Algorithms for GIS:

Computing visibility on terrains

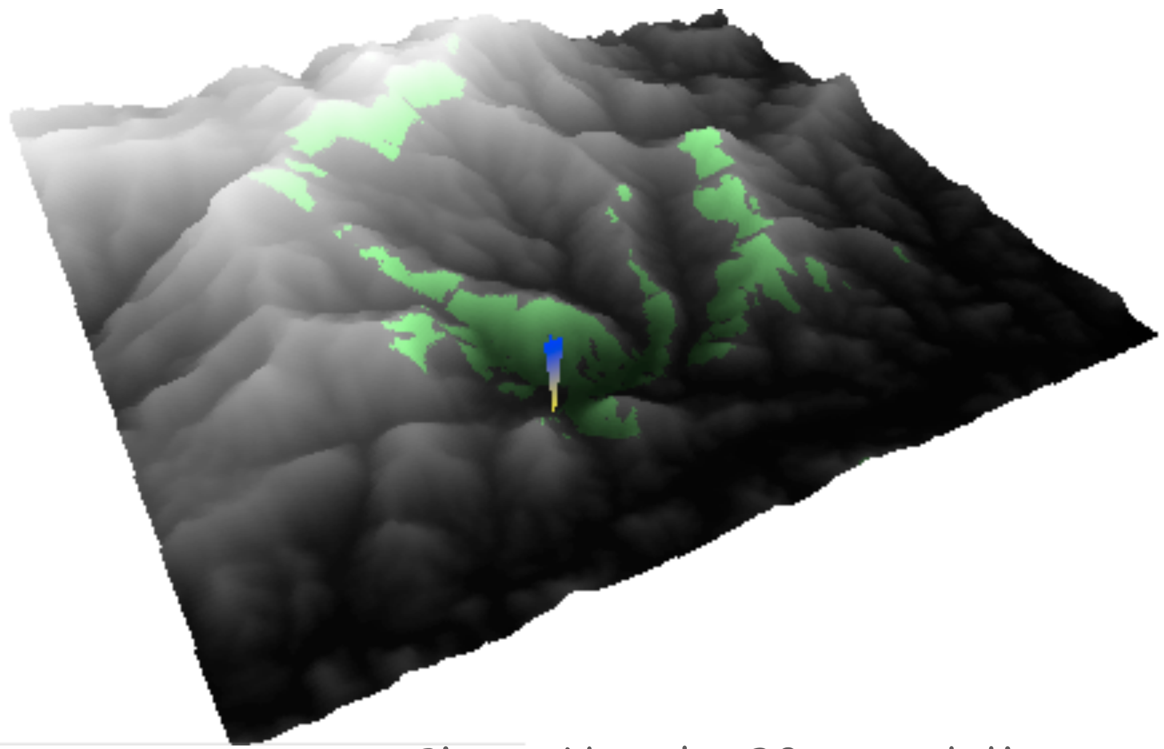
Visibility on terrains

- Are two points (on a terrain) visible to each other?
- What can one see from a given point (on a terrain)?
- How much does the visible area increase if we stand on a 10ft ladder?
- What is the point with largest visibility?
- What is the point with lowest visibility?
- How to place an ugly pipe in a scenic area?
- How to place a scenic highway?
- What is the cumulative visible area from these set of cell towers?
- Find a set of tower locations to cover the terrain
- …

Visibility on terrains

Problem:

- Terrain T + viewpoint v
- Compute the **viewshed** of v: the set of points in T visible from v



Sierra Nevada, 30m resolution

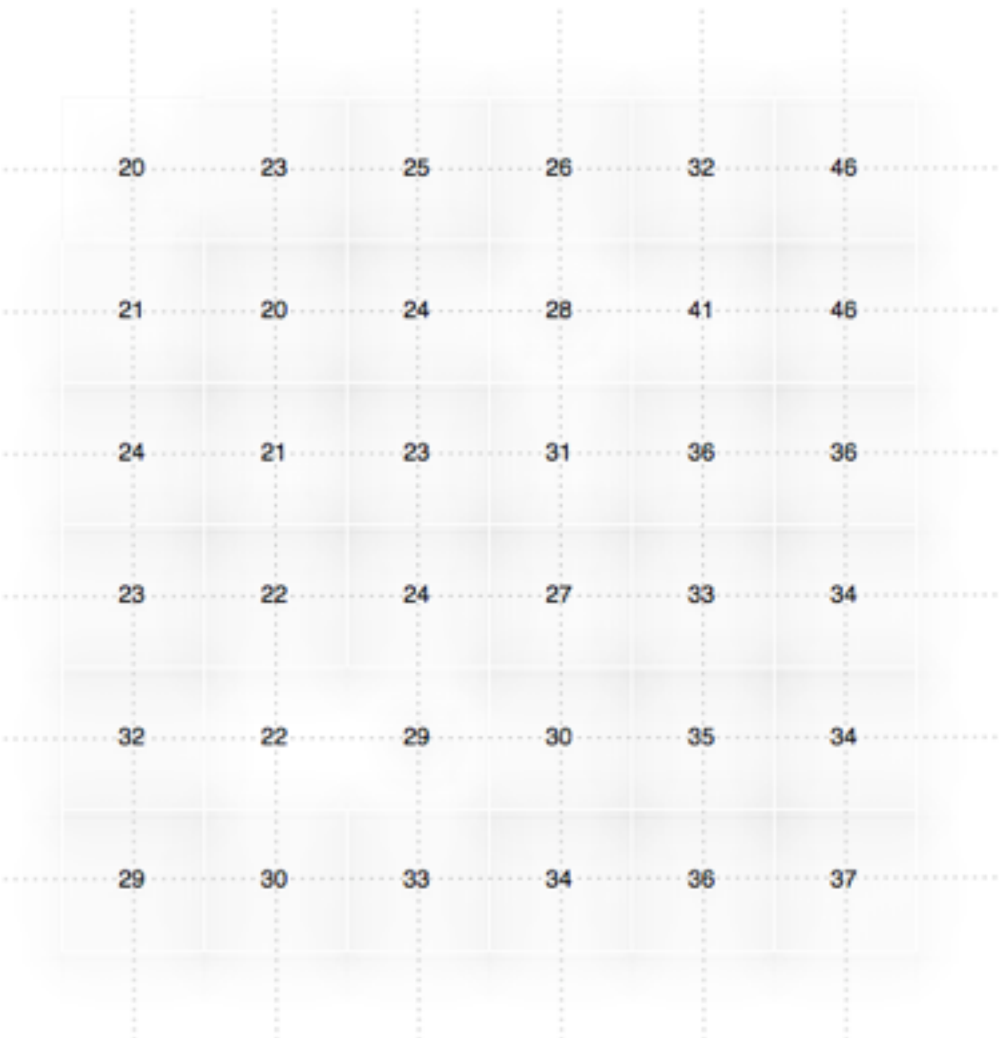
Visibility on terrains

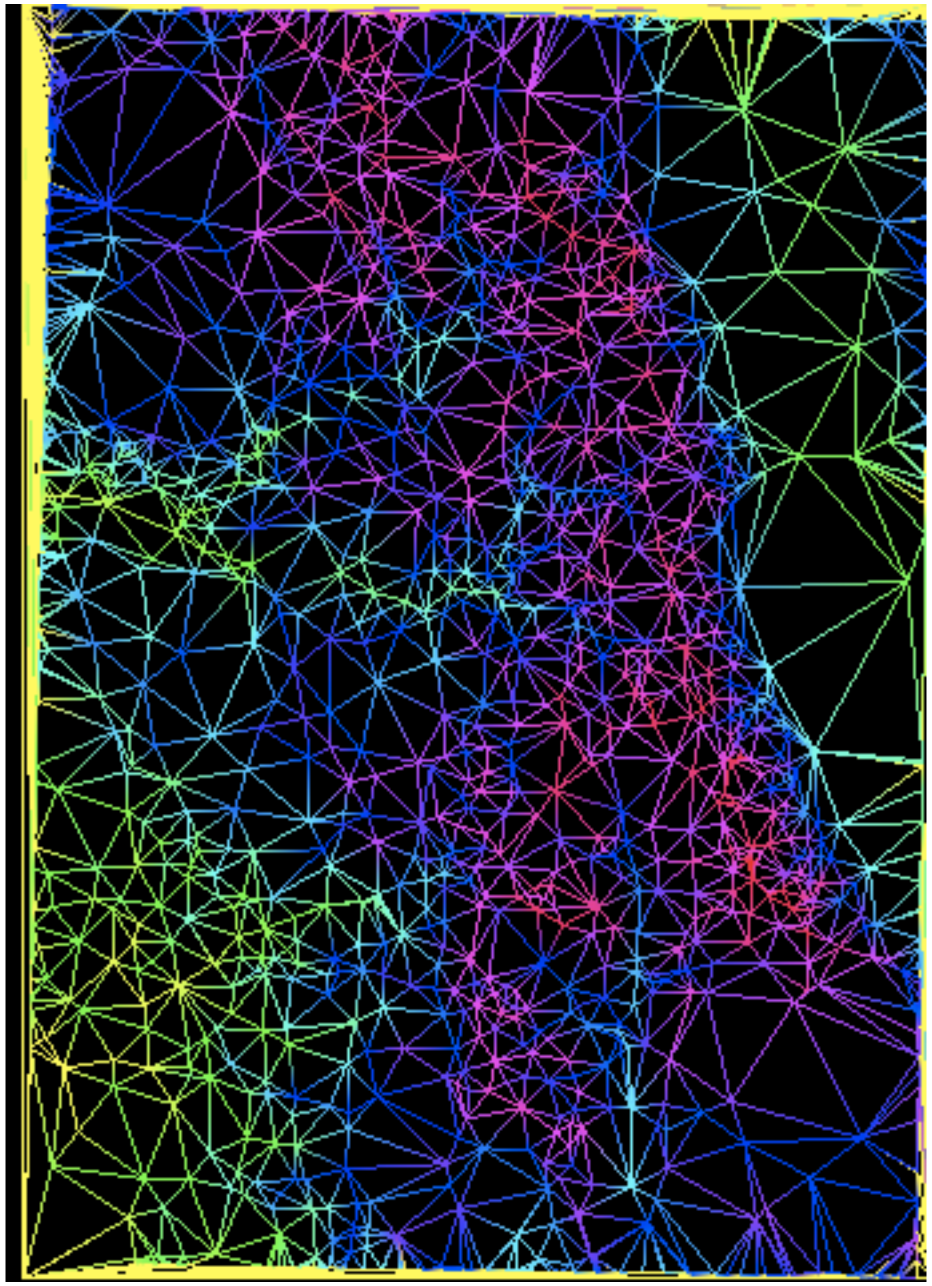
Input: terrain model (DEM = digital elevation model)

- grid
- TIN (triangulation)

