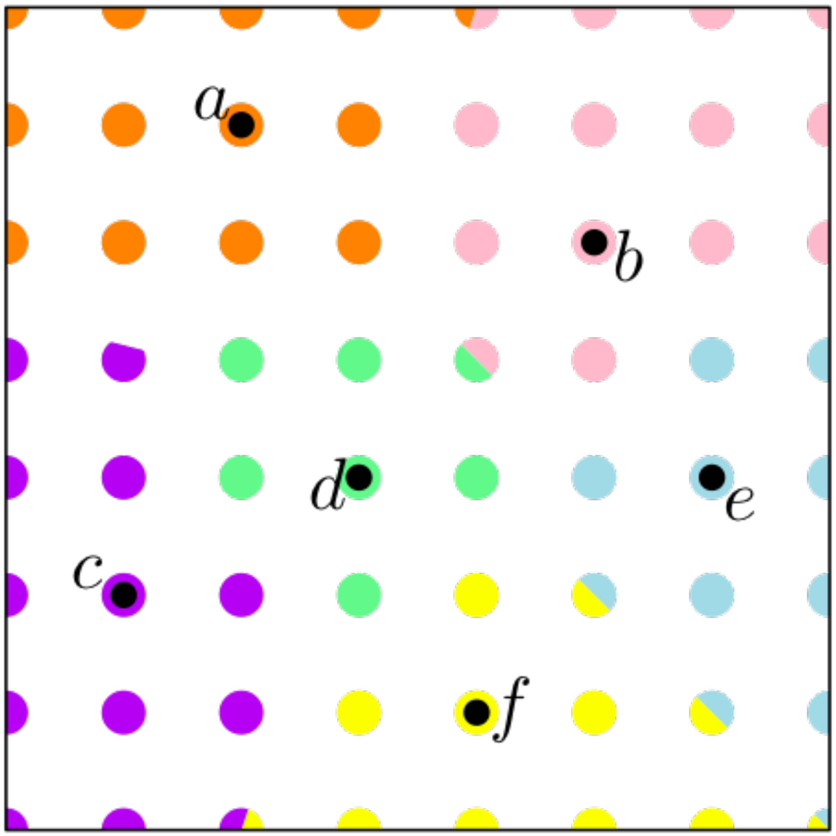
Degree-driven Geometric Algorithm Design

David L. Millman

University Of North Carolina - Chapel Hill

ITCS'12
9 January 2012

An Example Problem



Given n sites on a pixel grid, what is the closest site to each pixel?

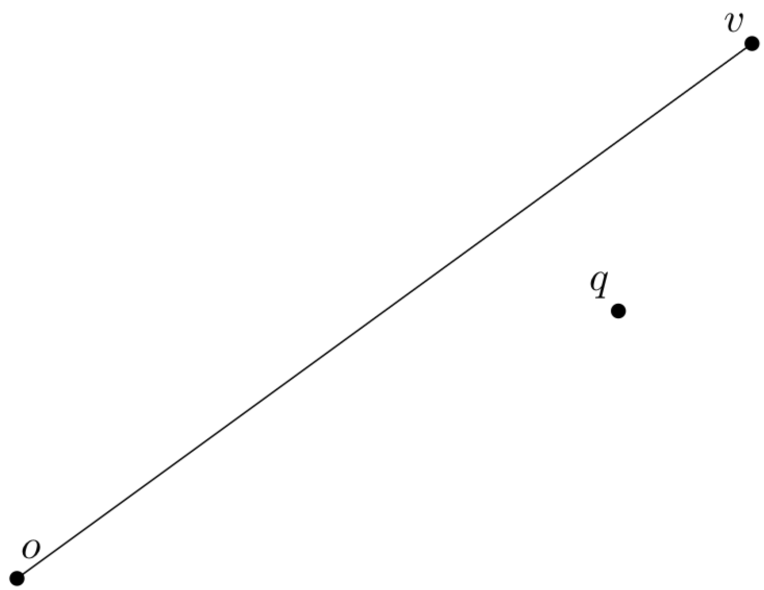
How much precision is needed to determine this?

