Algorithms for GIS
csci3225

Laura Toma

Bowdoin College

Visibility on terrains

Are two points visible to each other?

What can one see from a given viewpoint?

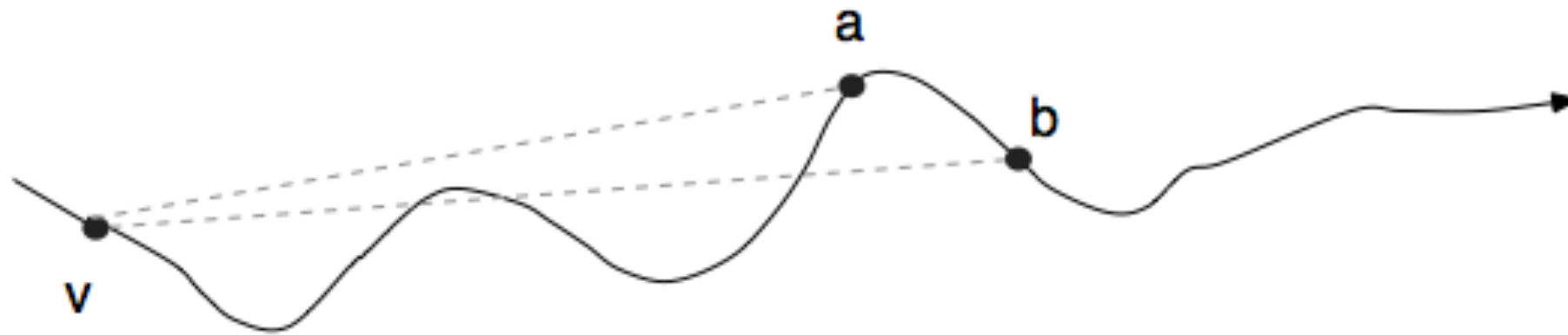What is the cumulative visible area from a set of viewpoints?

Visibility on terrains

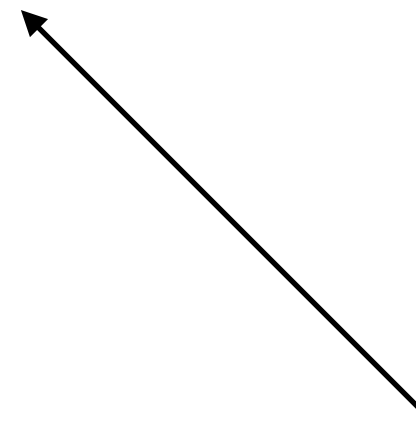What is the point with largest/smallest visibility?

Find a set of tower locations that "covers" the terrain

How to place an ugly pipe in a scenic area?
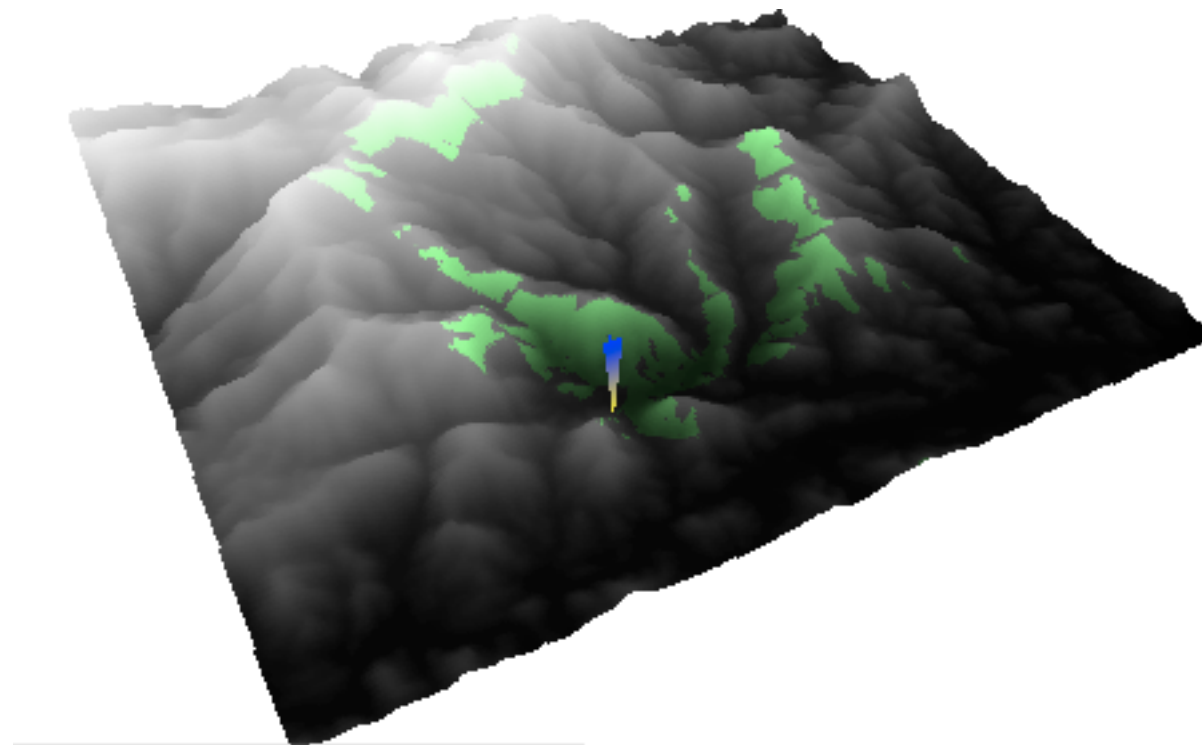
How to place a scenic highway?

Points u,v are visible if segment uv does not intersect the terrain

uv is called line-of-sight (LOS)

What can one see from a given point on a terrain?

- Terrain model T + viewpoint v

- Compute the **viewshed** of v: what part of T is visible from v

Sierra Nevada, 30m resolution

What Could Joshua Chamberlain See at Gettysburg?

What Could Joshua Chamberlain See at Gettysburg? is a project that re-examines the decisions made during the Battle of Little Round Top in 1863 using digital humanities methods. Chamberlain is well known for ordering a bayonet charge that helped the Union line secure Little Round Top. Until now, Chamberlain's line of sight during the battle has not been investigated using digital tools. Digital tools create data that supplements the pre-existing historical narrative of that fateful day. One of the main objectives is to create data visualizations that show us not only what Chamberlain could actually see from his vantage point, but also uncover the external influences and forgone decisions that affected the outcome at Little Round Top. The visualization software Gephi was used to create digital networks of Chamberlain's overall correspondence based on over 500 original letters. Gephi enabled us to repurpose the original letters to gain insight into the humanistic side of Chamberlain's decisions during the battle. GIS, a mapping software, was then used for georeferencing and geoprocessing. Georeferencing compares the accuracy of the historical maps Chamberlain and his line used to modern topo maps. Using geoprocessing tools, it was possible to create an image that shows Chamberlain's visibility at Little Round Top, which suggests that the southern region of Little Round Top was the least visible from Chamberlain's position. This limited visibility challenges Chamberlain's strategy and could determine if his success in the environment of Little Round Top can be attributed to good

S. Topouzi, S. Soetens, A. Gkiourou & A. Sarris
The Application of Viewshed Analysis in Greek Archaeological Landscape.

6th Annual Meeting of the European Association of Archaeologists, Lisbon, Portugal - 10th-17th September 2000.

The Application of Viewshed Analysis in Greek Archaeological Landscape.

S. Topouzi, S. Soetens, A. Gkiourou & A. Sarris

Laboratory of Geophysical-Satellite Remote Sensing and Archaeo-environment,
Institute for Mediterranean Studies - Foundation of Research & Technology, Hellas (F.O.R.T.H.),
Melissinou & Nik. Foka 130, PO. Box 119, Rethymnon 74100, Crete, Greece,
tel. ++30-831-25146, 56627, fax. ++30-831-25810

Abstract

Viewshed analysis has been extensively used in spatial modeling or landscape reconstruction of the ancient environmental settings and has been established as one of the most useful modules in the archaeological applications of Geographical Information Systems. The application of the technique has been proven especially effective in modeling spatial patterns within an uneven terrain.

Viewshed analysis was applied in the exploration of the defense system of the Mantineian plain in central Peloponnese, the study of the communication network among the historical towers of the island of Amorgos and the modeling of the location of the Minoan peak sanctuaries of the Lasithi district in NE Crete.

In the first case, the purpose of the analysis was to determine the visibility coverage of the plain surrounding the ancient city of Mantineia from watchtowers situated in the nearby hills and mountains and compare it with the corresponding visibility coverage resulting from each one of the gates of the city walls. The data were correlated to the results of previous surface surveys that had located ancient roads and paths. In this way, a thematic map of the potential defensive region has been created, suggesting either possible gaps or candidate archaeological sites that may have been used to complete the defense system of the region.
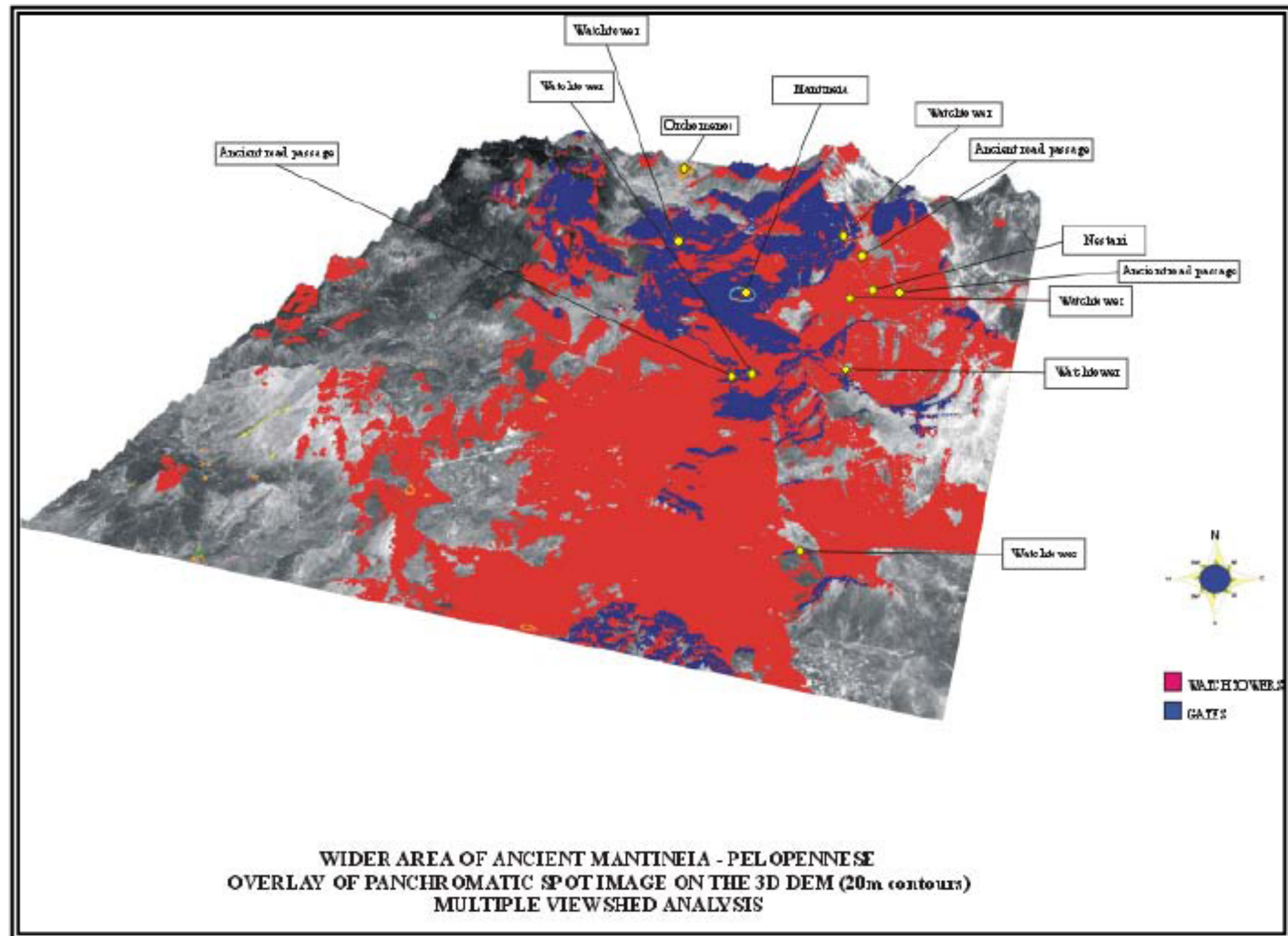
S. Topouzi, S. Soetens, A. Gkiourou & A. Sarris
The Application of Viewshed Analysis in Greek Archaeological Landscape.

6th Annual Meeting of the European Association of Archaeologists, Lisbon, Portugal - 10th-17th September 2000.

WIDER AREA OF ANCIENT MANTINEIA - PELOPENNESE
OVERLAY OF PANCHROMATIC SPOT IMAGE ON THE 3D DEM (20m contours)
MULTIPLE VIEWSHED ANALYSIS

Figure 1-3. Overlay of the multiple viewshed analysis of the defensive network of the Mantineia plain, the outline of archaeological features and the SPOT PAN imagery on the 3D DEM.

Problem: Given terrain model T + viewpoint v, compute the **viewshed of v**

↑

(what part of T is visible from v)

Terrain models

Rasters

TINs (vector)
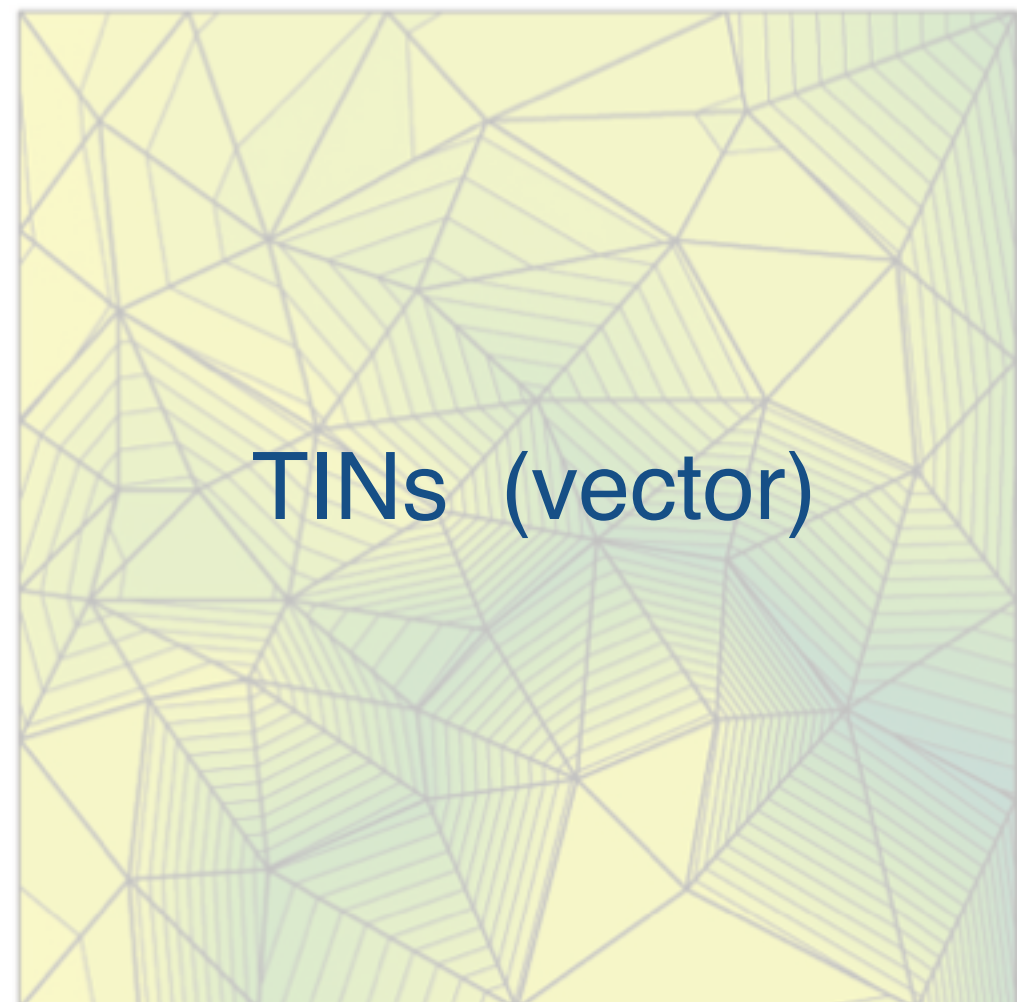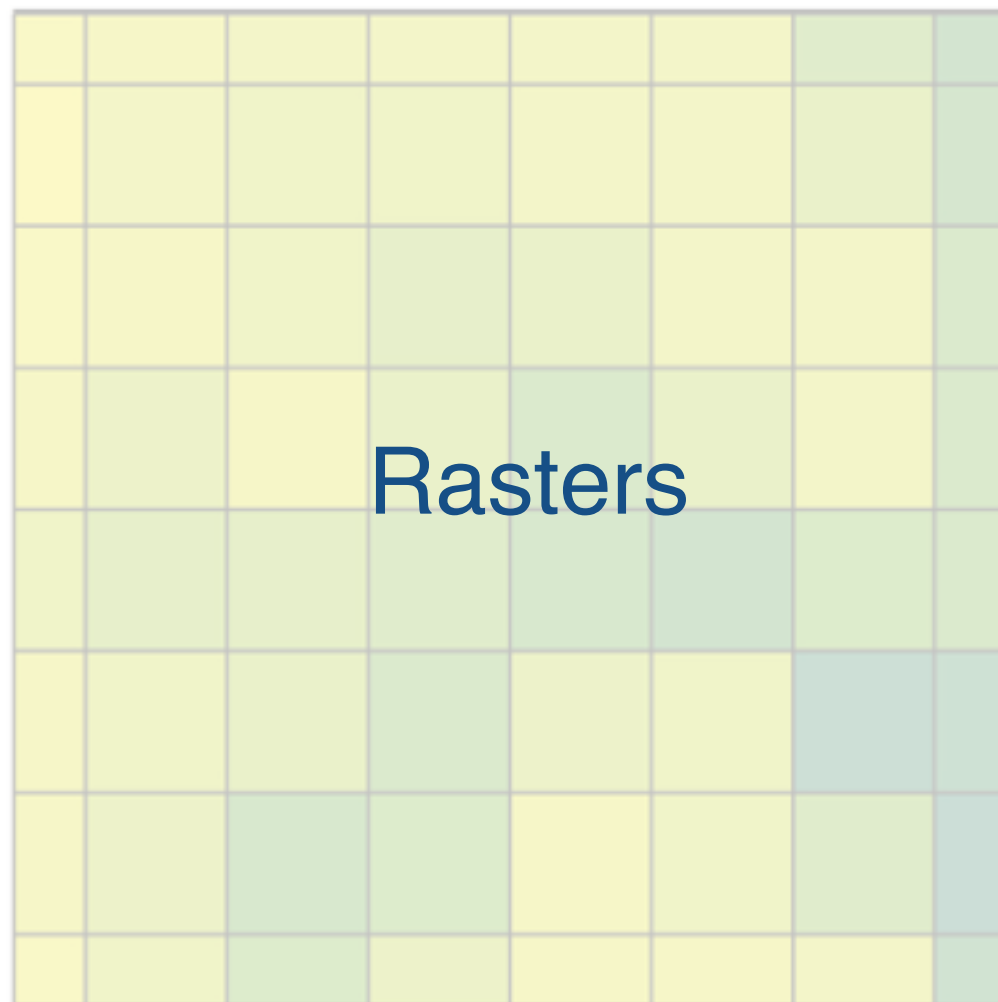
Viewsheds on grid terrains

