

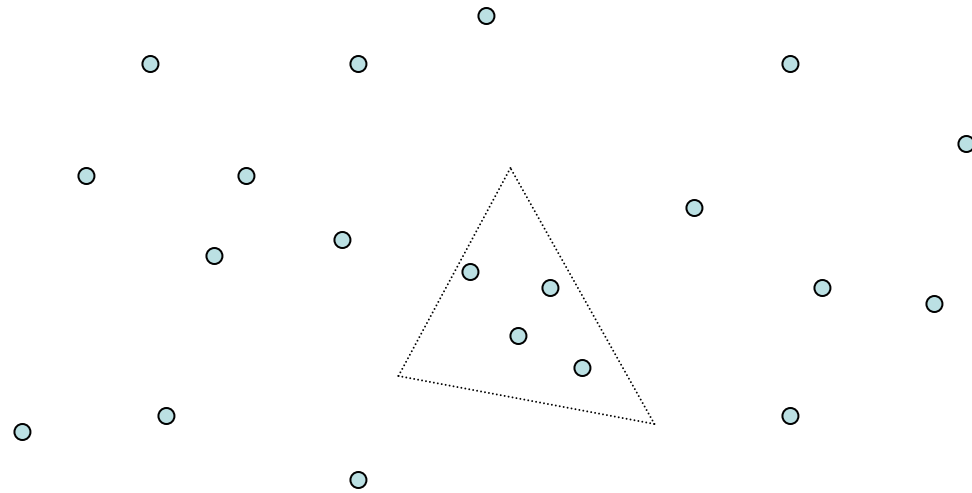
Optimal Partition Trees

Timothy Chan

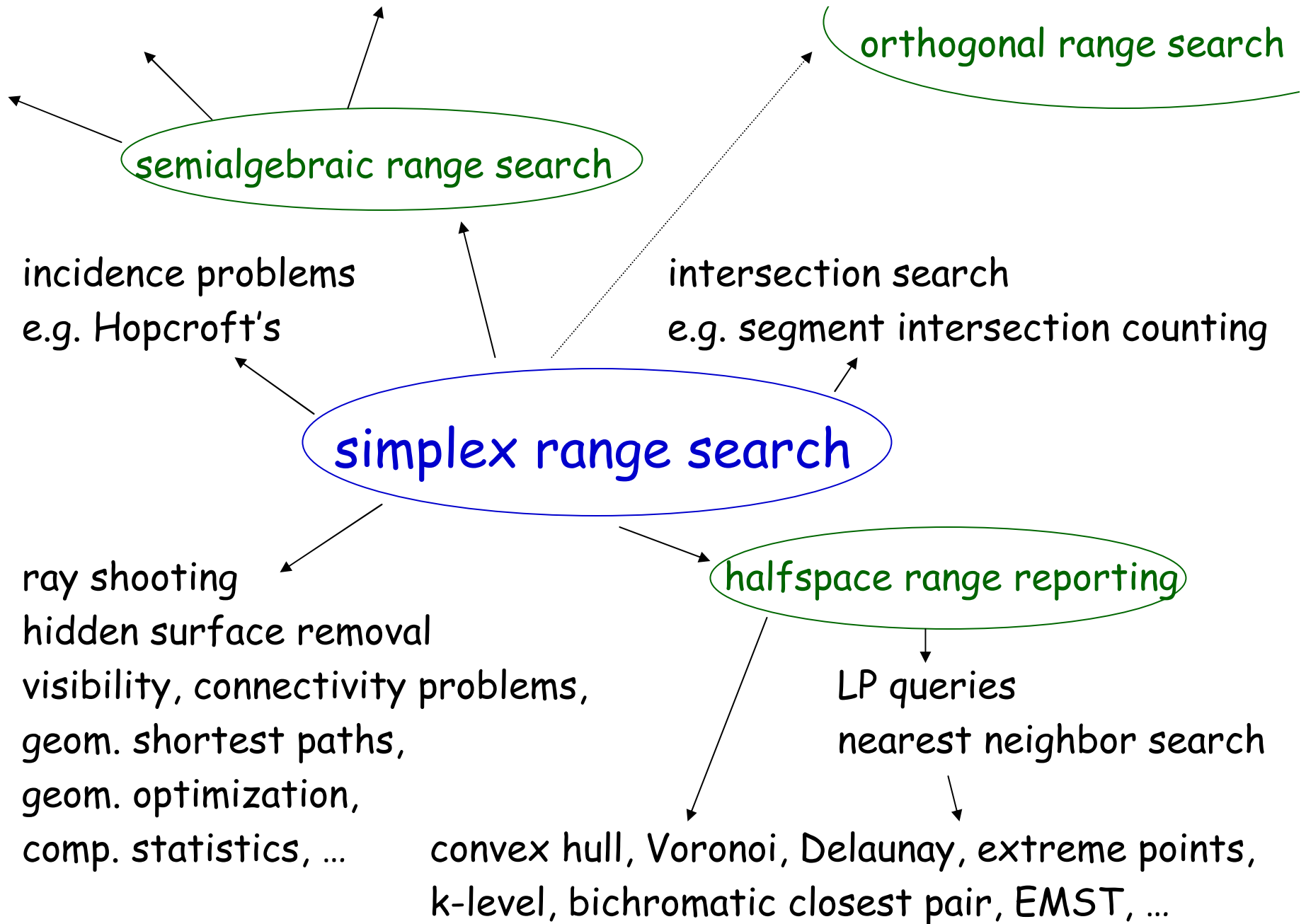
School of CS

U. Waterloo

Range Searching



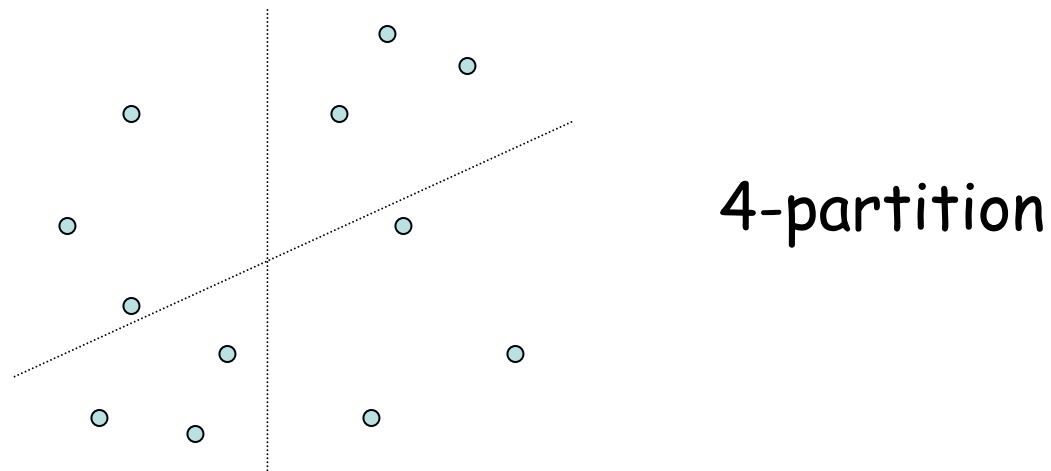
at the "core" of computational geometry...



History of Simplex Range Search

($O(n)$ -space data structures)

- 2-d: Willard'82 $O(n^{\log_4 3}) \approx O(n^{0.792})$



Willard'82 $O(n^{\log_6 4}) \approx O(n^{0.774})$
Edelsbrunner, Welzl'86 $O(n^{\log_2 \phi}) \approx O(n^{0.695})$

History (Cont'd)

- **3-d:** F. Yao'83 $O(n^{\log_8 7}) \approx O(n^{0.936})$
Dobkin, Edelsbrunner'84 $O(n^{0.916})$
Edelsbrunner, Huber'84 $O(n^{0.909})$
Yao, Dobkin, Edelsbrunner, Paterson'89 $O(n^{0.899})$
- **Higher-d:** Yao, Yao'85 $O(n^{\log_2(2^d-1) / d})$

