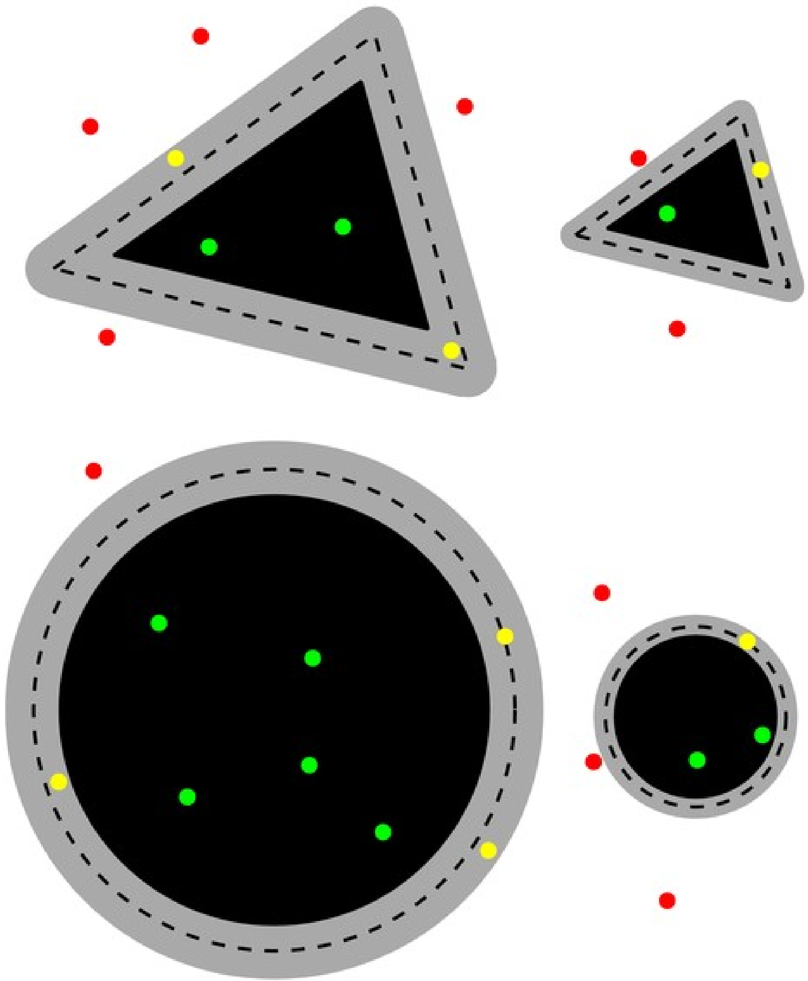
Tradeoffs in Approximate Range Searching Made Simpler



- Sunil Arya

  Hong Kong University of Science and Technology

- Guilherme D. da Fonseca

  Universidade Federal do Rio de Janeiro

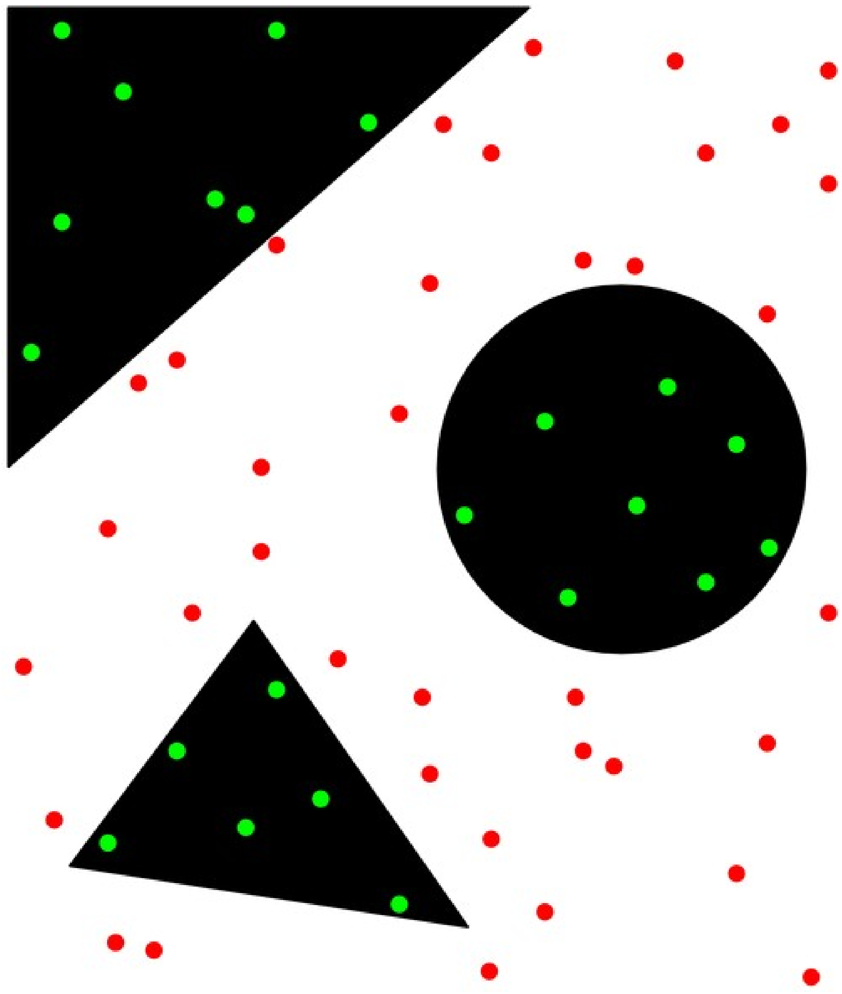
- David M. Mount

  University of Maryland

SIBGRAPI, Campo Grande, MS

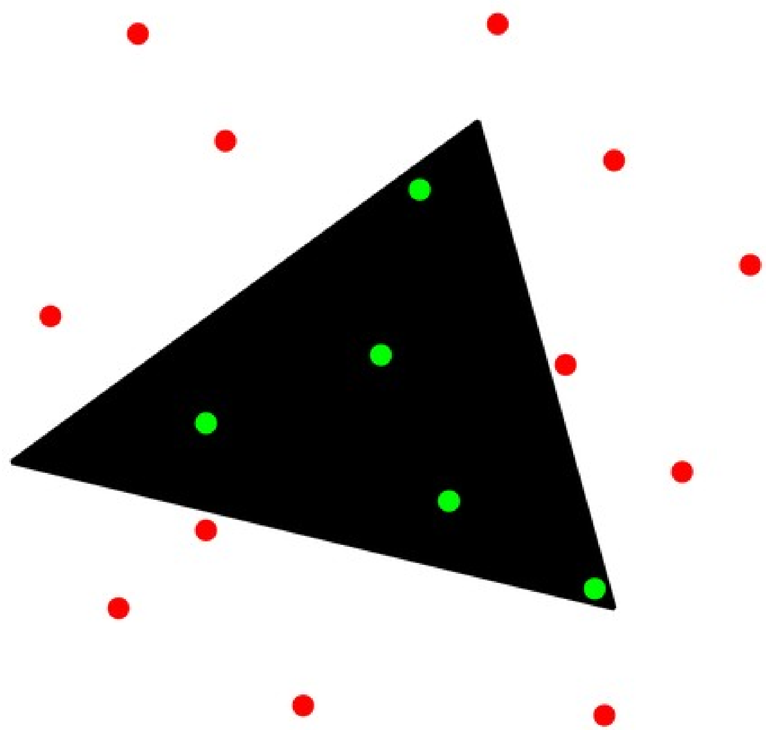
10/2008

Contents



- Range Searching
- Quadtrees
- Range Sketching
- Halfspaces
- Spheres
- Simplices
- Future research

Exact Range Searching



P: Set of n points in d-dimensional space.