Compute a **discrete** viewshed

- each grid point is marked visible/invisible ==> viewshed is a grid

- ignore the precise shape of the viewshed in between the grid points

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
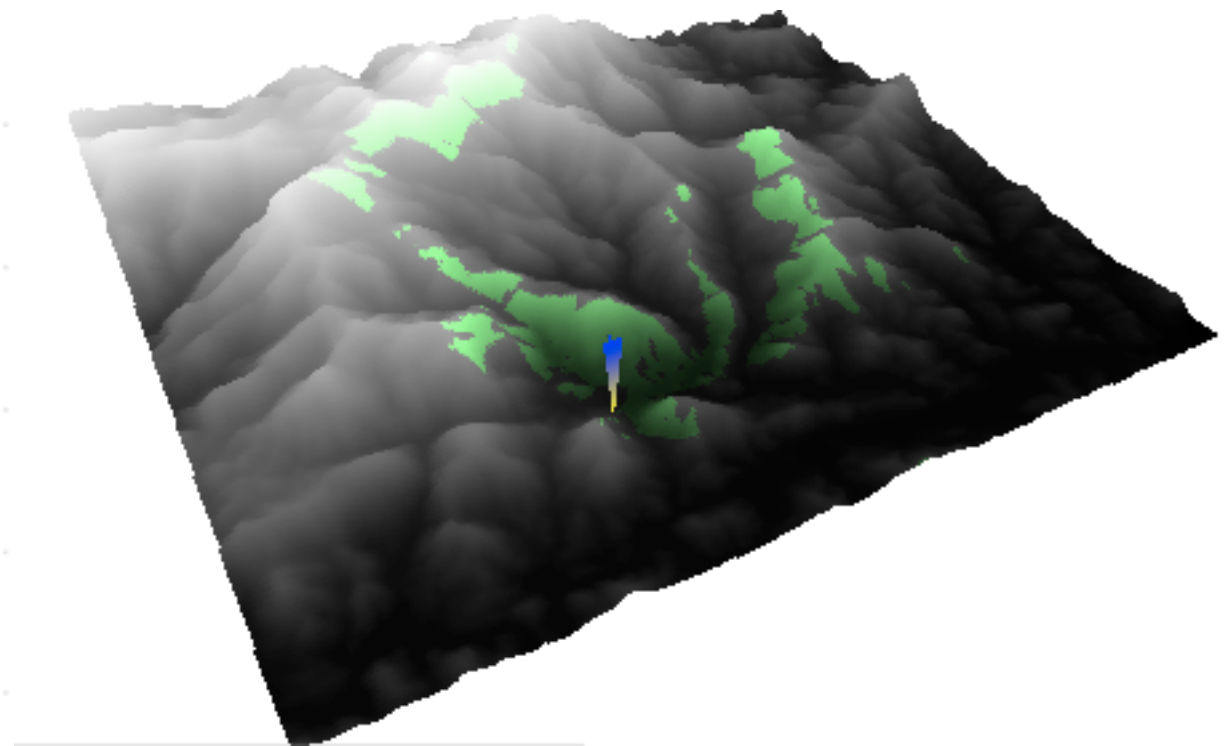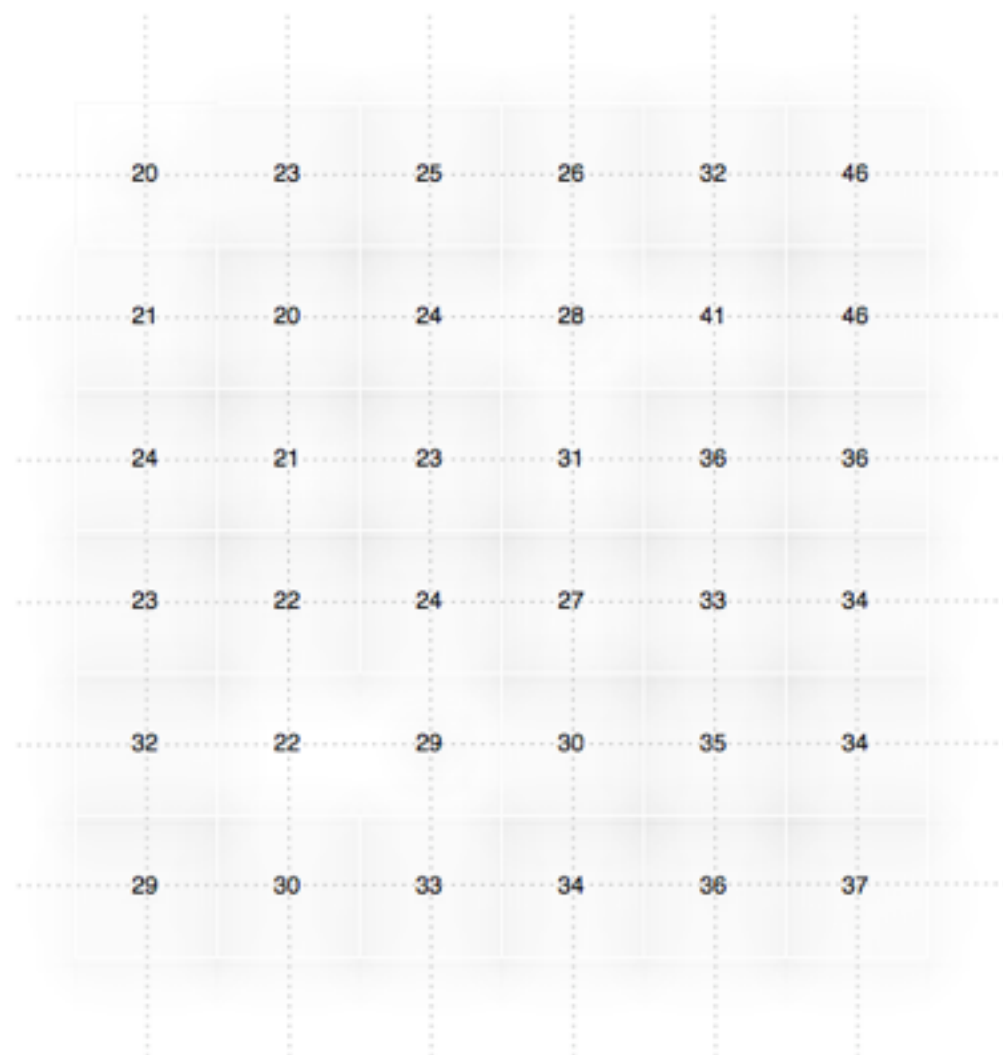
Sierra Nevada, 30m resolution

Viewsheds on TIN terrains

Compute a **continuous** viewshed

- compute the exact shape of the viewshed

- viewshed is a (set of) polygons
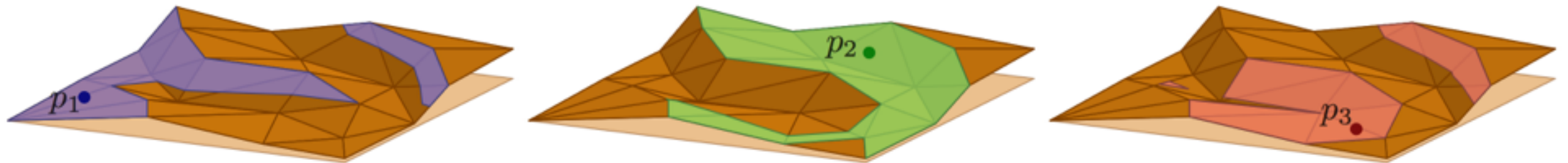
from: http://arxiv.org/pdf/1309.4323.pdf

Outline

- Viewsheds on grid terrains

  1. straightforward algorithm

  2. radial sweep algorithm

  3. viewshed via horizon

- Viewsheds on TIN terrains

  - quadratic size, worst-case construction

Viewsheds on Grids:

1. Basic (naive) algorithm

Are v,p visible?
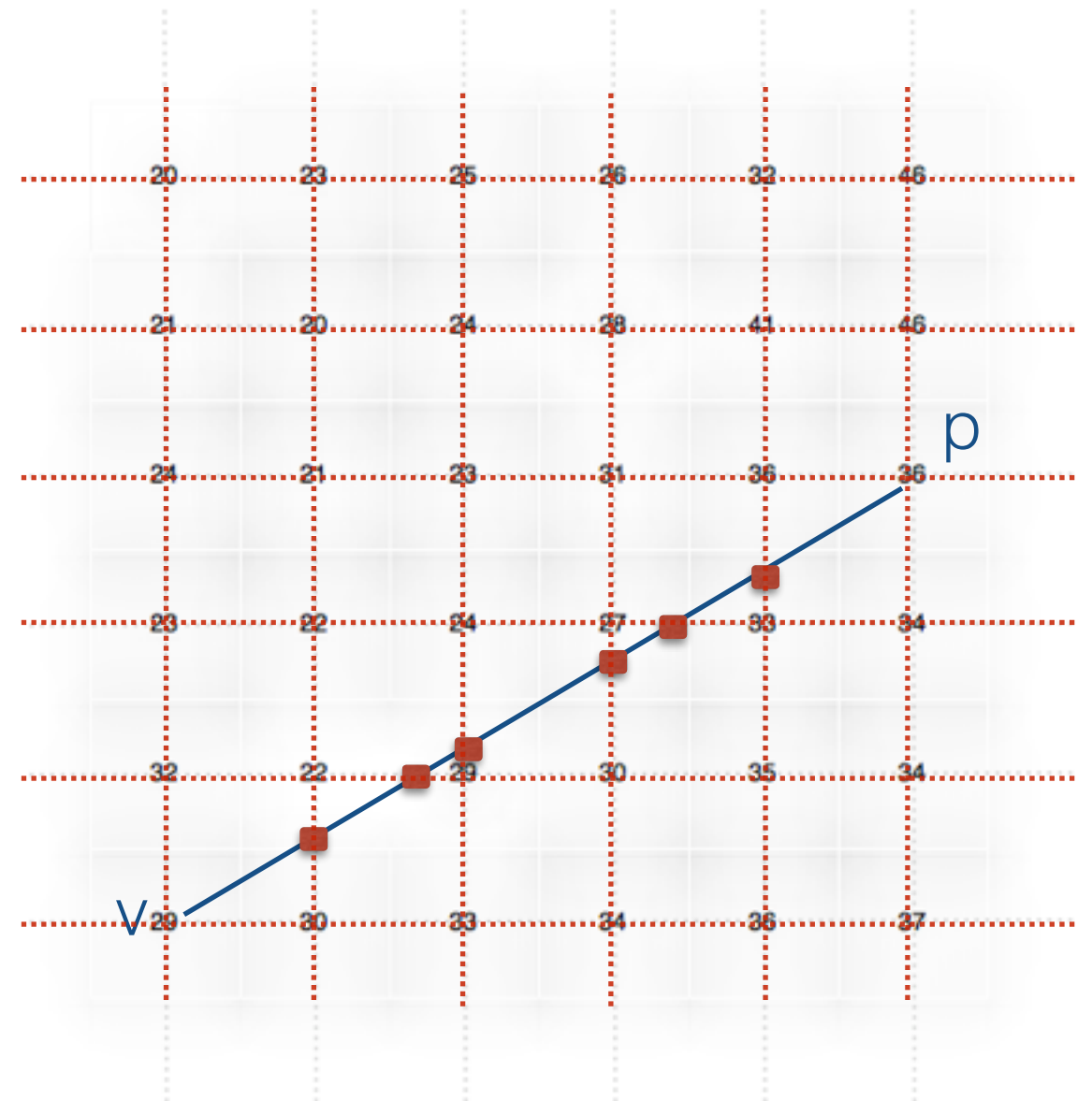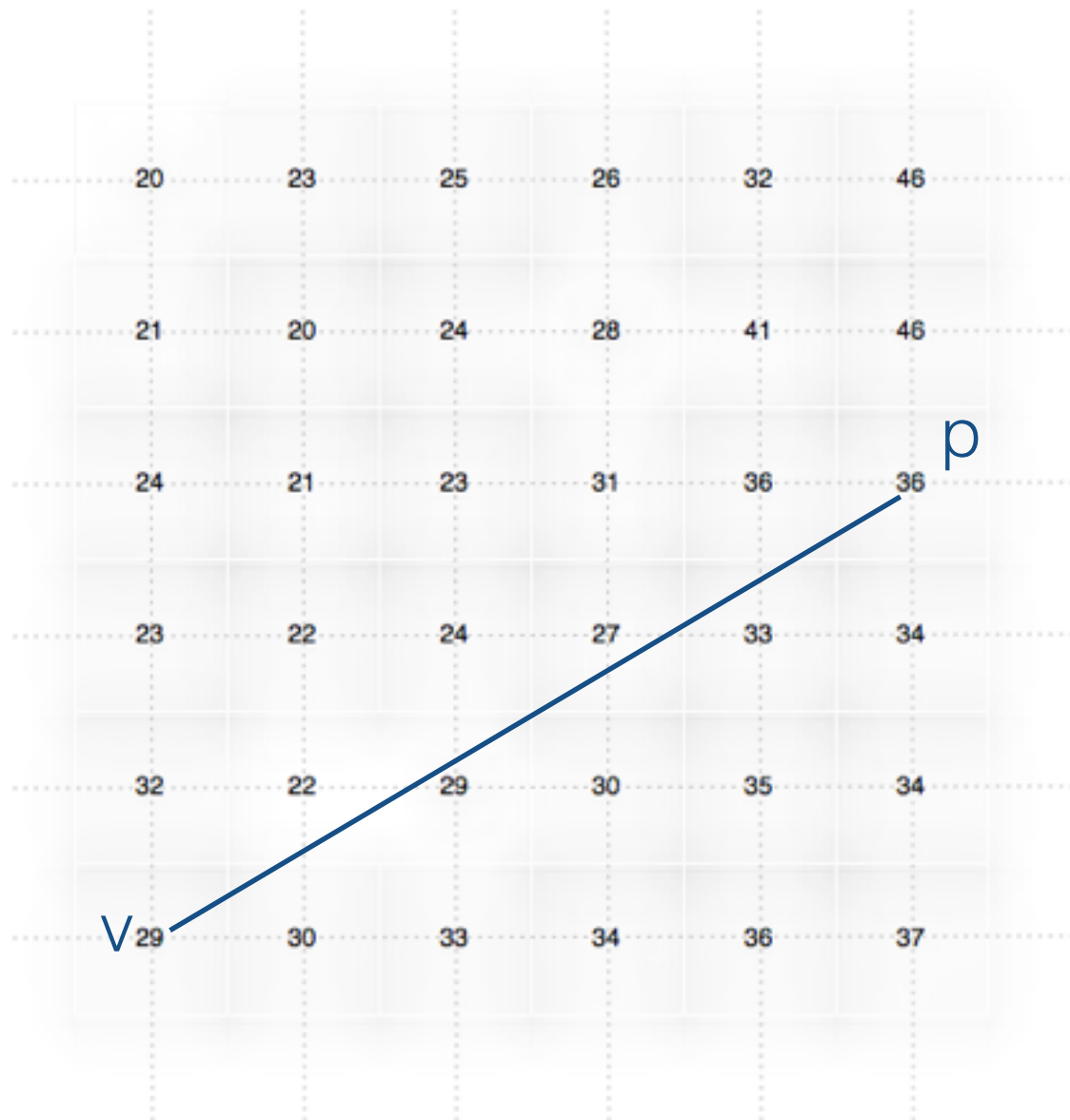
What is the height of the terrain along the LOS?

