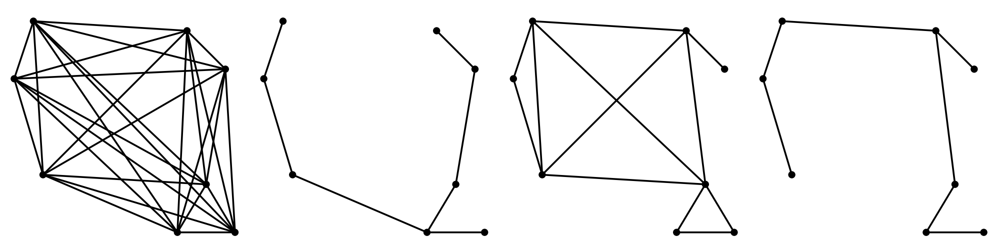

Fig. 5: Approximating the Euclidean MST.

To see why this works, consider any pair of points $\{x,y\}$, and let $w(x,y) = \|x-y\|$ denote the weight of the edge between them in the complete Euclidean graph. Let T denote the edges of the Euclidean minimum weight spanning tree, and $w(T)$ denote the total weight of its edges. For each edge $\{x, y\} \in T$, let $\pi_G(x, y)$ denote the shortest path (as a set of edges) between x and y in the spanner, G. Since G is a spanner, we have

$$w(\pi_G(x, y)) \;=\; \delta_G(x, y) \;\leq\; (1+\varepsilon)\|x - y\|.$$

Now, consider the subgraph $G' \subseteq G$ formed by taking the union of all the edges of $\pi_G(x, y)$ for all $\{x, y\} \in T$. That is, G and G' have the same vertices, but each edge of the MST is replaced by its spanner path. Clearly, G' is connected (but it may not be a tree). We can bound the weight of G' in terms of the weight of the Euclidean MST:

$$\begin{aligned} w(G') &= \sum_{\{x,y\}\in T} w(\pi_G(x, y)) \;\leq\; \sum_{\{x,y\}\in T} (1+\varepsilon)\|x - y\| \\ &= (1+\varepsilon) \sum_{\{x,y\}\in T} \|x - y\| \;=\; (1+\varepsilon)w(T). \end{aligned}$$

However, because G and G' share the same vertices, and the edge set of G' is a subset of the edge set of G, it follows that $w(\mathrm{MST}(G) \leq w(\mathrm{MST}(G'))$. (To see this, observe that if you have fewer edges from which to form the MST, you may generally be forced to use edges of higher weight to connect all the vertices.) Combining everything we have

$$w(\mathrm{MST}(G)) \;\leq\; w(\mathrm{MST}(G')) \;\leq\; w(G') \;\leq\; (1+\varepsilon)w(T),$$

yielding the desired approximation bound.