Computing the Implicit Voronoi Diagram in Triple Precision

David L. Millman Jack Snoeyink

University Of North Carolina - Chapel Hill

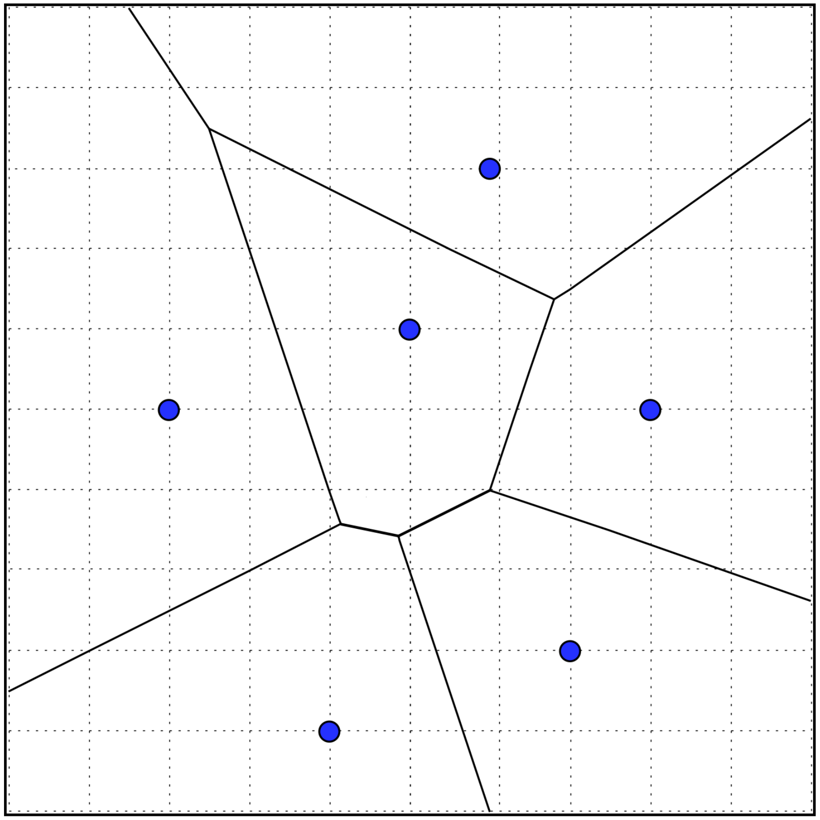
I will…

- define Voronoi, arithmetic degree, implicit Voronoi [LPT99]
- define a more structured *reduced-precision Voronoi diagram*
- describe its incremental construction
- present open problems created by this work

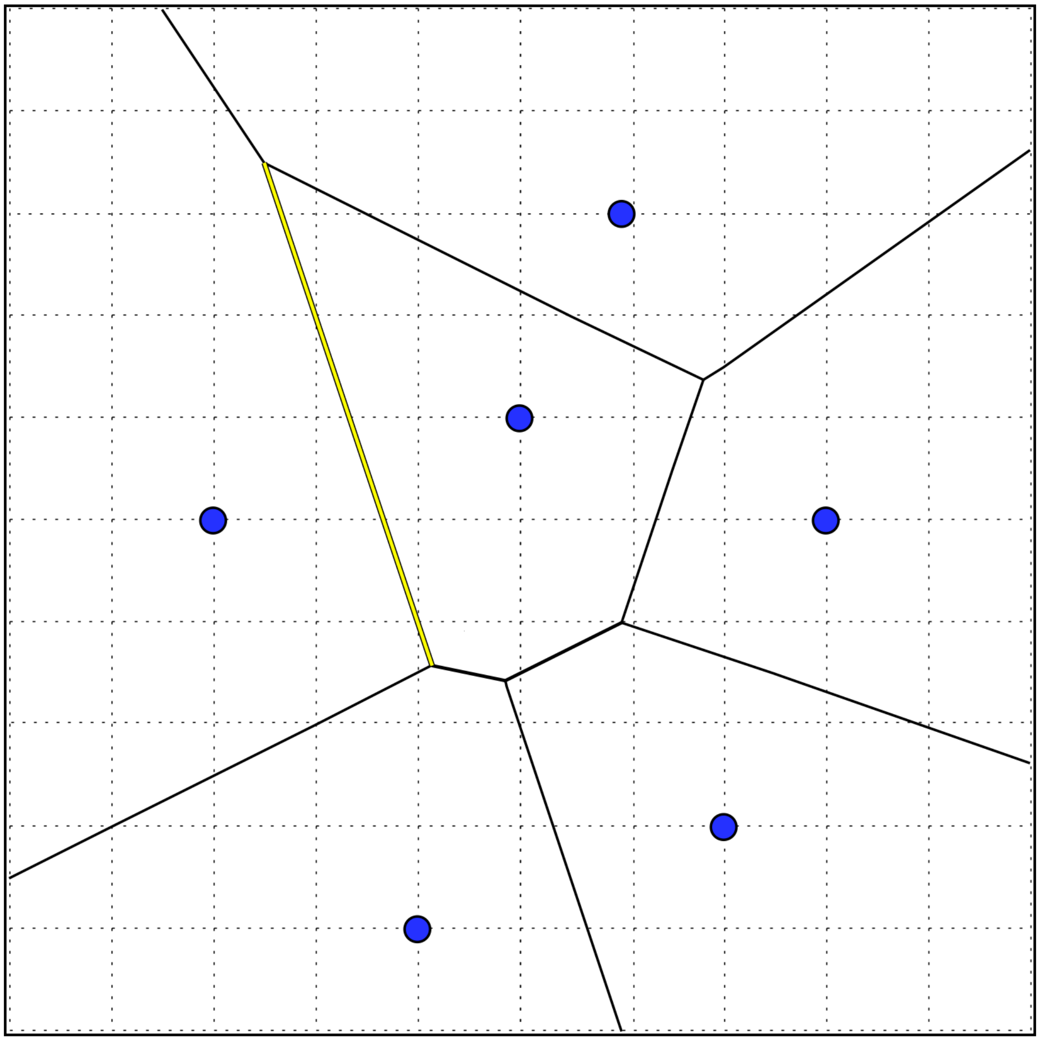
WADS09
August 23, 2009

Voronoi Diagram

Given a finite set of sites on a grid, the *Voronoi diagram* is the partition of the plane into maximally connected regions having the same set of closest sites