Output: viewshed model

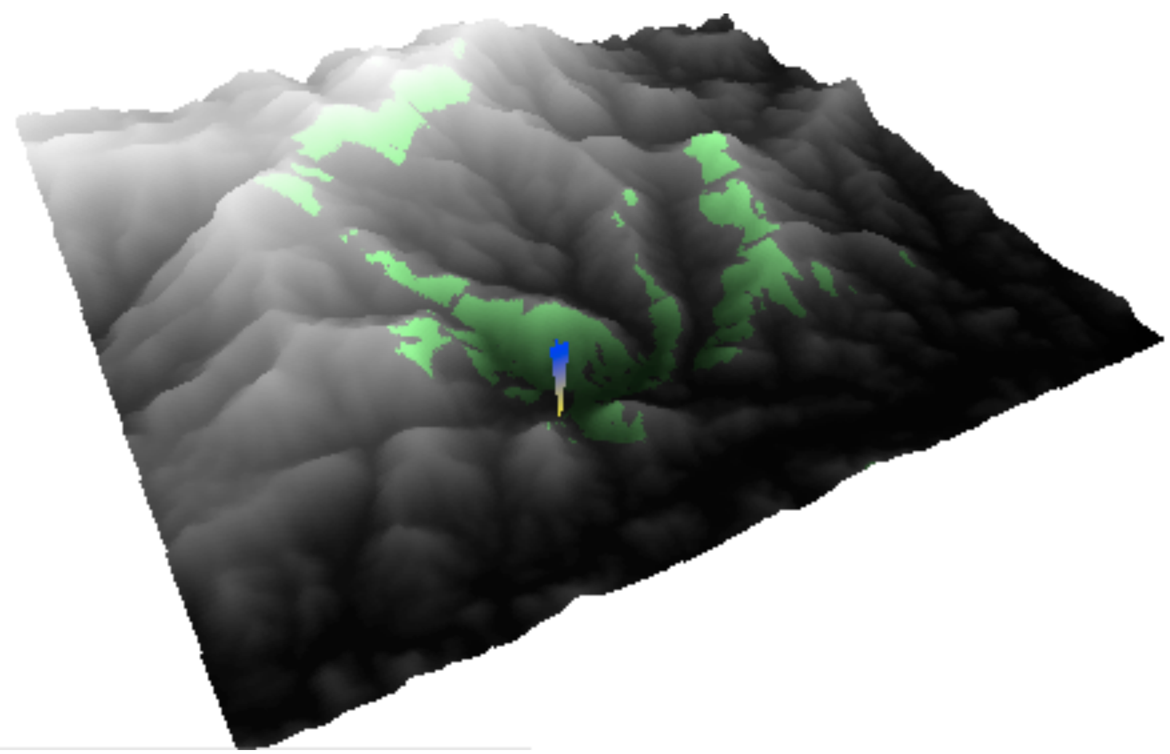
- grid elevation model ==> grid viewshed
- TIN elevation model ==> TIN viewshed





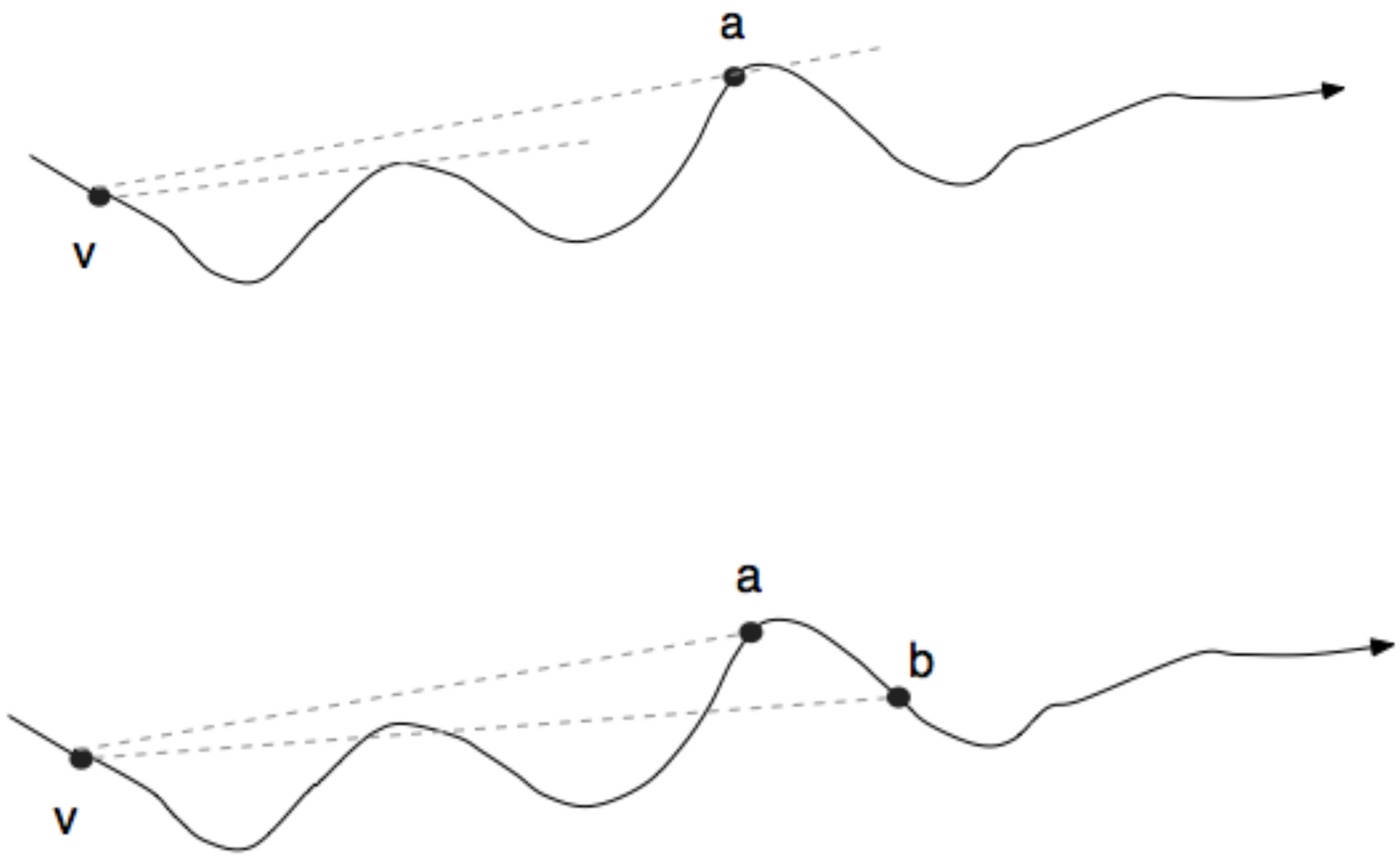
Visibility on grid terrains

20	23	25	26	32	46
21	20	24	28	41	46
24	21	23	31	36	36
23	22	24	27	33	34
32	22	29	30	35	34
29	30	33	34	36	37



Sierra Nevada, 30m resolution

Visibility



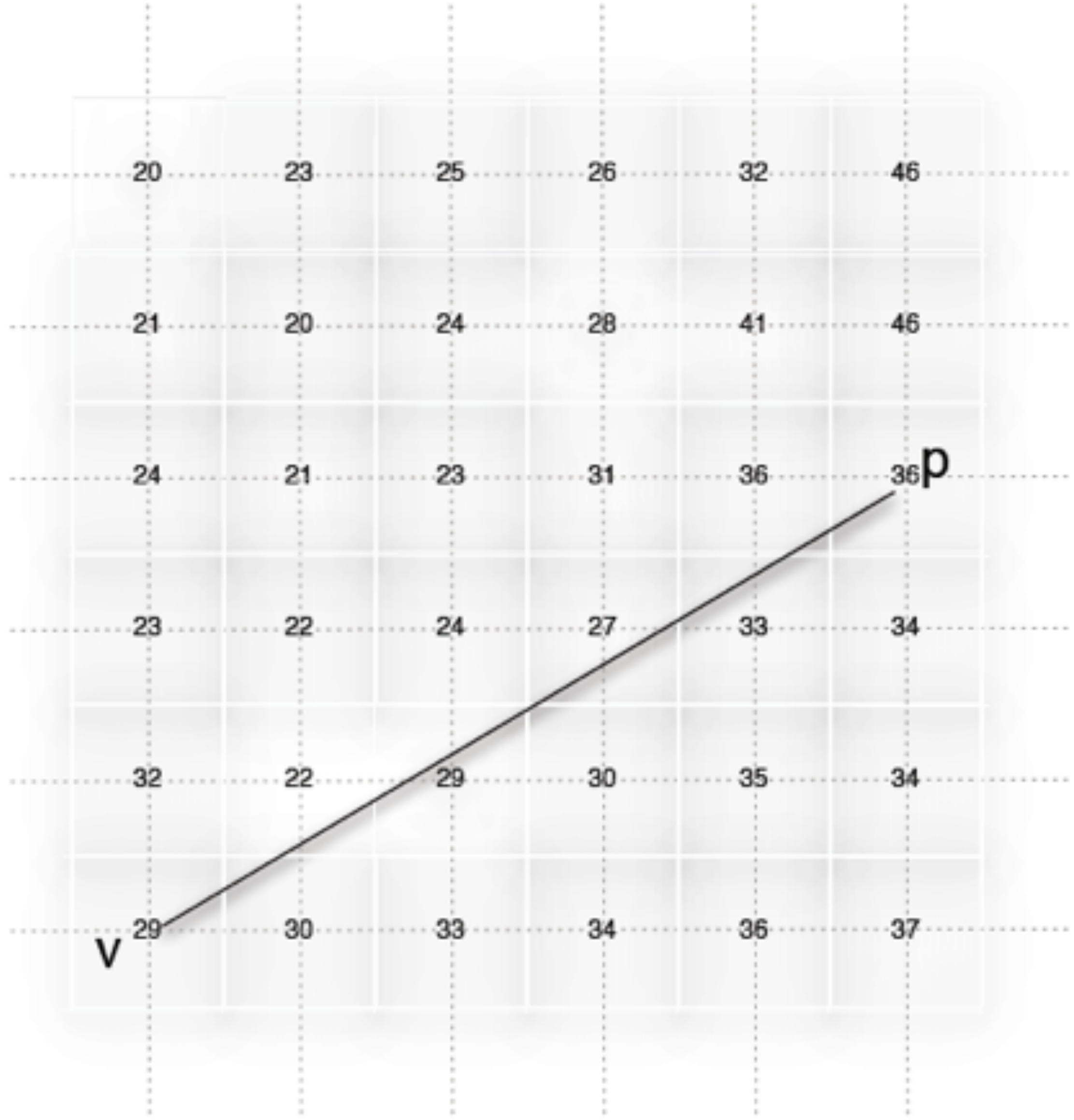
(u,v) visible iff segment uv does not intersect T

uv is called line-of-sight (LOS)

Basic viewshed algorithm

Input: elevation grid
Output: visibility grid, each point marked visible/invisible