Why Precision [Y97]

$$\text{Geometric Computing} = \text{Numerical} + \text{Combinatorial Computing}.$$

Analyzing Precision [LPT99]

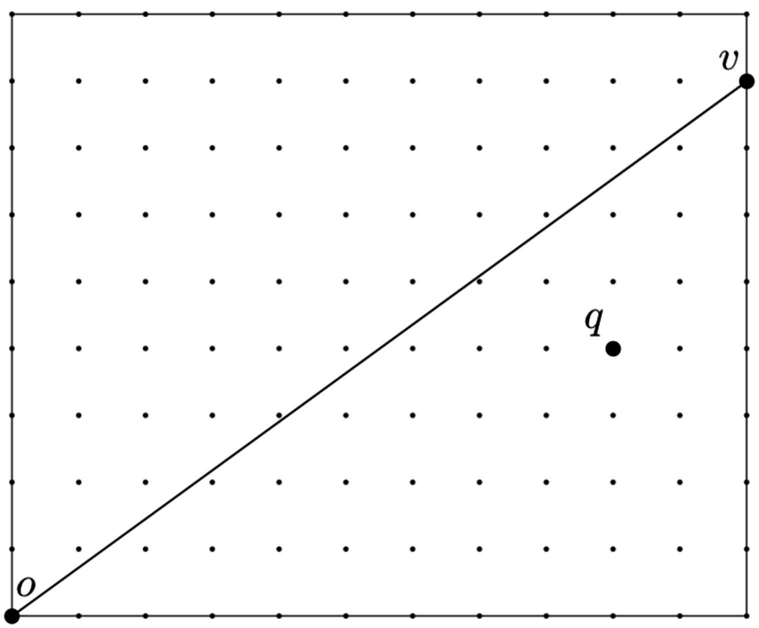
e.g. Precision of the Orientation test:



$\texttt{orientation}(o, v, q)$

Analyzing Precision [LPT99]

e.g. Precision of the Orientation test:

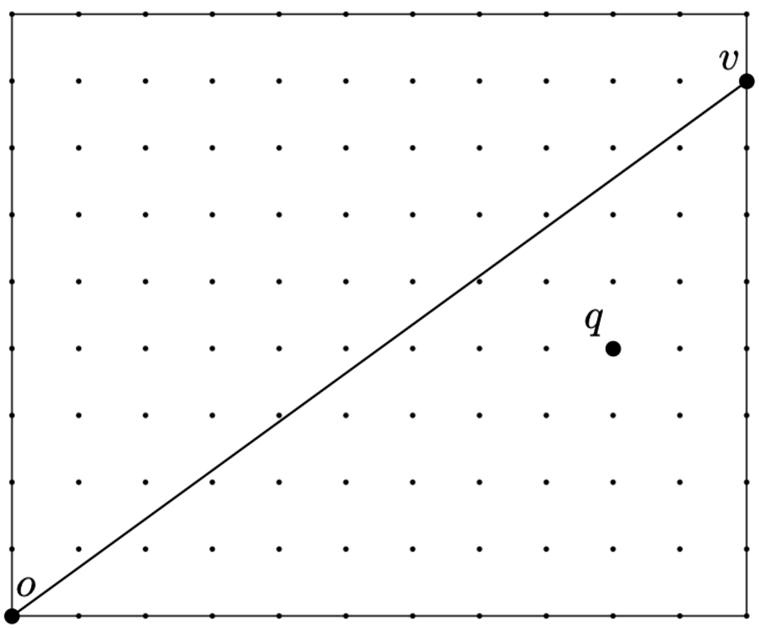


$$\mathbb{U} = \{1, \ldots, U\}^2$$
$$o, v, q \in \mathbb{U}$$
$$o = (o_x, o_y)$$
$$v = (v_x, v_y)$$
$$q = (q_x, q_y)$$

$\texttt{orientation}(o, v, q)$

Analyzing Precision [LPT99]

e.g. Precision of the Orientation test:

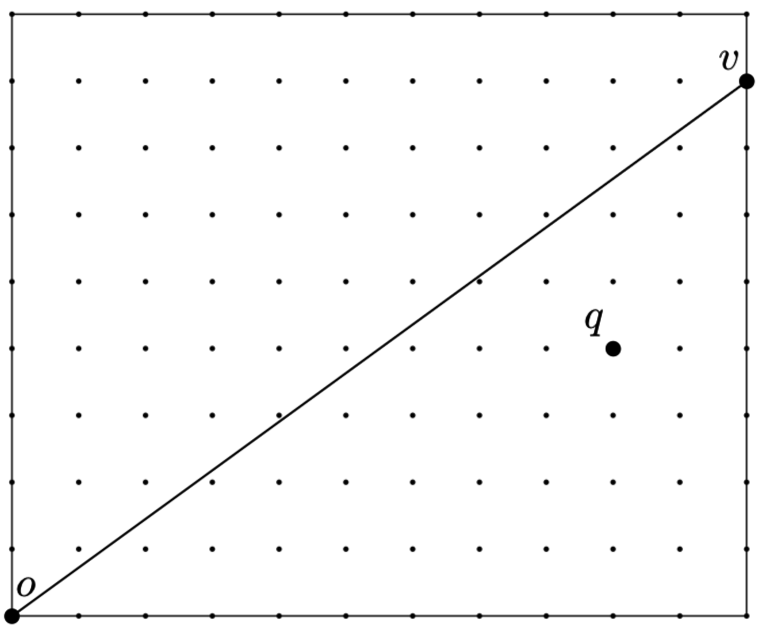


$$\mathbb{U} = \{1, \ldots, U\}^2$$
$$o, v, q \in \mathbb{U}$$
$$o = (o_x, o_y)$$
$$v = (v_x, v_y)$$
$$q = (q_x, q_y)$$

$$\texttt{orientation}(o, v, q) = \begin{vmatrix} 1 & o_x & o_y \\ 1 & v_x & v_y \\ 1 & q_x & q_y \end{vmatrix}$$