Voronoi Diagram



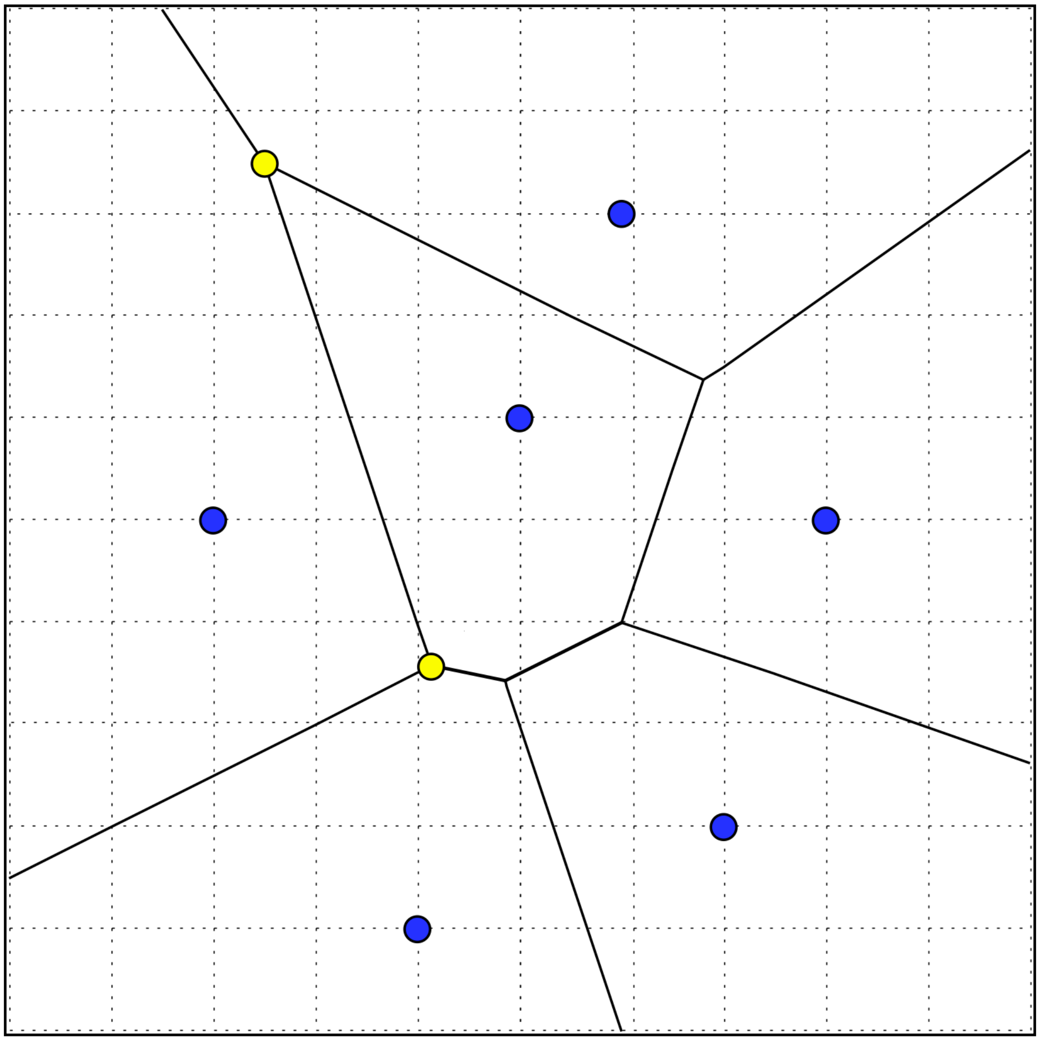
Voronoi edges

Voronoi vertices

Voronoi cell

Voronoi diagram

Voronoi Diagram



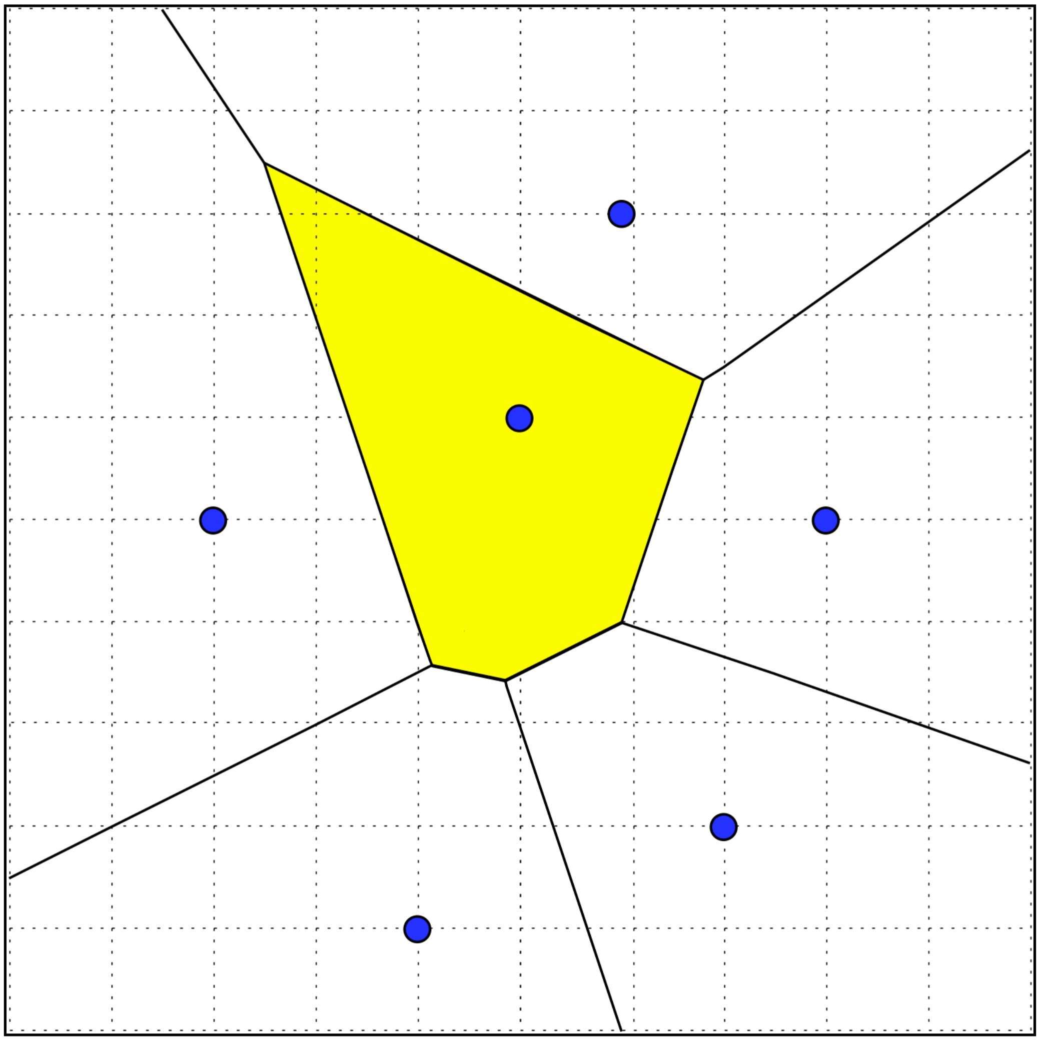
Voronoi edges

Voronoi vertices

Voronoi cell

Voronoi diagram

Voronoi Diagram



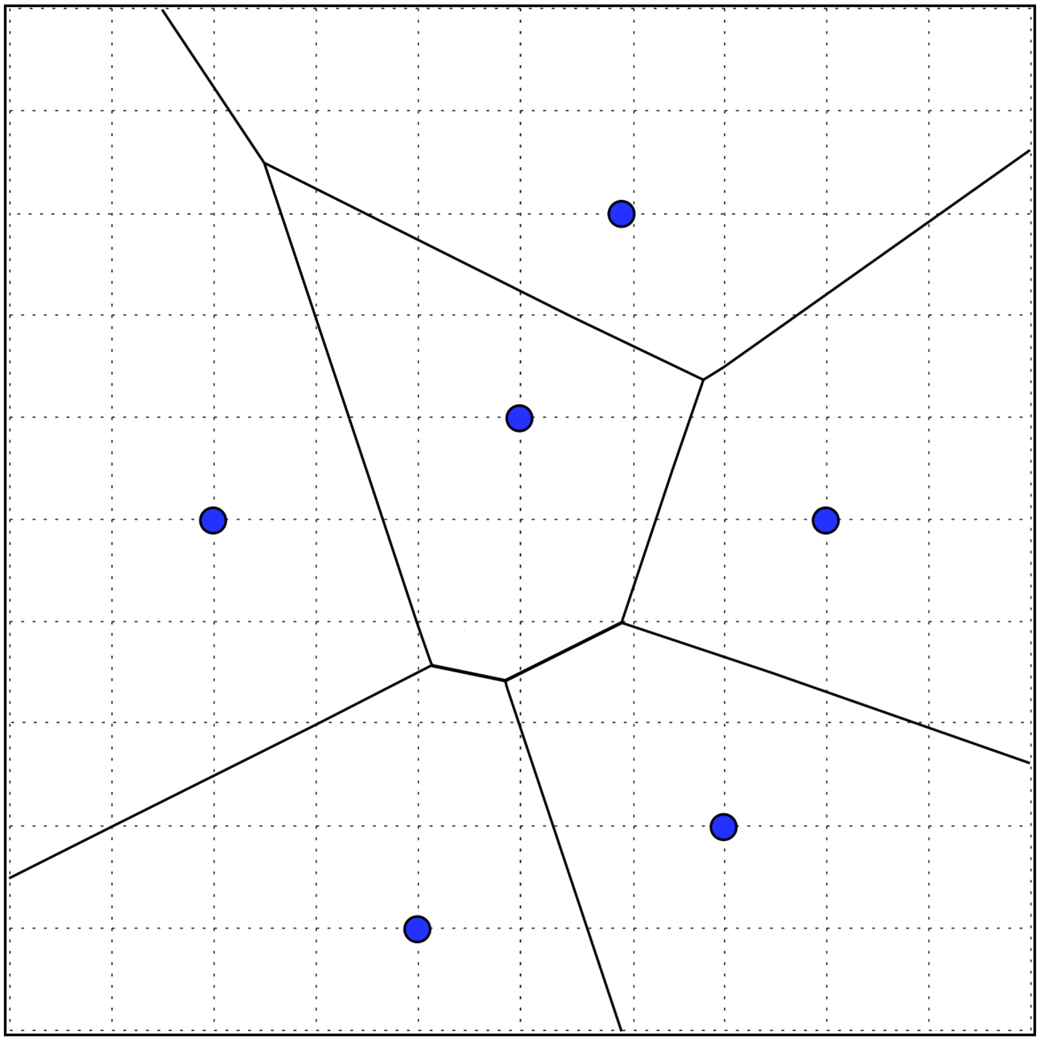
Voronoi edges

Voronoi vertices

Voronoi cell

Voronoi diagram

Voronoi Diagram



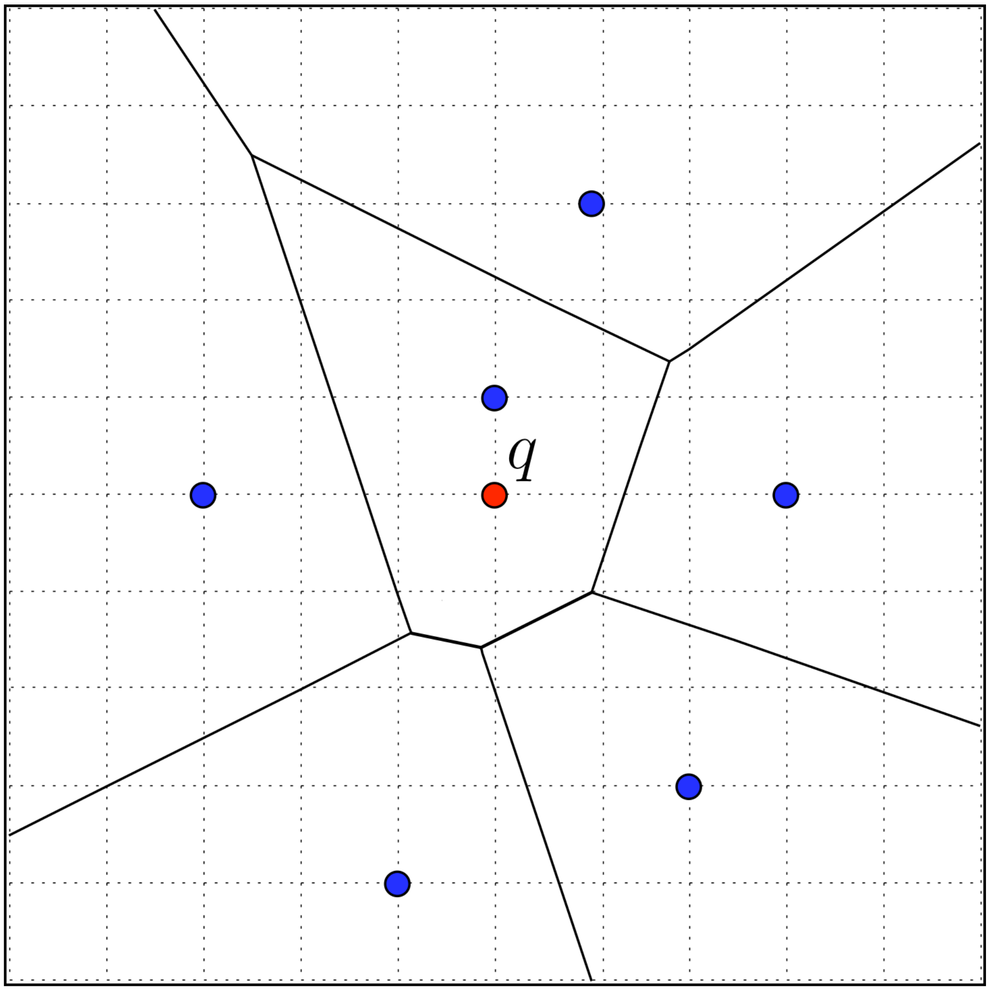
Voronoi edges

Voronoi vertices

Voronoi cell

Voronoi diagram

Point location returns nearest neighbor



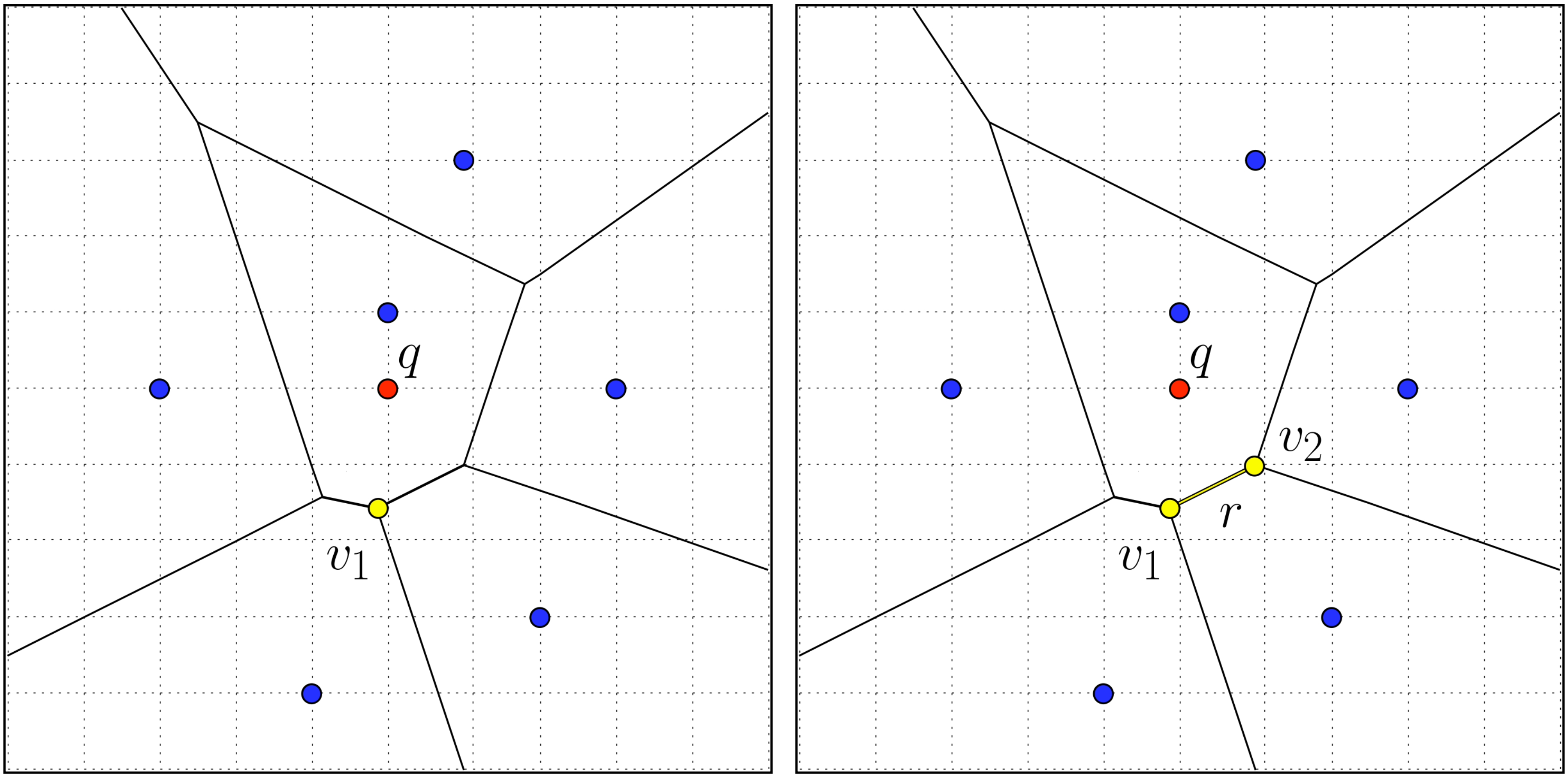
Given:

A set of sites S the Voronoi diagram of S & query point q

Determine:

the site of S closest to q

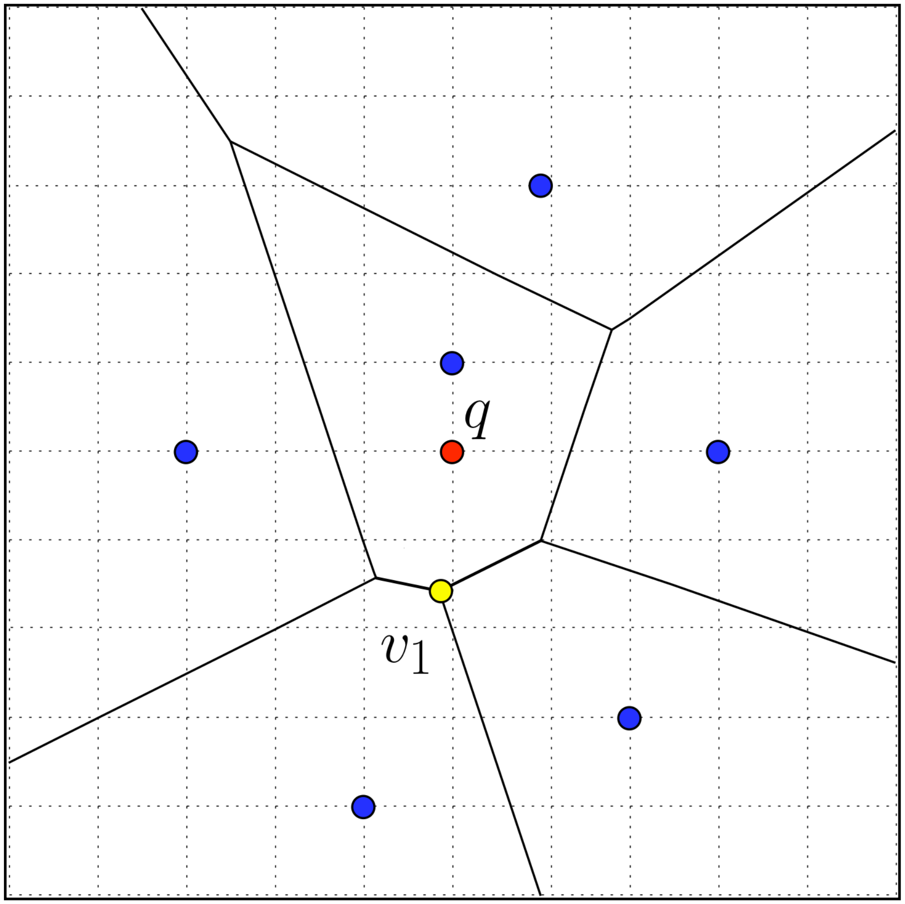
Predicates and their precision

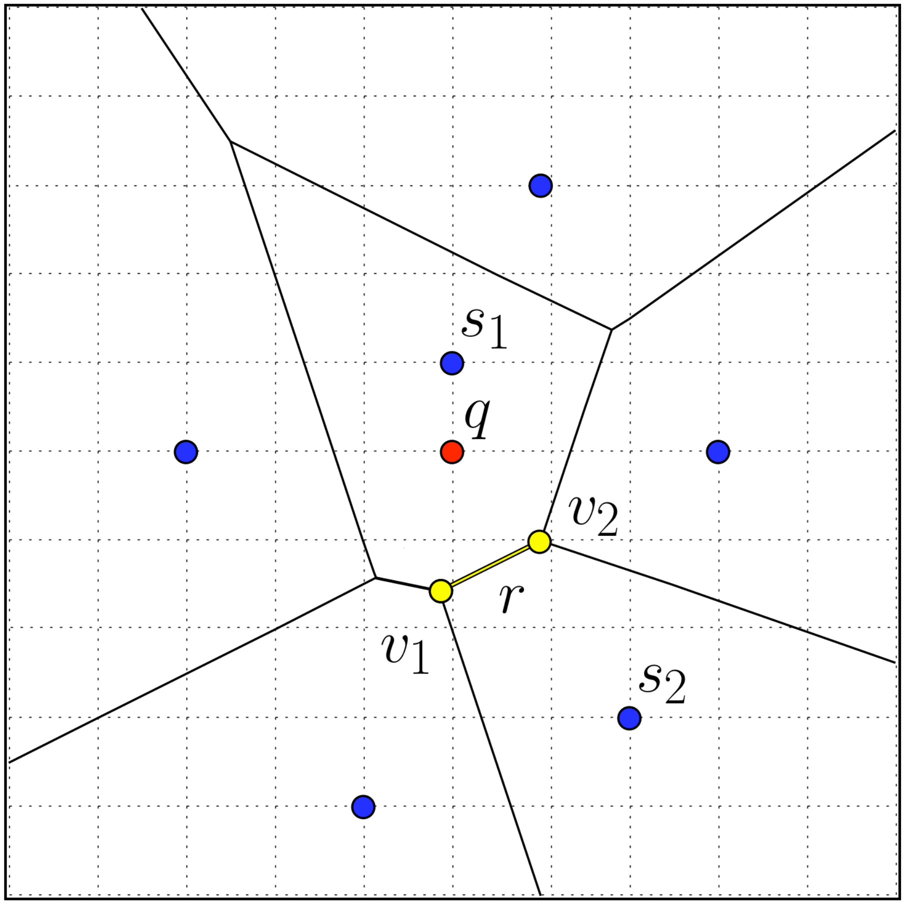


Is $q.x$ left/right of vertex $v_1.x$?
3× precision

Is q above/below segment r?
6× precision [LPT99]

Predicates and their precision



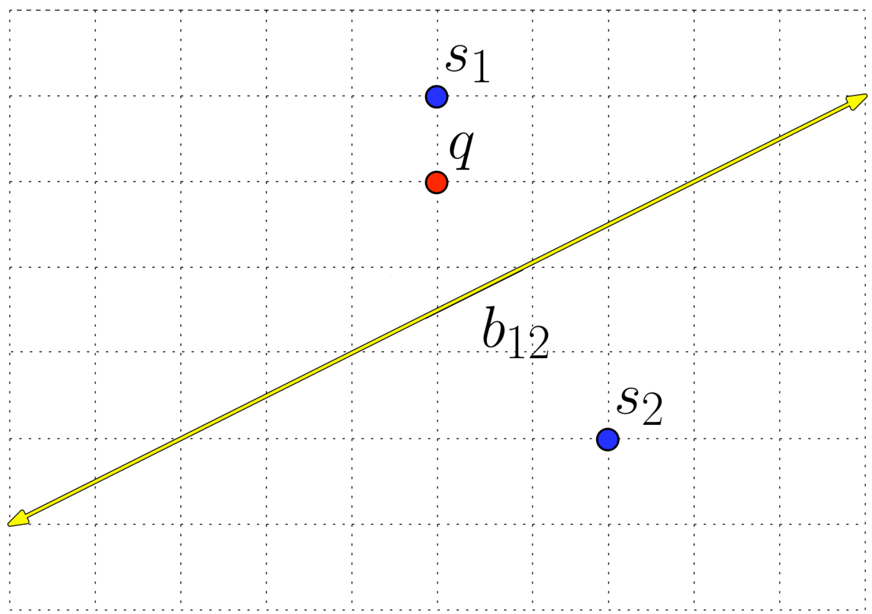


Is *q.x* left/right of vertex $v_1.x$?
3× precision

Is *q* above/below segment *r*?
6× precision [LPT99]

Arithmetic Degree – Side of Bisector

Is q closer to s_1?

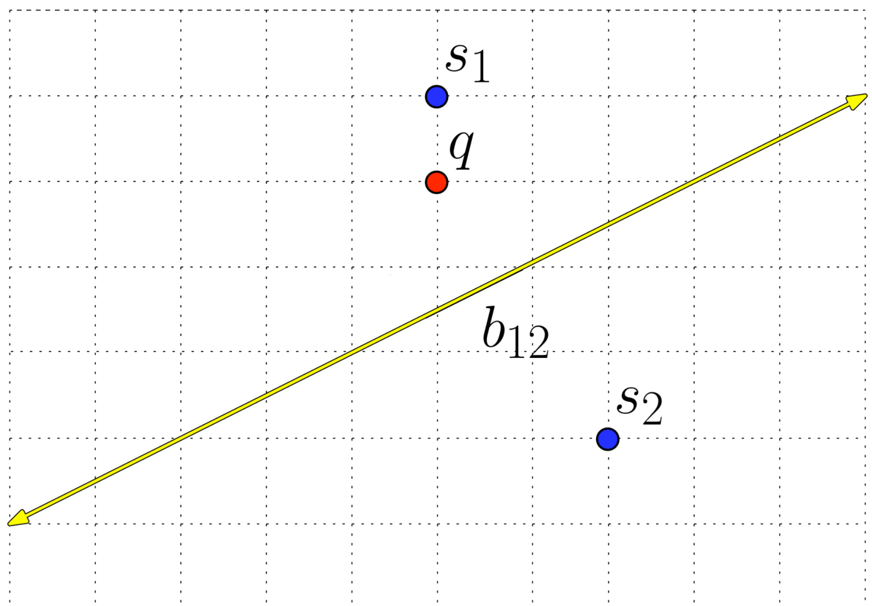


$$\begin{aligned} q &= (x_q, y_q) \\ s_1 &= (x_1, y_1) \\ s_2 &= (x_2, y_2) \end{aligned}$$

$$\|q - s_1\|_2 \lesseqgtr \|q - s_2\|_2$$

Arithmetic Degree – Side of Bisector

Is *q* closer to s_1?



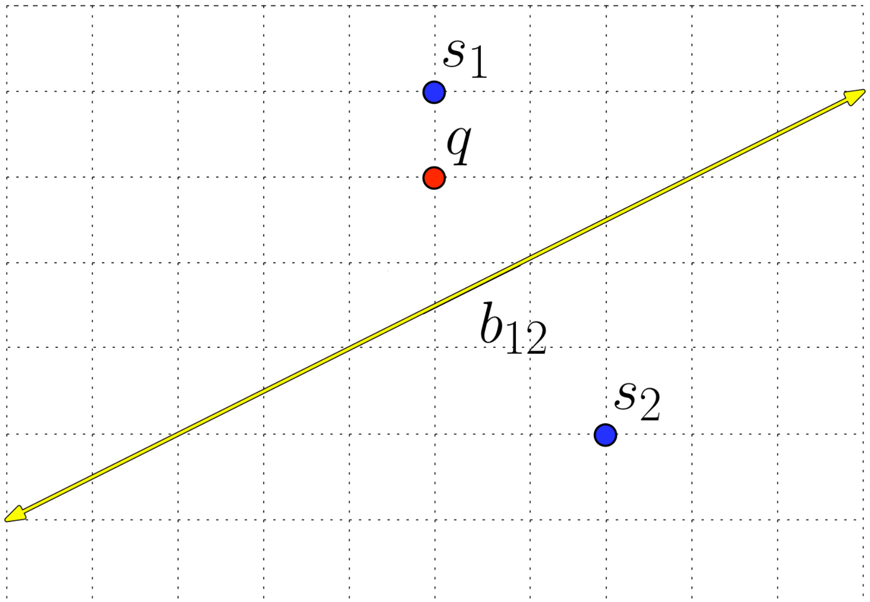
$$q = (x_q, y_q)$$
$$s_1 = (x_1, y_1)$$
$$s_2 = (x_2, y_2)$$

$$\|q - s_1\|_2 \lesseqgtr \|q - s_2\|_2$$

$$(x_q - x_1)^2 + (y_q - y_1)^2 \lesseqgtr (x_q - x_2)^2 + (x_q - y_2)^2$$

Arithmetic Degree – Side of Bisector

Is q closer to s_1?