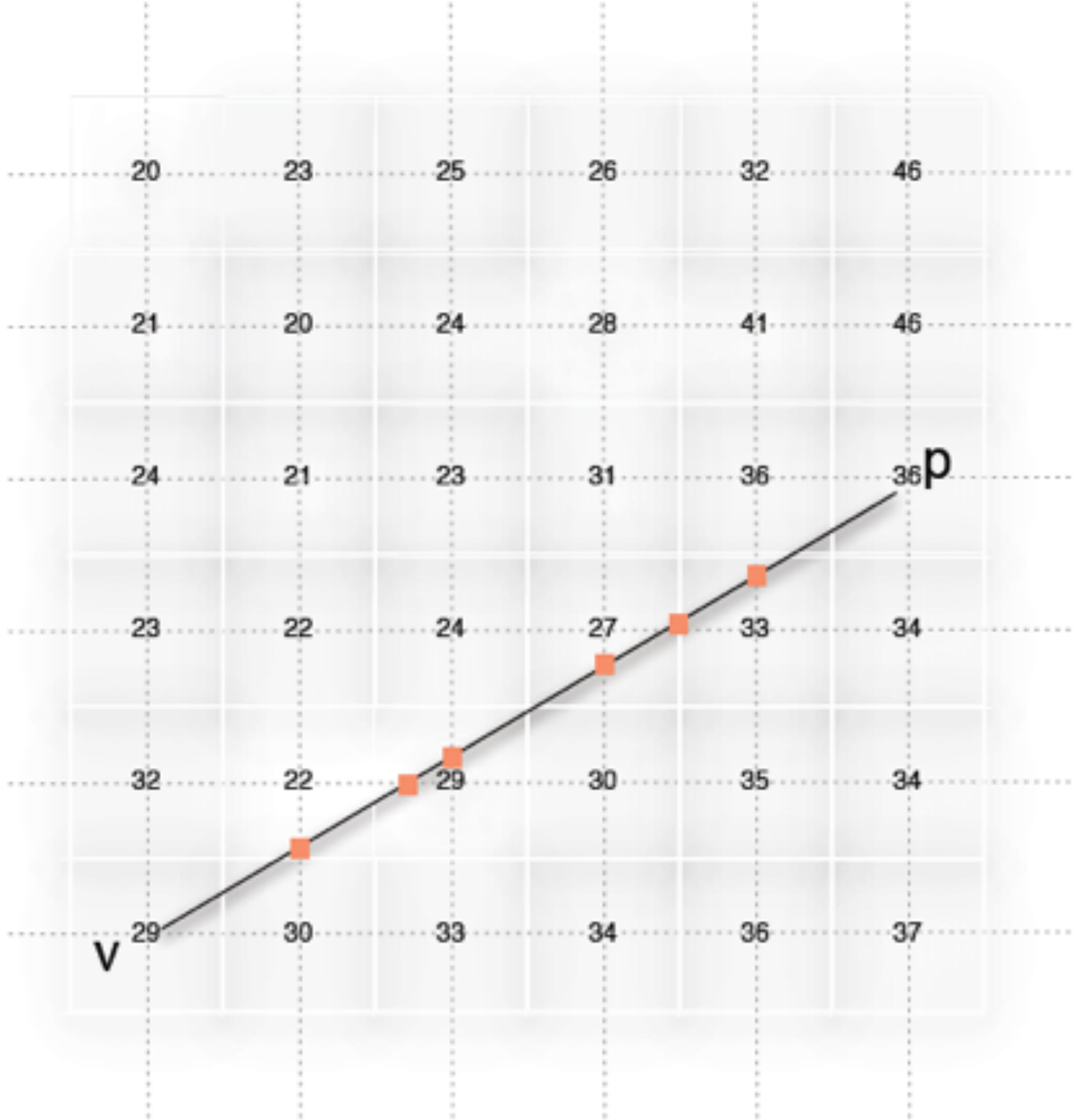
- For each p in grid
  - compute intersections between *vp* and grid lines
  - if all these points are below *vp* then *p* is visible



Basic viewshed algorithm

Input: elevation grid
Output: visibility grid, each point marked visible/invisible

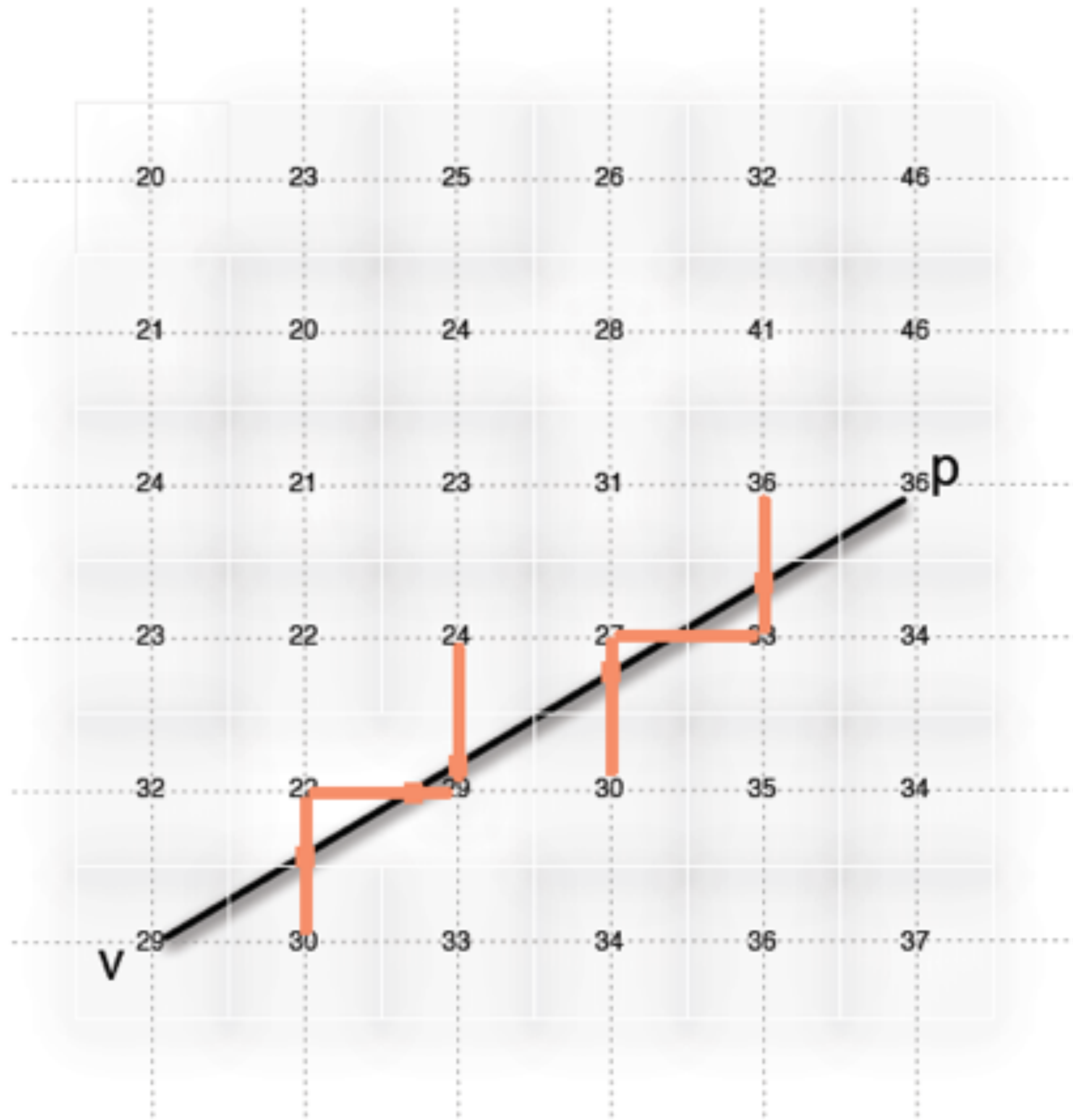
- For each p in grid
  - compute intersections between *vp* and grid lines
  - if all these points are below *vp* then *p* is visible



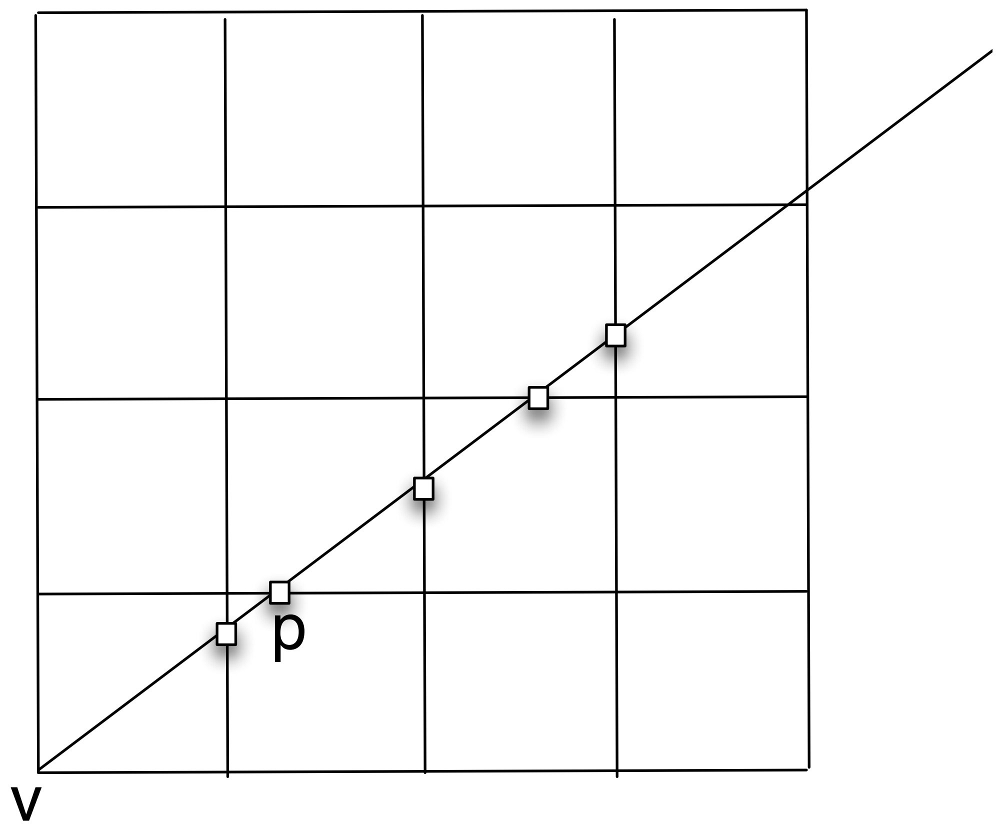
Basic viewshed algorithm

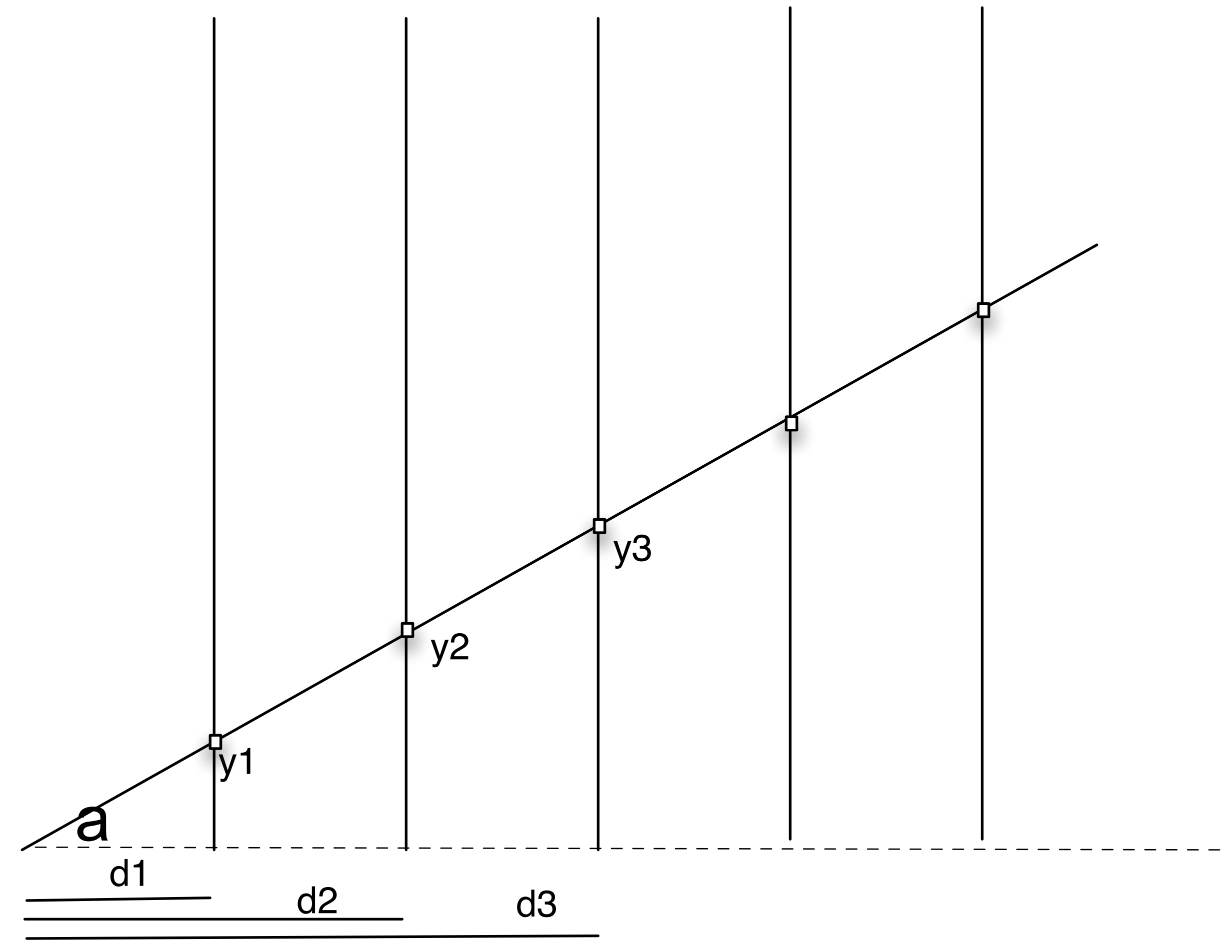
Input: elevation grid
Output: visibility grid, each point marked visible/invisible