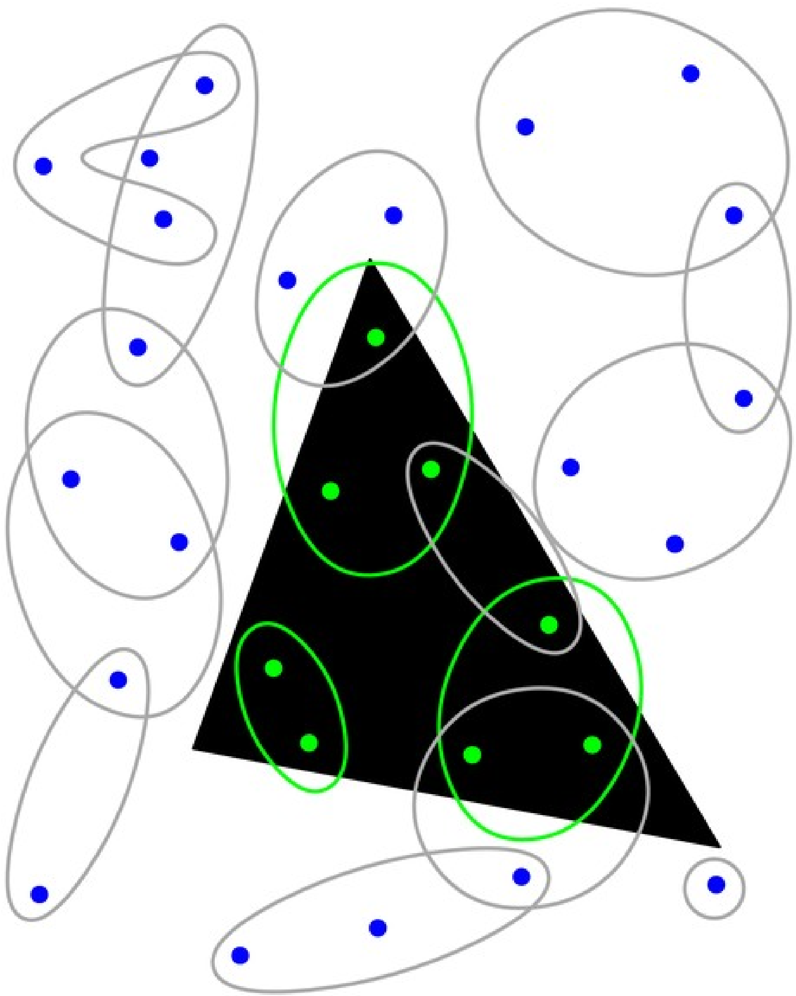
w: Weight function.

$\mathcal{R}$: Set of regions (ranges).

- Preprocess P such that, given $R \in \mathcal{R}$, we can quickly compute:

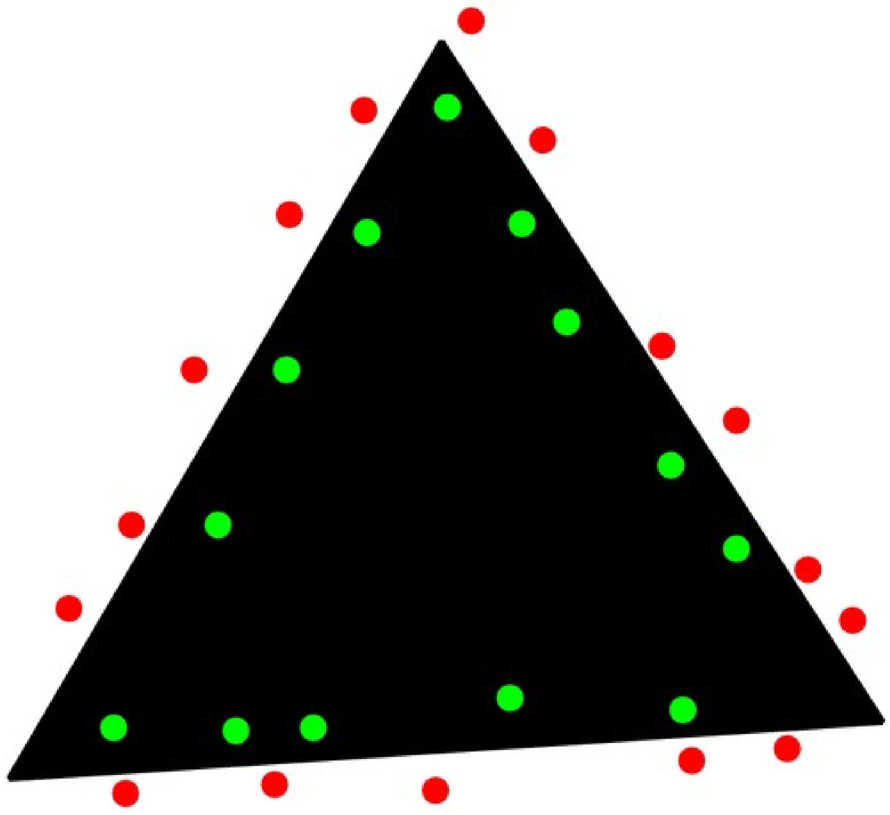
$$\sum_{p \in P \cap R} w(p)$$

Generators and Tradeoffs



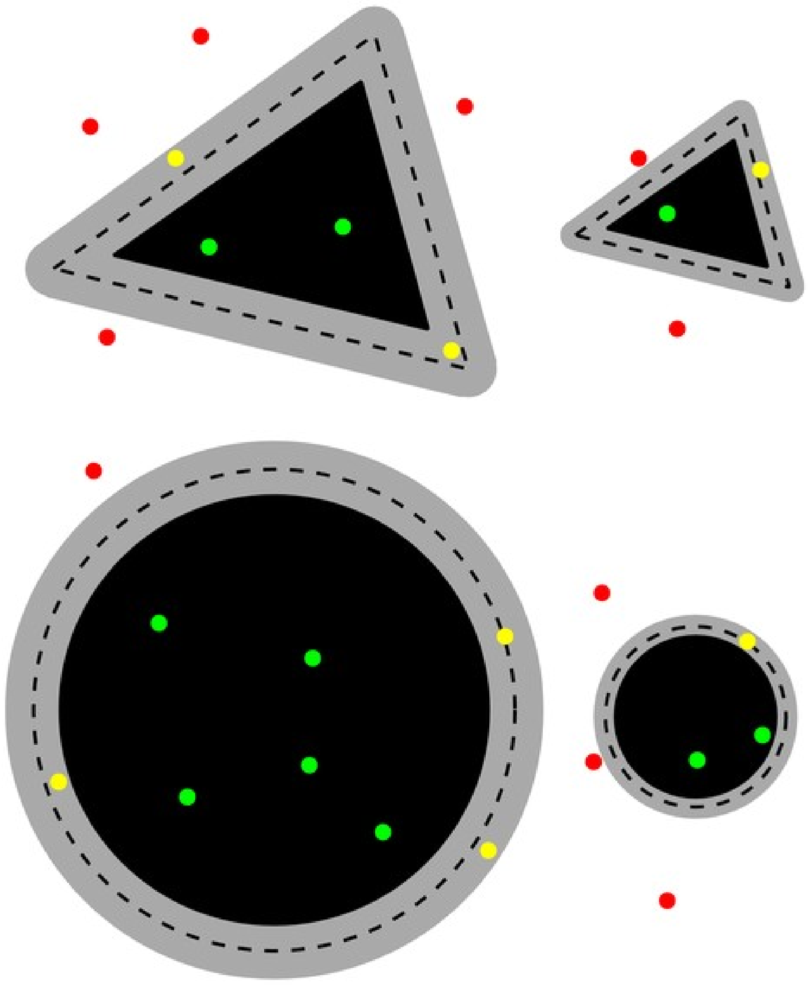
- A generator is a set of point whose sum is precomputed.
- We answer a query by adding generators.
- Tradeoff:
  - Many large generators: High storage, low query time.
  - Few small generators: Low storage, high query time.