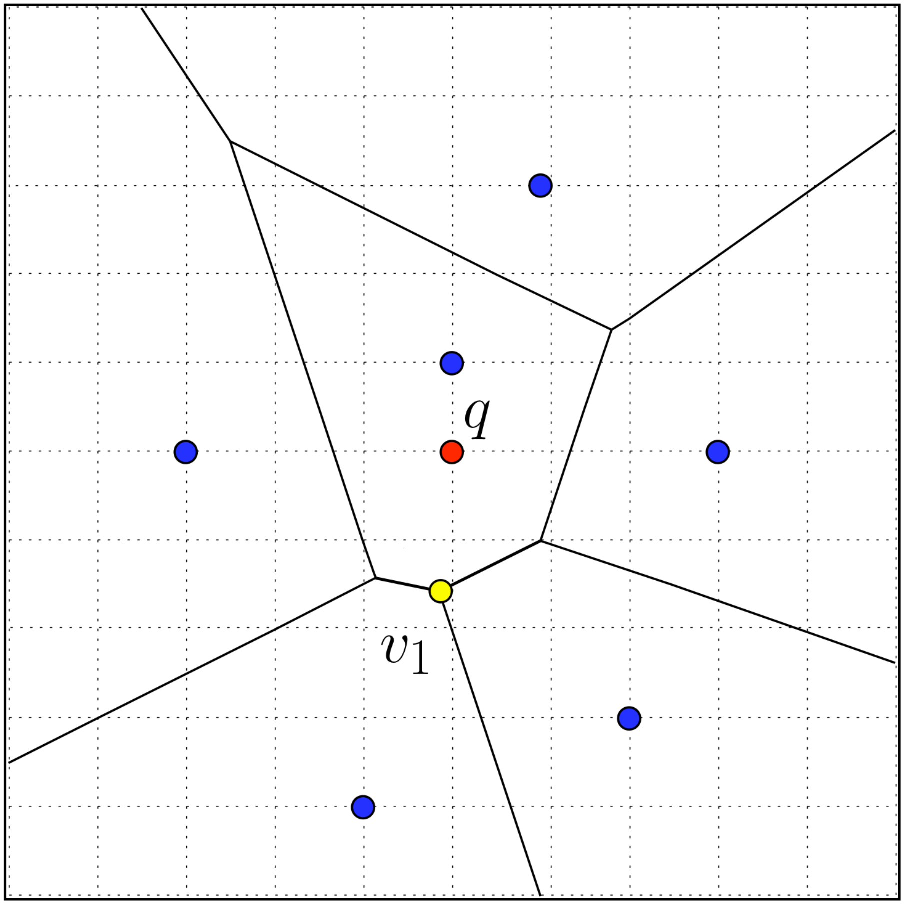
$$
\begin{aligned}
q &= (x_q, y_q) \\
s_1 &= (x_1, y_1) \\
s_2 &= (x_2, y_2)
\end{aligned}
$$

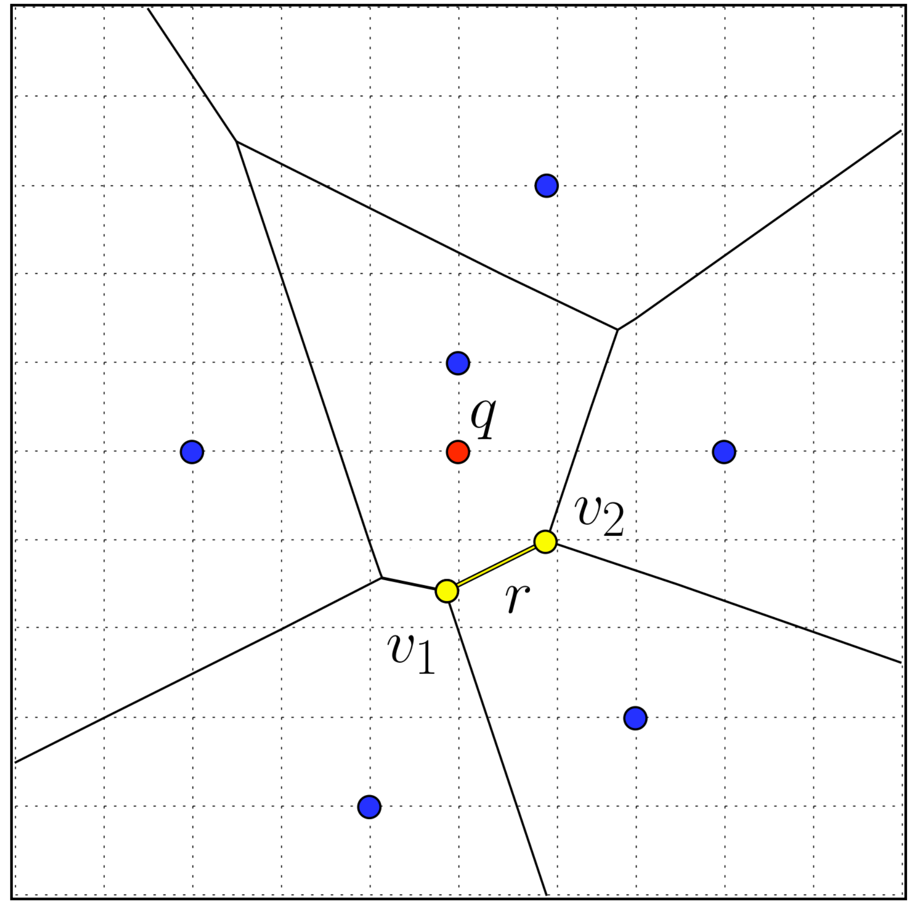
$$\|q - s_1\|_2 \lesseqgtr \|q - s_2\|_2$$