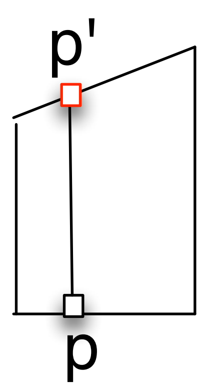
- For each p in grid
  - compute intersections between *vp* and grid lines
  - if all these points are below *vp* then *p* is visible



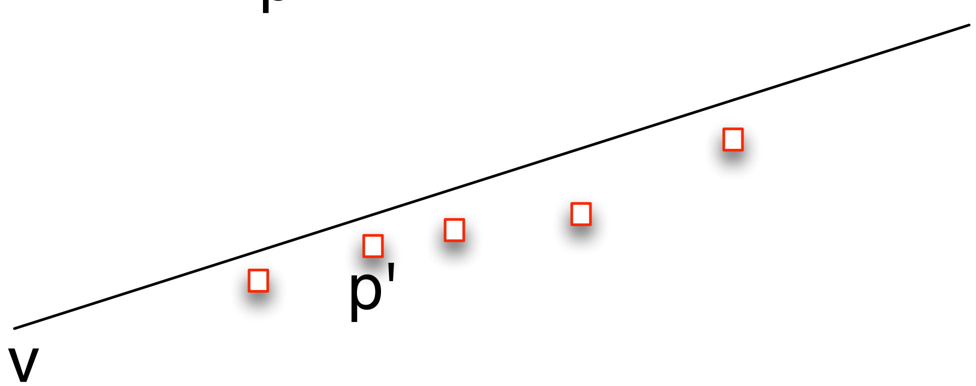
Assume grid of *n* points
($\sqrt{n}\ x\ \sqrt{n}$)
Running time: $O(n\sqrt{n})$







$$y1 = d1 \tan a$$
$$y2 = d2 \tan a$$

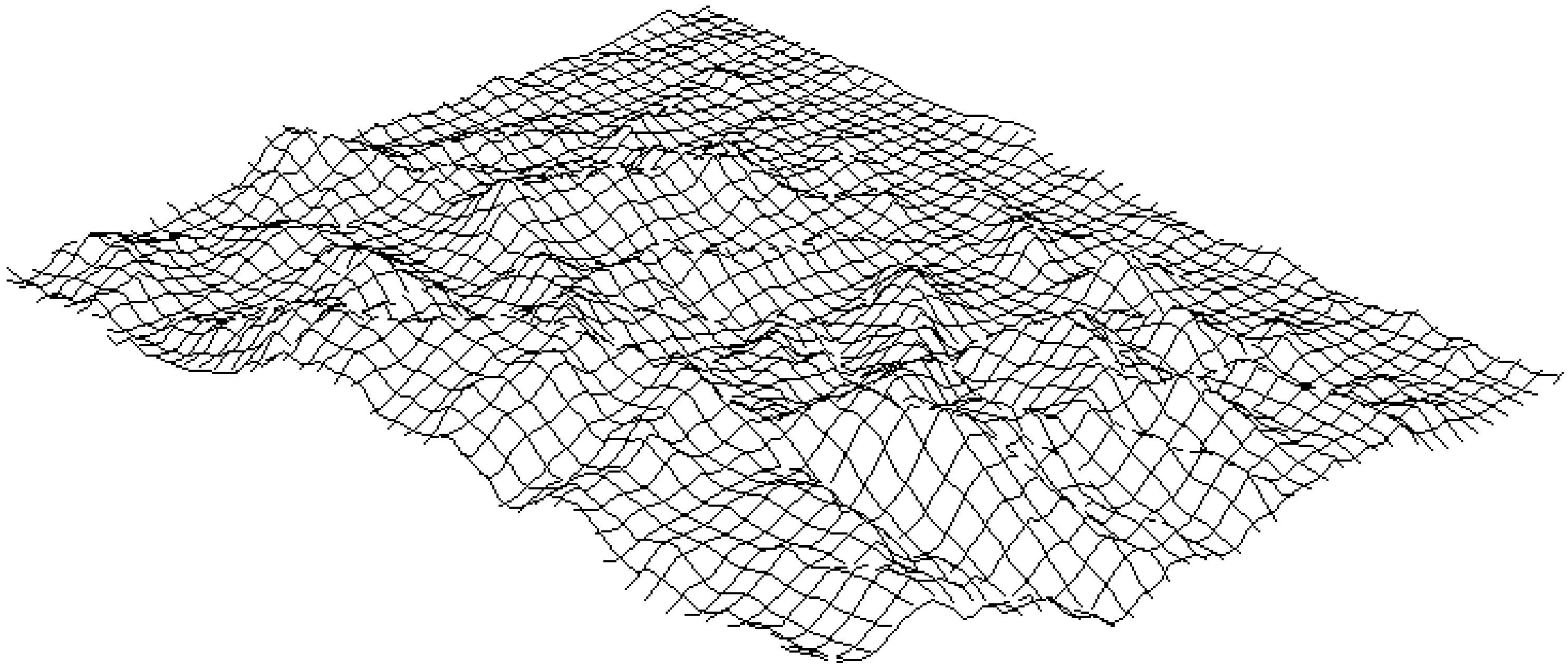


Viewshed on grids

Grid of n points:
sqrt n x sqrt n

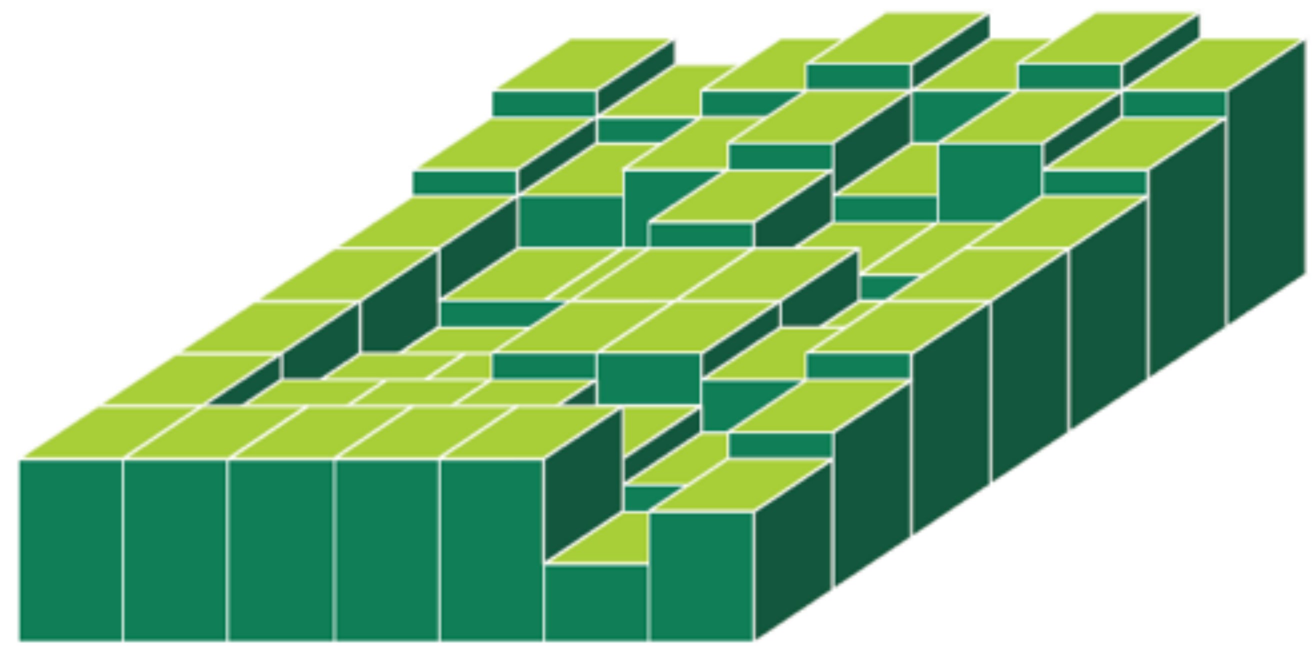
- The straightforward O(n sqrt n) algorithm
  - uses linear interpolation
  - “exact” as much as data allows
- Better?
- Van Kreveld, using different model
  - nearest neighbor interpolation
  - O (n lg n)

