Computational Geometry Lecture 33
(30 B following lecture 30)

Topic

Construction of ϵ-WSPD

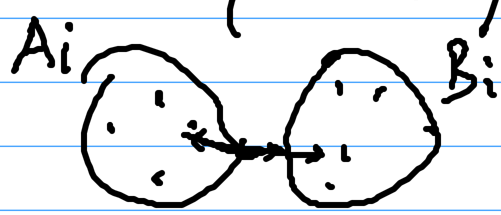
Given a point set S, we want to generate pairs of subsets (A_i, B_i) s.t.

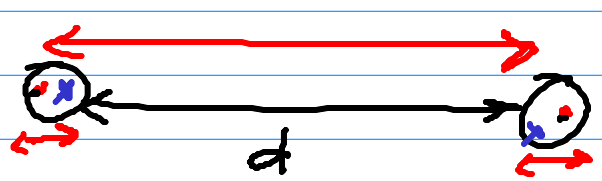
(i) $A_i \cap B_i = \emptyset$ (disjoint)

(ii) $\forall$ all $(p_i, p_j) \in S$, there exists some (A_k, B_k), s.t.
$p_i \in A_k$ $p_j \in B_k$

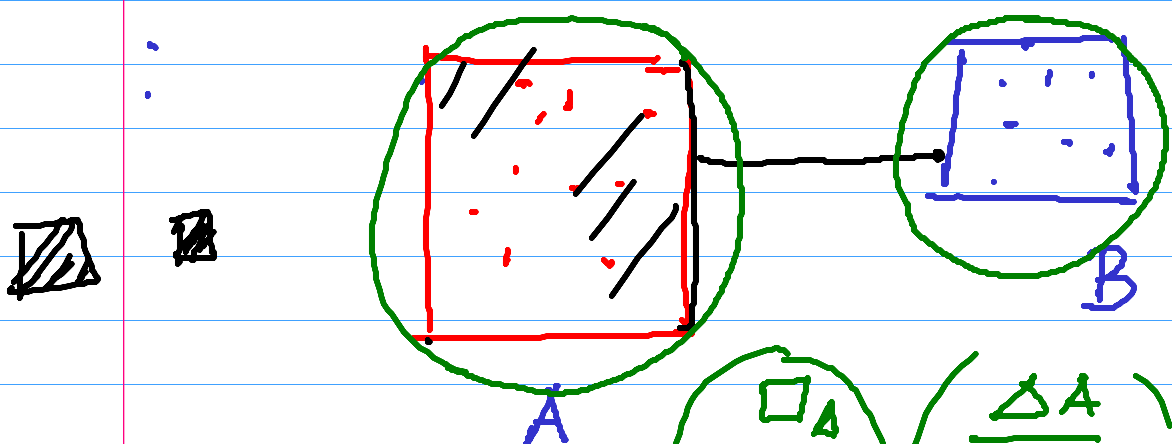
(iii) The diameter of the subsets should be "small" compared to the distance $d(A_i, B_i)$

$$\max\{\Delta(A_i), \Delta(B_i)\} \leq \epsilon \cdot d(A_i, B_i)$$





For the given point set S, we have constructed a compressed quadtree T of $O(n)$ nodes and $O(\log n)$ depth. This construction takes $O(n \log n)$ time.



$\Delta(A)$, $\Delta(B)$ $\qquad \frac{\square_A}{d(\square_A, \square_B)} \le \left(\frac{\Delta A}{d(A,B)} \right) \le \varepsilon$

$d(A,B)$

Suppose we approximate:

$\Delta(A)$ by $\square_A$: the cell enclosing points of A

$\square_A \geq \Delta(A)$

$\Delta(B)$ by $\square_B$

$d(A,B) \geq d(\square_A, \square_B)$

If we can ensure $\square_A \leq \varepsilon \cdot d(\square_A, \square_B)$

Alg WSPD (u, v) (T is quadtree on P)

If $u = v$ and $\Delta(u) = 0$ then return
[a single point doesn't generate any pairs]

else
if $\square_u < \square_v$ then exchange (u, v)
(i.e. $\square_u \geq \square_v$)

If $\square_u \leq \varepsilon \cdot d(\square_u, \square_v)$
then return $\{(u, v)\}$
(u, v is a valid pair)

else If $u_1, u_2 \ldots u_r$ are children of u in T
return $\bigcup_{i=1}^{r}$ AlgWSPD (u_i, v)

—— end ——

Initially $u = v = S$: root of T

<u>Claim</u> : Alg WSPD (root(T), root(T))

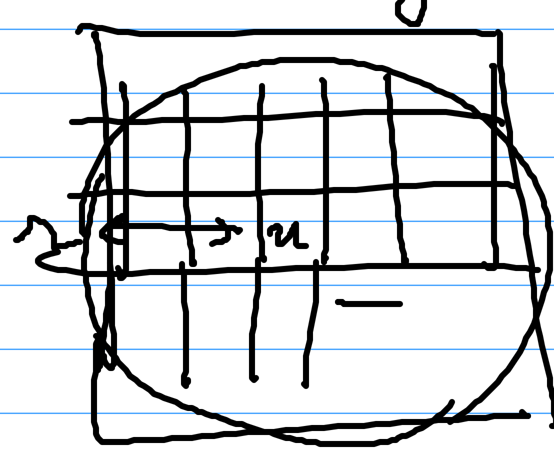
produces $O\left(\frac{n}{(\epsilon)^d}\right)$ pairs

and runs in $O\left(n\log n + \frac{n}{(\epsilon)^d}\right)$ steps

where d is the dimension

<u>Observation 1</u>: For a given level of the canonical grid the number of cells at a distance r from a fixed cell $\Box_u$ can be bounded by

$$\left(\frac{2r}{\Box_u}\right)^d$$



Look at any sequence of recursive calls

$$(u_1, v_1) \to \underbrace{(u_2, v_2) \to (u_3, v_3) \cdots}_{\text{No pairs are generated}} \to (u_s, v_s)$$

(u_s, v_s) is a pair generated

recursive calls can be charged to the number of pairs generated

We want to count the number of pairs generated.
For that we will charge the pair generated to the node of T that was the last recursive call.

<u>Lemma</u>: The total number of charges in any node of T is $O\left(\frac{2^{2d}}{\epsilon^{d}}\right)$