$$(x_q - x_1)^2 + (y_q - y_1)^2 \lesseqgtr (x_q - x_2)^2 + (x_q - y_2)^2$$

Degree 2

Predicates and their precision

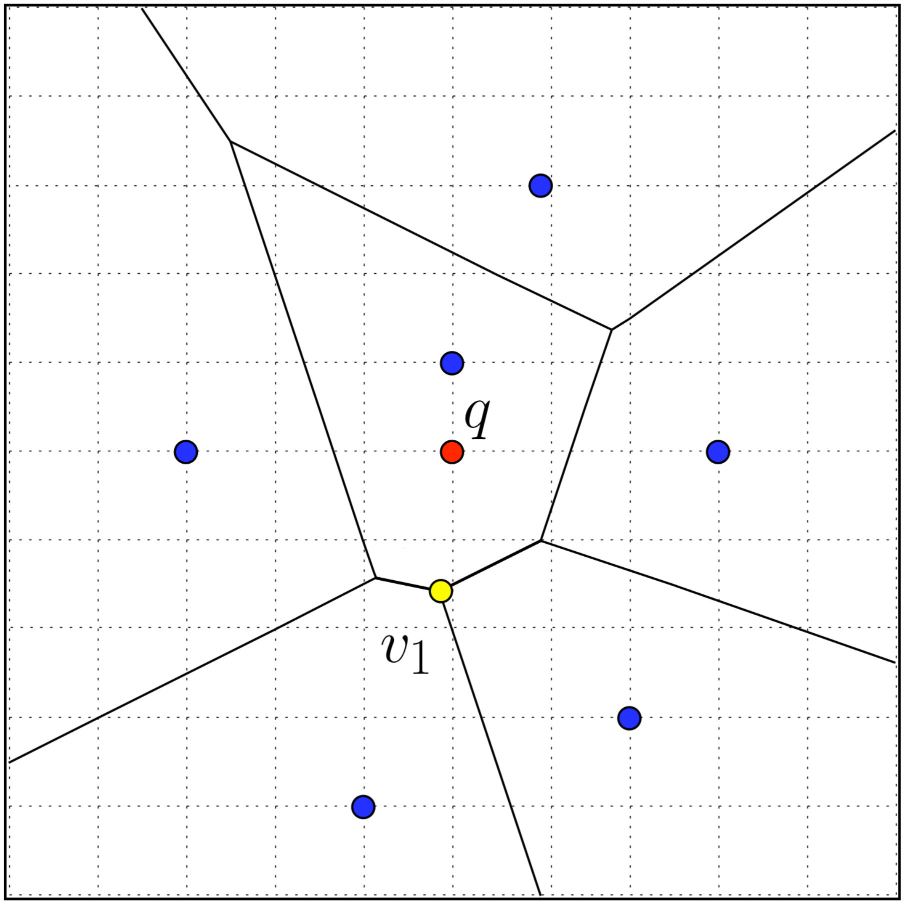


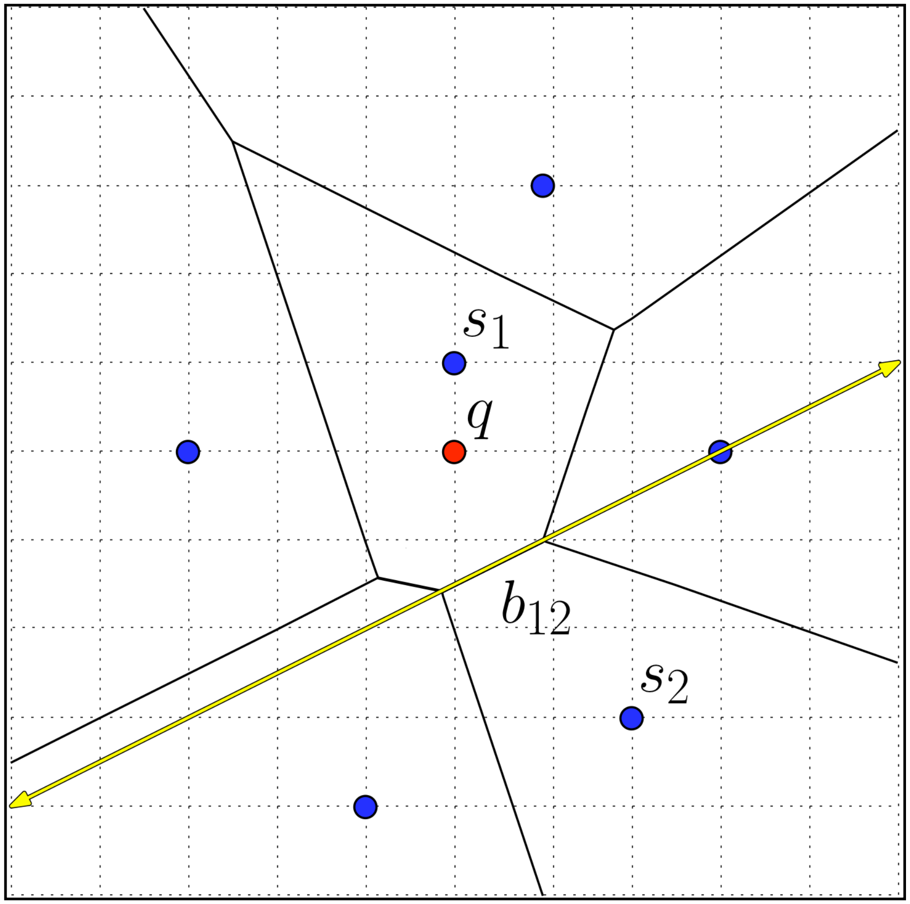


Is *q.x* left/right of vertex $v_1.x$?
degree 3

Is *q* above/below segment *r*?
degree 6

Predicates and their precision

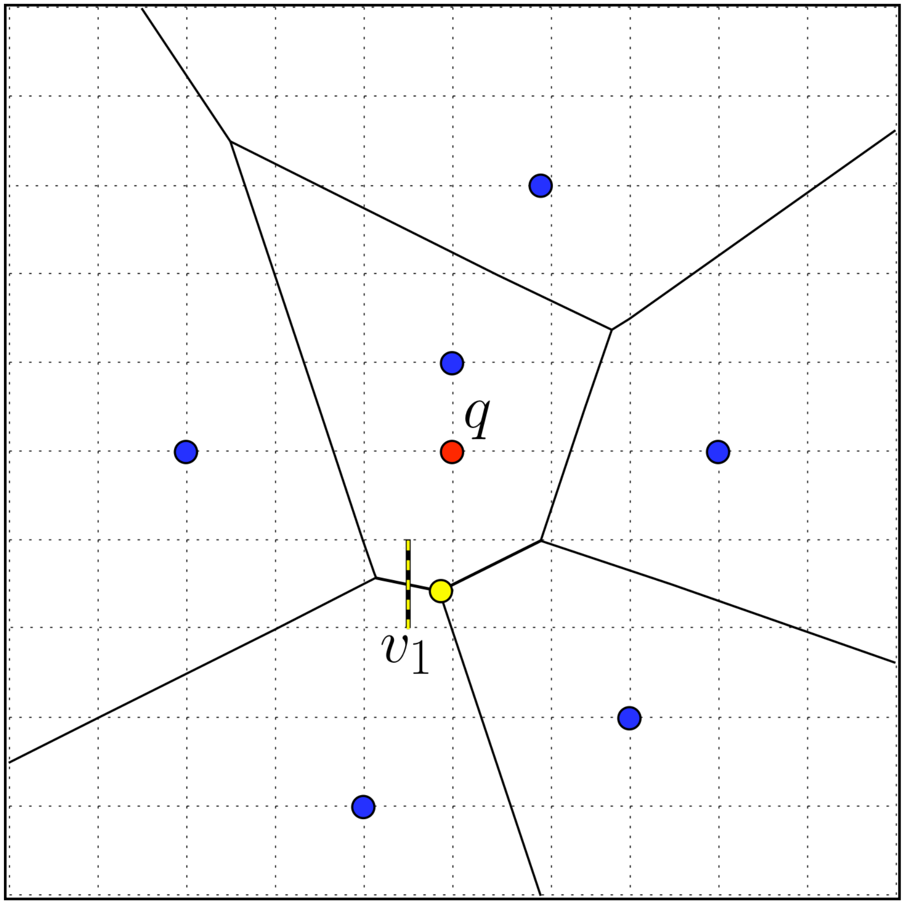


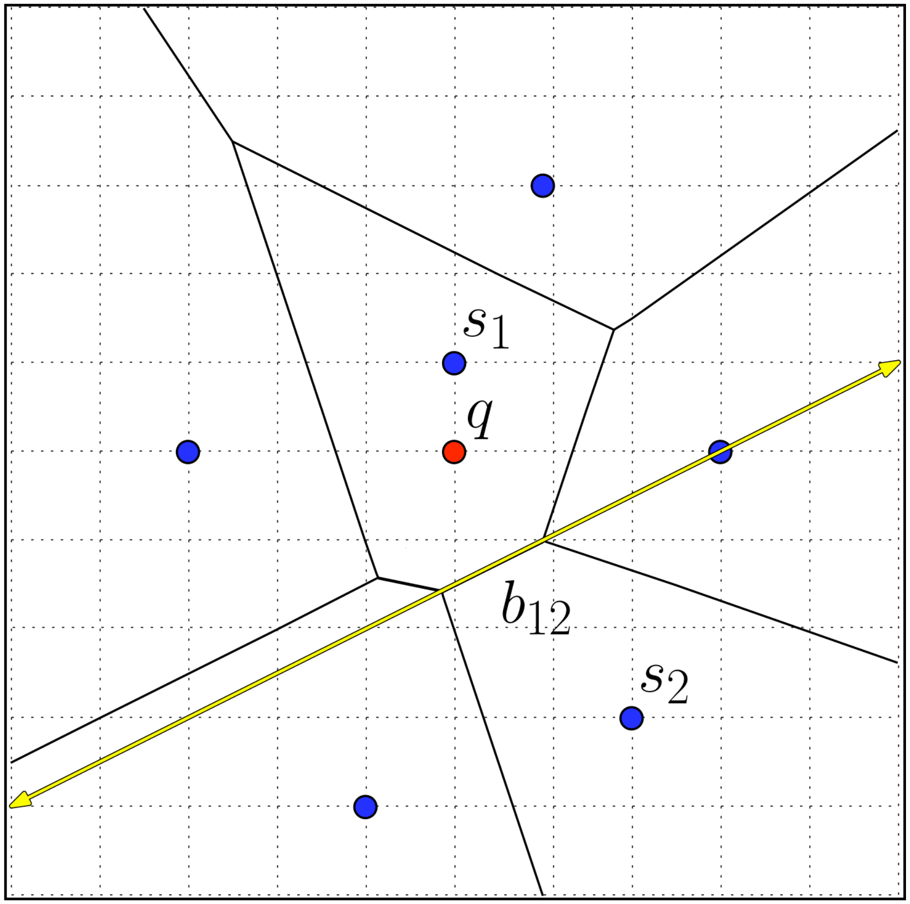


Is *q.x* left/right of vertex $v_1.x$?
degree 3

Is *q* closer to s_1 or s_2?
degree 2

Predicates and their precision

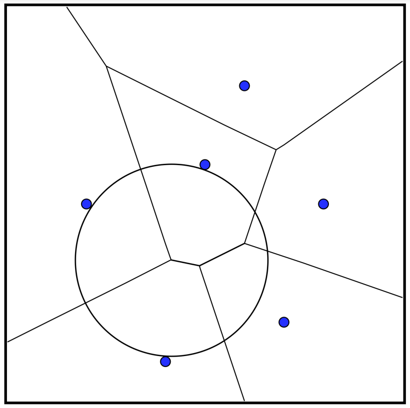




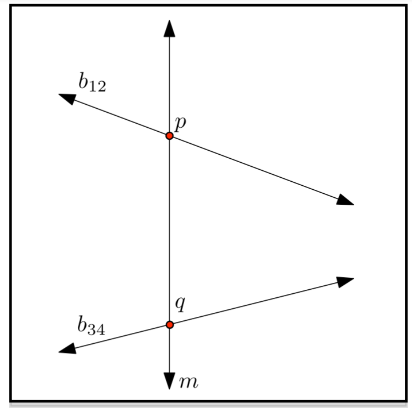
Is $q.x$ left/right of the grid cell containing v_1? [LPT99]
degree 1

Is q closer to s_1 or s_2?
degree 2

Preds, Ops, Constructions and their precision

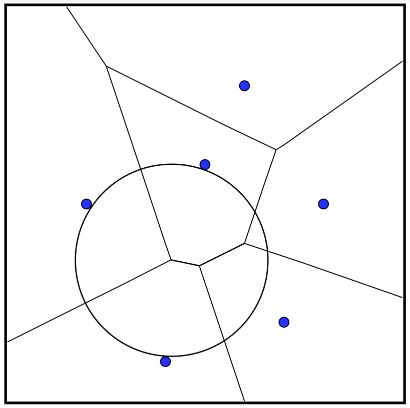


InCircle
degree 4

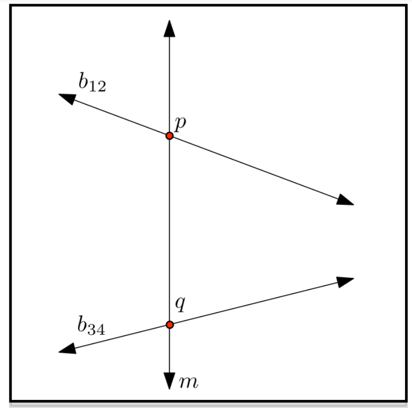


Vertical order
degree 3

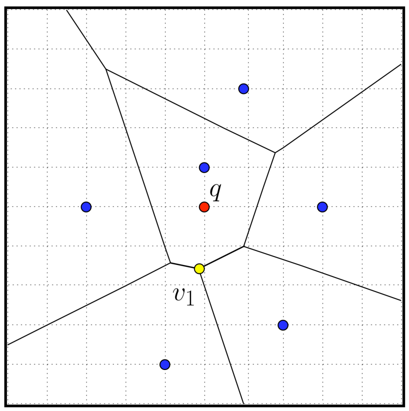
Preds, Ops, Constructions and their precision