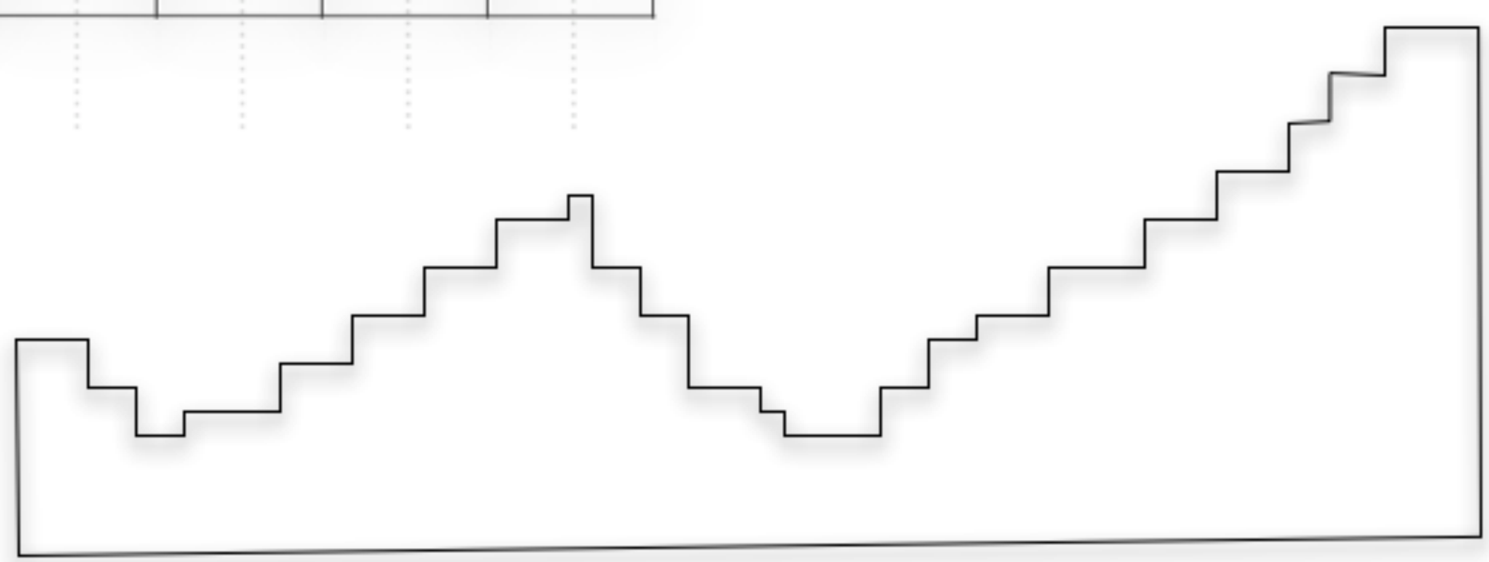
Grids with linear interpolation



Grids with nearest neighbor interpolation

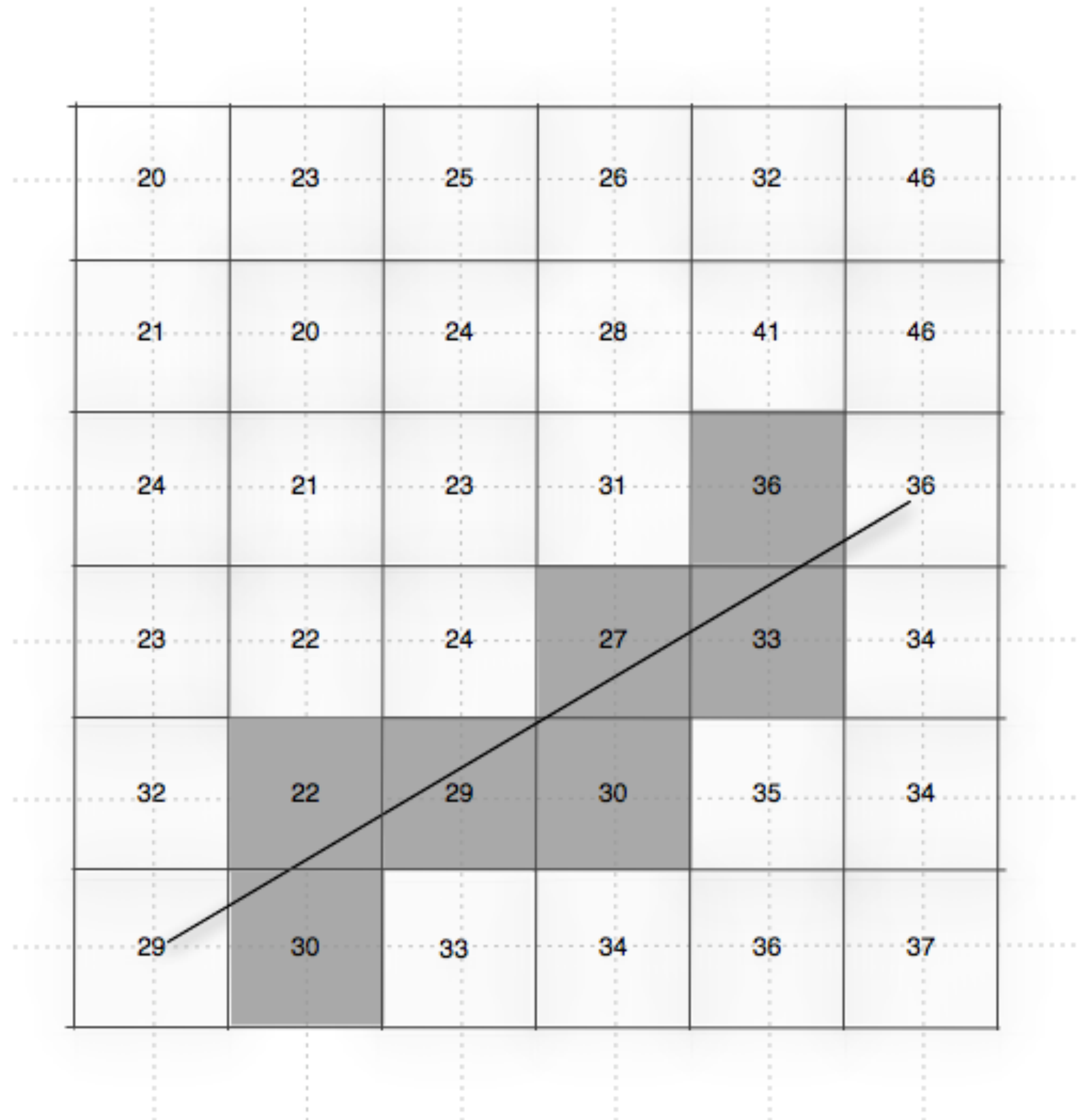
20	23	25	26	32	46
21	20	24	28	41	46
24	21	23	31	36	36
23	22	24	27	33	34
32	22	29	30	35	34
29	30	33	34	36	37

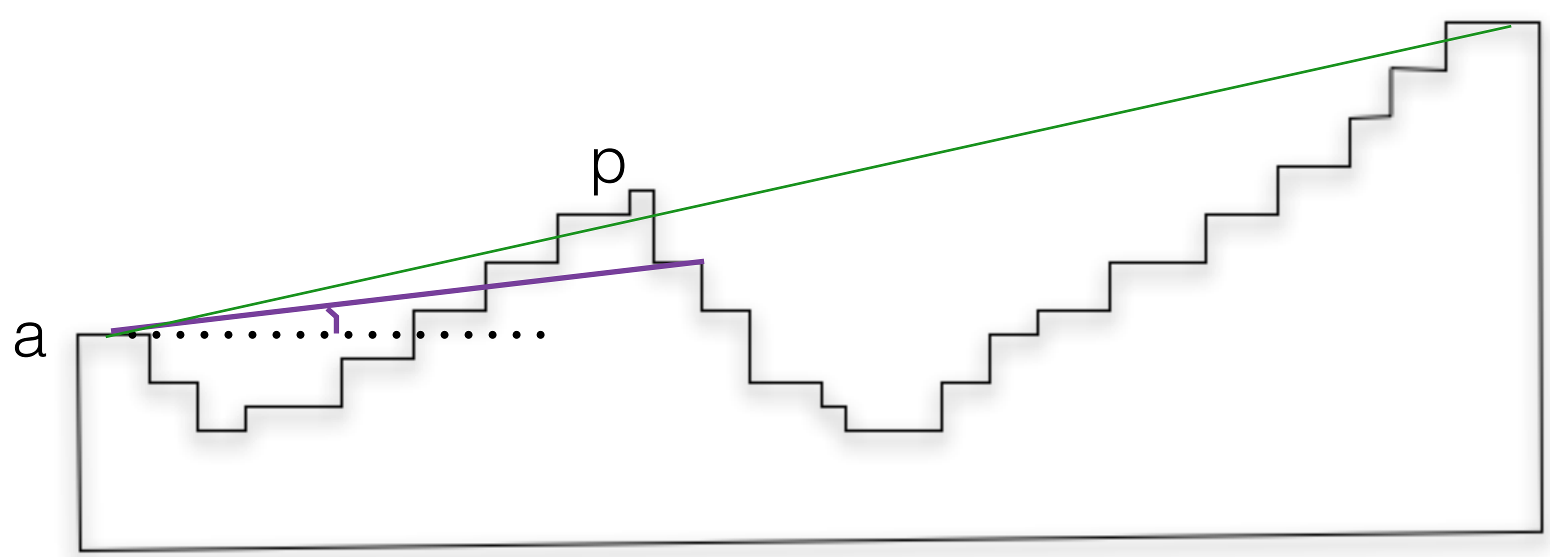




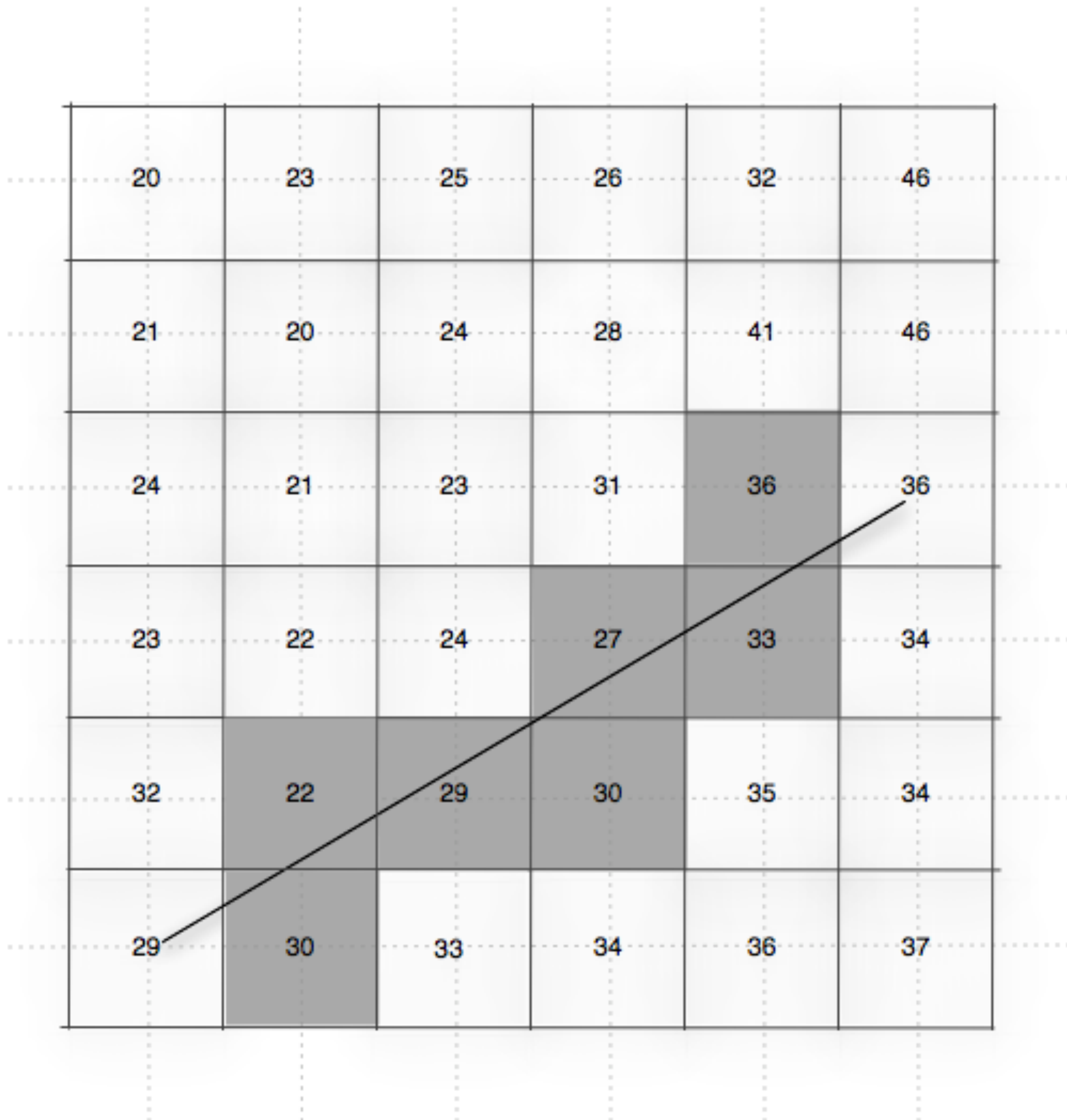
20	23	25	26	32	46
21	20	24	28	41	46
24	21	23	31	36	36
23	22	24	27	33	34
32	22	29	30	35	34
29	30	33	34	36	37

20	23	25	26	32	46
21	20	24	28	41	46
24	21	23	31	36	36
23	22	24	27	33	34
32	22	29	30	35	34
29	30	33	34	36	37

