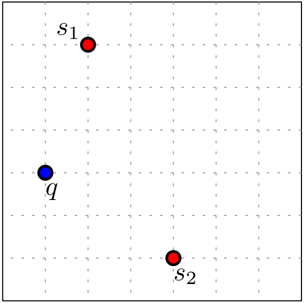
Discrete Voronoi Diagrams and Post Office Query Structures without the InCircle Predicate

Timothy M. Chan David L. Millman Jack Snoeyink

November 13, 2009

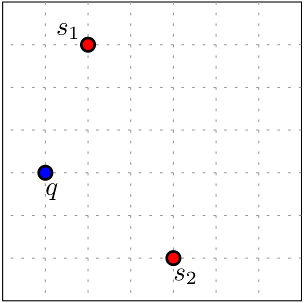


Given

sites $S = \{s_1, \ldots, s_n\}$ and
query points
on a $U \times U$ grid

Compute

using double precision,
a data structure
supporting post office queries
in $O(\log n)$ time
and double precision



Given

sites $S = \{s_1, \ldots, s_n\}$ and
query points
on a $U \times U$ grid

Compute

using double precision,
a data structure
supporting post office queries
in $O(\log n)$ time
and double precision

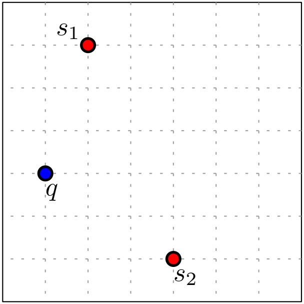
Result

Construction takes $O(U^2)$ expected time
assuming $O(n \log n) < O(U^2)$

The Algorithm Uses

Discrete Voronoi diagram, Voronoi Polygon Set,
Proxy Segments and Trapezoidation

Is q closer to s_1?

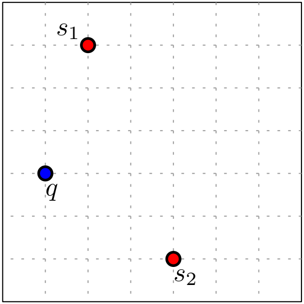


$$
\begin{aligned}
\mathbb{U} &= \{1, 2, \ldots, U\} \\
s_1, \quad s_2, &\quad q \in \mathbb{U}^2 \\
s_1 &= (x_1, y_1) \\
s_2 &= (x_2, y_2) \\
q &= (x_q, y_q)
\end{aligned}
$$

$$\|q - s_1\|^2 \gtrless \|q - s_2\|^2$$

Analyzing Precision[LPT99]

Is q closer to s_1?



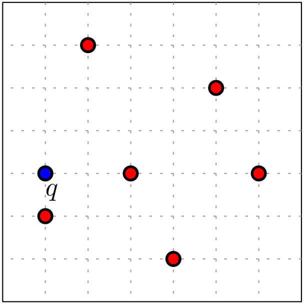
$$
\begin{aligned}
\mathbb{U} &= \{1, 2, \ldots, U\} \\
s_1, \quad s_2, &\quad q \in \mathbb{U}^2 \\
s_1 &= (x_1, y_1) \\
s_2 &= (x_2, y_2) \\
q &= (x_q, y_q)
\end{aligned}
$$

$$\|q - s_1\|^2 \quad \gtrless \quad \|q - s_2\|^2$$

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

$\boxed{\text{Degree 2}}$

Which site is q closest to?



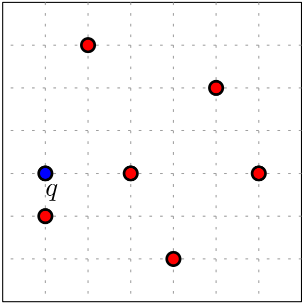
Given

Sites $S = \{s_1, \ldots, s_n\}$ and query point q with $s_i, q \in \mathbb{U}^2$

Determine

The site of S closest to q

Which site is q closest to?