Use linear interpolation.

Are v,p visible?
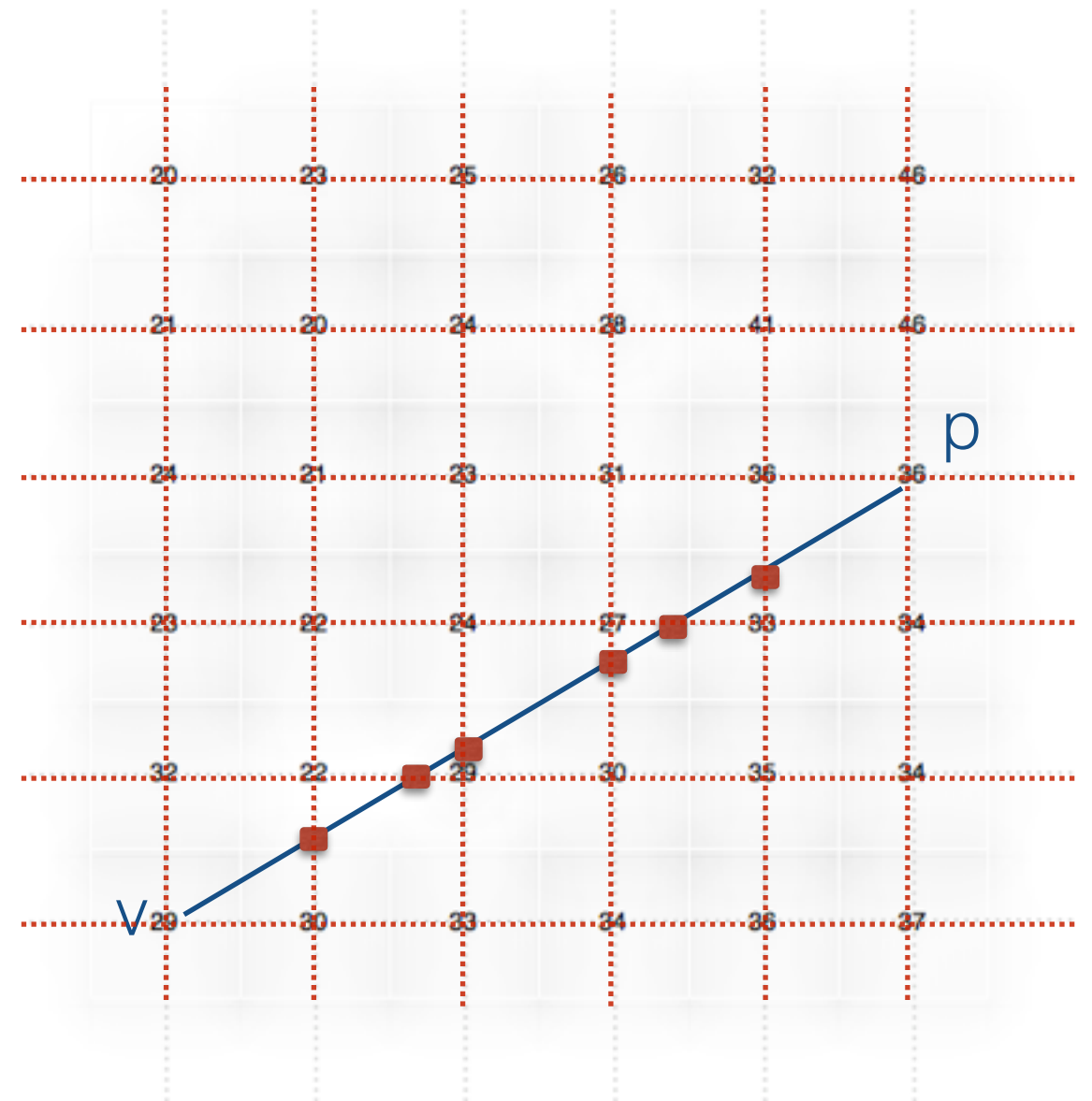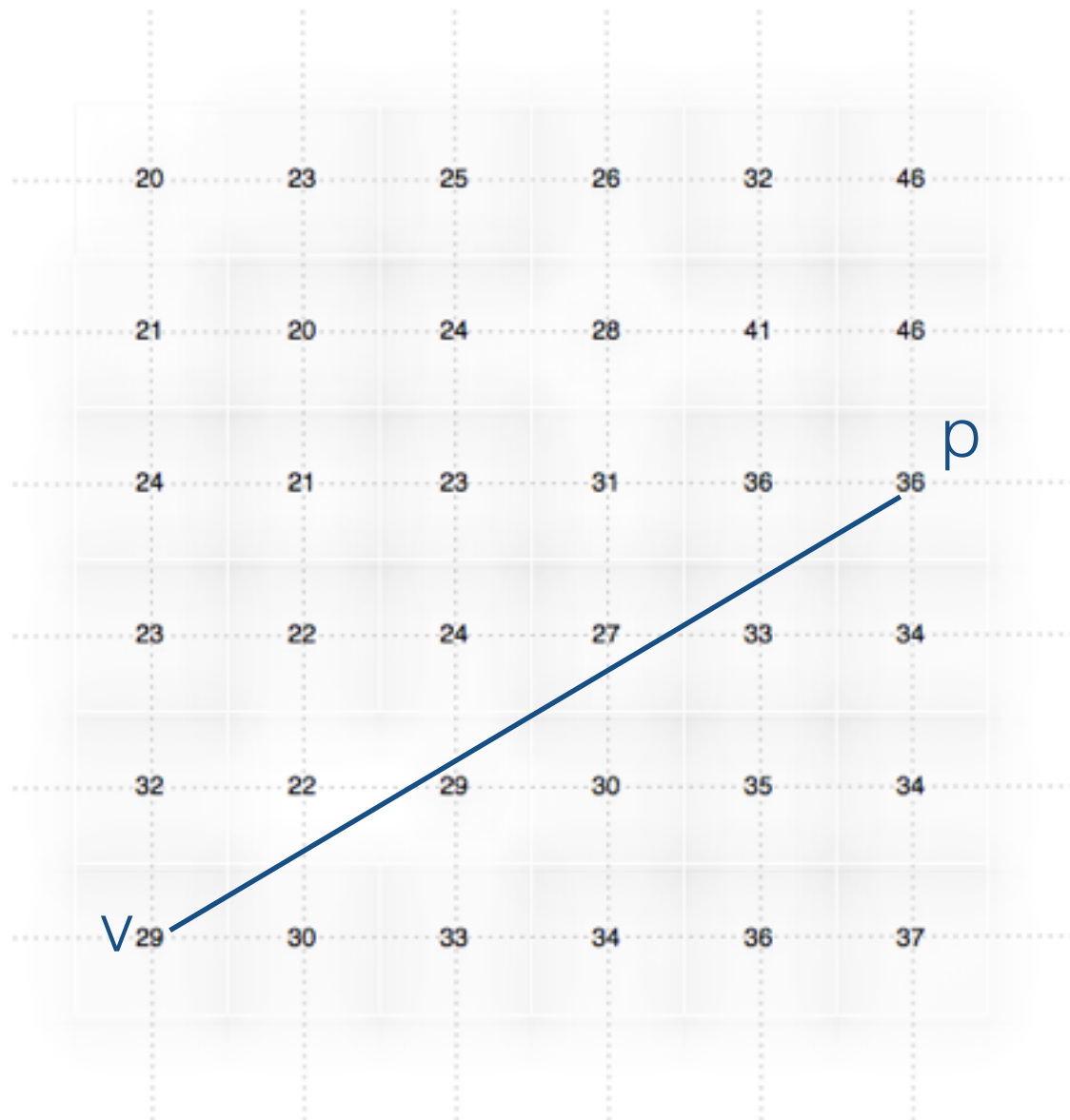
20	23	25	26	32	46
21	20	24	28	41	46
24	21	23	31	36	36
23	22	24	27	33	34
32	22	29	30	35	34
29	30	33	34	36	37

p

v

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

p

v

1. Compute intersections (in xy-plane) of LOS with the grid lines

2. Lift to 3D: Interpolate their elevations linearly

Are v,p visible?

y1 = d1 tan a
y2 = d2 tan a

1. Compute intersections (in xy-plane) of LOS with the grid lines
2. Lift to 3D: Interpolate their elevations linearly

Are v,p visible?

When is p visible?
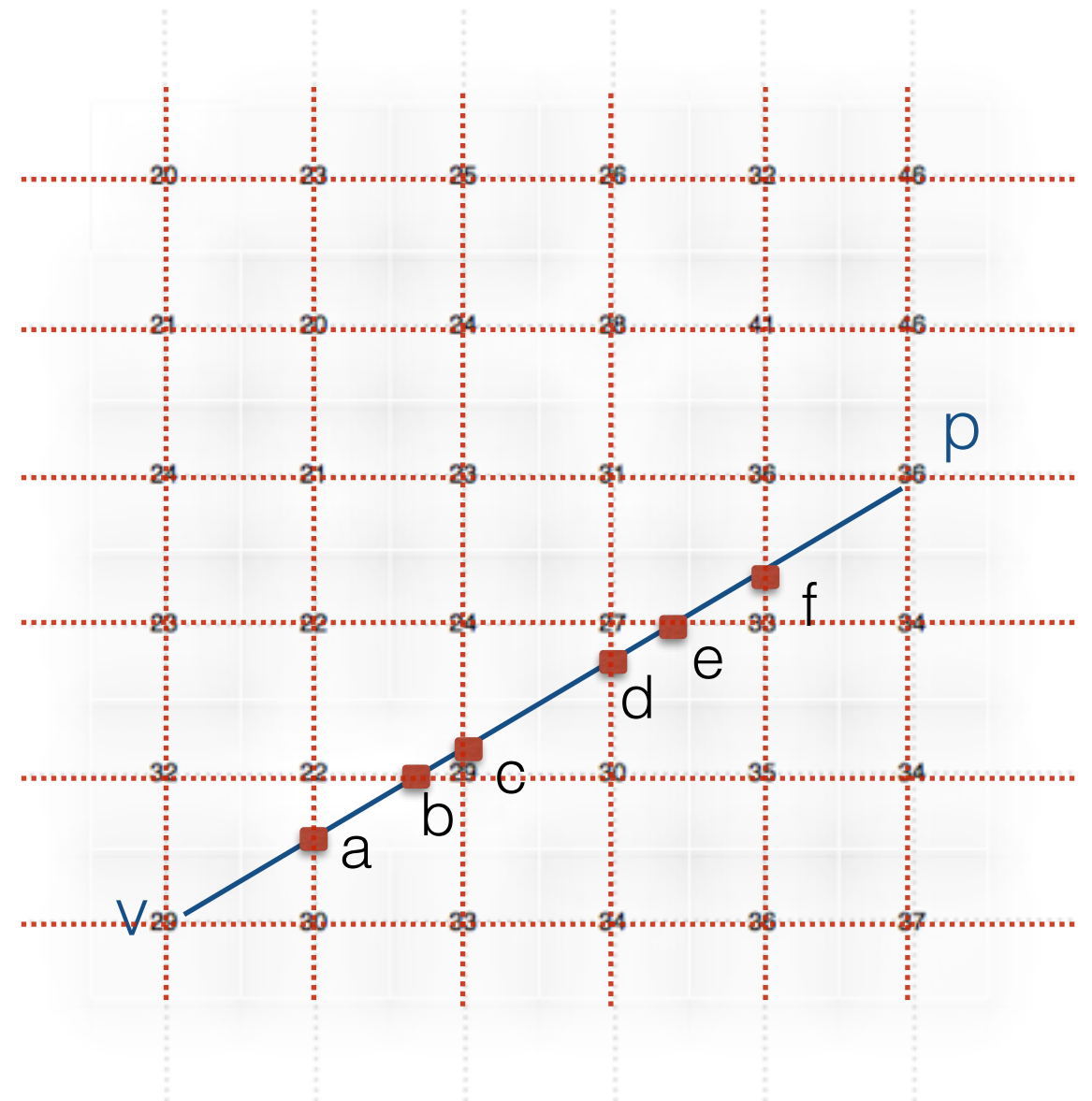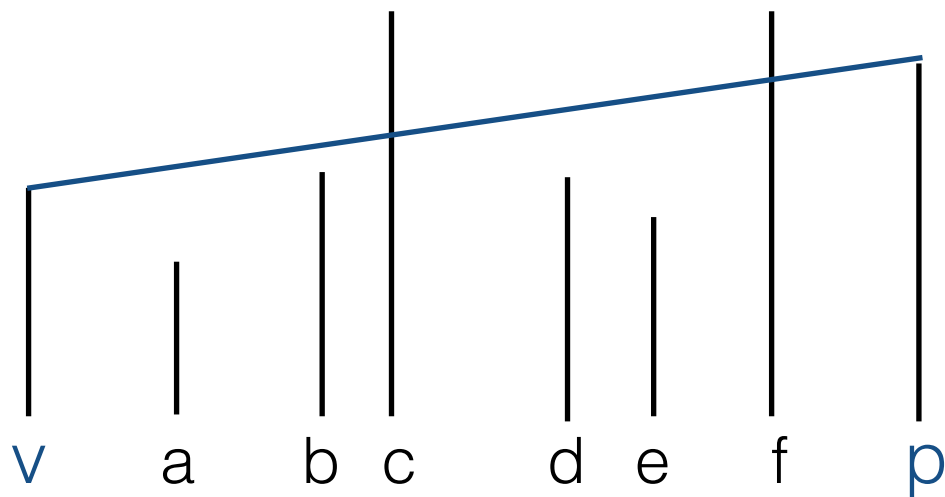
1. Compute intersections (in xy-plane) of LOS with the grid lines
2. Lift to 3D: Interpolate their elevations linearly

Vertical angle of a point p with respect to v

$$verticalAngle_v(p) = atan\ (h_p - h_v) / d(v,p)$$

LOS

v

p
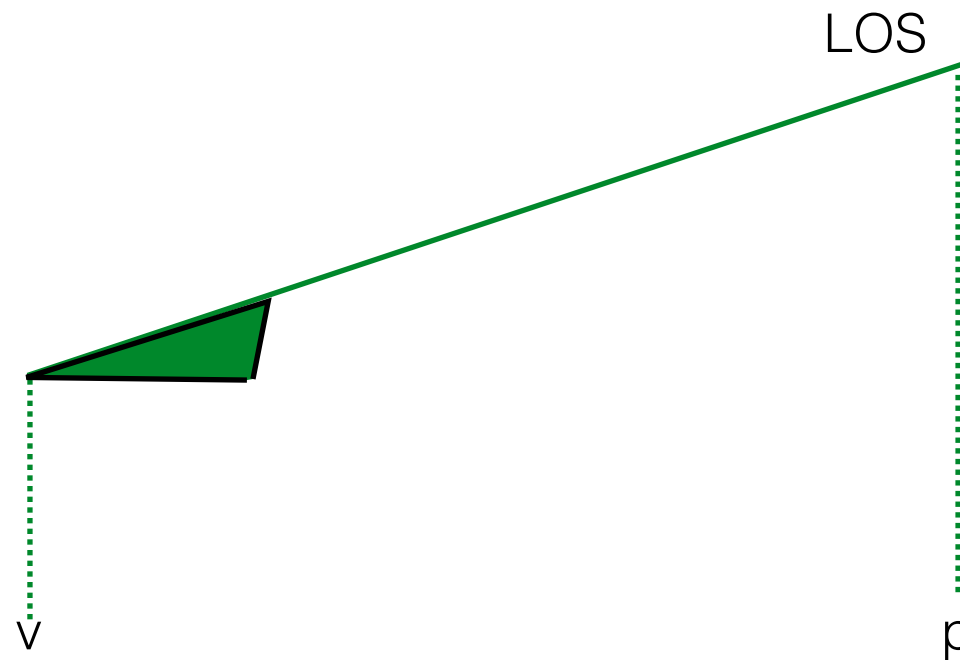
How tall p appears from v

Are v,p visible?