Why approximate?



- Exact solutions are complicated and inefficient.
- Polylogarithmic time requires n^d space.
- With linear space, the query time approaches O(n) as d increases.
- Troublemakers: Points close to the boundary of the query region.

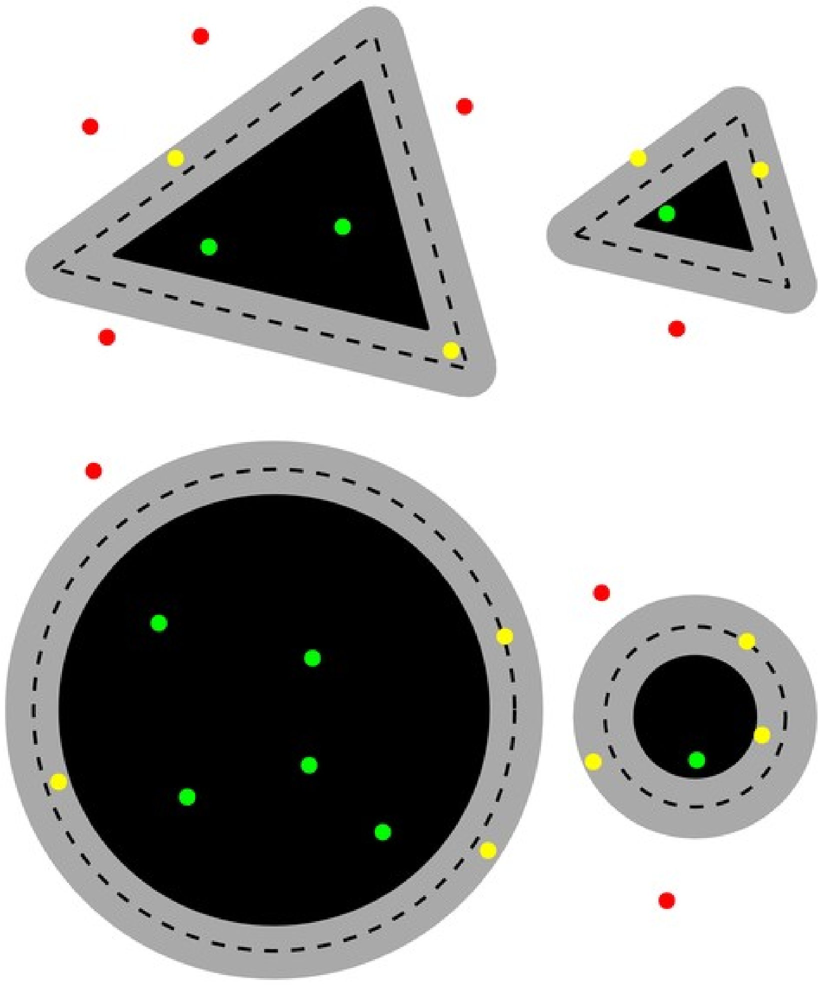
Relative Model



- In the **relative** model, points within distance ε $diam(R)$ of the range boundary may be counted or not. [AM00]
- No unbounded regions such as halfspaces.
- Original data structures based on Approximate Voronoi Diagrams (AVDs).

[AM00] Sunil Arya, David M. Mount. Approximate range searching, CGTA, 2000.

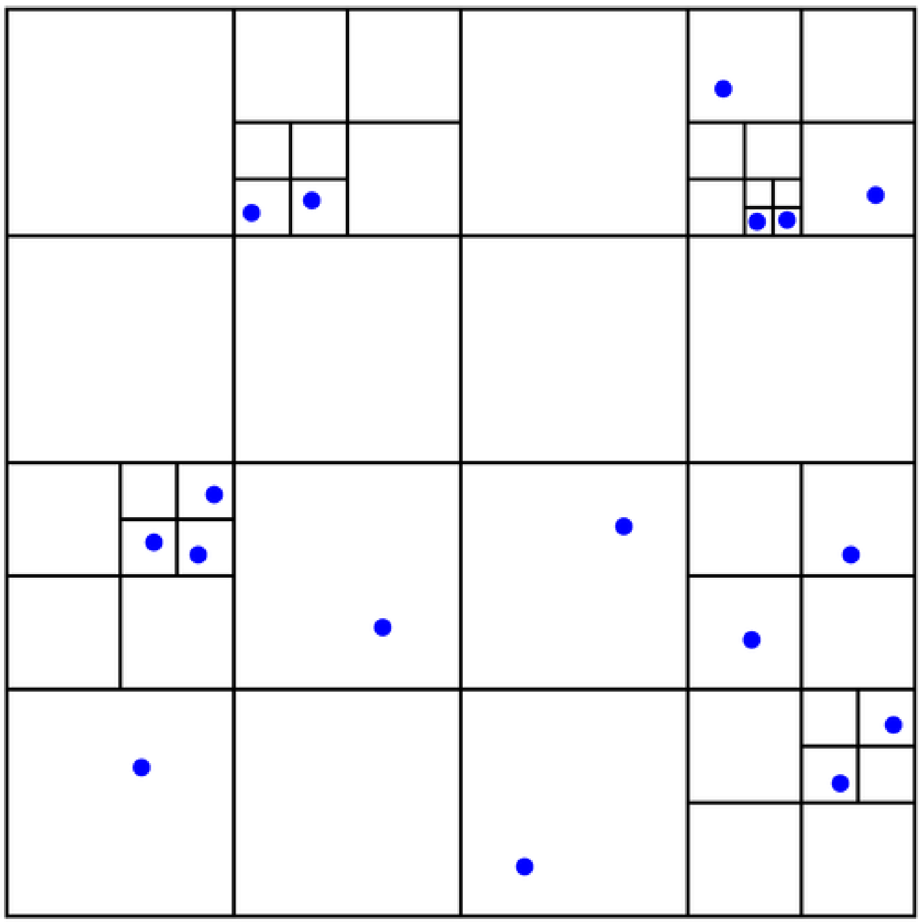
Absolute Model



- In the **absolute** model, points within distance ε from the range boundary may be counted or not. [Fo07]
- All points inside $[0,1]^d$.
- We use absolute model data structures to build our relative model data structures.

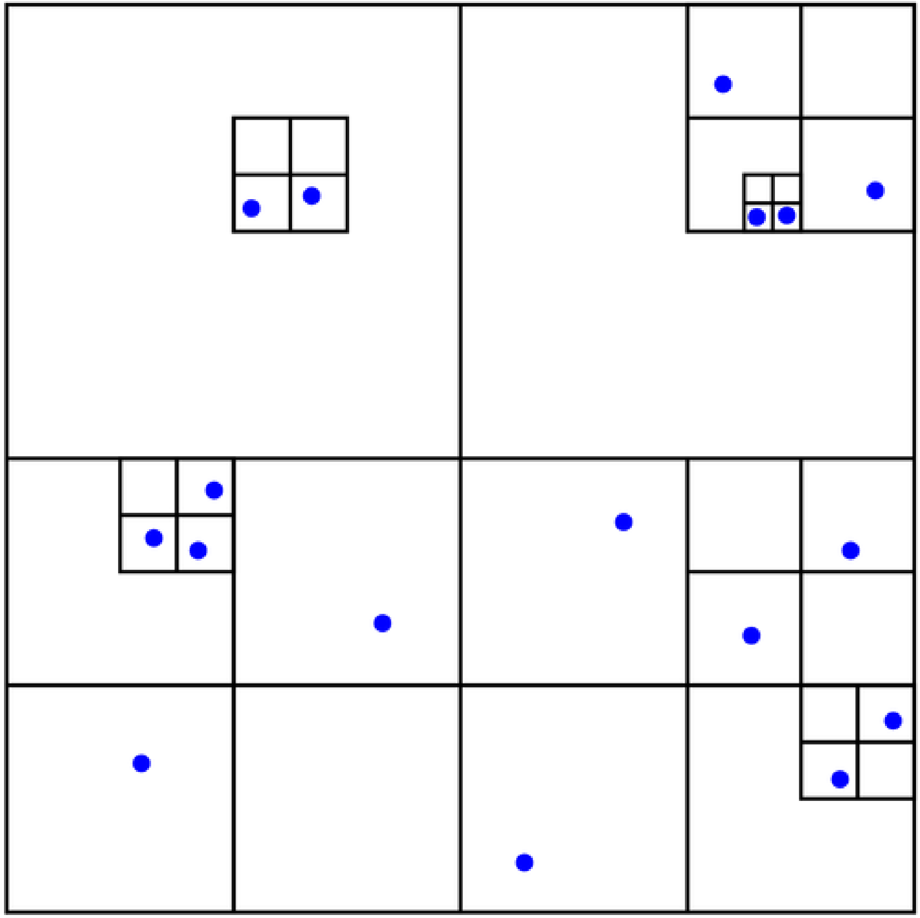
[Fo07] Guilherme D. da Fonseca, Approximate range searching: the absolute model, WADS, 2007