History (Cont'd)

- Haussler, Welzl [SoCG'86]

$$O(n^{1-1/[d(d-1)+1]+\epsilon})$$

e.g. 2-d: $O(n^{2/3+\epsilon})$, 3-d: $O(n^{0.858})$

key idea: random sampling (ϵ -nets)

History: Turning Point

- Welzl [SoCG'88], Chazelle-Welzl'89

2-d: $O(n^{1/2} \log n)$ query, $O(n)$ space

3-d: $O(n^{2/3} \log^2 n)$ query, $O(n \log n)$ space

$O(n^{1-1/d} \log n)$ query, $O(n)$ space in semigroup model
(but not "algorithmic")

spanning trees with low crossing number

key idea: iterative reweighting

History: Last Stretch

- Chazelle, Sharir, Welzl [SoCG'90]
 $O(n^{1-1/d+\epsilon})$ query, $O(n^{1+\epsilon})$ space
idea: multiple cuttings
- Matoušek [SoCG'91]
 $O(n^{1-1/d} \log^{O(1)} n)$ query, $O(n)$ space
"partition thm"
idea: iterative reweighting + cuttings
- Matoušek [SoCG'92]
 $O(n^{1-1/d})$ query, $O(n)$ space
"final method"
idea: iterative reweighting + Chazelle's hierarchical cuttings

Loose Ends

Matoušek's final method is great, but...

- has large preprocessing time: $O(n^{1+\varepsilon})$
(in algorithmic applications, we usually switch back to partition-thm method, which has $O(n \log n)$ preproc.)
- is not good for multilevel data structures
(Matoušek switched to multiple-cuttings method:
cost $1 \log$ in query, $2 \log$'s in space, n^ε in preproc. per level,
or $\log^{O(1)} n$ in query/space/preproc. per level)
- is complicated !!

New Result

$O(n^{1-1/d})$ query, $O(n)$ space

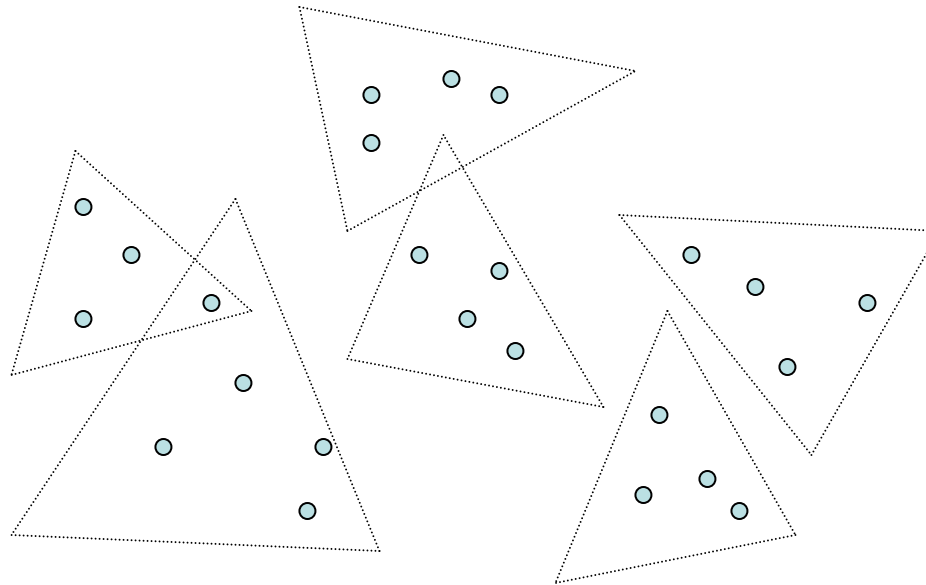
- with $O(n \log n)$ preprocessing time (rand. w.h.p.)
- good for multilevel data structures
(cost only 1 log in query/space/preproc. per level)
- simpler !!