1. Find the intersections (in the xy-plane) between LOS and the grid lines

2. Lift to 3D: find their elevations by linear interpolation

3. If **all** verticalAngle$_v$(a) are below verticalAngle$_v$(p) ==> p visible
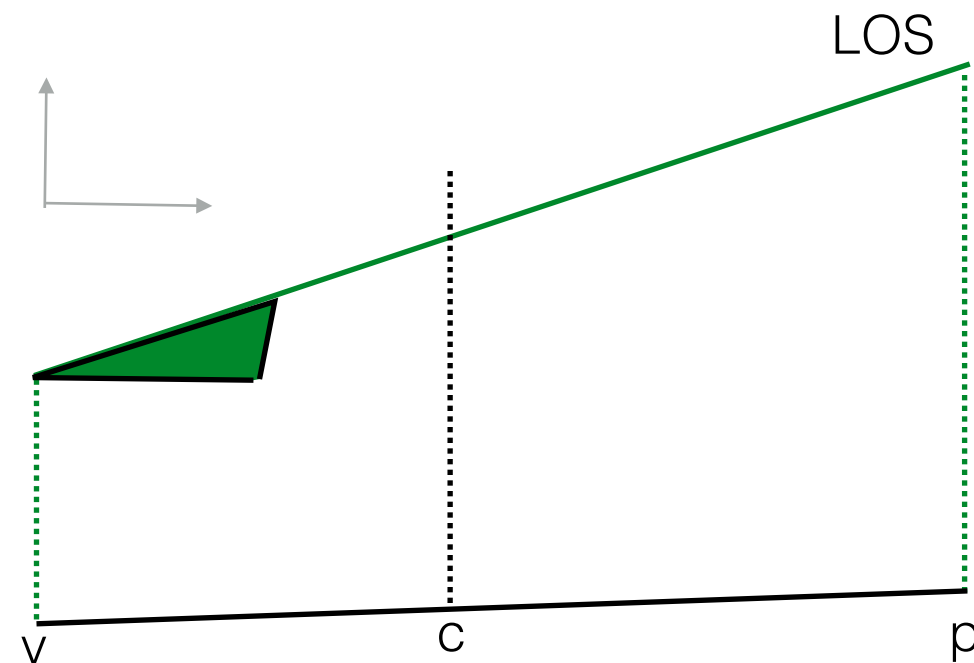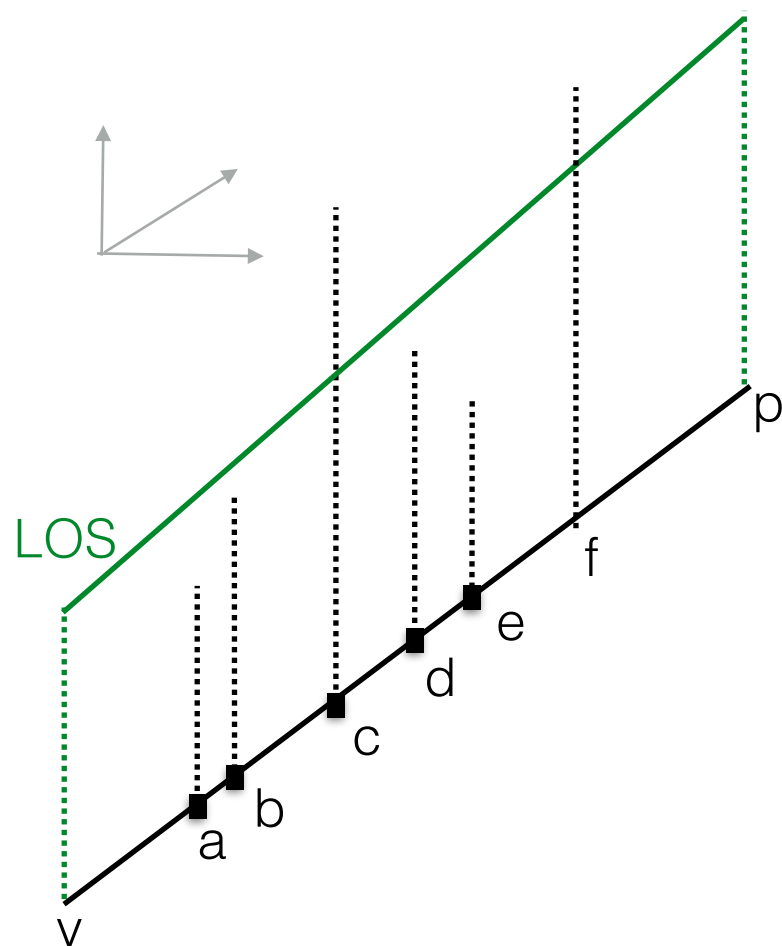
This takes $O(\sqrt{n})$ per point in the worst case

Basic viewshed algorithm

- For every point p=(i,j) in the grid

  - Find if p is visible from v

- Analysis: O(n$\sqrt{n}$)

- Uses linear interpolation

Linear interpolation
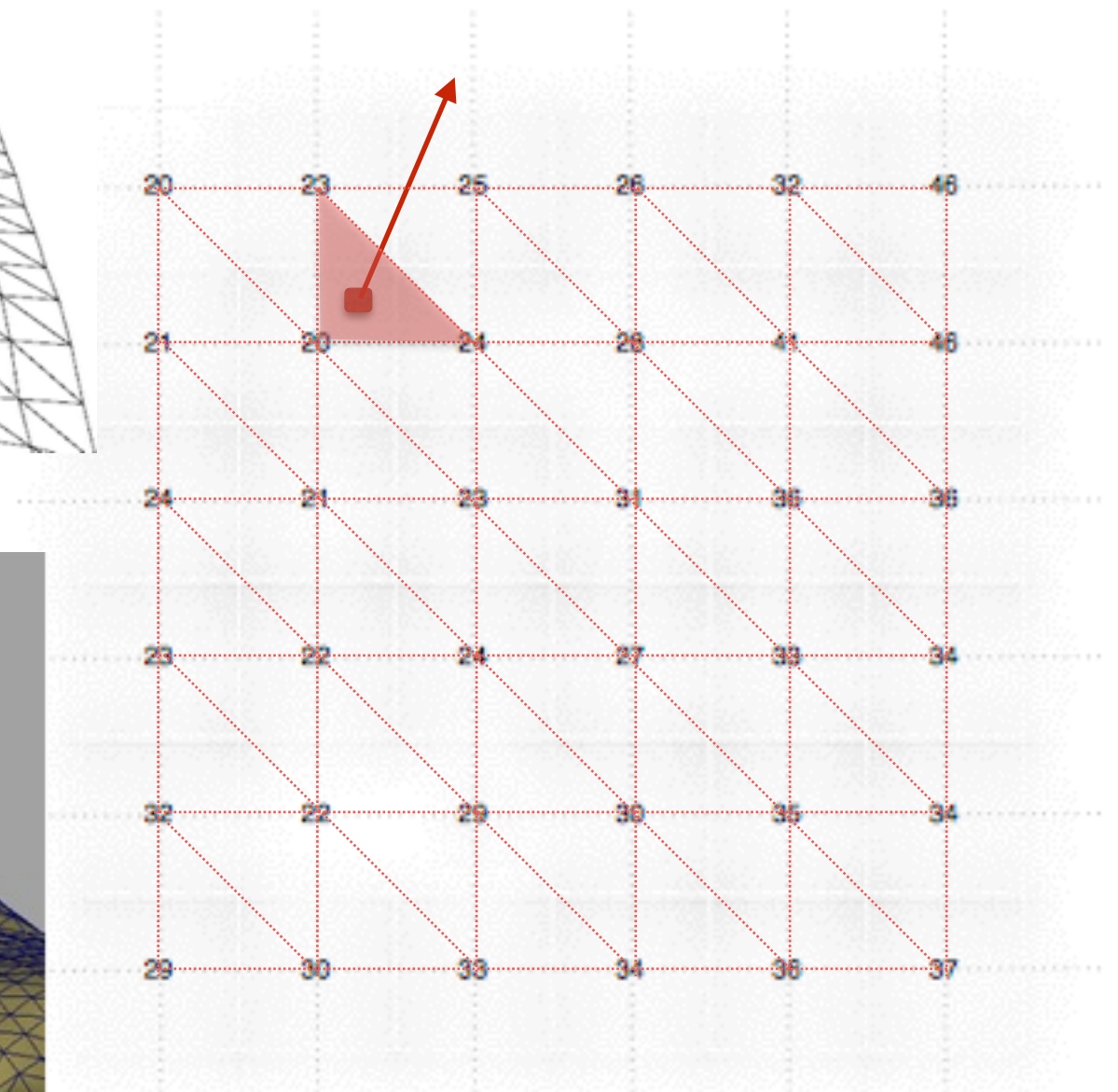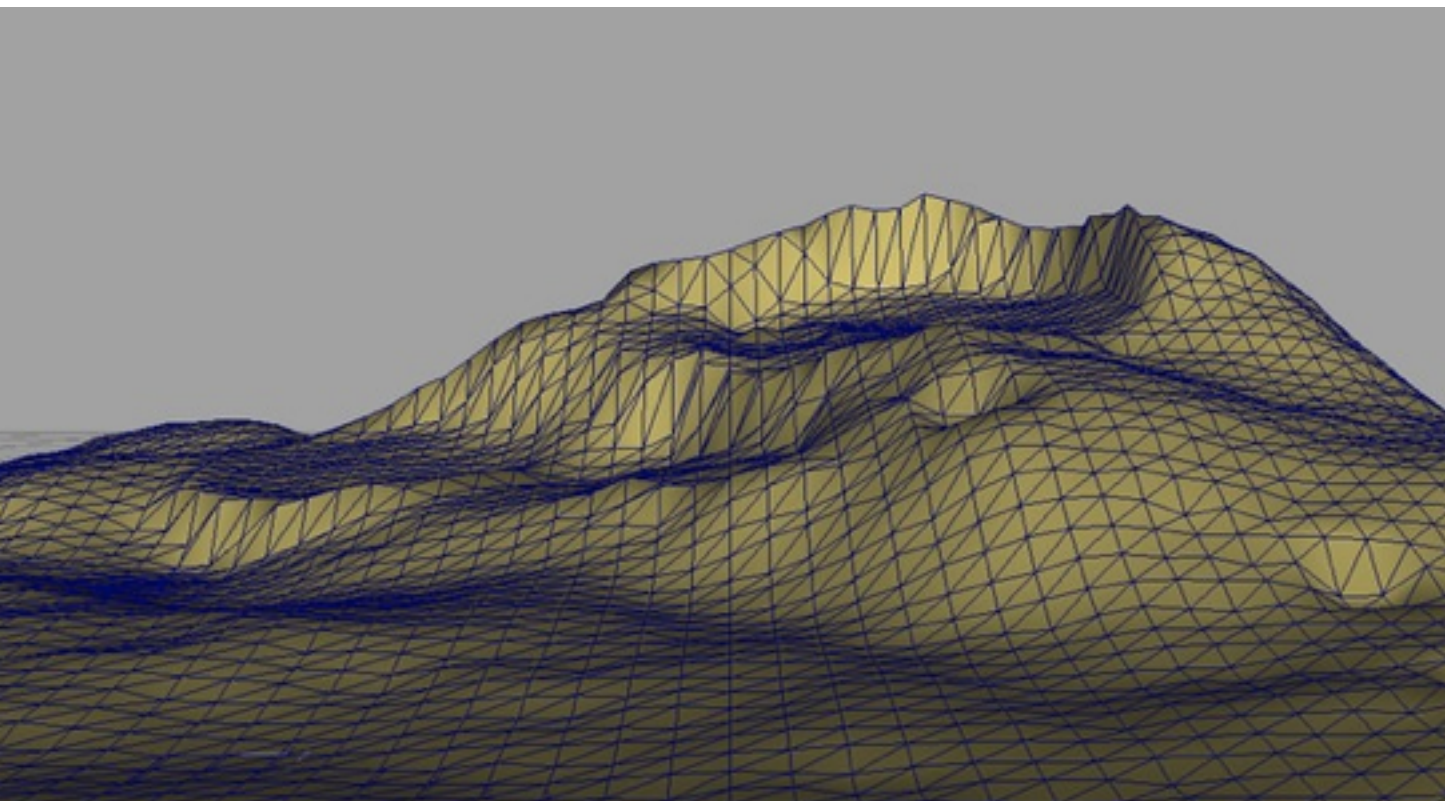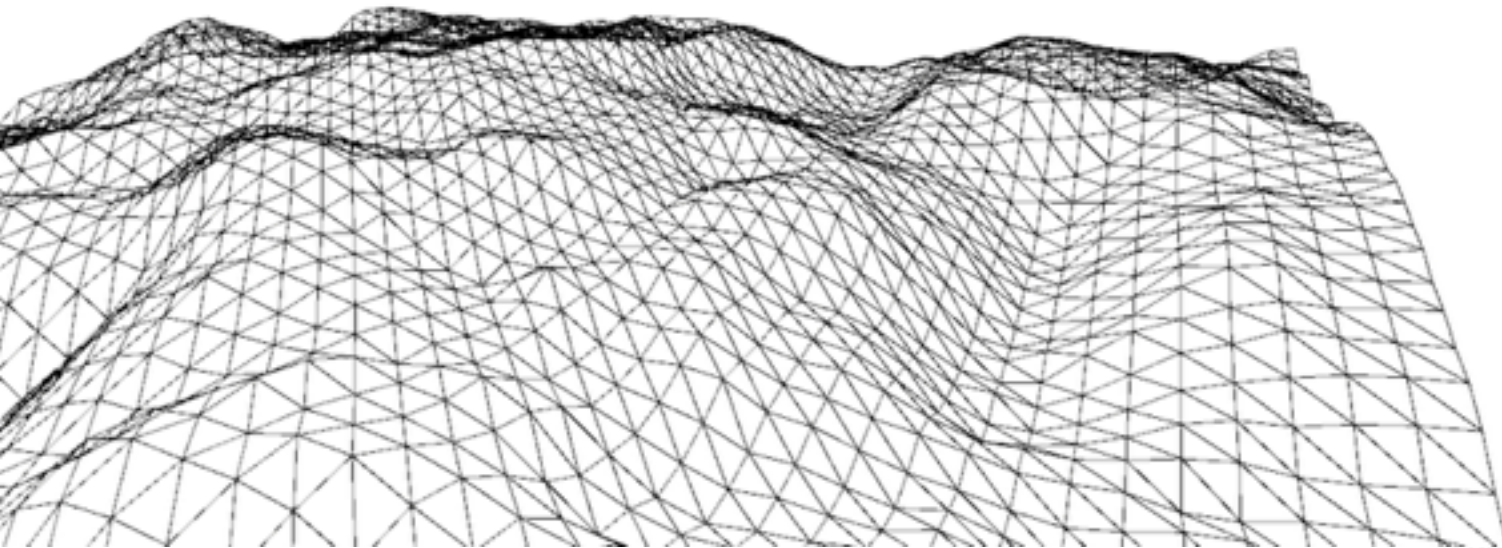
Terrain: mesh of triangles on grid points