Quadtrees



- A quadtree is a recursive subdivision of the bounding box into 2^d equal boxes.
- Subdivisions are called quadtree boxes.
- We recursively subdivide boxes with more than 1 point.
- Problems: Size is unbounded in terms of n and ε. Also, height is $\Theta(n)$.

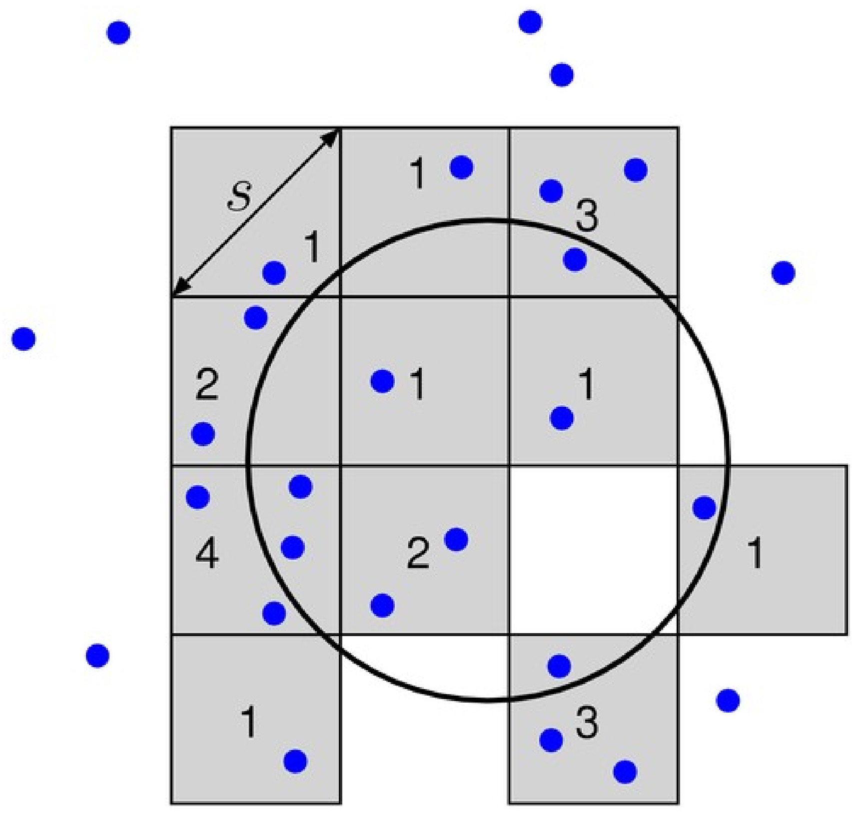
Compressed Quadtree



- Compression reduces storage to O(n), but height remains Θ(n).
- Pointers can be added to allow searching the quadtree in O(log n) time. [HP08]
- Preprocessing takes O(n log n) time. [HP08]

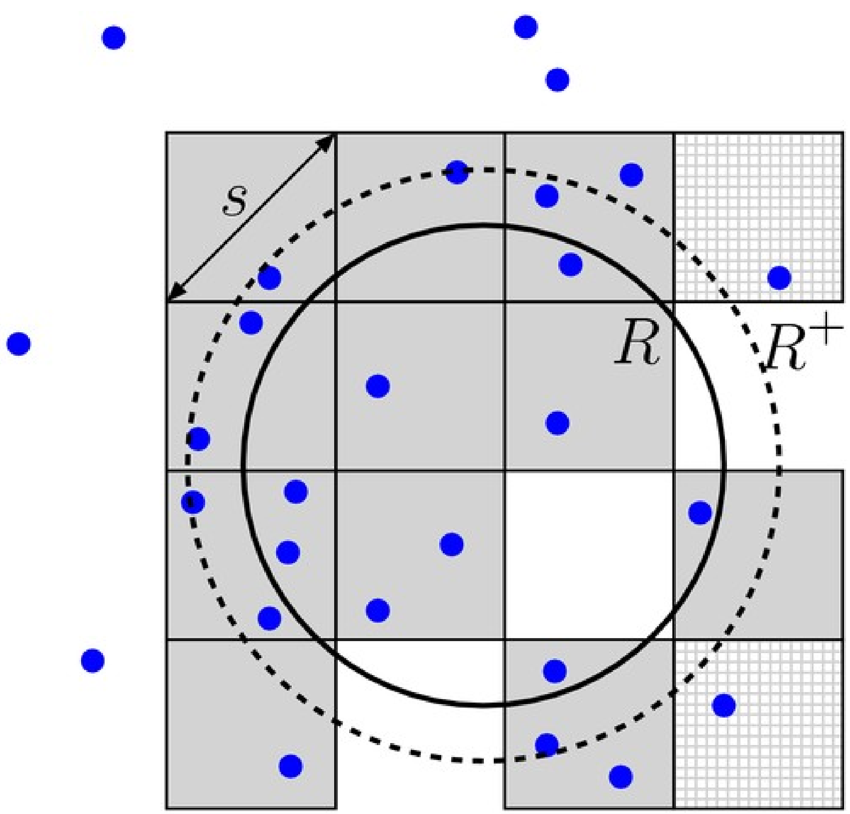
[HP08] Sariel Har-Peled. Geometric approximation algorithms, available online, 2008.

Range Sketching



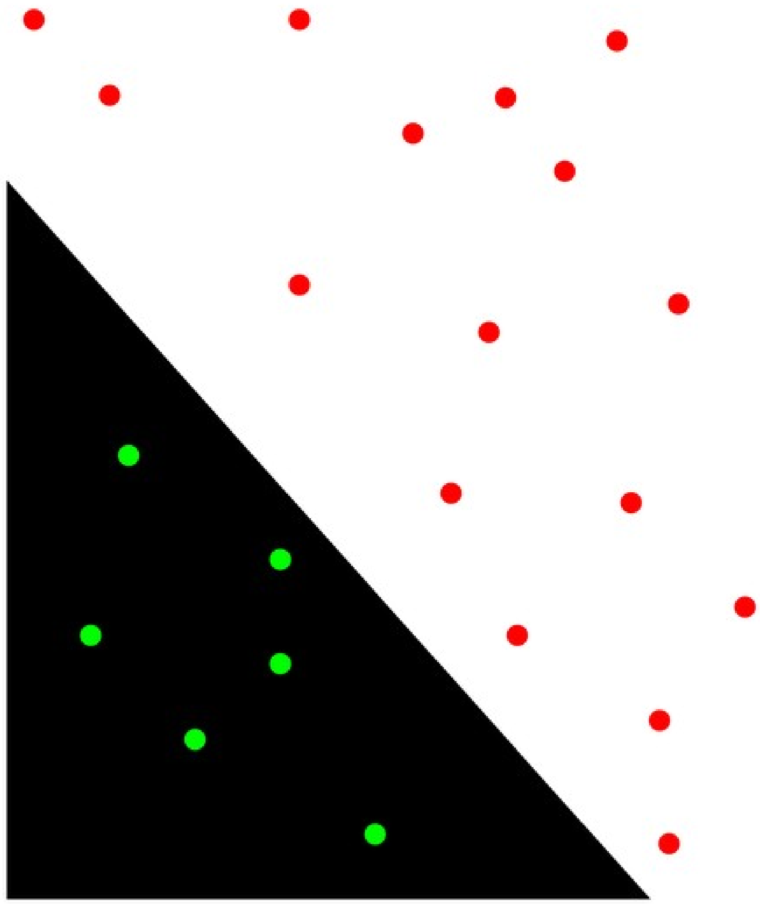
- Range counting: very limited information.
- Range reporting: very verbose information.
- Range sketching: offers a resolution tradeoff.
- Returns the counts of points inside each quadtree box of diameter s that intersect the query range.