Recap: Matoušek's Partition Thm

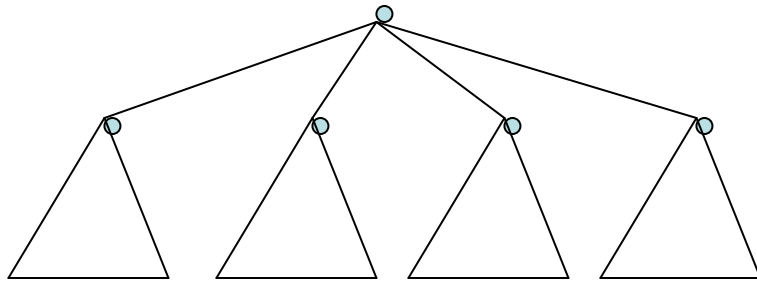
Let P be n -point set in \mathbb{R}^d , $t \leq n$.

Can partition P into t subsets of $O(n/t)$ points & enclose each subset in a (simplicial) cell s.t.

max # cells crossed by any hyperplane is $O(t^{1-1/d})$



Matoušek's Partition Tree



recurse

$$Q(n) = O(t^{1-1/d}) Q(n/t) + O(t)$$

$$\text{set } t = \text{large const} \Rightarrow Q(n) = O(n^{1-1/d+\epsilon})$$

$$\text{set } t = n^\epsilon \Rightarrow Q(n) = O(n^{1-1/d} \log^{O(1)} n)$$

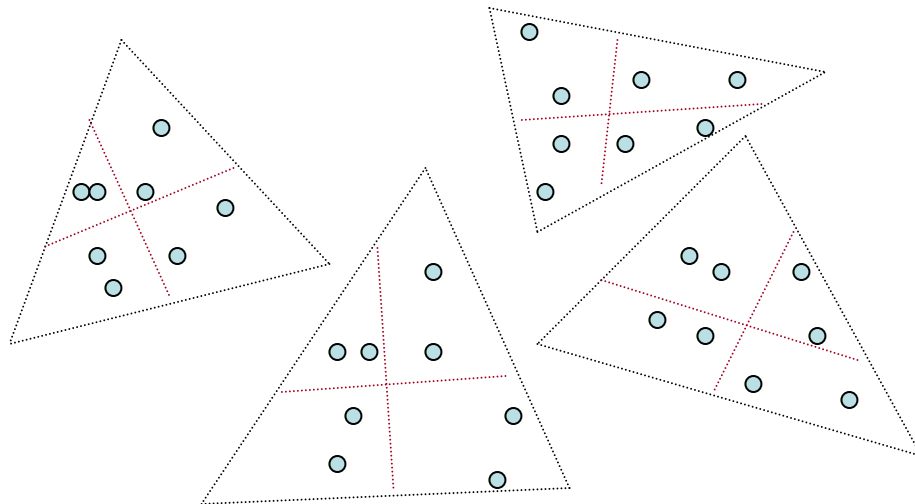
A New "Partition Refinement Thm"

Let P be n -point set in \mathbb{R}^d , $bt \leq n$.

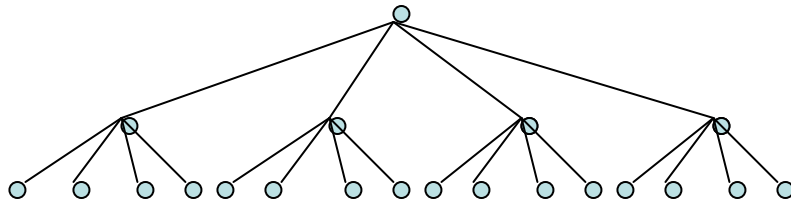
Given a partition with t disjoint cells each with $O(n/t)$ points s.t. max # cells crossed by any hyperplane is ℓ .