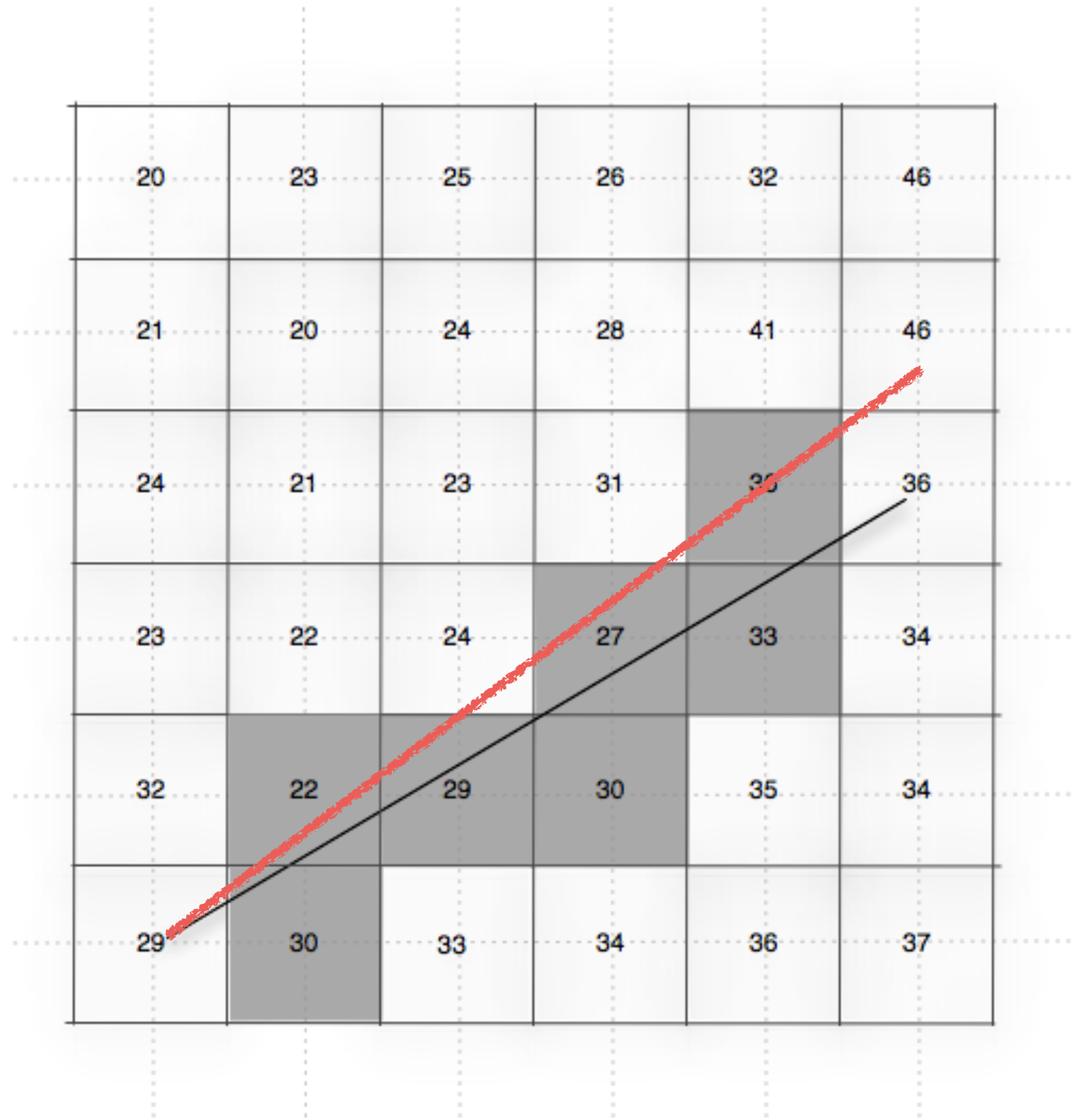

20 23 25 26 32 46
21 20 24 28 41 46
24 21 23 31 36 36
23 22 24 27 33 34
32 22 29 30 35 34
29 30 33 34 36 37

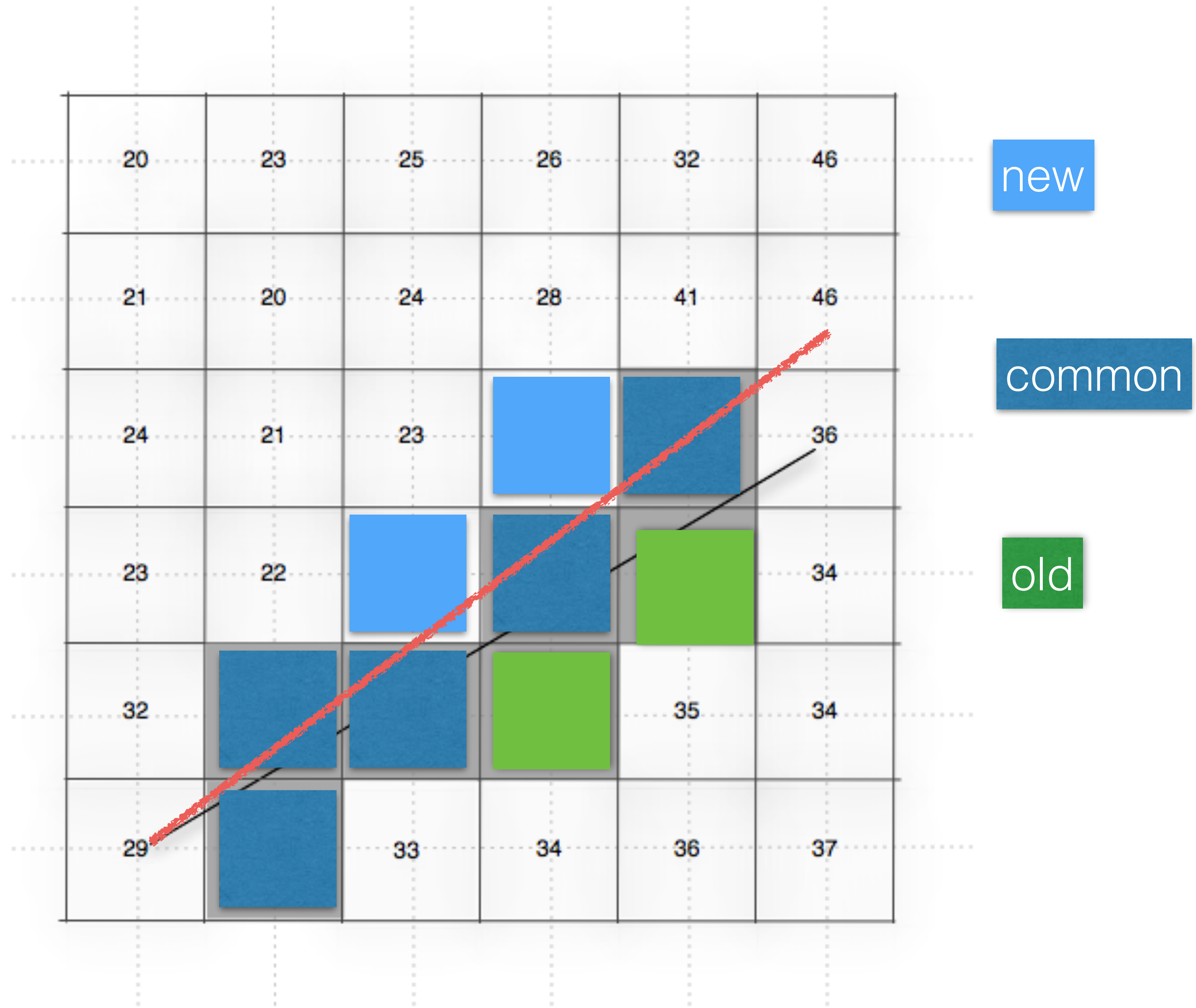


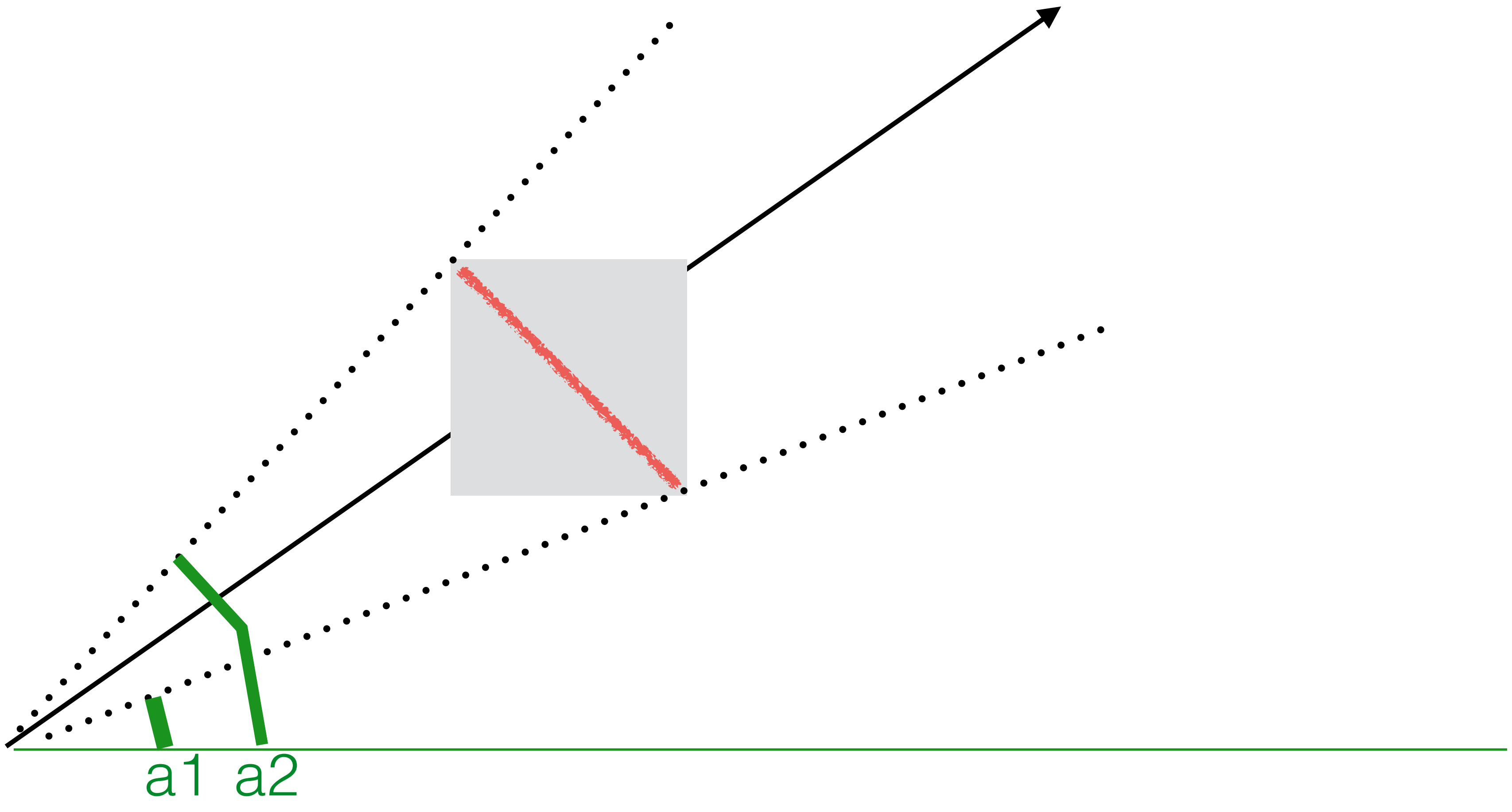
vertical slope(p,a) = (h_p - h_a) / d(a,p)