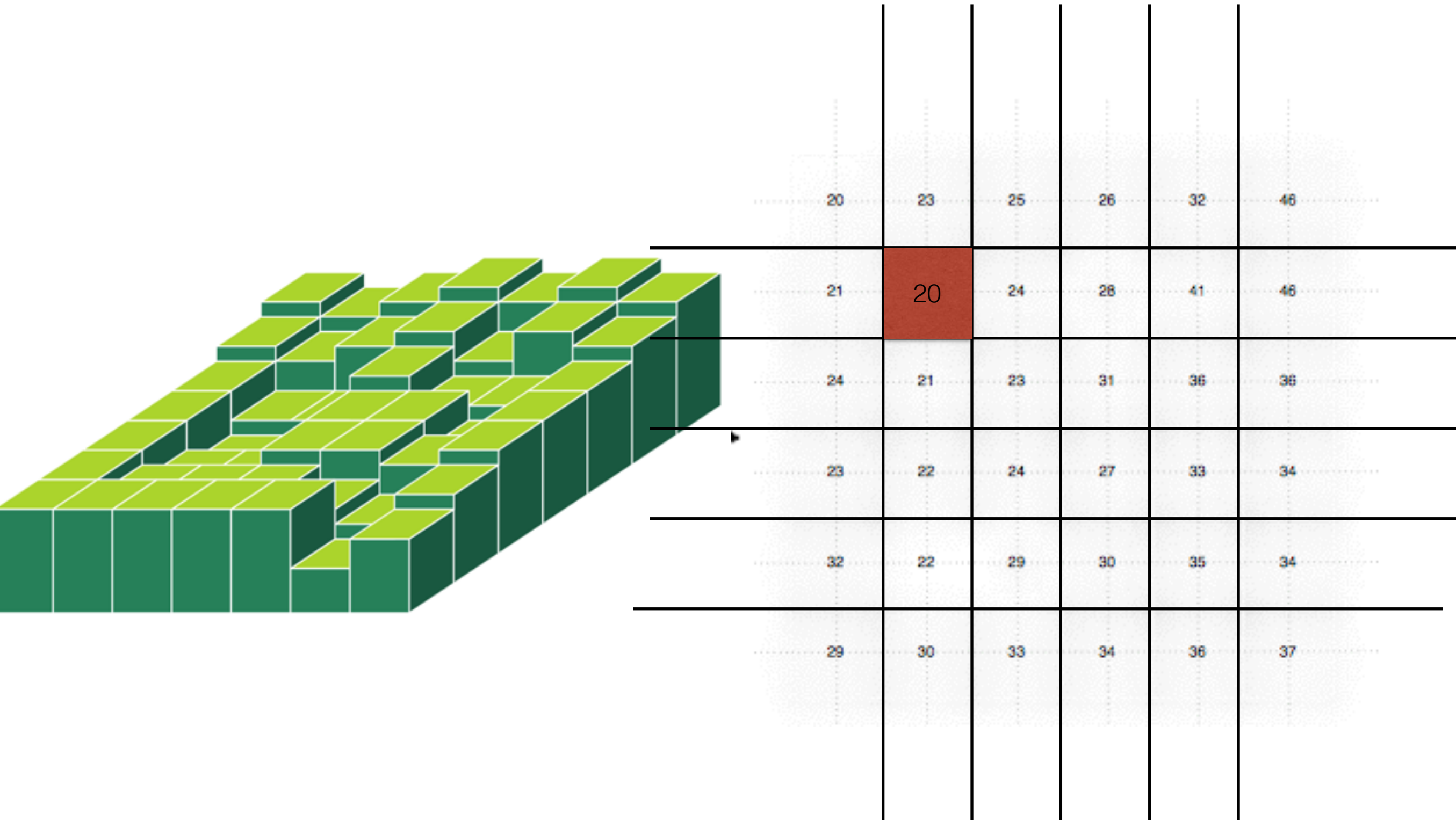# Are v,p visible?
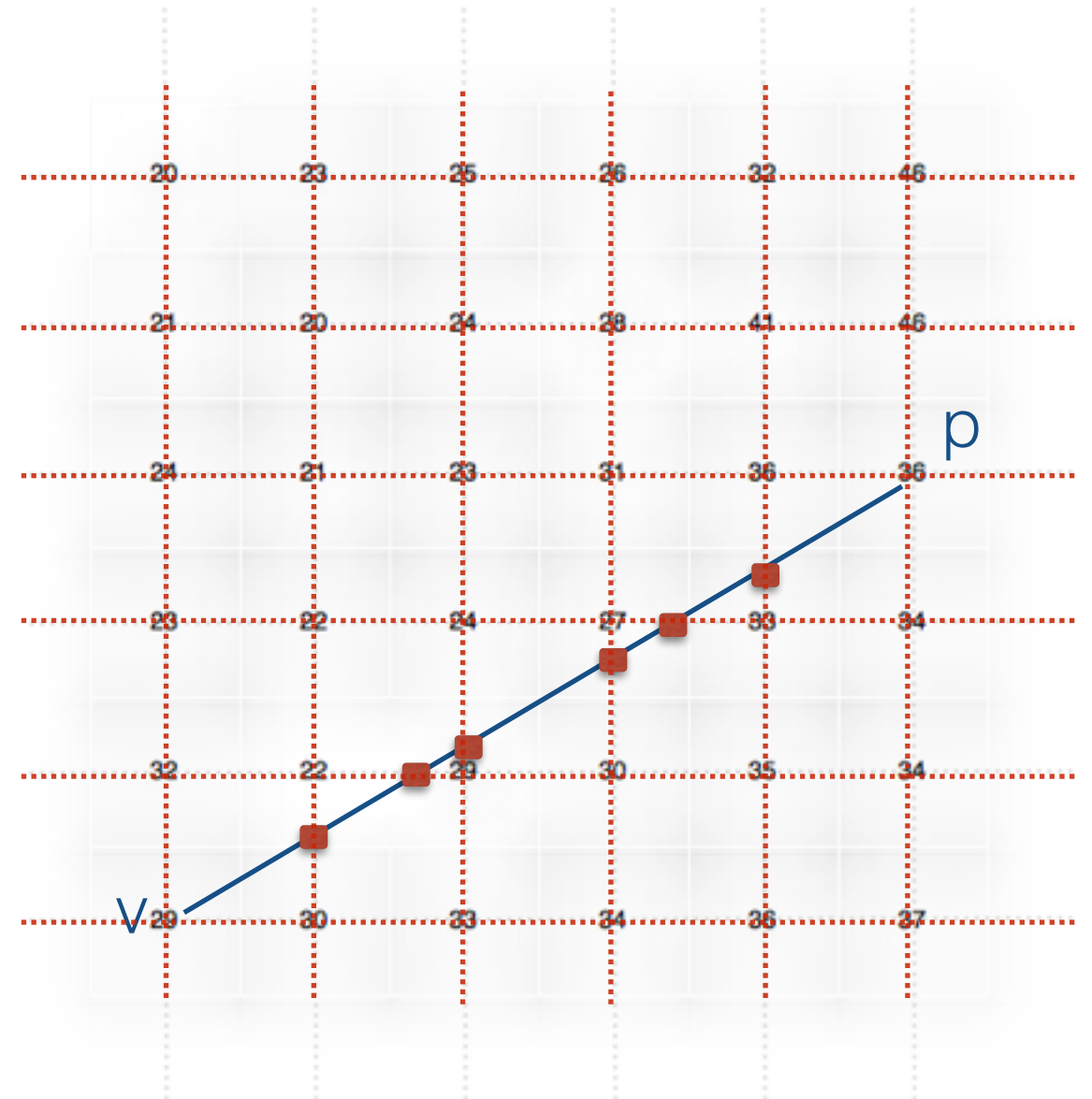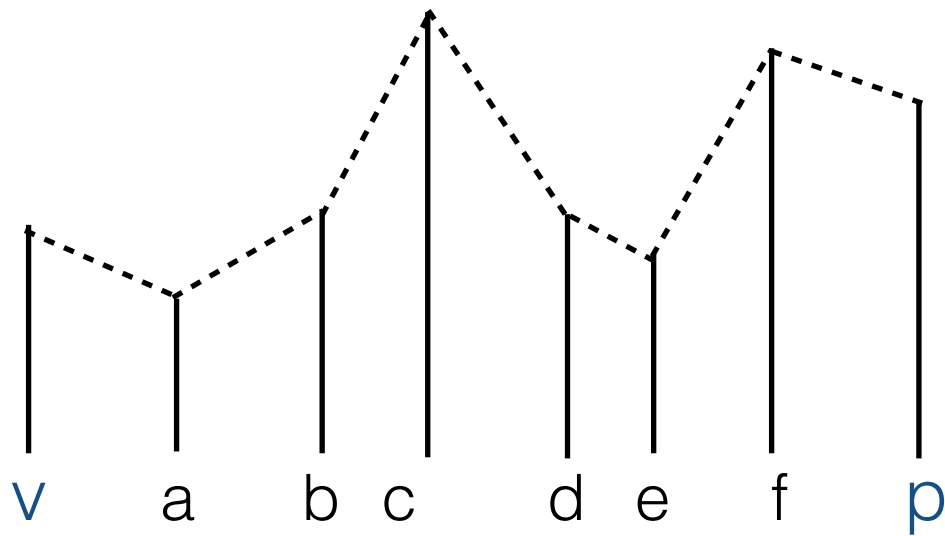
Basic viewshed algorithm

- For every point p=(i,j) in the grid

  - Find if p is visible from v

- Analysis: O($n\sqrt{n}$)

- Uses linear interpolation

- Can we do better (faster)?

  - without skipping points/introducing approximation

- A better algorithm is known, but uses a different/simpler interpolation
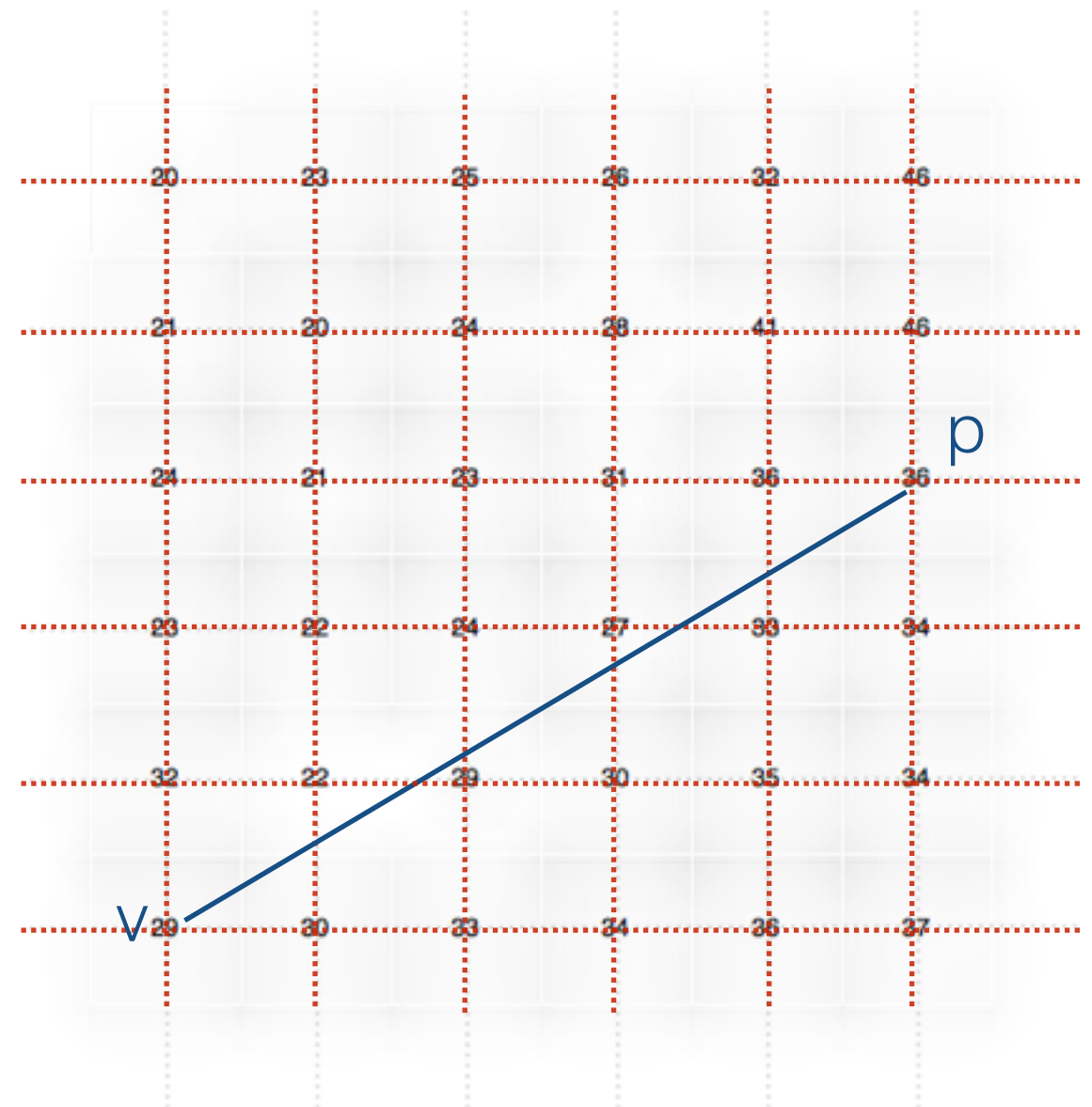
Grids with nearest neighbor interpolation

Viewsheds on Grids:

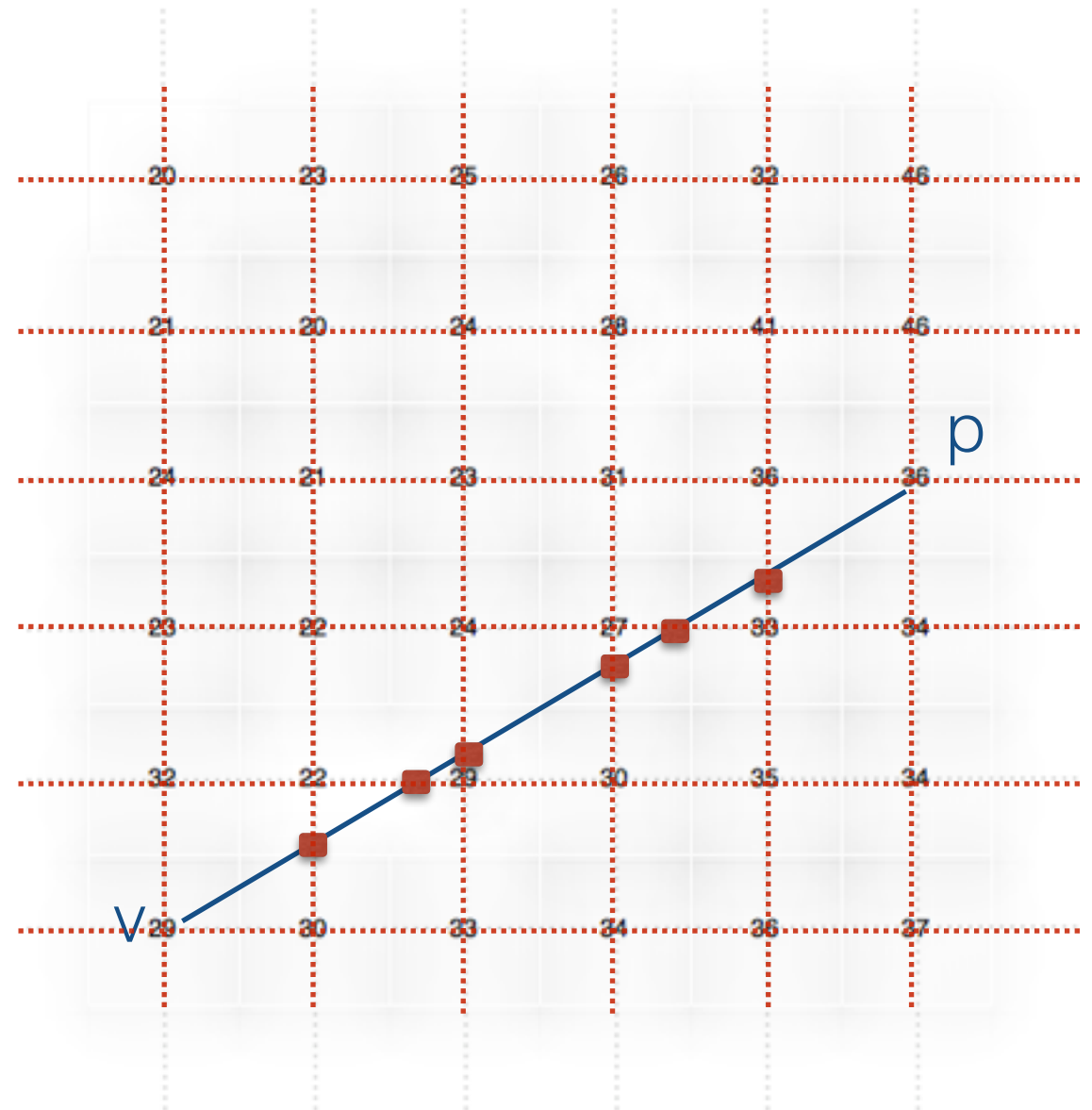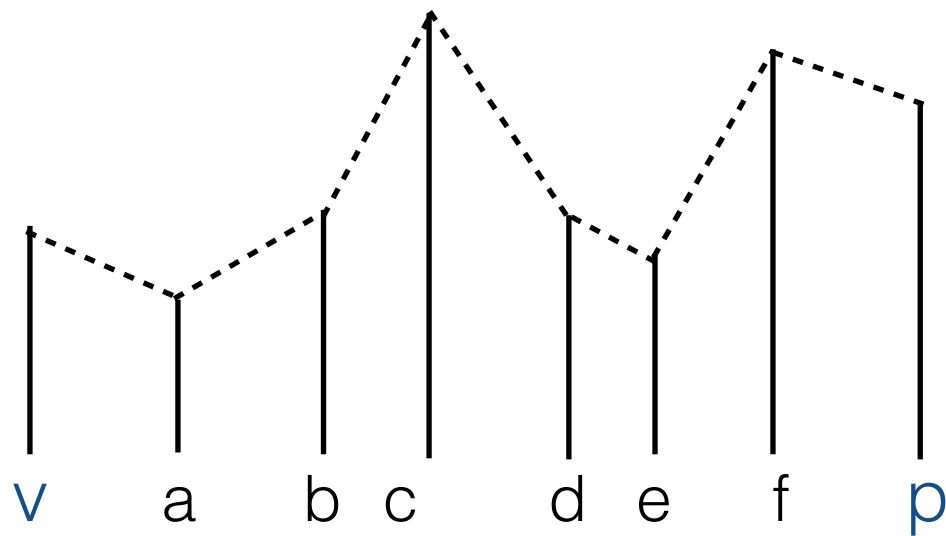2. Van Kreveld's radial sweep viewshed algorithm [VK'96]

Are v,p visible?