Can subdivide each cell into $O(b)$ disjoint subcells each with $O(n/bt)$ points s.t.

max total # subcells crossed by any hyperplane is
 $O((bt)^{1-1/d} + b^{1-1/(d-1)} \ell + b \log^{O(1)} n)$



The New Partition Tree



build level by level

...

$$\ell(bt) \approx O((bt)^{1-1/d} + b^{1-1/(d-1)} \ell(t))$$

$$\text{set } b = \text{large const} \Rightarrow \ell(n) = \boxed{O(n^{1-1/d})}$$

Proof Sketch of Partition Refinement Thm

Preliminaries

- Suffices to work with a finite set of $n^{O(1)}$ "test hyperplanes"
- **Cutting Lemma:** [Clarkson, Shor/Chazelle, Friedman]
Given m hyperplanes in \mathbb{R}^d & cell Δ containing X vertices, can divide Δ into $O(X(r/m)^d + r^{d-1})$ disjoint subcells s.t. each subcell is crossed by $O(m/r)$ hyperplanes

The New Algorithm

Idea: iterative reweighting

Initialize **multiplicity** ("weight") of each hyperplane to 1

For $i = t$ to 1 do:

0. Among the i remaining cells, pick a good cell Δ
1. Apply cutting lemma to subdivide Δ into $O(b)$ subcells
2. Further subdivide Δ s.t. each subcell has $O(n/bt)$ points
3. For each hyperplane h , multiply multiplicity of h by $(1+1/b)^{\lambda(h)}$ where $\lambda(h) = \#$ subcells of Δ crossed by h

Analysis

Let M = total multiplicity of all hyperplanes

0. Among the i remaining cells, pick a good cell Δ
with $X \leq O(M^d/i)$ vertices,
crossed by $m \leq O(M\ell/i)$ hyperplanes

1. Apply cutting lemma with $r = \min \{ m(b/X)^{1/d}, b^{1/(d-1)} \}$
to subdivide Δ into $O(X(r/m)^d + r^{d-1}) = O(b)$ subcells

$$\begin{aligned} \Rightarrow \# \text{ hyperplanes crossing each subcell} &\leq O(m/r) \\ &\leq O((X/b)^{1/d} + m/b^{1/(d-1)}) \\ &\leq M \cdot O(1/(bi)^{1/d} + \ell / (b^{1/(d-1)}i)) \end{aligned}$$

Analysis (Cont'd)

2. Further subdivide Δ s.t. each subcell has $O(n/bt)$ points
3. For each hyperplane h , multiply multiplicity of h by $(1+1/b)^{\lambda(h)}$ where $\lambda(h) = \#$ subcells of Δ crossed by h

$$\sum_h \lambda(h) \leq O(bm/r)$$

$$\Rightarrow \text{increase in } M \leq \sum_h [(1+1/b)^{\lambda(h)} - 1]$$

$$\leq O(\sum_h \lambda(h)/b)$$

$$\leq O(m/r)$$

$$\leq M \cdot O(1/(bi)^{1/d} + \ell / (b^{1/(d-1)}i))$$

Analysis (Cont'd)

- Final value of M

$$\begin{aligned} &\leq n^{O(1)} \prod_{i=t, \dots, 1} [1 + O(1/(bi)^{1/d} + \ell / (b^{1/(d-1)}i))] \\ &\leq n^{O(1)} \exp(O(t^{1-1/d}/b^{1/d} + \ell \ln t / b^{1/(d-1)})) \end{aligned}$$

- Final multiplicity of $h = (1+1/b)^{\text{crossing-number}(h)}$
 \leq final value of M

$$\begin{aligned} \Rightarrow \text{crossing-number}(h) &\leq b \log(\text{final value of } M) \\ &\leq \boxed{O((bt)^{1-1/d} + b^{1-1/(d-1)} \ell \ln t + b \log n)} \end{aligned}$$

Concluding Remarks

- Simple tree structure, with const degree, disjoint cells (in 2-d, can make it a BSP tree)
- Optimal crossing number at essentially all levels of tree
- **An open problem:** halfspace range reporting for odd d with $O(n^{1-1/\lfloor d/2 \rfloor} + k)$ query, $O(n)$ space?