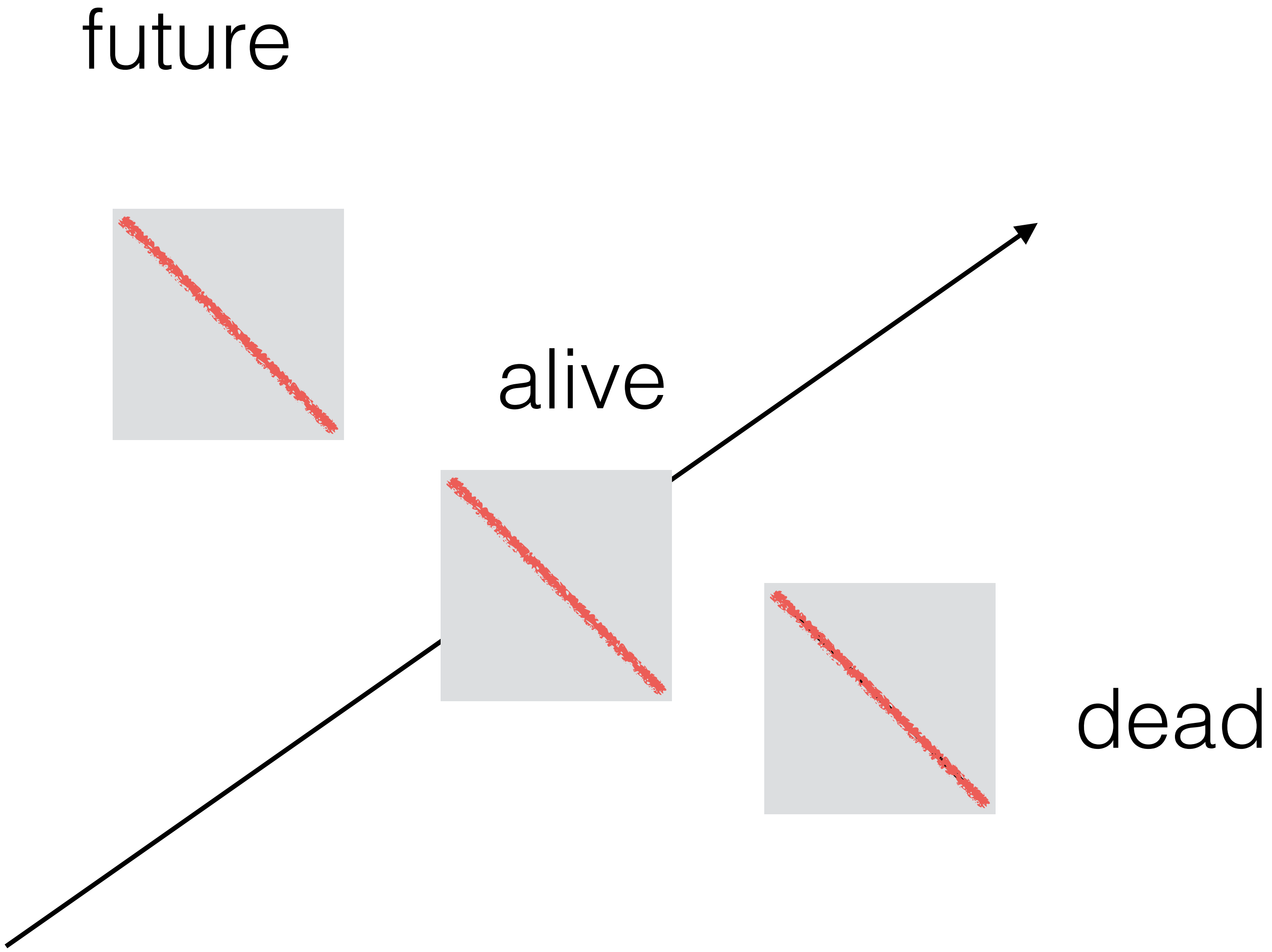


20 23 25 26 32 46
21 20 24 28 41 46
24 21 23 31 36 36
23 22 24 27 33 34
32 22 29 30 35 34
29 30 33 34 36 37

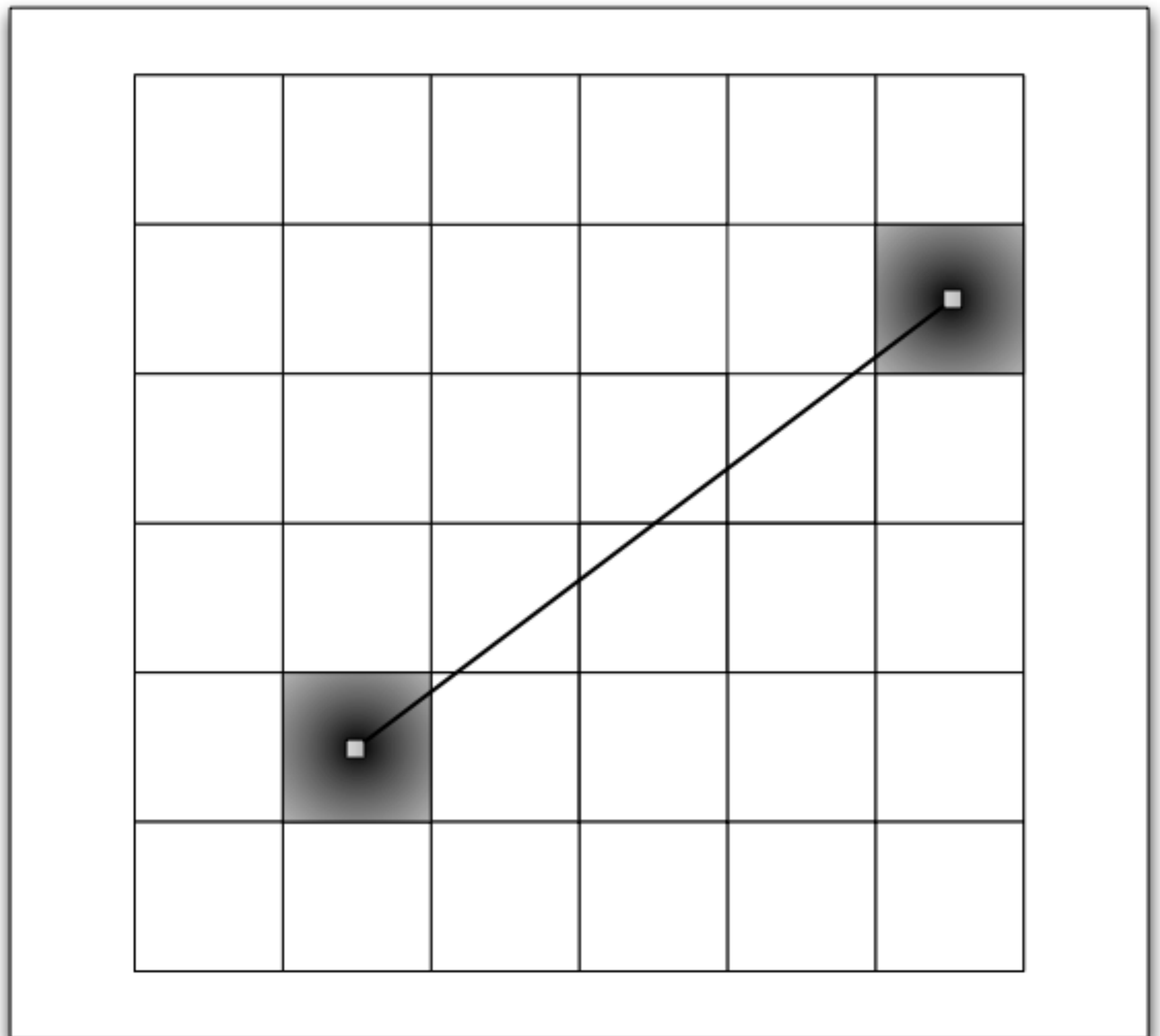

20 23 25 26 32 46
21 20 24 28 41 46
24 21 23 31 36
23 22 24 27 33 34
32 22 29 30 35 34
29 30 33 34 36 37


20 23 25 26 32 46
21 20 24 28 41 46
24 21 23 36
23 22 34
32 35 34
29 33 34 36 37
new
common
old