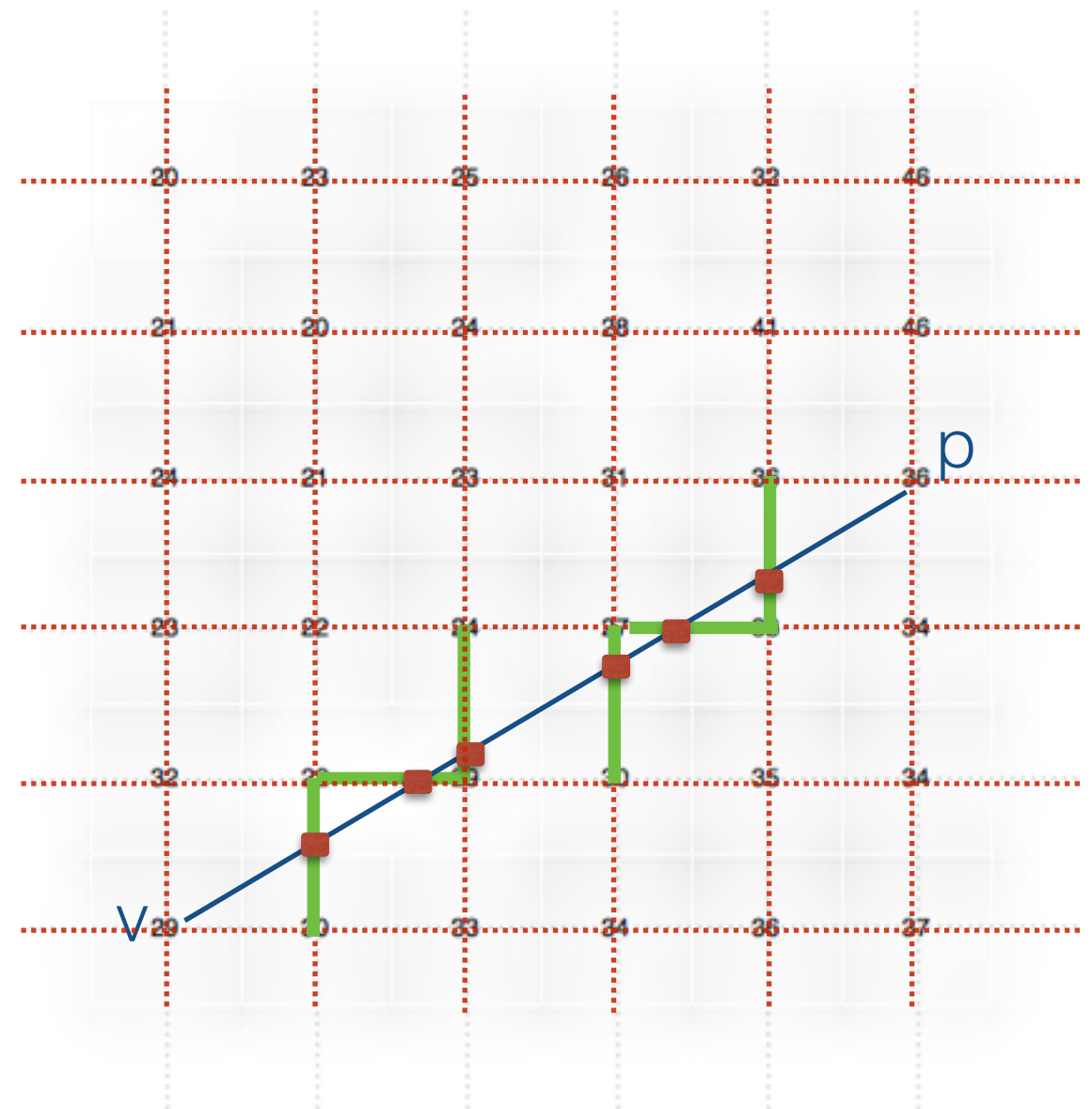# Are v,p visible?

Need the profile of the terrain along LOS vp

Grid of n points:
$\sqrt{n} \times \sqrt{n}$

v   a   b   c   d   e   f   p

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

If the terrain is linear, a straightforward implementation leads to O($\sqrt{n}$) per point

is_visible(v,p) deconstructed
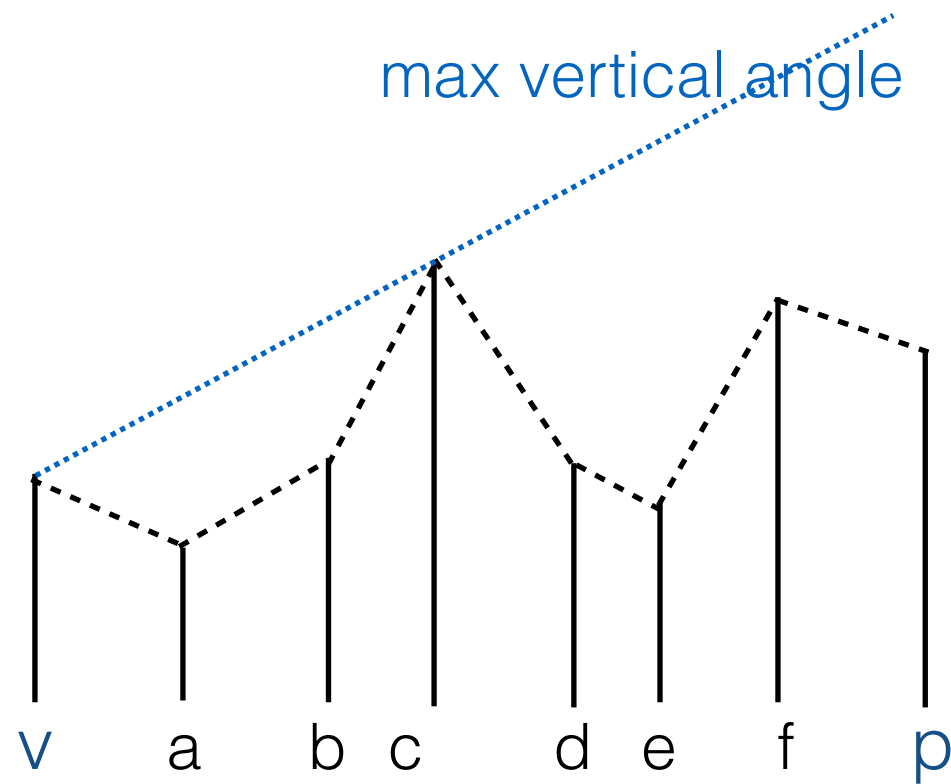
max vertical angle

v   a   b c   d   e   f   p

v

p

- Find all grid segments that intersect LOS vp
- Find maximum vertical angle
- p above/below ==> visible/invisible

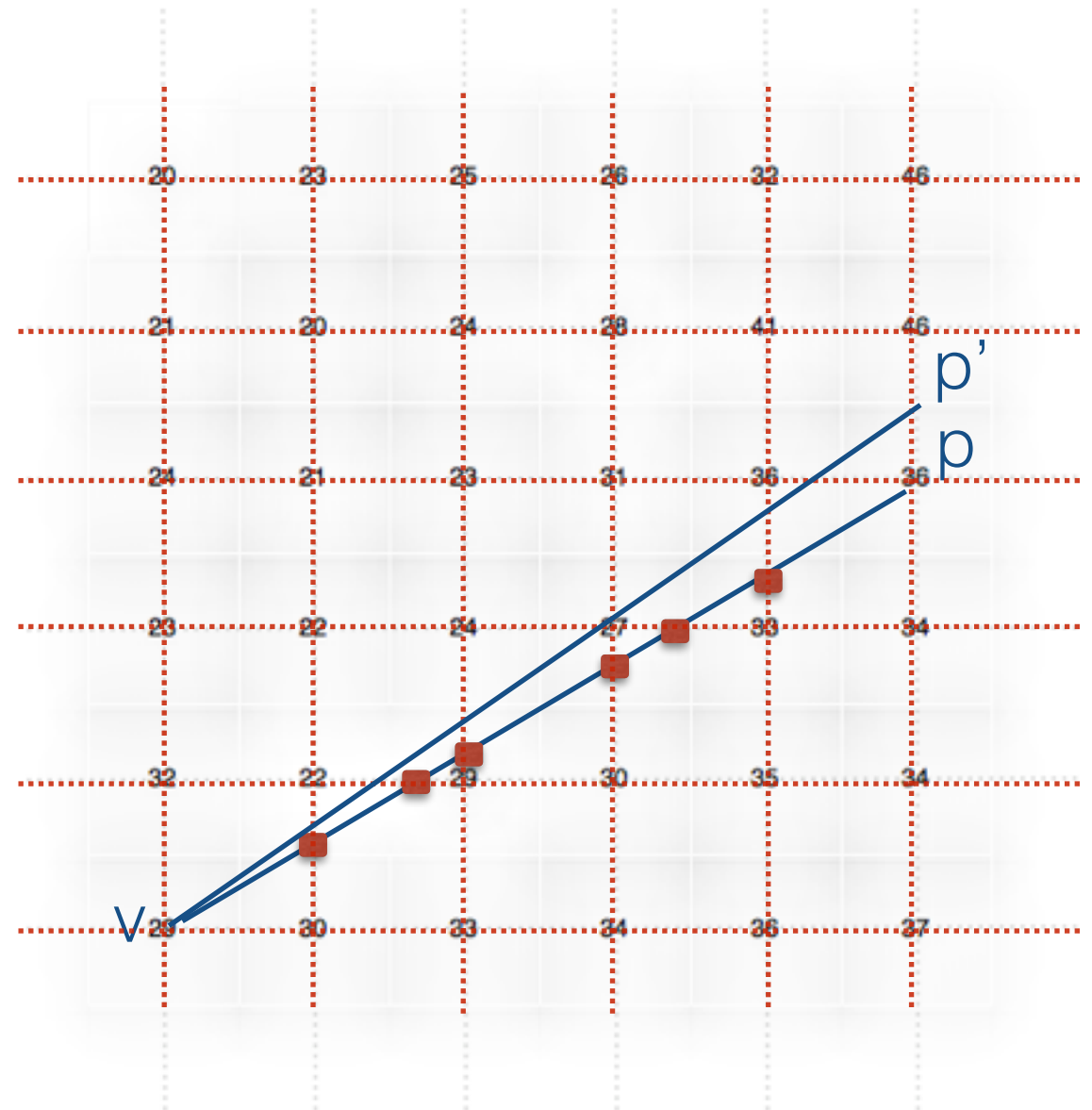# is_visible(v,p) deconstructed

max vertical angle

v   a   b   c   d   e   f   p

p

v
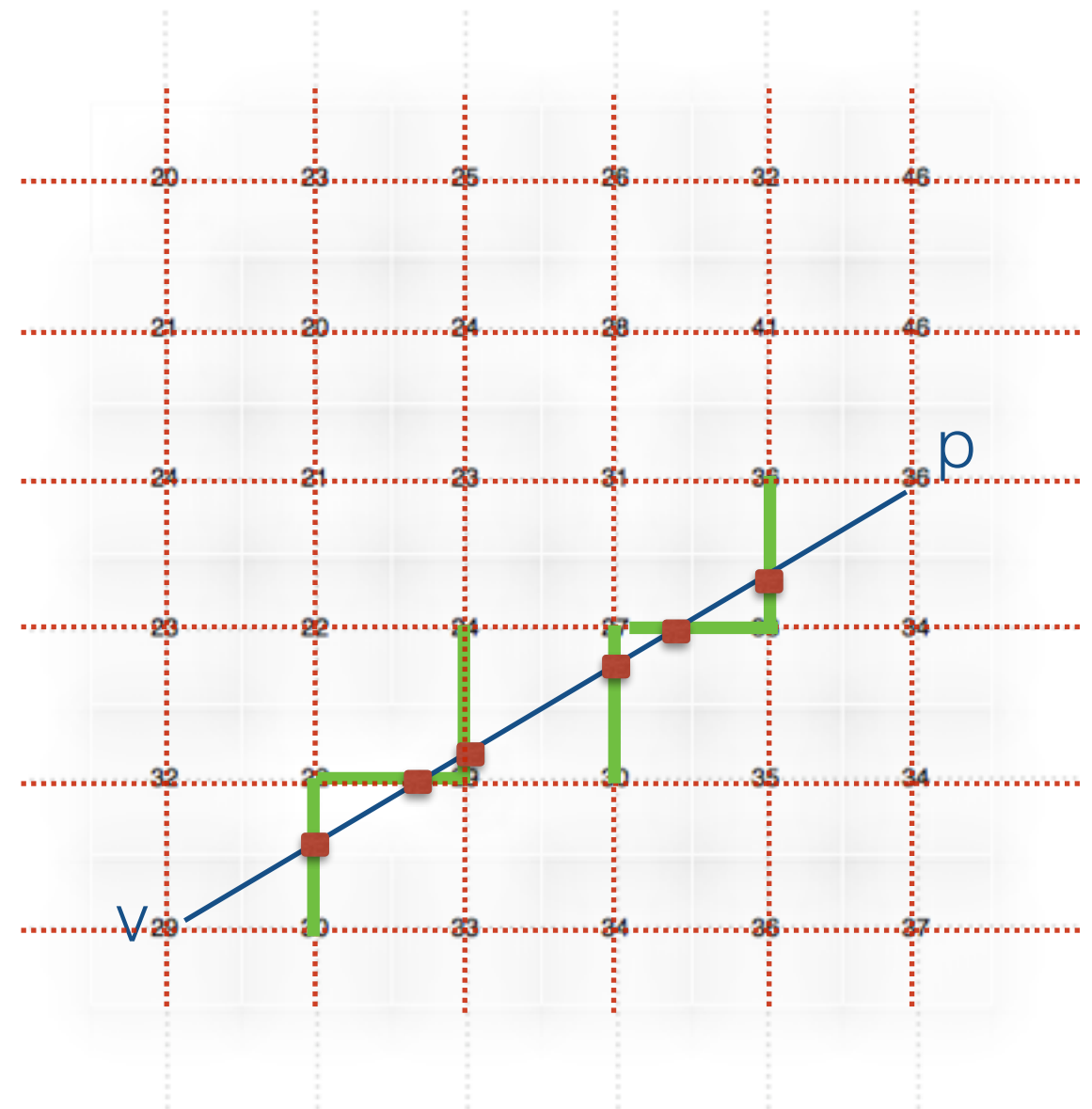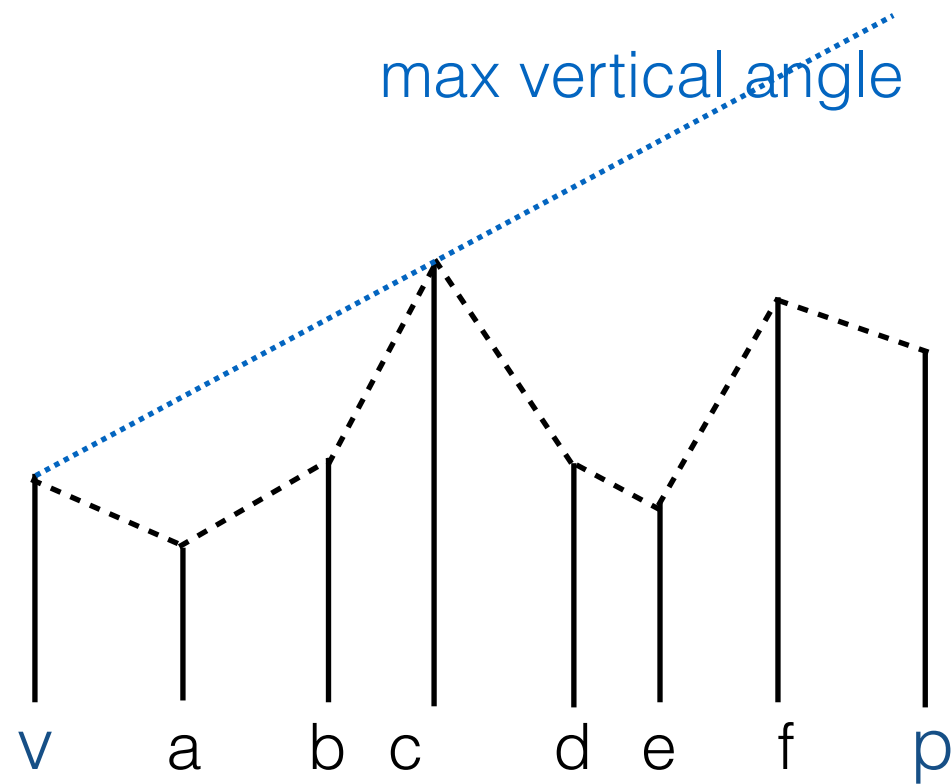
- Find all grid segments that intersect LOS vp ⟵ $O(\sqrt{n})$
- Find maximum vertical angle ⟵ $O(\sqrt{n})$
- p above/below ==> visible/invisible ⟵ $O(1)$
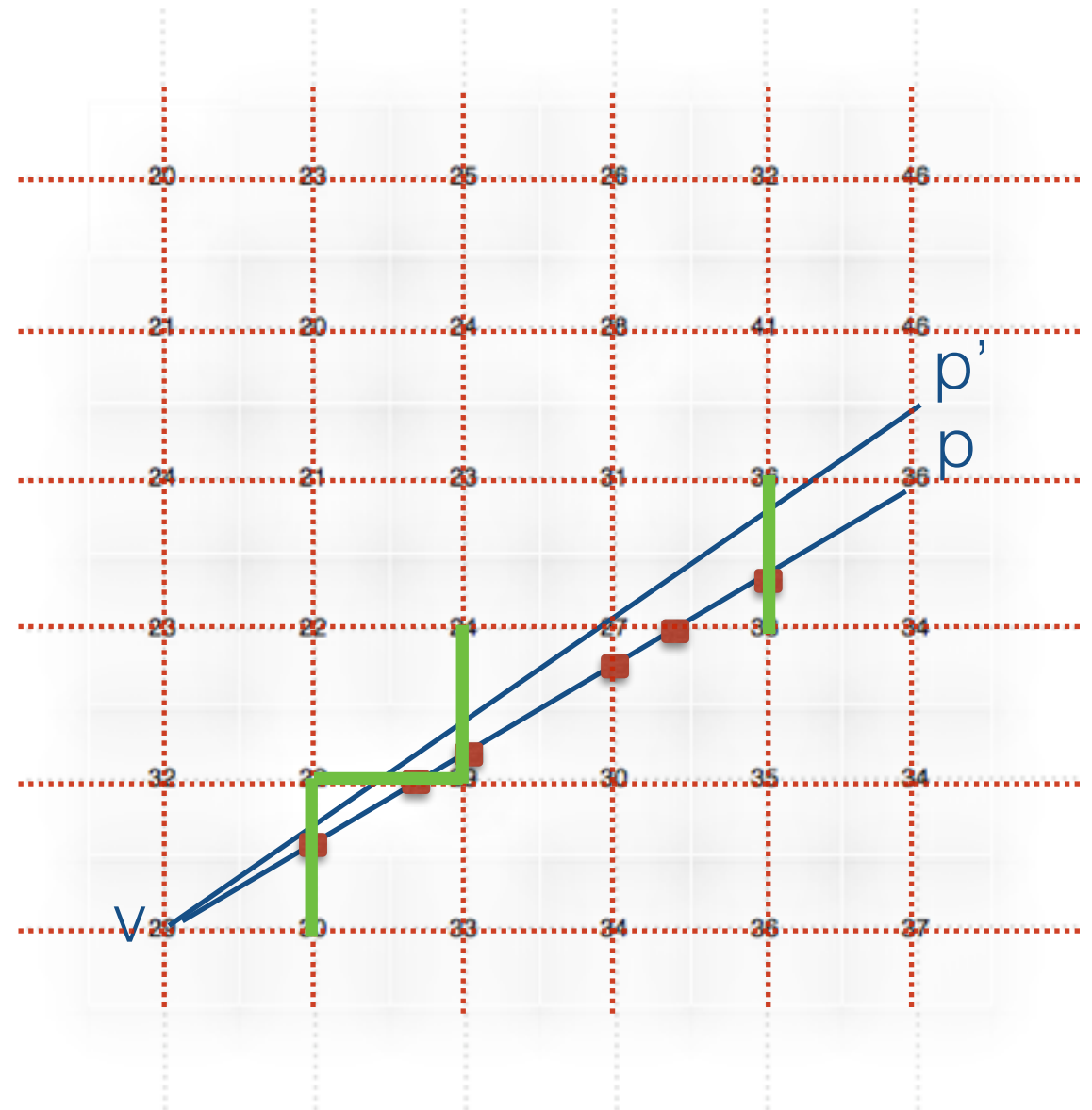
is_visible(v,p) deconstructed