Range Sketching



- Consider a slightly larger range R^+.
- Let k and k' respectively be the number of non-empty quadtree boxes of diameter s that intersect R and R^+.
- The compressed quadtree answers sketching queries in O(log n + k') time.
- The query result has size k.

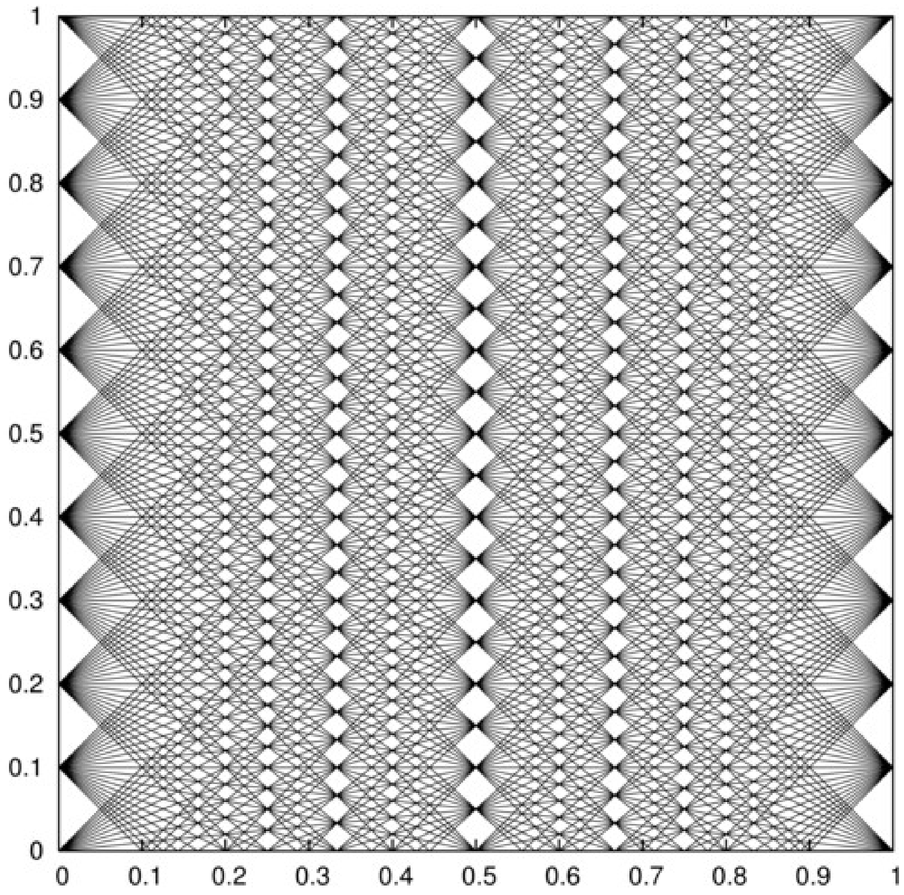
Halfspace Range Searching



- Ranges are d-dimensional halfspaces.
- Exact [Ma93]:
  - Query time: $O(n^{1-1/d})$.
  - Storage: $O(n)$.
- Absolute model [Fo07]:
  - Query time: O(1).
  - Storage: $O(1/\varepsilon^d)$.

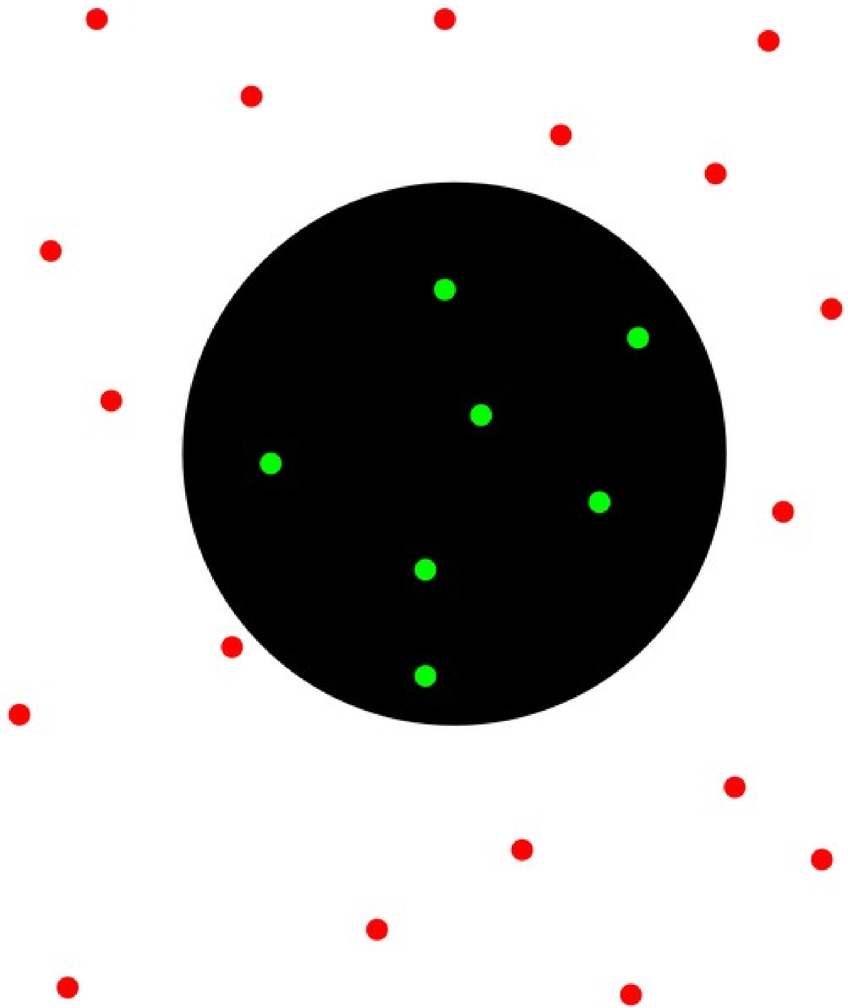
[Ma93] Jirí Matousek, Range searching with efficient hiearchical cutting, DCG, 1993.