Given

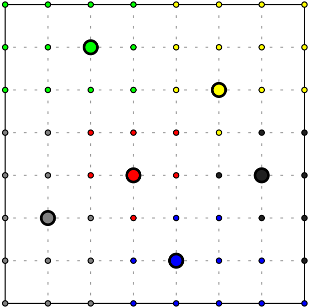
Sites $S = \{s_1, \ldots, s_n\}$ and query point q with $s_i, q \in \mathbb{U}^2$

Determine

The site of S closest to q

	Preprocess		Query	
	Alg	Time	Alg	Time
Brute force	-	-	deg 2	$O(n)$
Voronoi diagram	deg 4	$O(n \log n)$	deg 6	$O(\log n)$
Imp Voronoi [LPT99]	deg 4	$O(n \log n)$	deg 2	$O(\log n)$
RP-Voronoi [MS09]	deg 3	$O(n \log(Un))$	deg 2	$O(\log n)$

Discrete Voronoi Diagram



Given

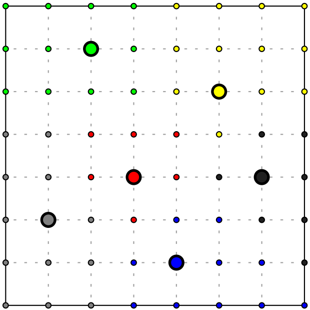
A grid of size U and

Sites $S = \{s_1, \ldots, s_n\} \subset \mathbb{U}^2$

Label

Each grid point of $\mathbb{U}^2$ with the closest site of S

Discrete Voronoi Diagram



Given

A grid of size U and

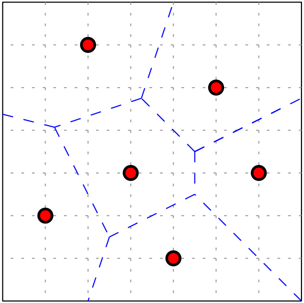
Sites $S = \{s_1, \ldots, s_n\} \subset \mathbb{U}^2$

Label

Each grid point of $\mathbb{U}^2$ with the closest site of S

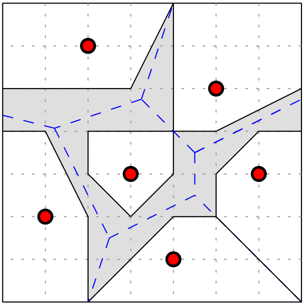
	Alg	Time
Brute Force	deg 2	$O(nU^2)$
Query the Voronoi diagram	deg 4	$O(U^2 \log n)$
Nearest Neighbor Trans. [B90]	deg 4	$O(U^2)$
Discrete Voronoi diagram [C06]	deg 3	$O(U^2)$
GPU Hardware [H99]	-	$\Theta(nU^2)$

Voronoi Polygon Set



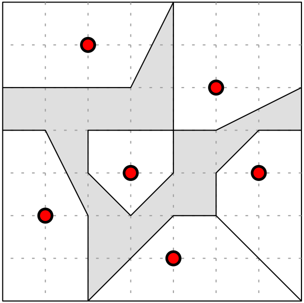
- Partition of $\mathbb{U}^2$
- n convex polygons $\{C(s_1), \ldots, C(s_n)\}$

Voronoi Polygon Set



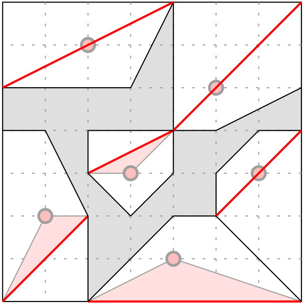
- Partition of $\mathbb{U}^2$
- n convex polygons $\{C(s_1), \ldots, C(s_n)\}$
- Where $C(s_i)$ is the convex hull of the grid points in the Voronoi cell of s_i