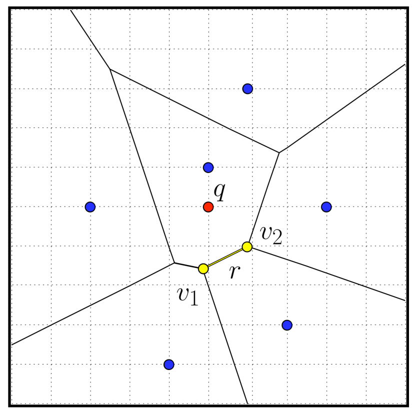
InCircle
degree 4



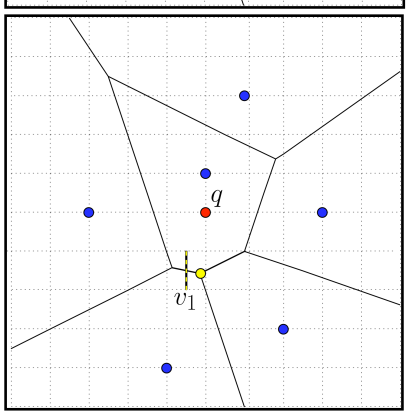
Vertical order
degree 3



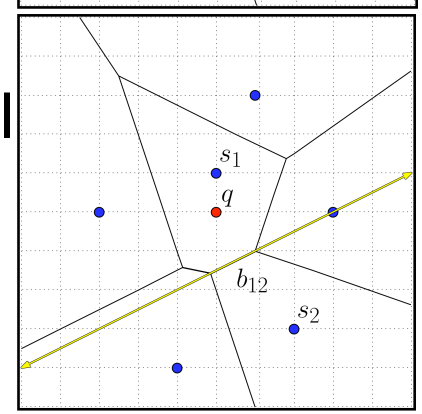
Left/right vertex
degree 3



Above/below seg
degree 6

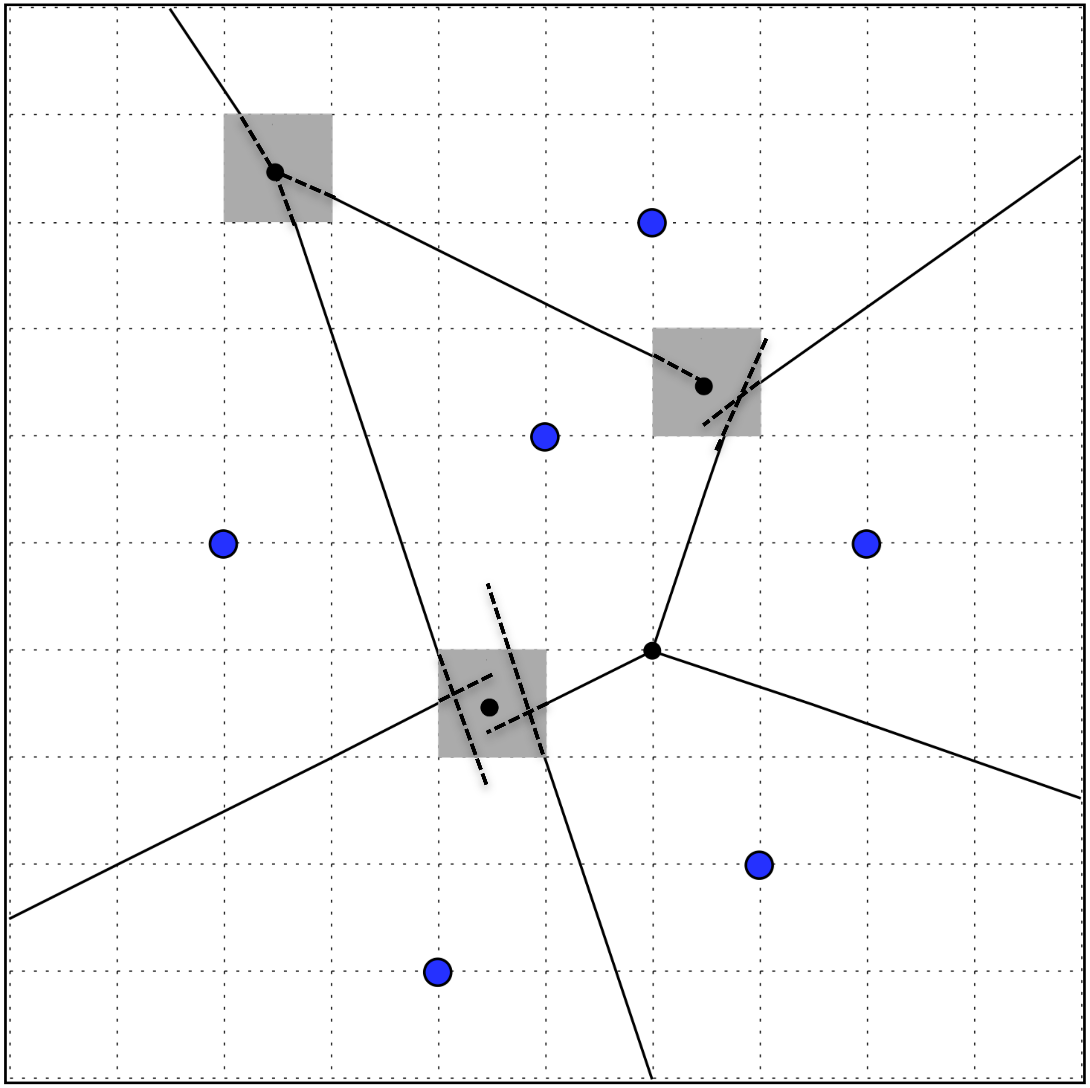


Left/right grid cell
[LPT99]
degree 1

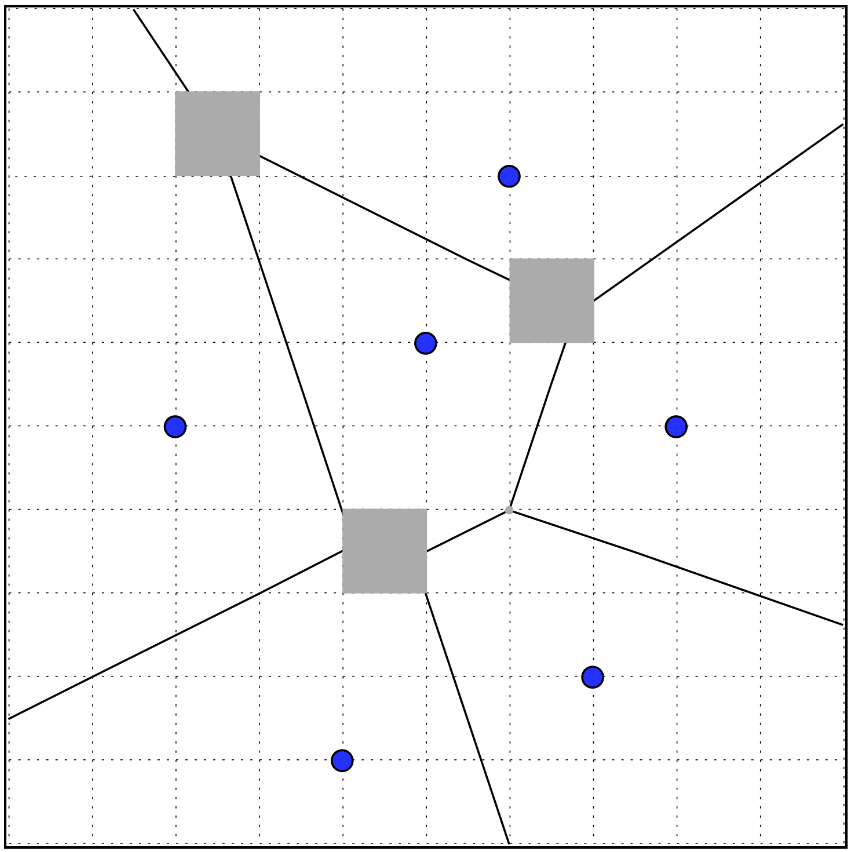


Side of a bisector
degree 2

Implicit Voronoi diagram [LPT99]



Proximity queries w/ min arith precision?



Given:

sites $S = \{s_1, s_2, \ldots, s_n\}$
w/ *b*-bit integer coords

Construct:

Implicit Voronoi with
minimum precision.

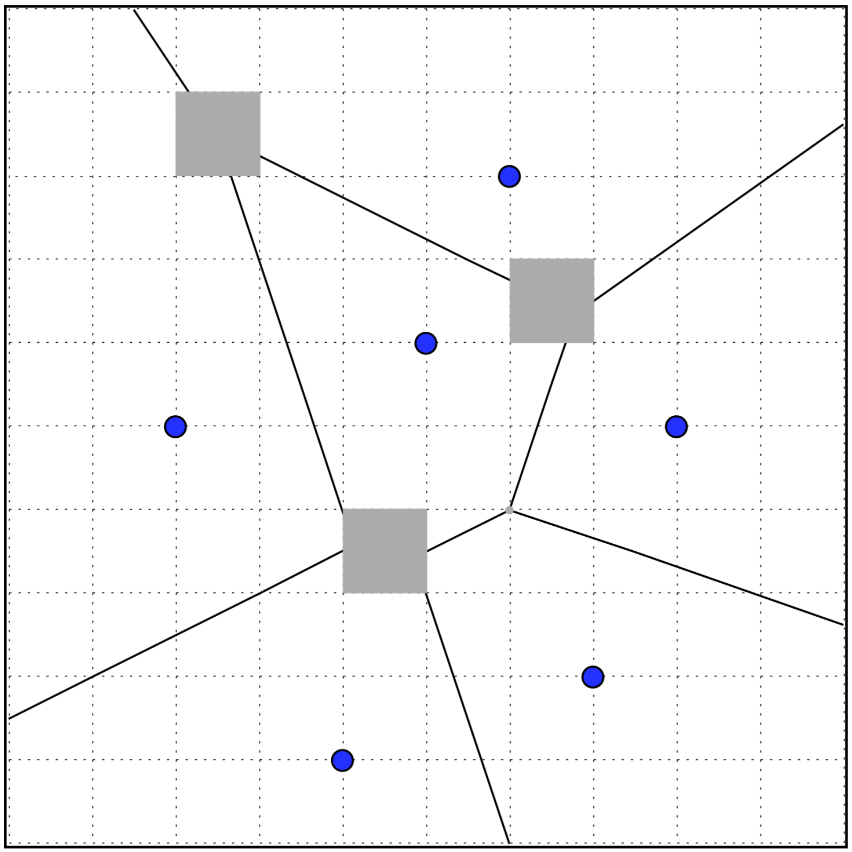
Note: precision < 4*b* bits
precludes computing the
Voronoi Diagram…

Previous work on handling numerics in CG

Implementing geometric algorithms in finite precision computer arithmetic:

- Rely on machine precision (+epsilon)
- Exact Geometric Computation [Y97]
- Arithmetic Filters [FW93][DP99]
- Adaptive Predicates [P92][S97]
- Topological Consistency [SI92]
- Restricted precision algorithm design [LPT99]

Proximity queries w/ min arith precision?



Given:

sites $S = \{s_1, s_2, \ldots, s_n\}$
w/ b-bit integer coords

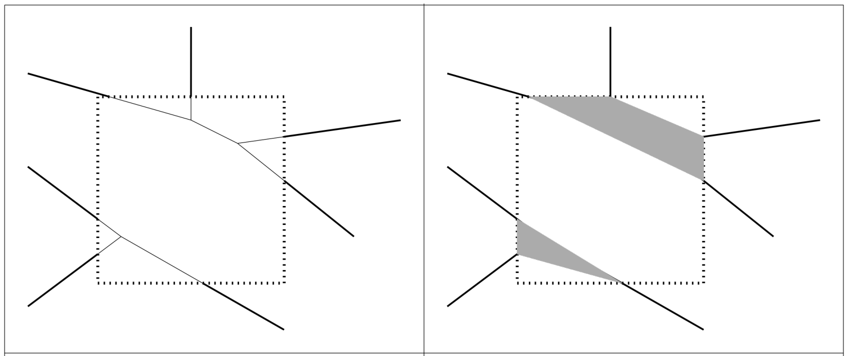
Construct:

Implicit Voronoi with minimum precision.

Note: precision < $4b$ bits precludes computing the Voronoi Diagram…

Reduced-precision Voronoi diagram

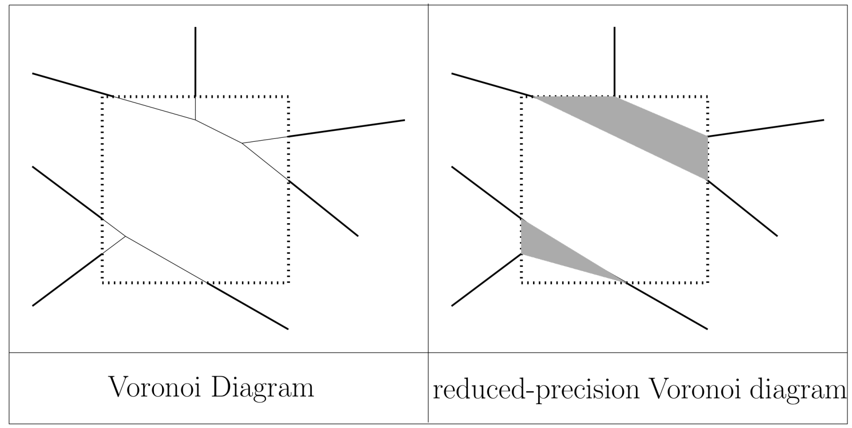
Replace connected subtrees of Vor edges inside a cell with their convex hulls.

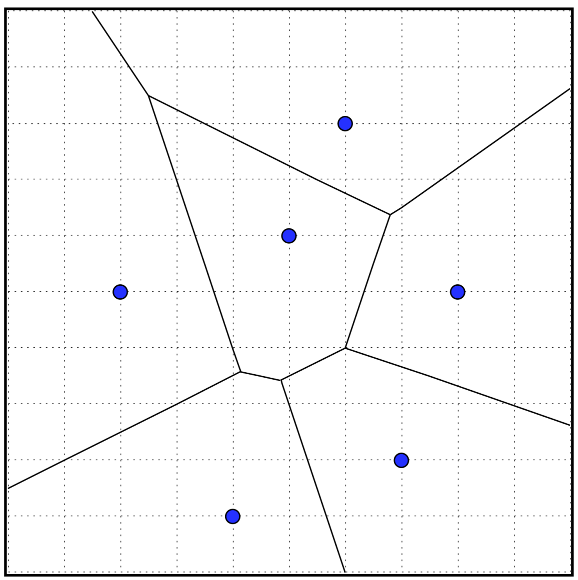


Voronoi Diagram	reduced-precision Voronoi diagram

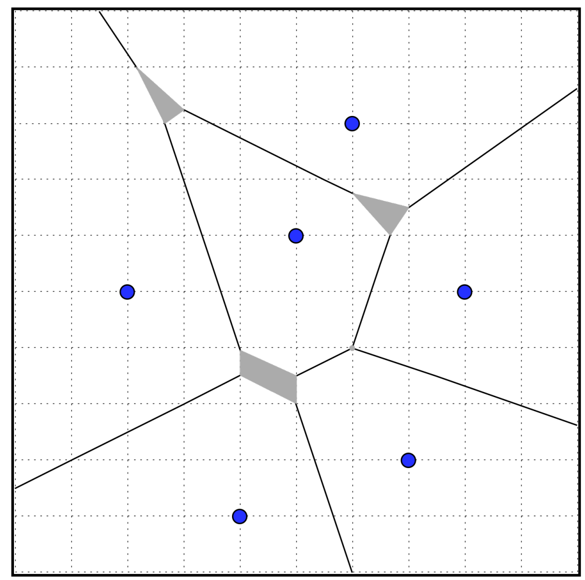
Reduced-precision Voronoi diagram

Replace connected subtrees of Vor edges inside a cell with their convex hulls.





Voronoi diagram



reduced-precision Voronoi

Results (preview)

Randomized Incremental Construction
of the *reduced-precision Voronoi diagram*
of n points on an $m \times m$ grid.

Time: $O(n (\log n + \log m))$ expected

Space: O(n) expected

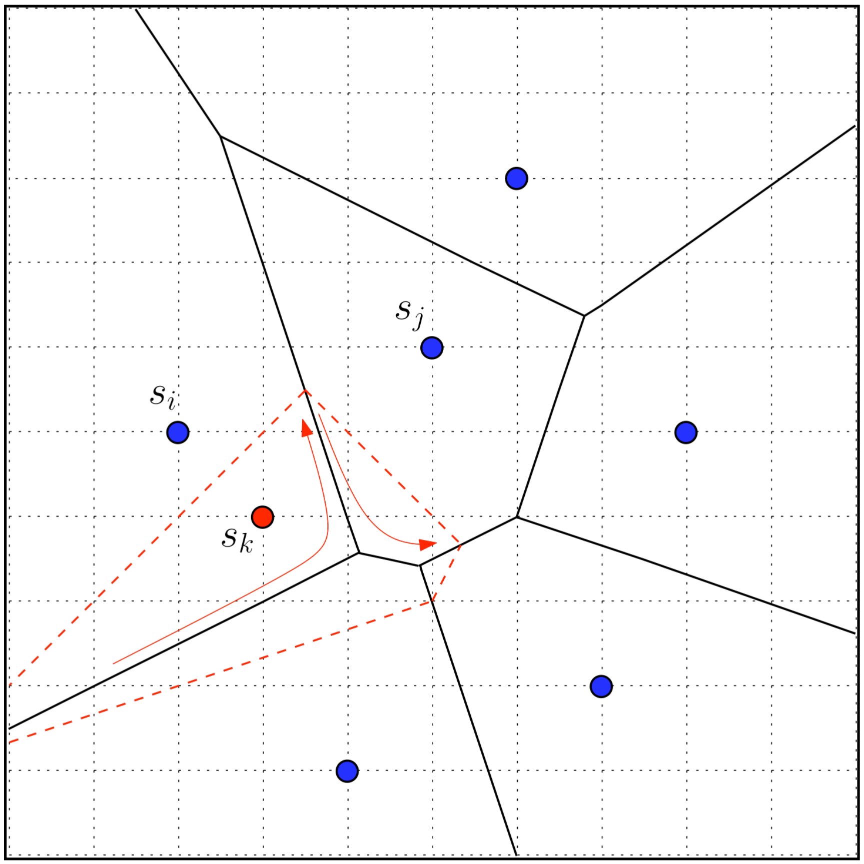
Precision: 3× precision of the input.