Halfspace Data Structure



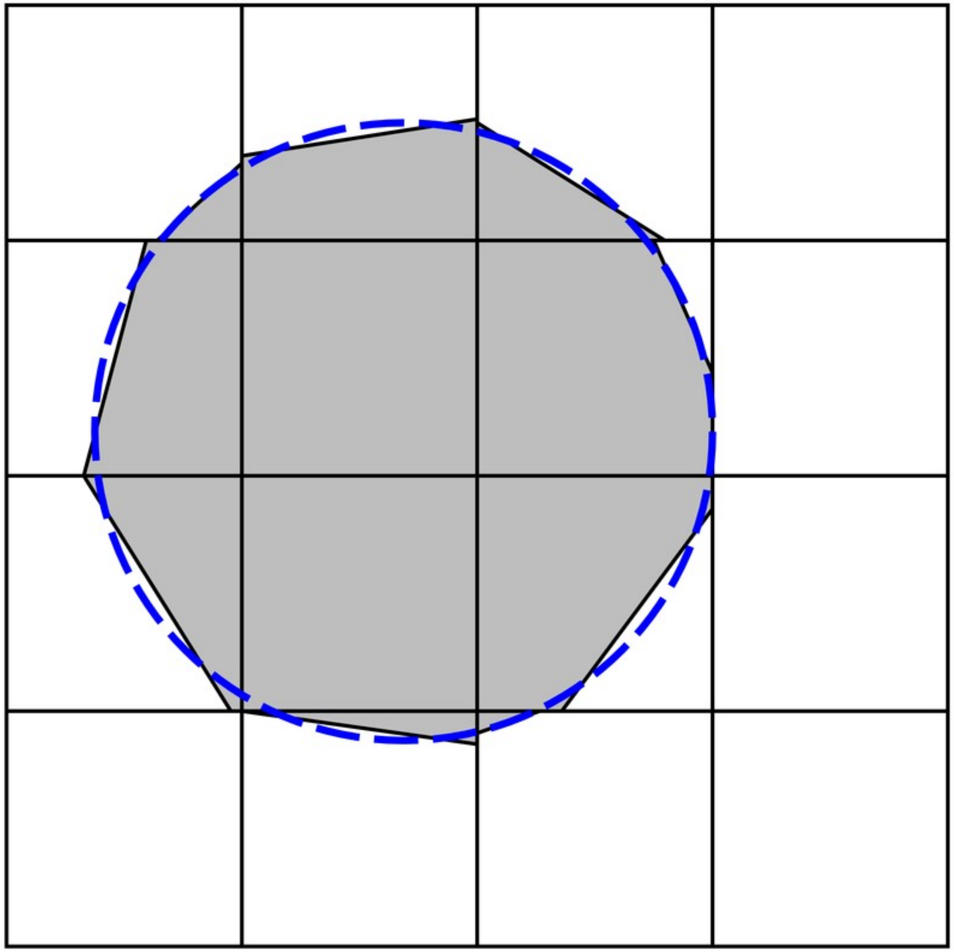
- We can ε-approximate every halfspace using $O(1/\varepsilon^d)$ halfspaces.
- Store query results in a table.
- Answer queries by rounding halfspace parameters and returning the corresponding value from the table.

Spherical Range Searching



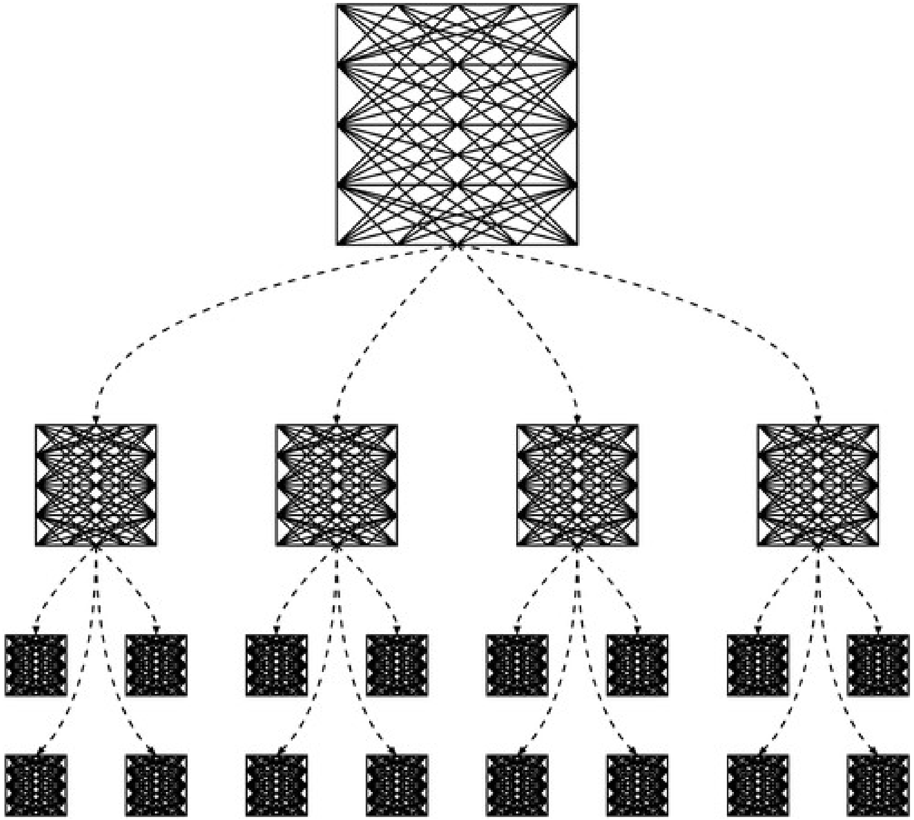
- Ranges are d-dimensional spheres.
- Exact version:
  - Project the points onto a (d+1)-dimensional paraboloid.
  - Use halfspace range searching.
- In the paper, we consider the more general *smooth ranges*.

Approximating Spheres with Halfboxes



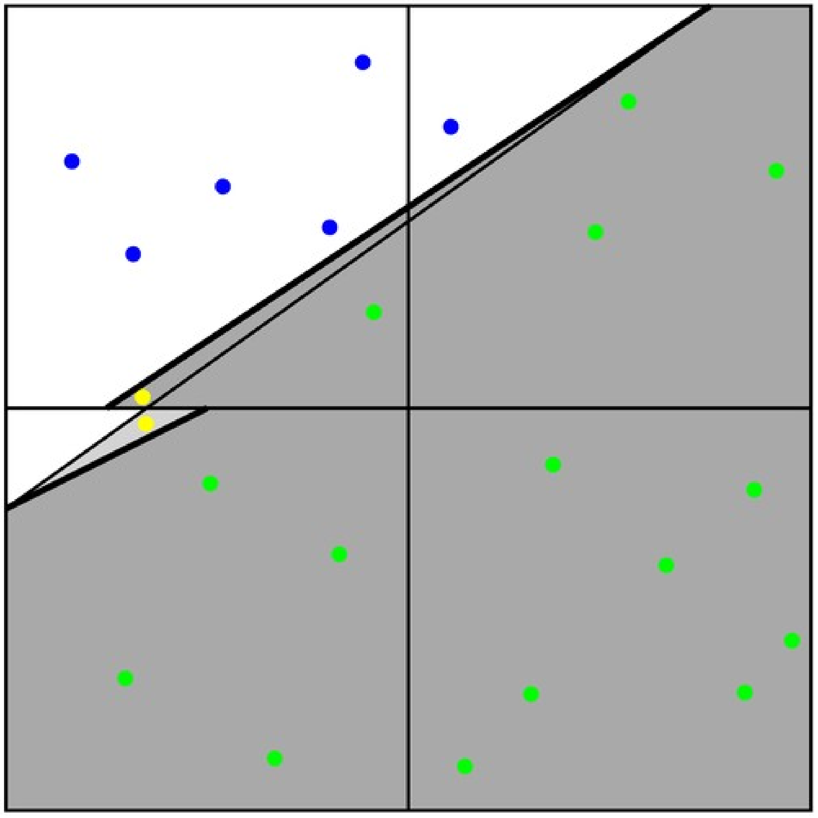
- A halfbox is the intersection of a quadtree box and a halfspace.
- We can ε-approximate a sphere with $O(1/\varepsilon^{(d-1)/2})$ halfboxes.
- We can associate halfspace data structures with quadtree nodes to obtain halfboxes.