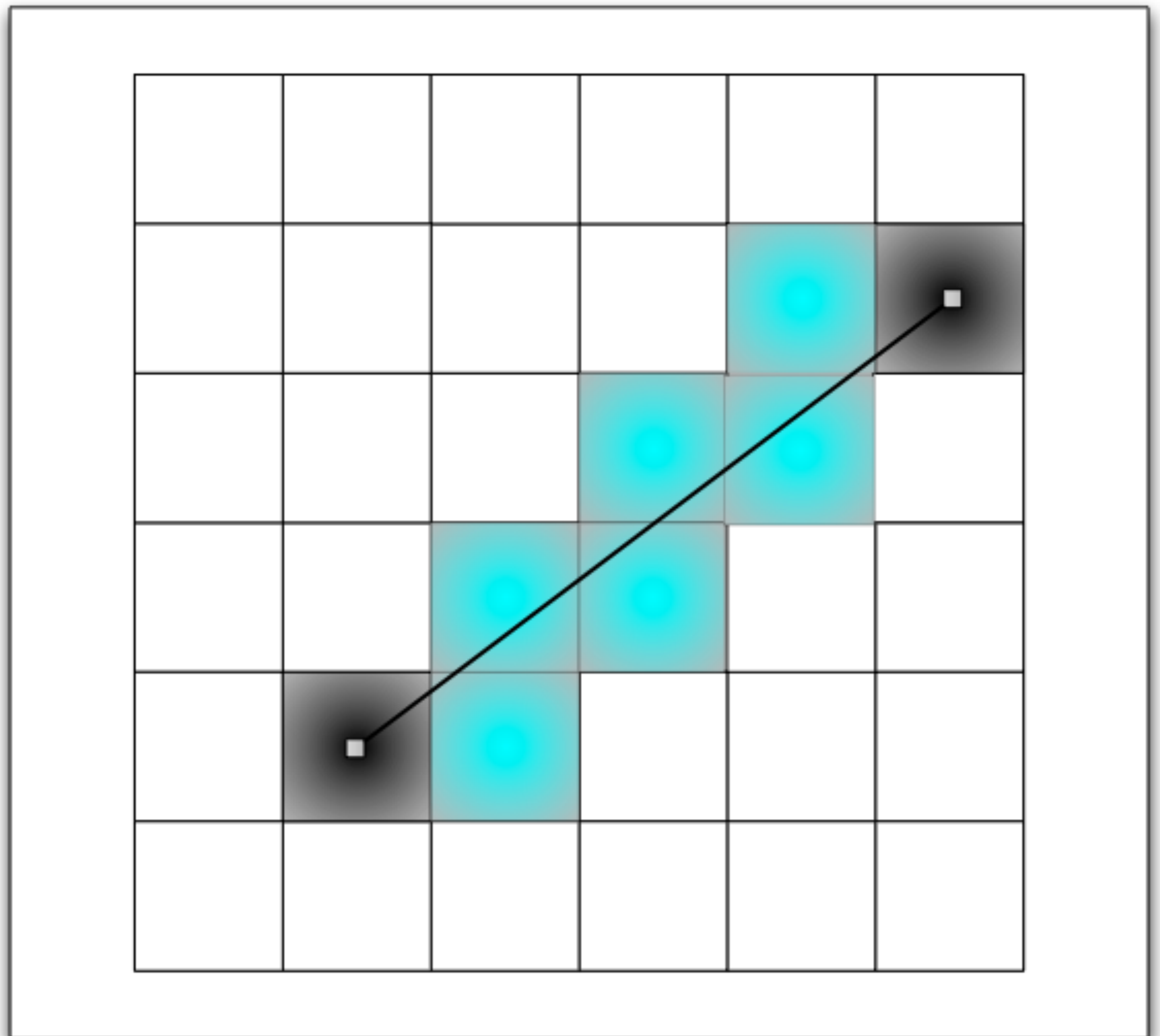

a1
a2


future
alive
dead

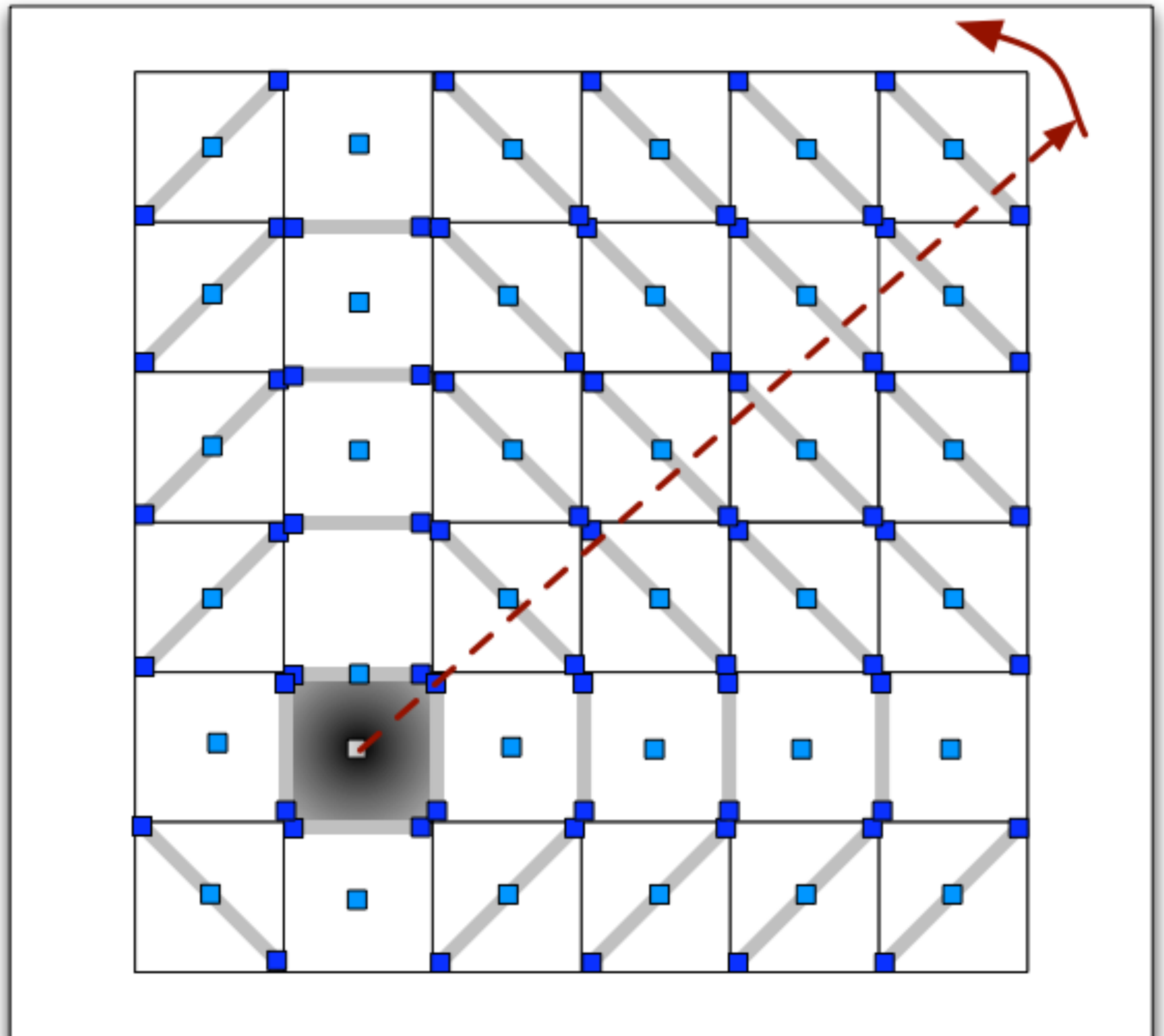
Van Kreveld's radial sweep algorithm



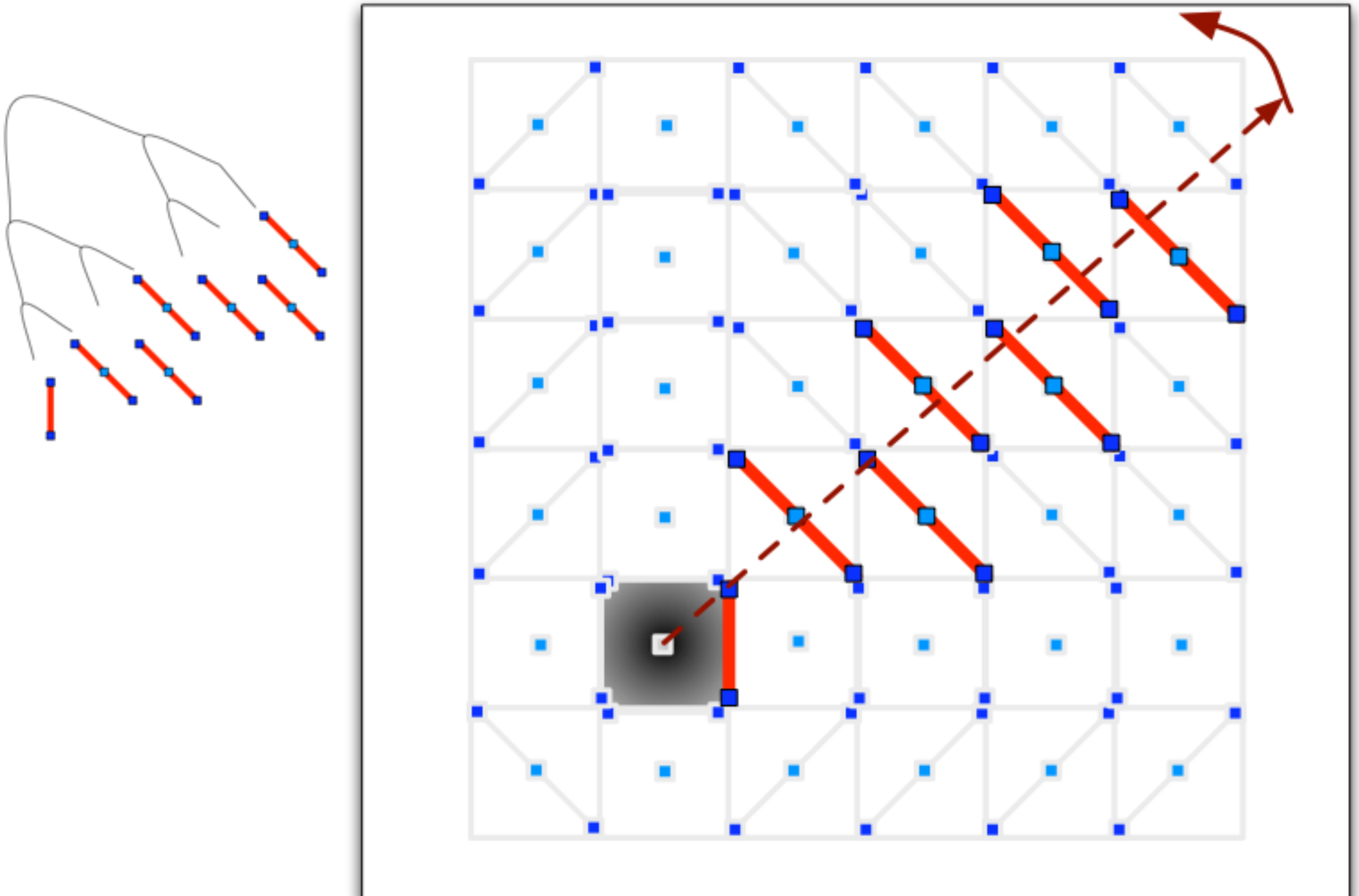
Van Kreveld’s radial sweep algorithm



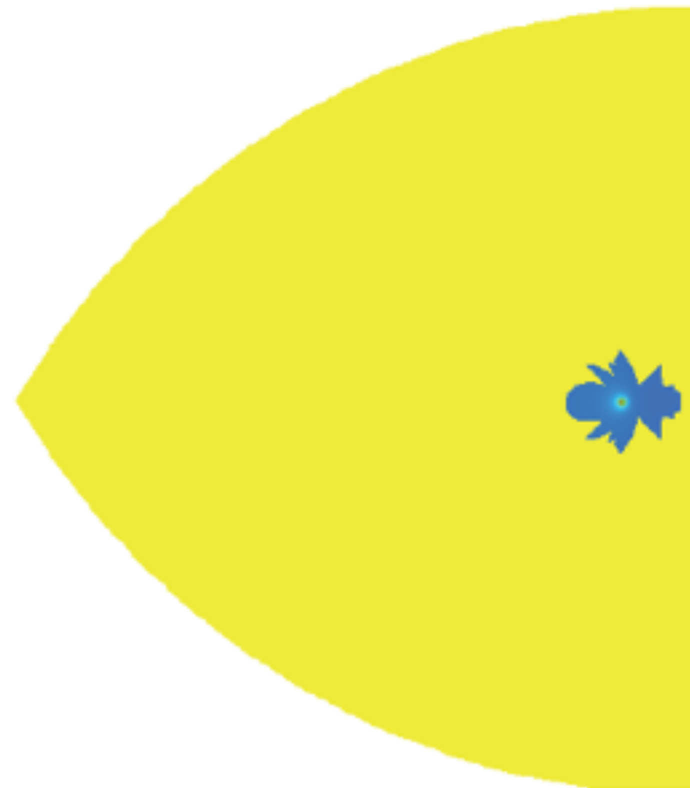
Van Kreveld's radial sweep algorithm



Van Kreveld's radial sweep algorithm

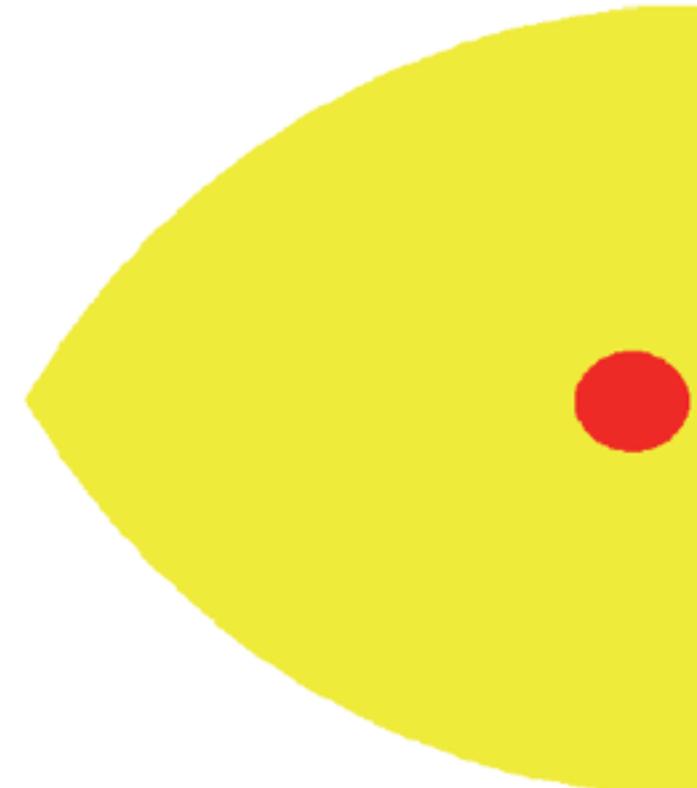


Accuracy!!



with ioradial from

Fishman et al 2009



with GRASS