Analyzing Precision [LPT99]

e.g. Precision of the Orientation test:



$$\mathbb{U} = \{1, \ldots, U\}^2$$

$$o, v, q \in \mathbb{U}$$

$$o = (o_x, o_y)$$

$$v = (v_x, v_y)$$

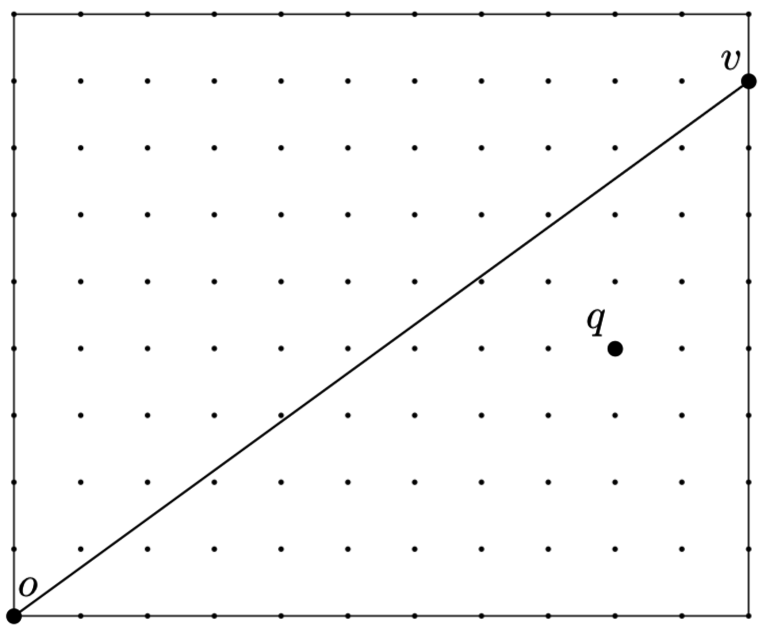
$$q = (q_x, q_y)$$

$$\texttt{orientation}(o, v, q) = \begin{vmatrix} 1 & o_x & o_y \\ 1 & v_x & v_y \\ 1 & q_x & q_y \end{vmatrix}$$

$$= v_x q_y - v_x o_y - o_x q_y + o_x o_y$$
$$- v_y q_x + v_y o_x + q_y q_x - q_y o_x$$

Analyzing Precision [LPT99]

e.g. Precision of the Orientation test:



$$\mathbb{U} = \{1, \ldots, U\}^2$$
$$o, v, q \in \mathbb{U}$$
$$o = (o_x, o_y)$$
$$v = (v_x, v_y)$$
$$q = (q_x, q_y)$$

$$\texttt{orientation}(o, v, q) = \begin{vmatrix} 1 & o_x & o_y \\ 1 & v_x & v_y \\ 1 & q_x & q_y \end{vmatrix}$$

$$= v_x q_y - v_x o_y - o_x q_y + o_x o_y$$
$$- v_y q_x + v_y o_x + q_y q_x - q_y o_x$$

Degree 2

Some Results of My Thesis

Given

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

Construct Proxy Trapezoidation

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

Compute 2D Nearest Neighbor Trans

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

Query Proxy Trapezoidation

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

Construct Gabriel Graph

- Time: $O(n^2)$
- Space: $O(n^2)$
- Precision: degree 2

Other Projects

Parallel computational geometry

Vicente H.F. Batista, David L. Millman, Sylvain Pion, and Johannes Singler. Parallel geometric algorithms for multi-core computers. *Computational Geometry*, 43(8):663 – 677, 2010.

Monte Carlo algos for the neutron transport equation

David L. Millman, David P. Griesheimer, Brian Nease and Jack Snoeyink. Robust volume calculation for constructive solid geometry (CSG) components in Monte Carlo Transport Calculations. *Under Review*

David P. Griesheimer, David L. Millman, and Clarence R. Willis. Analysis of distances between inclusions in finite binary stochastic materials. *Journal of Quantitative Spectroscopy and Radiative Transfer in Journal of Quantitative Spectroscopy and Radiative Transfer*, 112(4):577–598, March 2011.

Contact Information

David L. Millman

dave@cs.unc.edu

http://cs.unc.edu/~dave