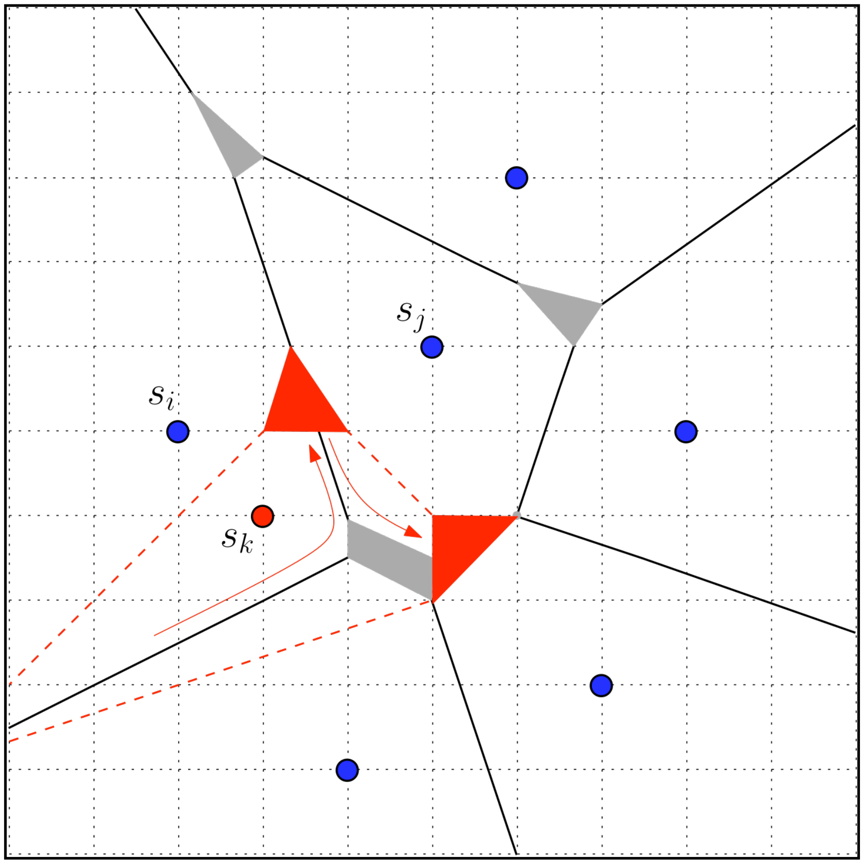
Construction of LPT's implicit Voronoi
w/o computing the full Voronoi diagram.

Rand. Incremental Construction

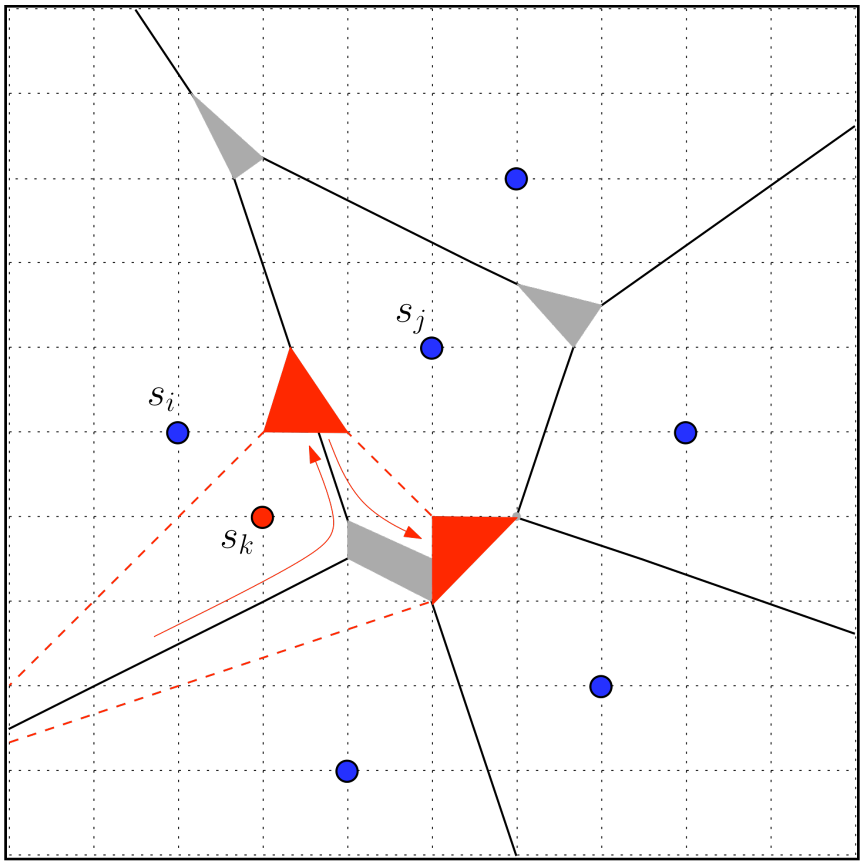
Invariant: Maintain reduced-precision Voronoi as each new site is added.

Update step: Extension of [SI92], walk the deleted tree.





Operations for RIC



What we need:

- Low precision construction for bisector intersection
- Determine the next edge in the tree walk

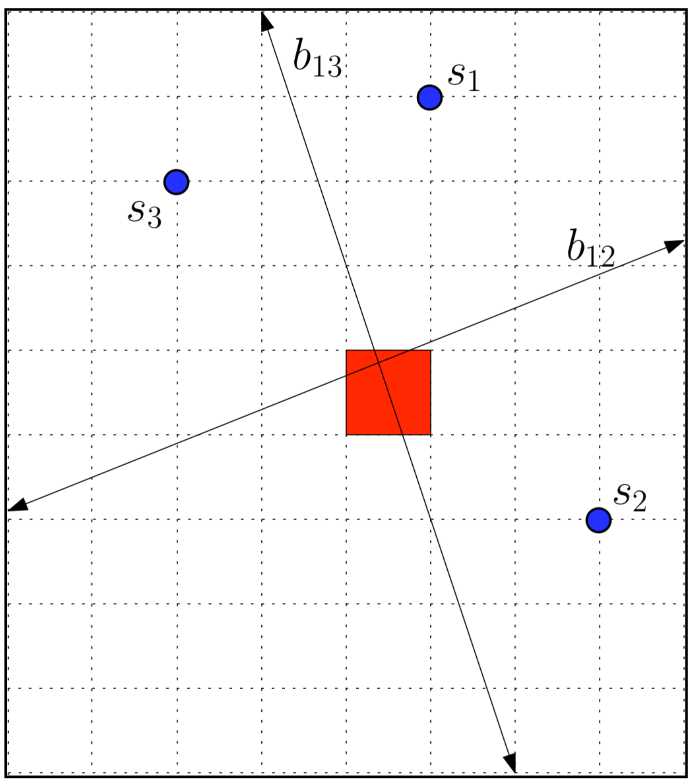
Low precision bisector intersection

Given:

Three sites s_1, s_2 and s_3

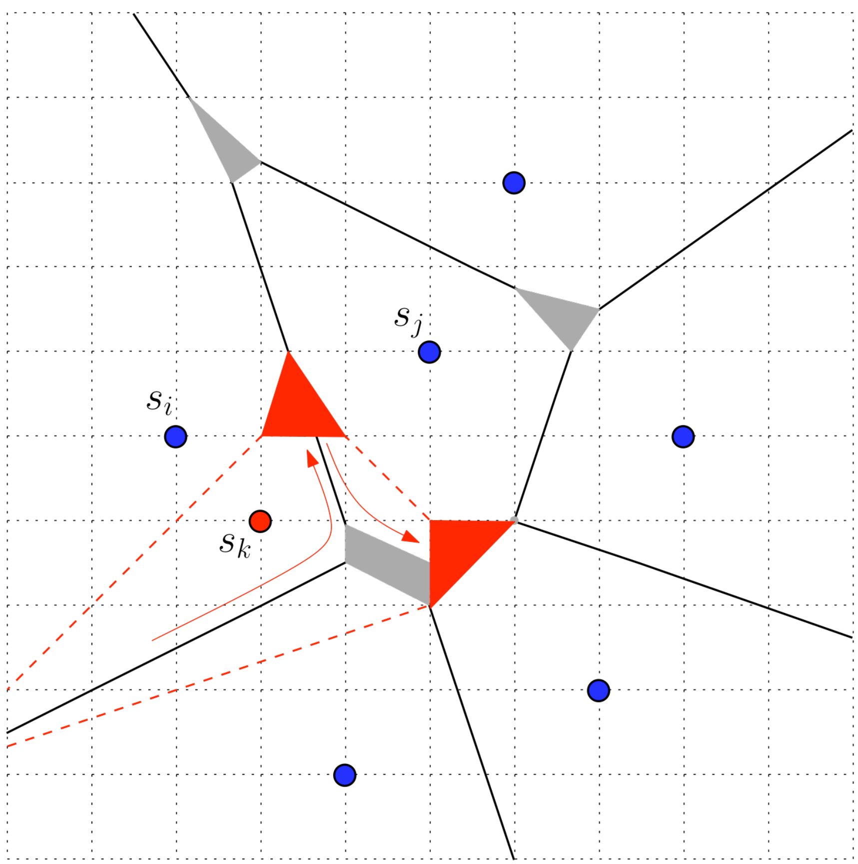
Find:

The grid cell that contains the intersection of bisectors b_{12} and b_{13}



Time: O(log *m*)
Precision: degree 3

Operations for RIC



What we needed:

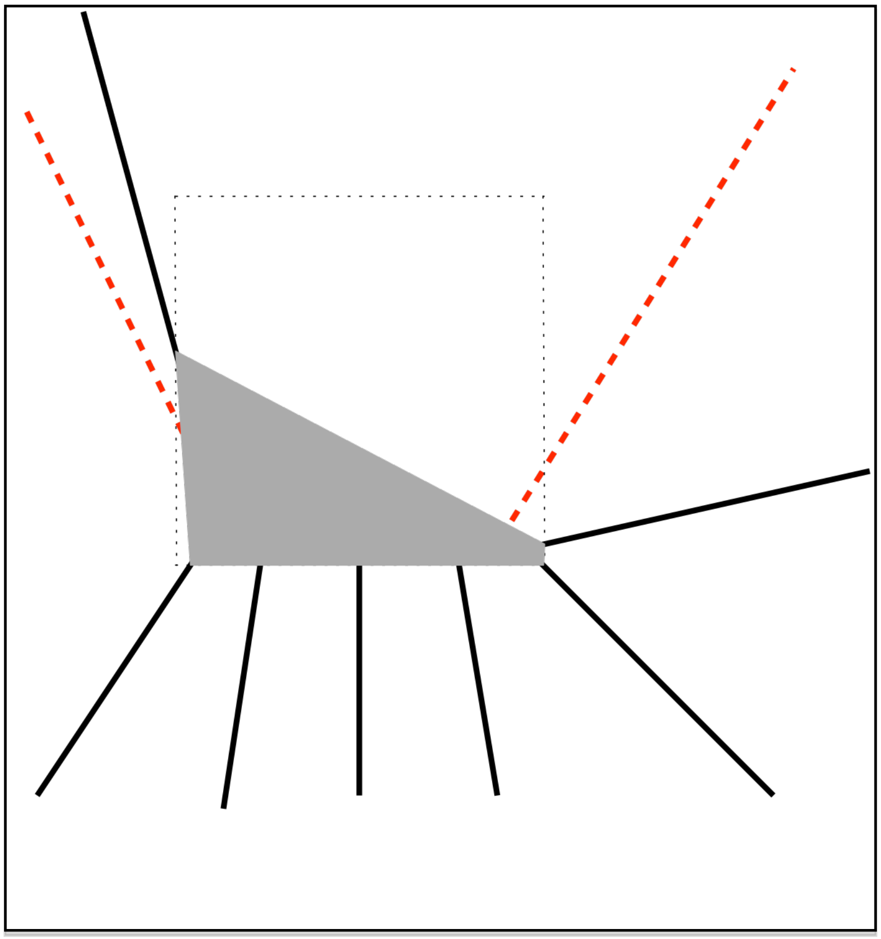
- Low precision construction for bisector intersection
- Determine the next edge in the tree walk