How to re-use some of this

computation for another point?

- Find all grid segments that intersect LOS vp
- Find maximum vertical angle
- p above/below ==> visible/invisible

is_visible(v,p) deconstructed

How to re-use some of this
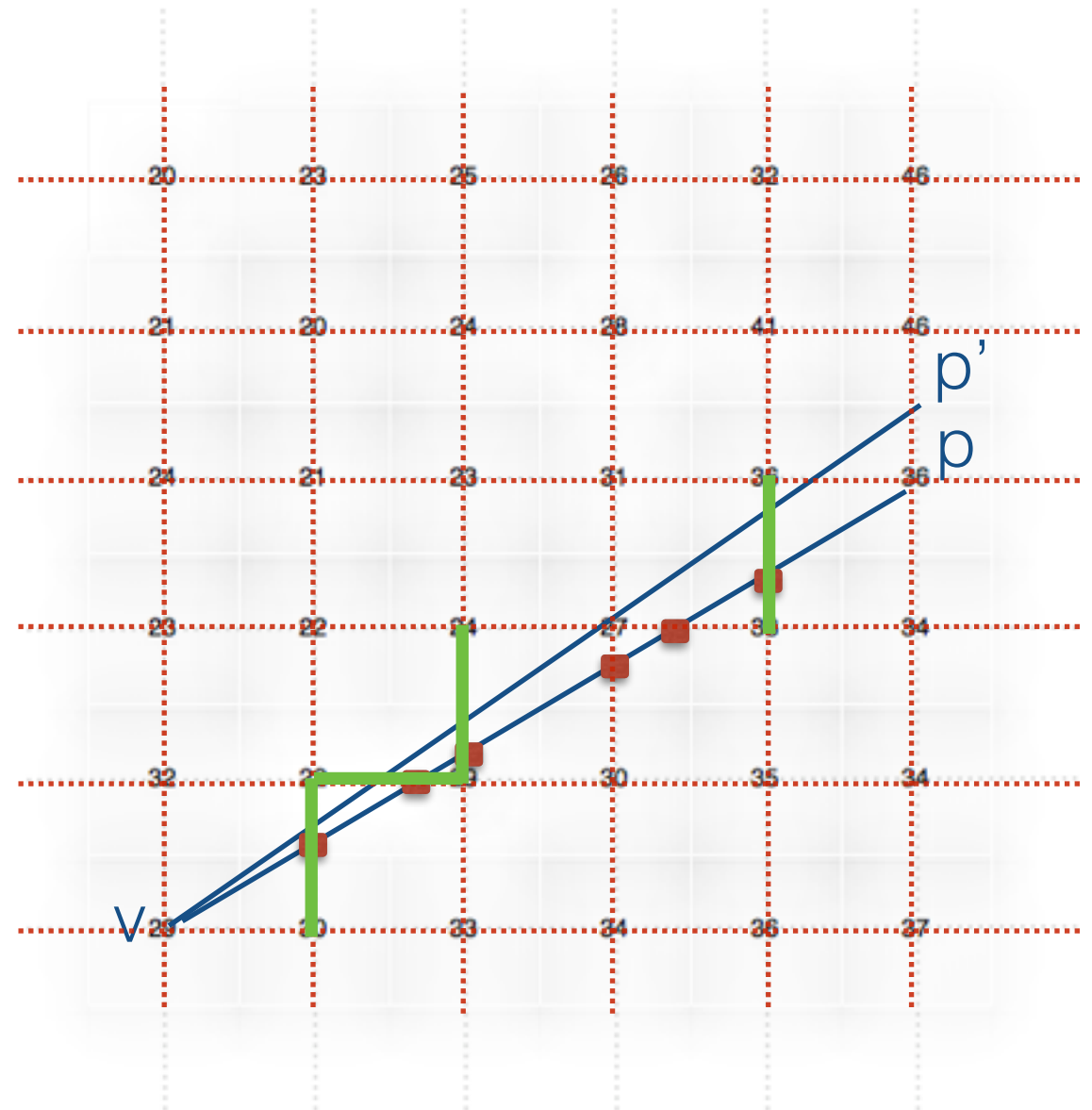
computation for another point?

Some segments intersected by both

The intersection point and its vertical angle is different

is_visible(v,p) deconstructed

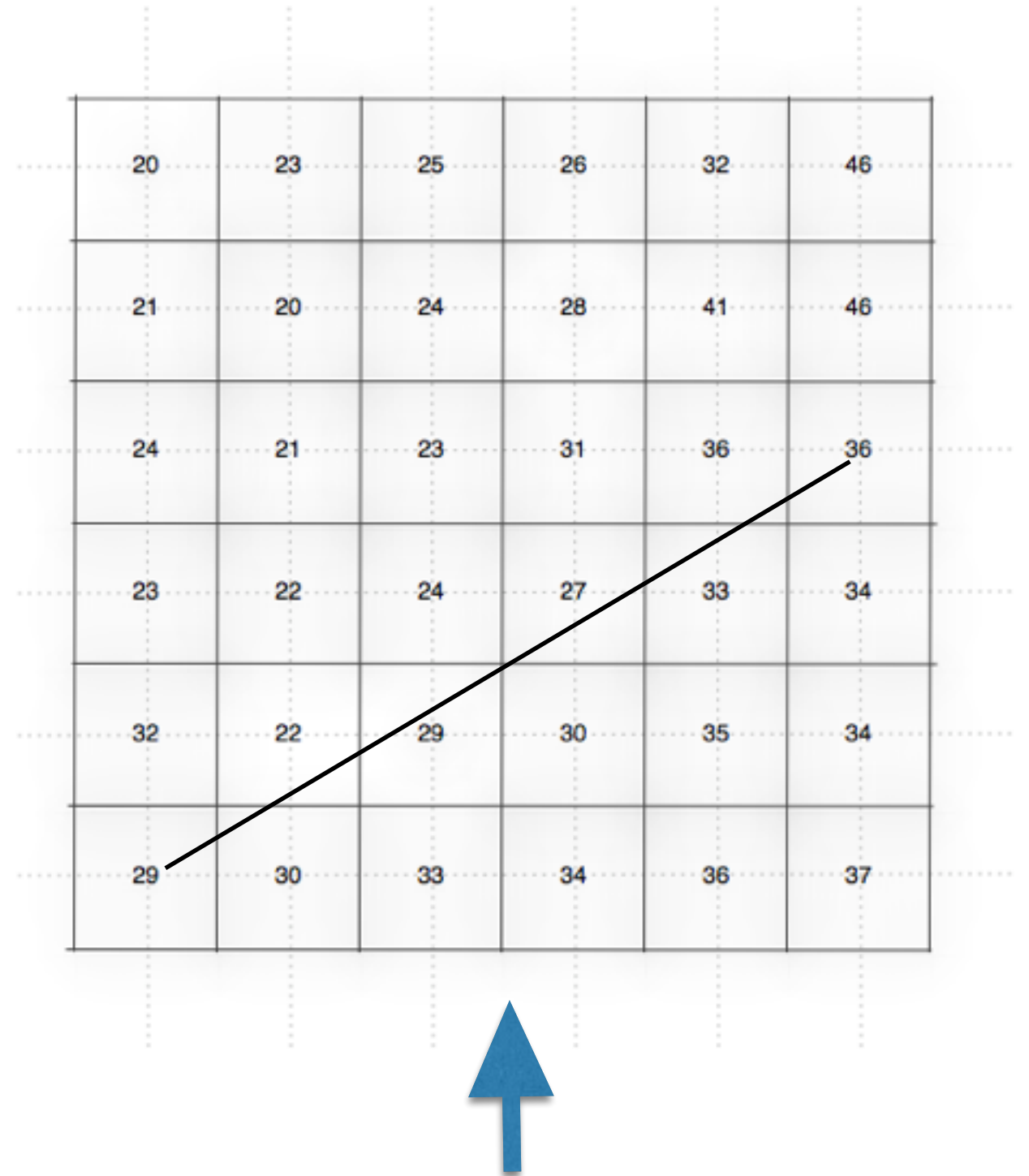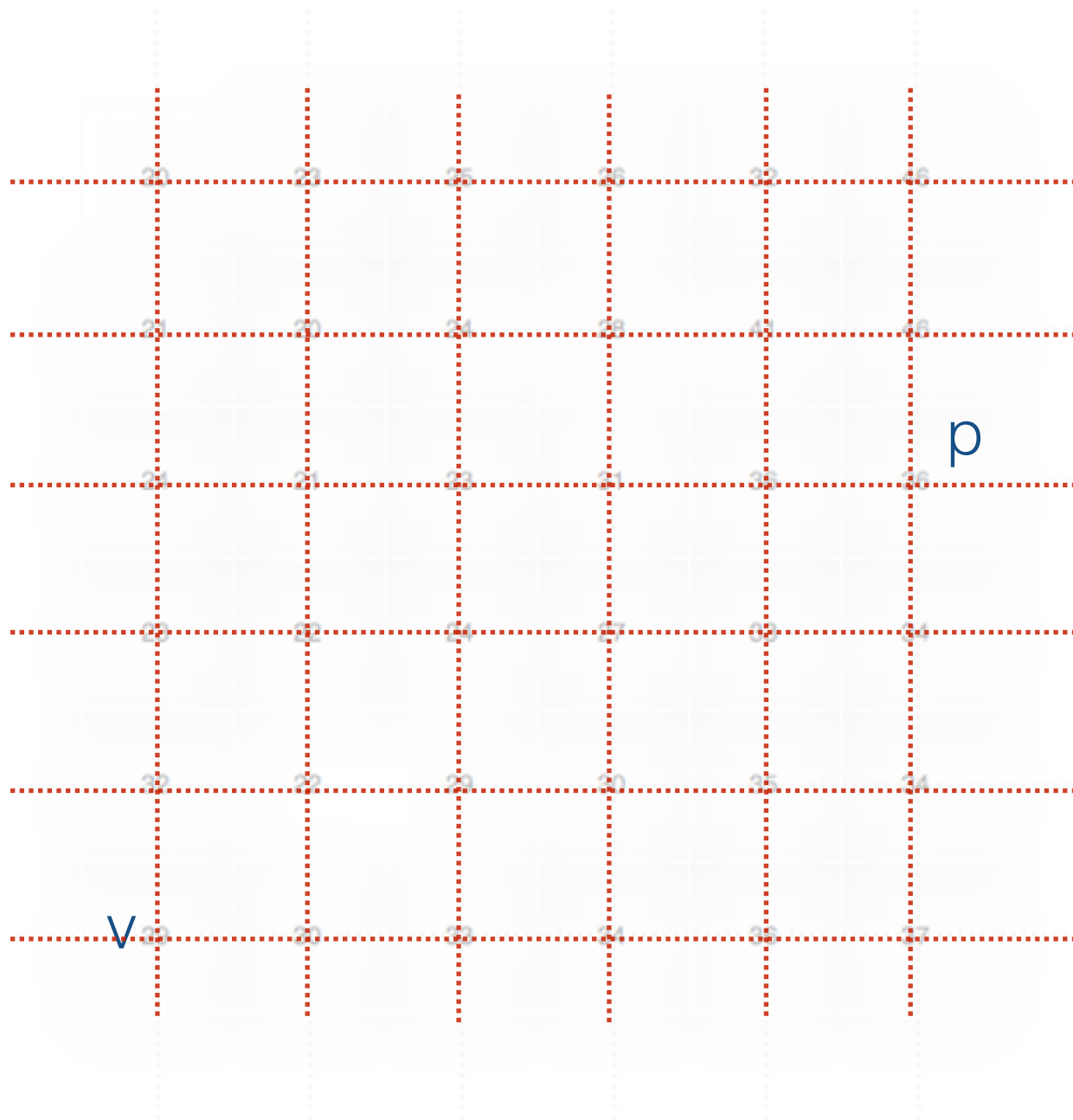How to re-use some of this

computation for another point?

p'
p

v

Some segments intersected by both

The intersection point and its vertical angle is different

But… if the segment was at constant height…

Are v,p visible?

Terrain consists of cells centered at the grid points.