Voronoi Polygon Set



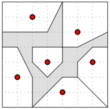
- Partition of $\mathbb{U}^2$
- n convex polygons $\{C(s_1), \ldots, C(s_n)\}$
- Where $C(s_i)$ is the convex hull of the grid points in the Voronoi cell of s_i
- Grey gaps

Voronoi Polygon Set

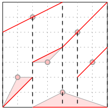


- Partition of $\mathbb{U}^2$
- n convex polygons $\{C(s_1), \ldots, C(s_n)\}$
- Where $C(s_i)$ is the convex hull of the grid points in the Voronoi cell of s_i
- Grey gaps
- Proxy segment

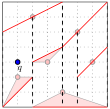
Construct and Query Post Office Structure



Compute Voronoi Polygon Set

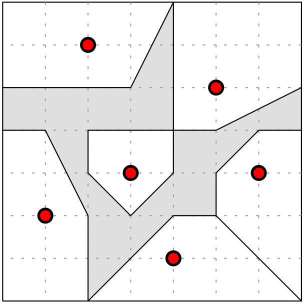


Compute Trapeziod graph of the proxy



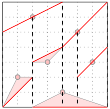
Query the post office structure

Construct and Query Post Office Structure

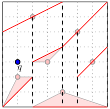


Compute *Voronoi Polygon Set*

- Time: $O(U^2)$ expected time
- Space: $O(n \log U)$
- Precision: Deg 2

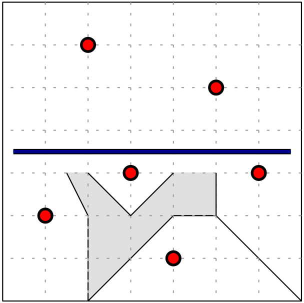


Compute Trapeziod graph of the proxy



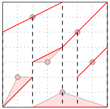
Query the post office structure

Construct and Query Post Office Structure

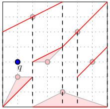


Compute *Voronoi Polygon Set*

- Time: $O(U^2)$ expected time
- Space: $O(n \log U)$
- Precision: Deg 2

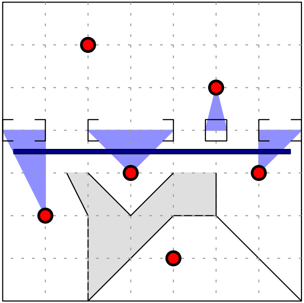


Compute Trapeziod graph of the proxy



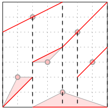
Query the post office structure

Construct and Query Post Office Structure

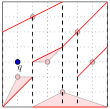


Compute *Voronoi Polygon Set*

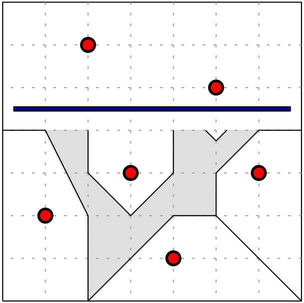
- Time: $O(U^2)$ expected time
- Space: $O(n \log U)$
- Precision: Deg 2



Compute Trapeziod graph of the proxy

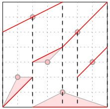


Query the post office structure

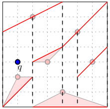


Compute *Voronoi Polygon Set*

- Time: $O(U^2)$ expected time
- Space: $O(n \log U)$
- Precision: Deg 2

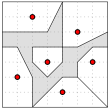


Compute Trapeziod graph of the proxy

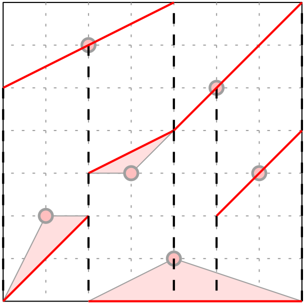


Query the post office structure

Construct and Query Post Office Structure

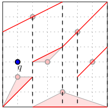


Compute Voronoi Polygon Set

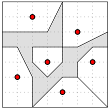


Compute Trapeziod graph of the proxy

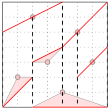
- Time: $O(n \log n)$ expected
- Space: $O(n)$ expected
- Precision: Deg 2