Halfbox Quadtree



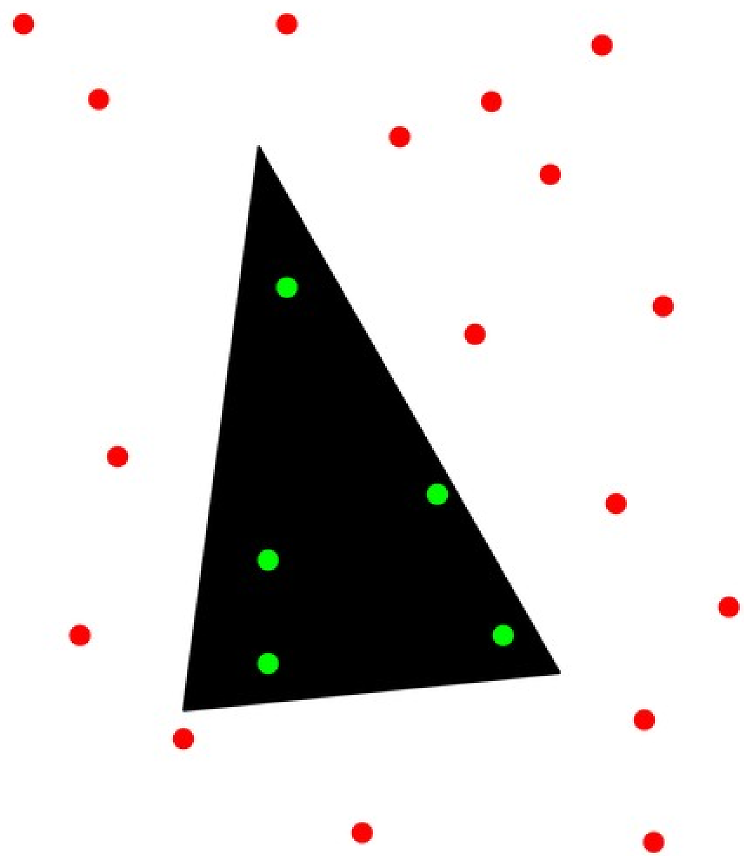
- Let γ between 1 and 1/√ε control the space-time tradeoff.
- Associate a (δ/γ)-approximate halfspace structure with each box of diameter δ.
- Storage: $O(n\gamma^d)$.
- Prepro.: $O(n\gamma^d + n \log n)$.
- Spherical queries: $O(\log n + 1/(\varepsilon\gamma)^{d-1})$.

Preprocessing



- Naive preprocessing takes $O(n^2 \gamma^d)$ time.
- Instead, we perform 2^d approximate queries among the children.
- Preprocessing takes contant time per unit of storage, after building the quadtree in $O(n \log n)$ time.
- Prepro.: $O(n\gamma^d + n \log n)$.

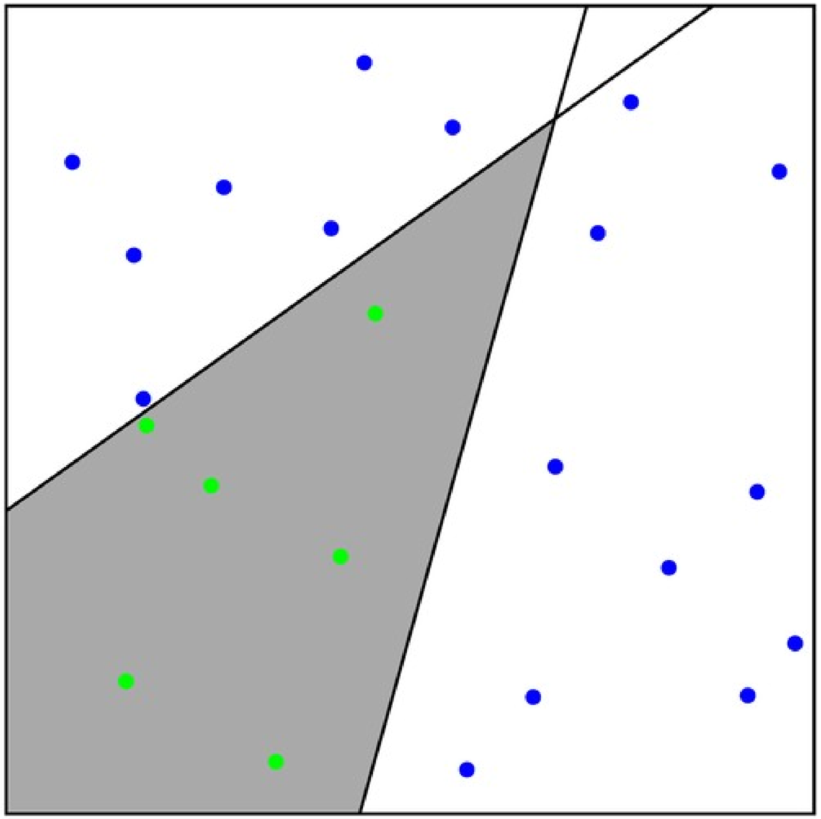
Simplex Range Searching



- Ranges are d-dimensional simplices: intersection of d+1 halfspaces.
- Exact version is similar to halfspaces: [Ma93]
  - Query time: O($n^{1\text{-}1/d}$).
  - Storage: O(n).
- Approximate version: use a multi-level variation of the halfbox quadtree.

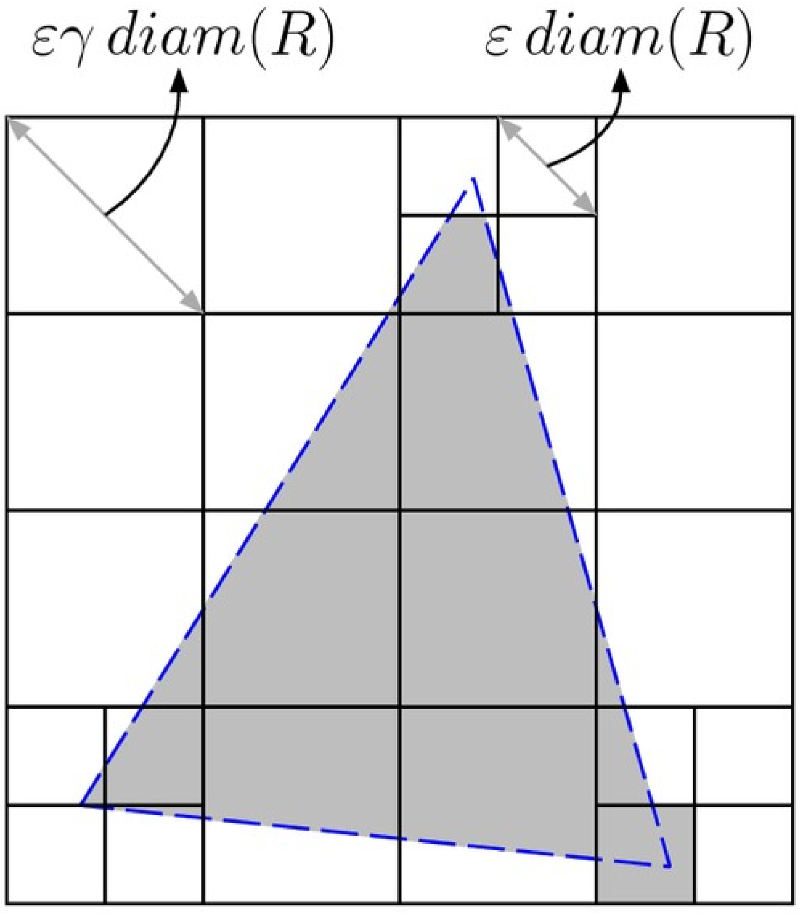
[Ma93] Jiří Matoušek, Range Searching with Efficient Hiearchical Cutting, DCG, 1993.

