CG
10/19/10

Nearest Neighbor Search

Input: $S \subseteq R^d$ $|S| = n$

Output: Search data structure DS

st

$\forall x \in R^d$ $DS(x) \in S$ closest point

Cost:

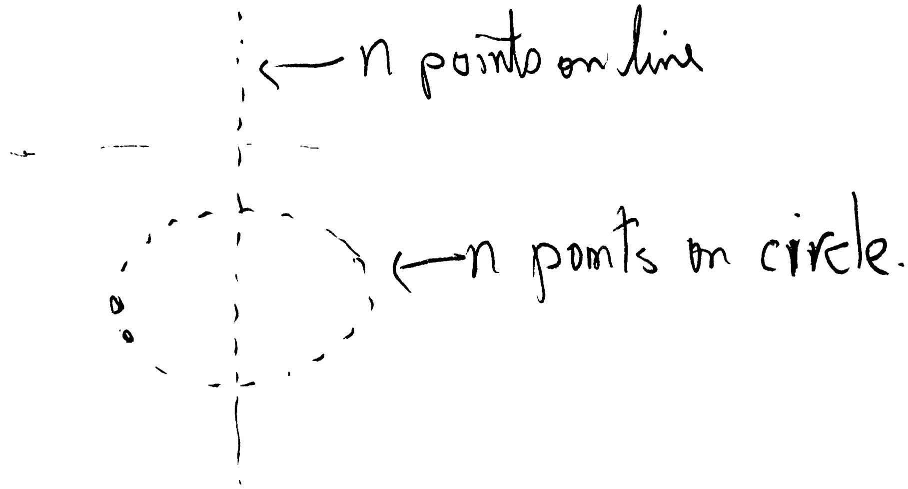
1) Time to compute DS
2) Size of DS
3) Query time

Quivalent: to Finding the Voronoi cell containing x.

Prob: # of Voronoi points on d maybe of size $O(n^{\lceil d/2 \rceil})$

# of Delaunay Simplices

Preparata's Ex.



Claim: neig pair on circle + neig pair on line form a Del-Tet

$\therefore n^2$ Tets $\quad d=3 \quad n^{\lceil d/2 \rceil} = n^2$

Naive DS may have size $\Omega(n^2)$ in 3D.

<u>Work around</u>: Approximate NN

<u>Def</u> $P \in S$ is a $(1+\varepsilon)$-approx NN of q if

$$\frac{dist(P,q)}{dist(P',q)} \leq (1+\varepsilon) \quad \forall P' \in S$$

<u>Goal!</u> $O(n \log n)$ preprocess

$O(n)$ space DS

$O(\log n)$ query time $(1+\varepsilon)$-approx

d fixed constants maybe expon in d or worst

---

set $d=2$.

DS $\equiv$ 1) A partition of space into boxes (cells)

2) Alg to determine box containing q

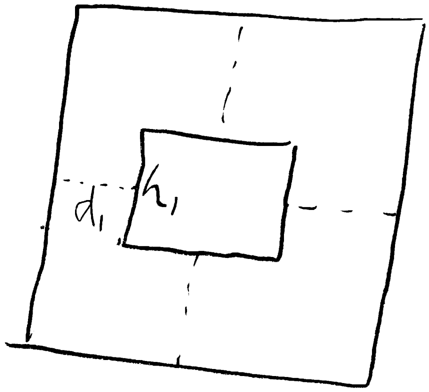
3) A seach of neig box for approx-NN

Box $\equiv$ Rect 1) $\frac{\text{long-side}}{\text{shortest-side}} \leq 3$ 2) aligned

Cell $\equiv$ 1) a Box

2) Diff of an outer-box & an inter-box. (sticky)
b_O b_I

<u>Def</u> b_I is <u>sticky</u> in b_O if



$d=0$ or $d \geq h$

$d_i = 0$ or $d_i \geq h_i$

Split(b, S) b: box $S \subseteq \mathbb{R}^2$

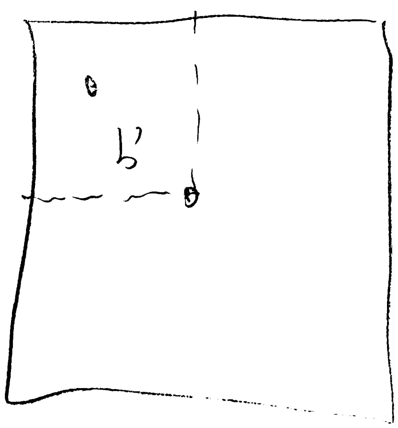
Let b_1, b_2 be an even split along longest side of b

if $b_1 \cap S \neq \emptyset$ & $b_2 \cap S \neq \emptyset$ then return b_1, b_2

else "not possible"

Shrink(b, S)

$\underset{b' \subseteq b}{\text{Minimal}} \{ S \subseteq b' \wedge b' \text{ is a box} \wedge \text{it is sticky} \}$



BoxDecomp(T, b)

if $|T| \leq$ BucketSize then create b.

else

if (split(b) = "not possible")

b' = shrink(b)

create cell $b \setminus b'$

$b \leftarrow b'$

(b_1, b_2) = split(b)

BoxDecomp$(T \cap b_1, b_1)$

BoxDecomp$(T \cap b_2, b_2)$

Def $\mathrm{dist}(q, \mathrm{cell}) = \mathrm{argmin}\{ \mathrm{dist}(q,p) \mid p \in \mathrm{cell} \}$

Search(q,b)

Set $p \in S$; cell = b

while($\mathrm{dist}(q, \mathrm{cell}) \leq (1+\varepsilon)\,\mathrm{dist}(q,p)$)

$p \leftarrow$ closest point in cell

cell = next closest cell (cell)