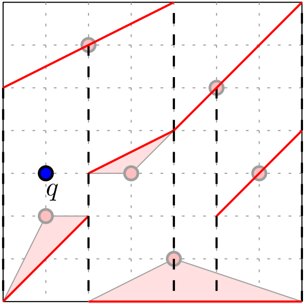
Query the post office structure



Compute Voronoi Polygon Set



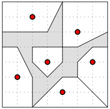
Compute Trapeziod graph of the proxy



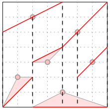
Query the post office structure

- Time: $O(\log n)$ expected
- Precision: Deg 2

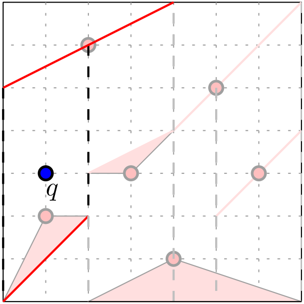
Construct and Query Post Office Structure



Compute Voronoi Polygon Set



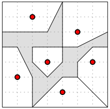
Compute Trapeziod graph of the proxy



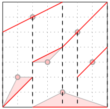
Query the post office structure

- Time: $O(\log n)$ expected
- Precision: Deg 2

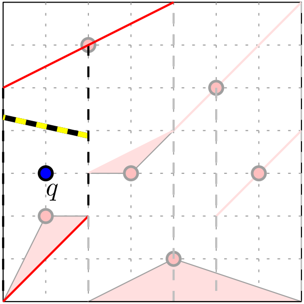
Construct and Query Post Office Structure



Compute Voronoi Polygon Set



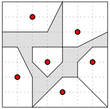
Compute Trapeziod graph of the proxy



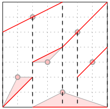
Query the post office structure

- Time: $O(\log n)$ expected
- Precision: Deg 2

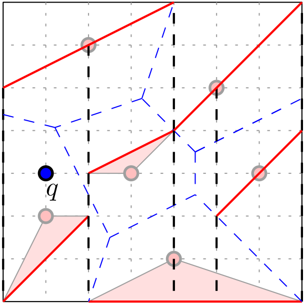
Construct and Query Post Office Structure



Compute Voronoi Polygon Set



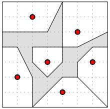
Compute Trapeziod graph of the proxy



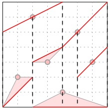
Query the post office structure

- Time: $O(\log n)$ expected
- Precision: Deg 2

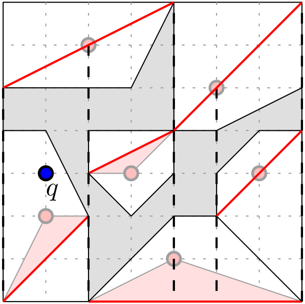
Construct and Query Post Office Structure



Compute Voronoi Polygon Set



Compute Trapeziod graph of the proxy



Query the post office structure

- Time: $O(\log n)$ expected
- Precision: Deg 2

Results

Assuming $O(n \log n) < O(U^2)$

Compute Voronoi Polygon Set

- Time: $O(U^2)$ expected time
- Space: $O(n \log U)$ and $O(n)$ for proxies only
- Precision: Deg 2

Query Post Office Structure

- Time: $O(\log n)$ expected
- Precision: Deg 2

Compute 2D discrete Voronoi

- Time: $O(U^2)$ expected time.
- Space: $O(n + U)$
- Precision: Deg 2

Open problems

Can we...

- build a post office query structure with double precision in sub-quadratic time?
- generalize to higher dimension?
- treat precision as a limited resource (like time and space) when solving other algorithmic problems?

Open problems

Can we...

- build a post office query structure with double precision in sub-quadratic time?
- generalize to higher dimension?
- treat precision as a limited resource (like time and space) when solving other algorithmic problems?