Multi-level Data Structure



- Let k be an integer parameter to control the space-time tradeoff.
- We build k levels of the halfspace data structure.
- Intersection of k hyperplanes can now be answered in O(1) time.
- Storage: O($n\gamma^{dk}$).
- Prepro.: O($n\gamma^{dk}$ + n log n).

Simplex Queries



Start querying with box v:

- Answer trivially if v is a leaf, or $v \cap R = \{\}$, or $diam(v)$ < ε $diam(R)$.
- If $diam(v)$ < εγ $diam(R)$ and v contains no (d-1-k)-face, then answer by subtracting all (d-k)-faces.
- Otherwise, answer recursively.

Simplex Range Searching Complexity

- Storage: $O(n\gamma^{dk})$.
- Preprocessing time: $O(n\gamma^{dk} + n \log n)$.
- Query time: $O(\log n + \log 1/\varepsilon + 1/(\varepsilon\gamma)^{d-1} + 1/\varepsilon^{d-1-k})$.

- Set $\gamma=1/\varepsilon^{k/(d-1)}$ to balance the last two terms.
- Storage: $O(n/\varepsilon^{k^2d/(d-1)})$.
- Preprocessing time: $O(n/\varepsilon^{k^2d/(d-1)} + n \log n)$.
- Query time: $O(\log n + \log 1/\varepsilon + 1/\varepsilon^{d-1-k})$.

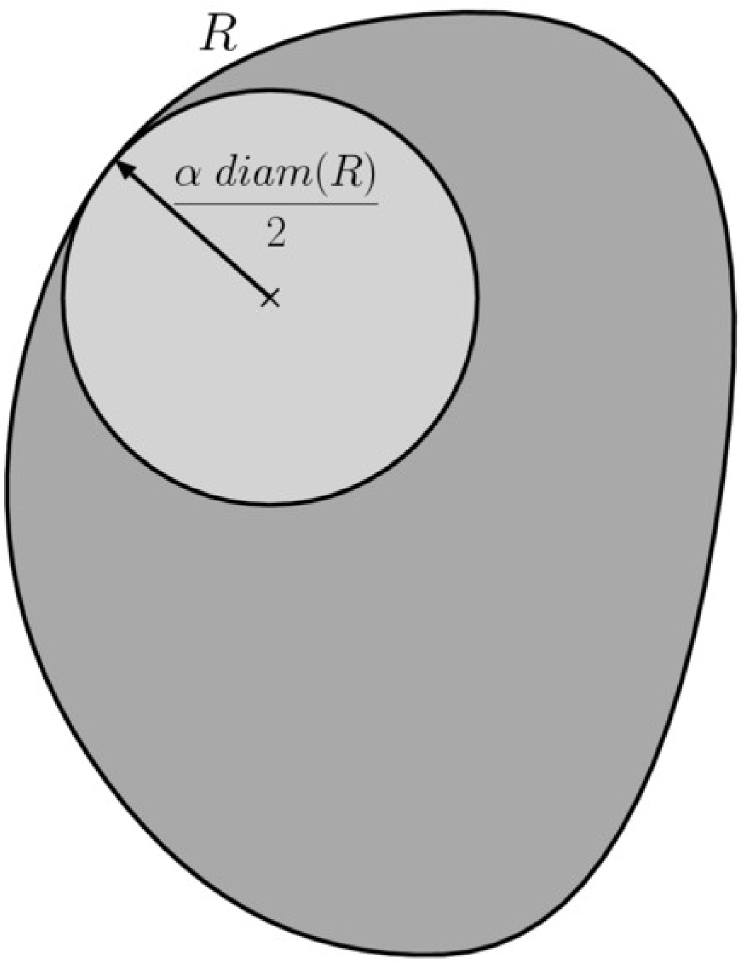
Future Research

SIBGRAPI 2009 will happen in Rio.

- More efficient data structures or tighter lower bounds? (Partially answered.)
- Data structures that benefit from idempotence? (Idempotent semigroup: $x + x = x$)
- Can we obtain simpler Approximate Voronoi Diagrams by extending these techniques?


Thanks!

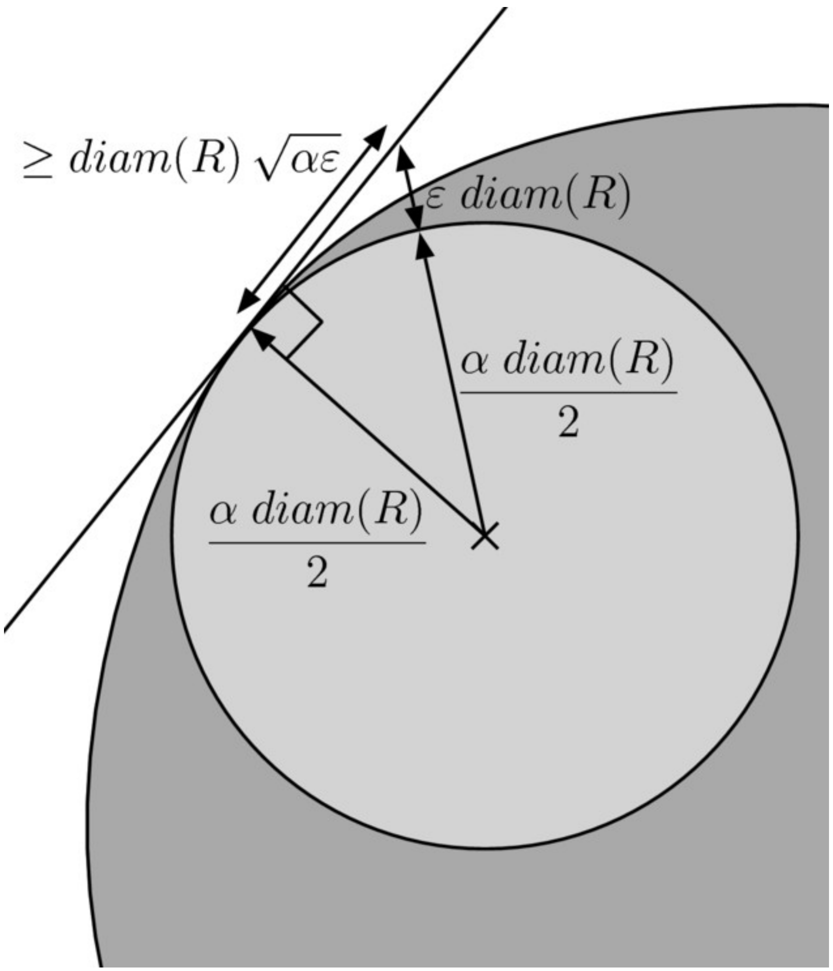
Smooth Region



- A convex region R is α-smooth if every point in the boundary of R is touched by a sphere of diameter α $diam(R)$ inside R.
- Spheres are 1-smooth.
- A region is smooth if it is α-smooth for constant α.

Smooth Range Searching

- Besides the unit-cost test assumption, we assume that a tangent hyperplane inside a quadtree box can be found in O(1) time.
- Use quadtree boxes of diameter at most $diam(R)\sqrt{\alpha\varepsilon}$ for the boundary.



Smooth Range Searching

- Besides the unit-cost test assumption, we assume that a tangent hyperplane inside a quadtree box can be found in O(1) time.
- Use quadtree boxes of diameter at most $diam(R)\sqrt{\alpha\varepsilon}$ for the boundary.
- Since each quadtree box of diameter δ contains a (δ/γ)-approximate data structure, use boxes of diameter at most $\varepsilon\gamma\, diam(R)$ for the boundary.
- By packing lemma, the number of boxes is $O(1/\varepsilon^{(d-1)/2} + 1/(\varepsilon\gamma)^{d-1})$.
- Query time: $O(\log n + 1/\varepsilon^{(d-1)/2} + 1/(\varepsilon\gamma)^{d-1})$.