What we have:

- Bisector intersection degree 3, $O(\log m)$

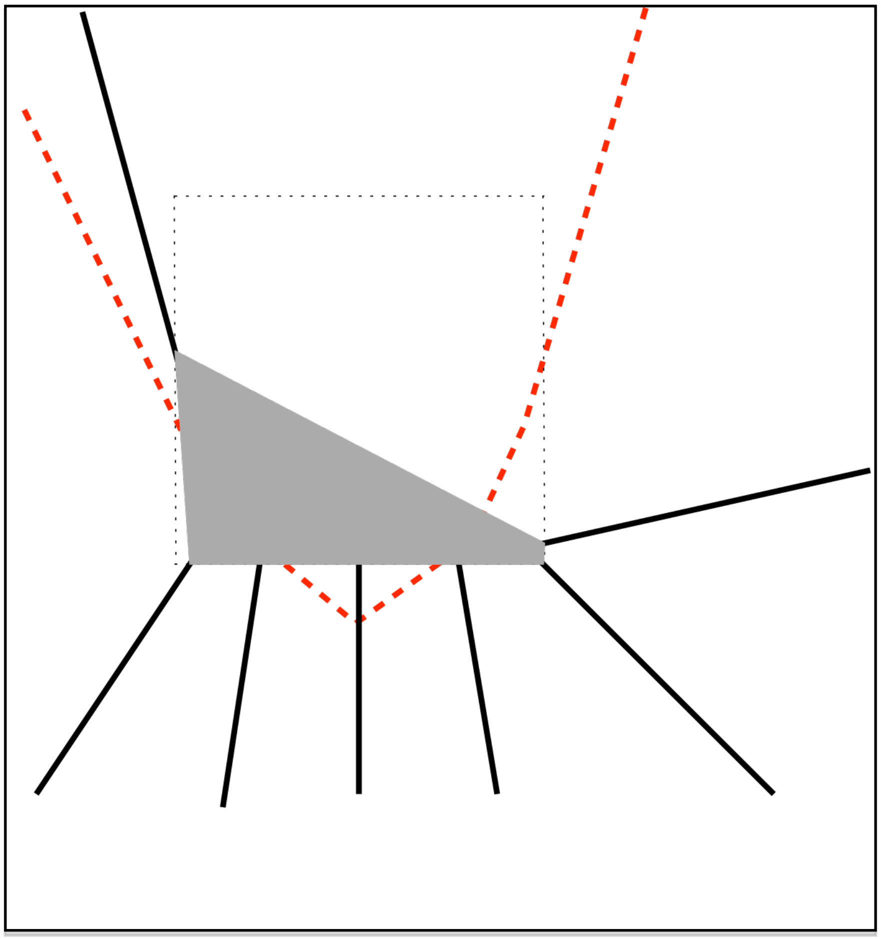
Where to walk next

Where do we walk once we have found an intersection?



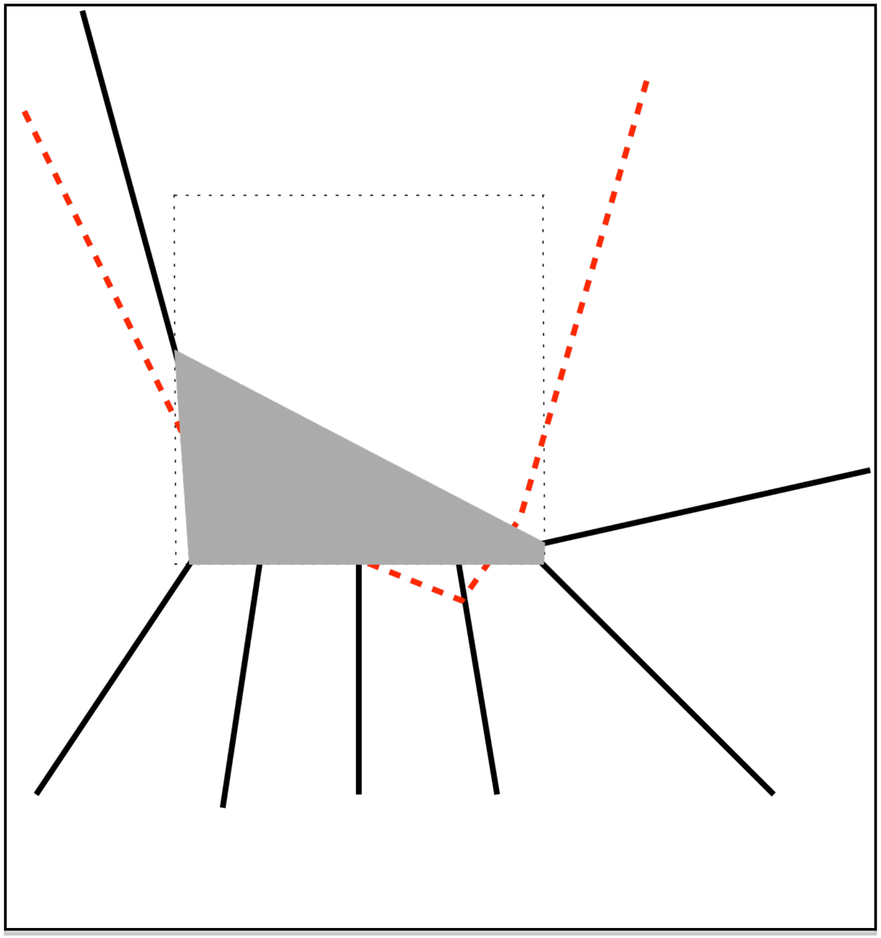
Where to walk next

Where do we walk once we have found an intersection?

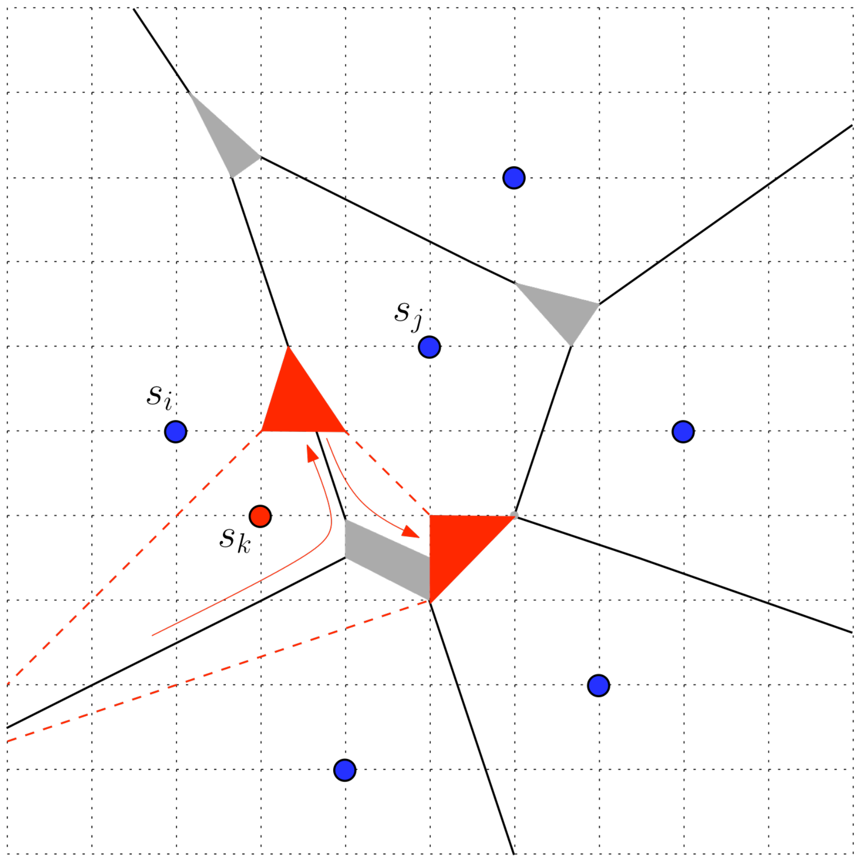


Where to walk next

Where do we walk once we have found an intersection?



Operations for RIC



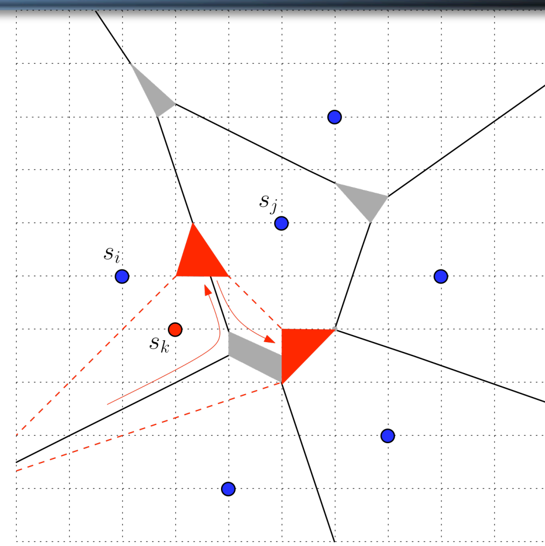
What we needed:

- Low precision construction for bisector intersection
- Determine the next edge in the tree walk

What we have:

- Bisector intersection degree 3, $O(\log m)$
- Next tree edge degree 3, $O(\log n)$

RIC alg for reduced-precision Voronoi



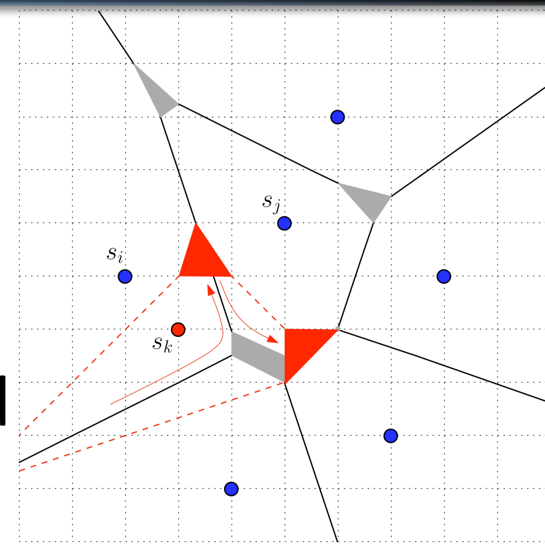
Update step to add a new site s_k

1. Find cell containing s_k
2. Identify bisectors and cells in the new cell of s_k
3. Walk tree inside cell of s_k

   3a. Binary search bisector for crossing

   3b. Compute grid intersection with cell s_k

Analysis



RIC Facts

- Creates θ(*n*) vertices expected
- Point location takes θ(*n* log *n*) expected

Charging scheme

- Binary searchers for (3a) (log *m*) and (3b) (log *n*) are charged to vertex creation.

Therefore,

- It takes O(*n*) space and O(*n* (log *n* + log *m*)) time to build the reduced-precision Voronoi diagram of *n* sites on a grid of size *m* using 3× input precision.

Results

Randomized Incremental Construction
of the *reduced-precision Voronoi diagram*
of n points on an $m \times m$ grid.

Time: $O(n (\log n + \log m))$ expected

Space: O(n) expected

Precision: 3× precision of the input.

Construction of LPT's implicit Voronoi
w/o computing the full Voronoi diagram.

Open problems

Can we…

- create a reduced-precision Voronoi diagram with $2\times$ precision of the input?
- generalize the algorithm to higher dimensions?
- Reduced-precision Voronoi diagrams with other input sites?
- treat precision as a limited resource (like time & space) when solving other algorithmic problems?

THANK YOU!