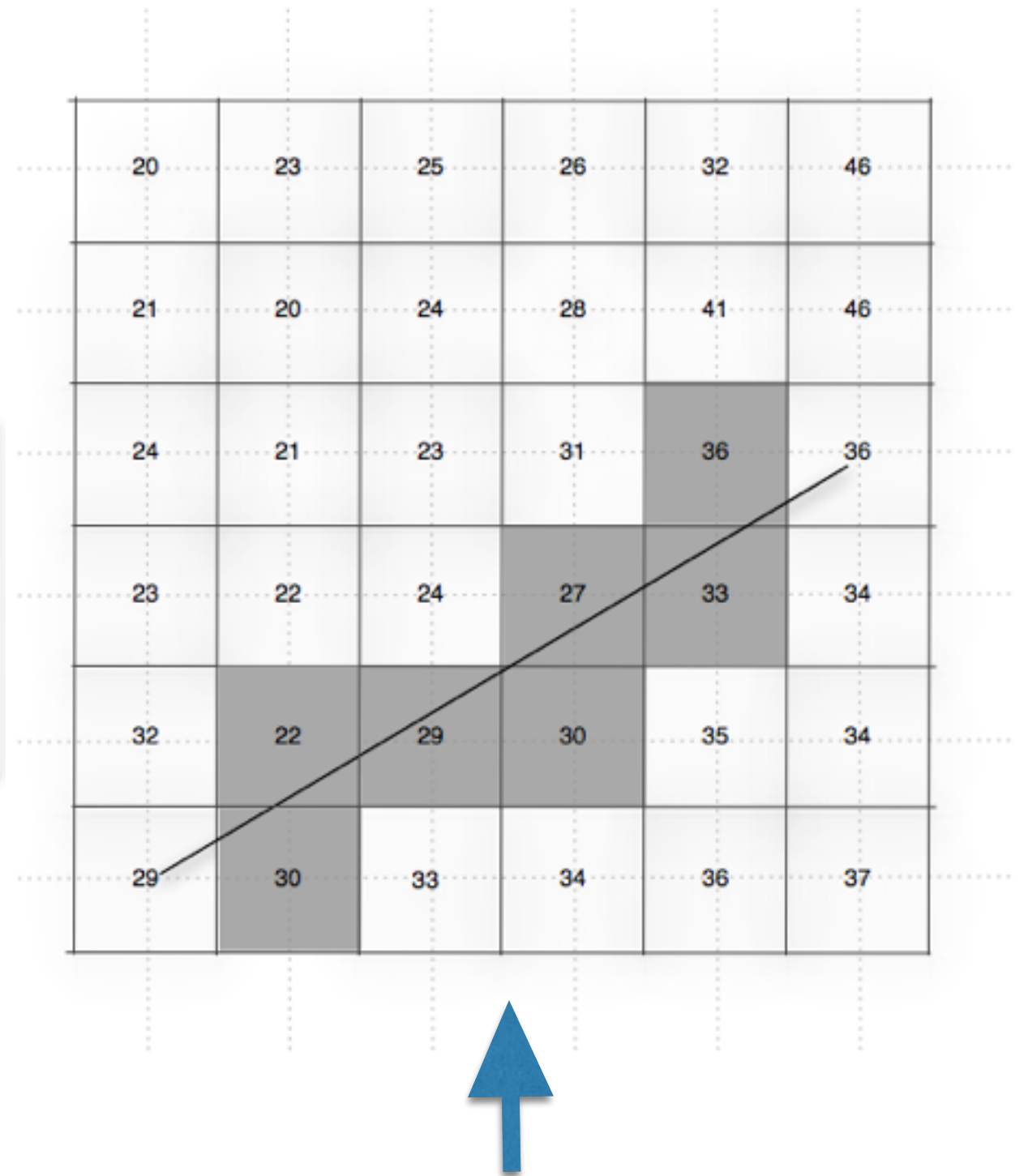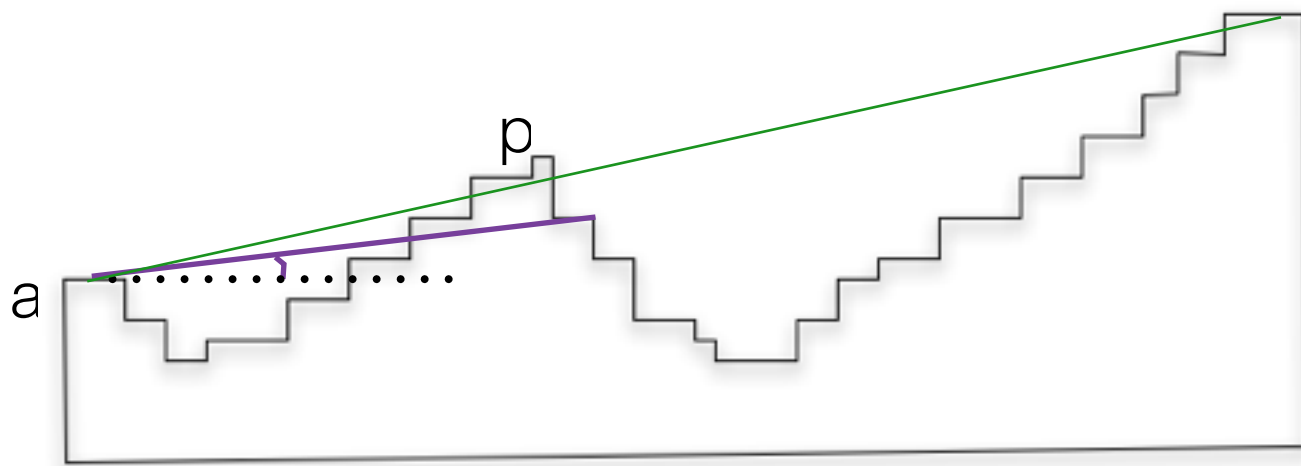**Assume that the vertical angle for a cell is the same throughout the cell.**
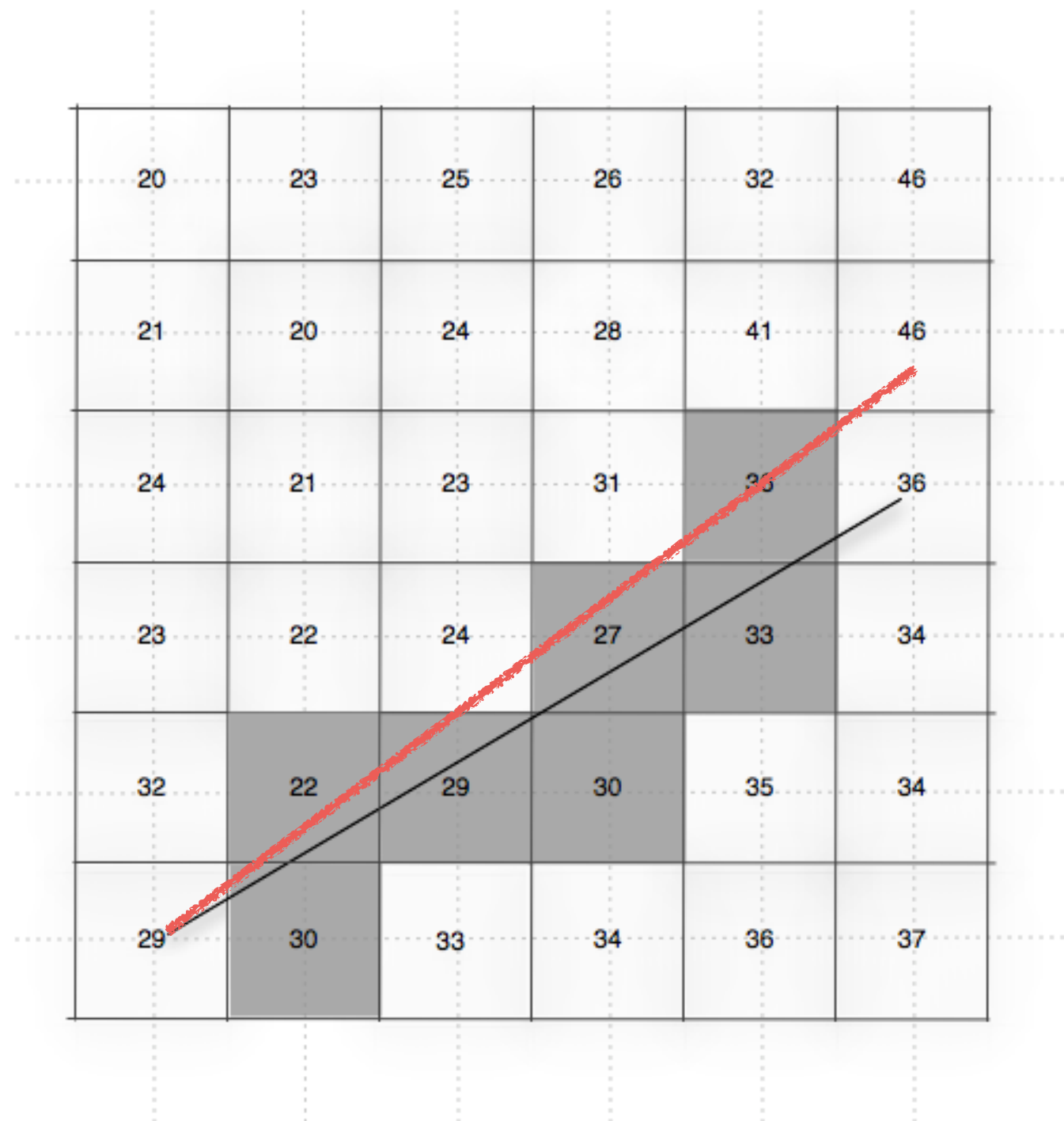
Are v,p visible?

Terrain consists of cells centered at the grid points.

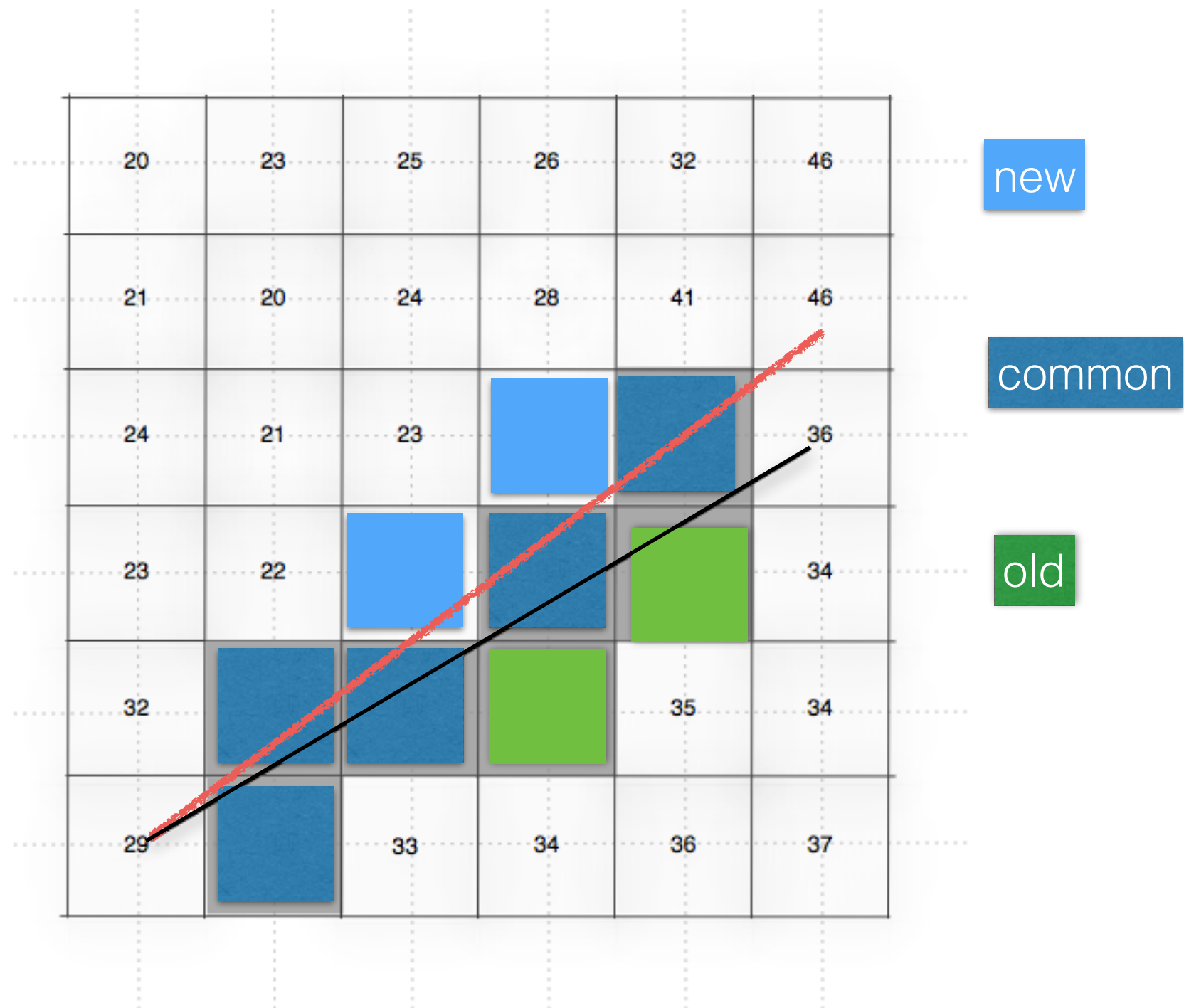Assume that the vertical angle for a cell is the same throughout the cell.

Towards a faster algorithm

- **Idea 1**: The lines-of-sight to two nearby points intersect a lot of same cells.

- **Idea 2**: Assume that the vertical angle for a cell is the same throughout the cell (i.e. nearest neighbor interpolation instead of linear)

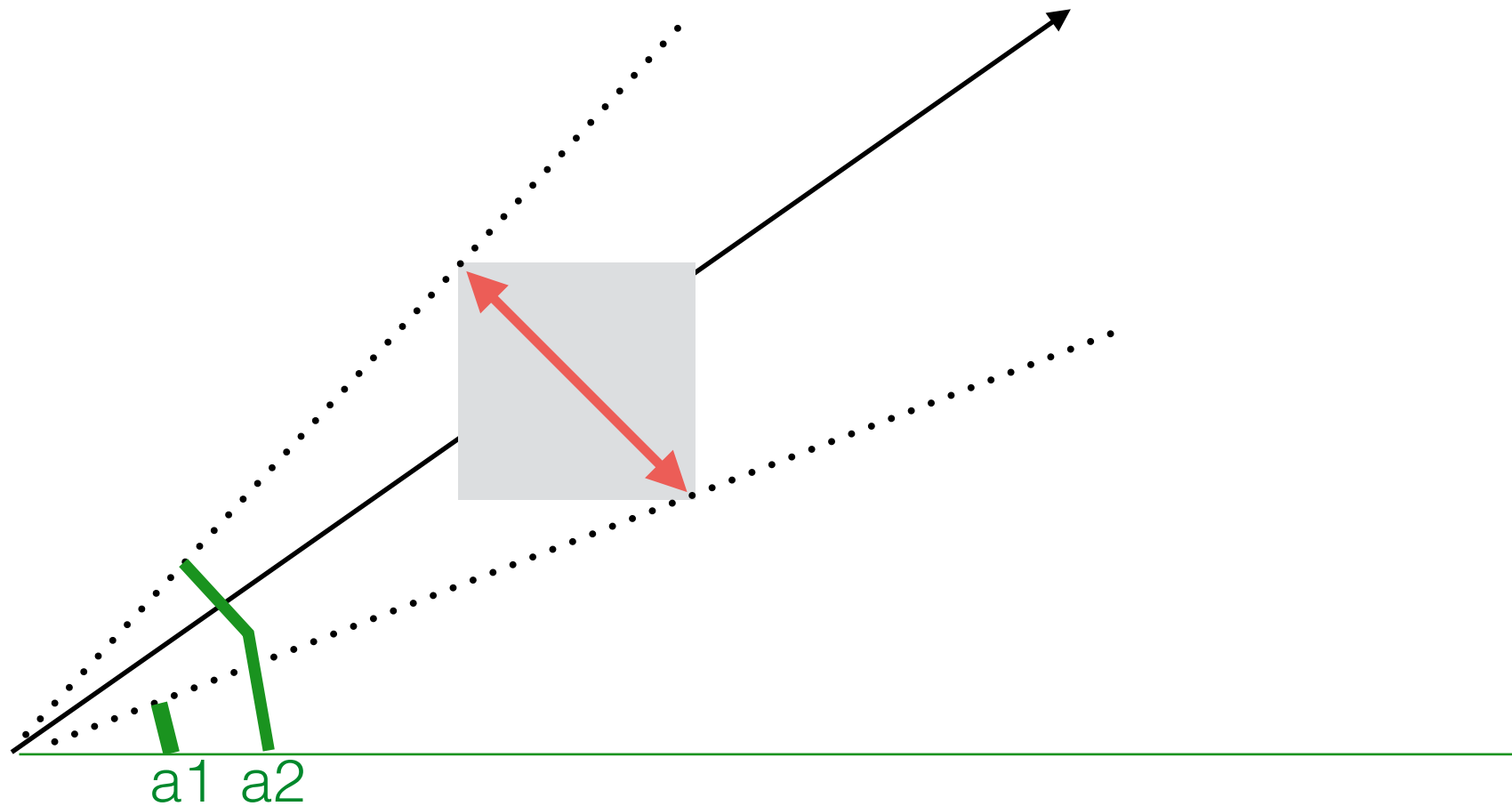# LOS to two nearby points intersect a lot of the same cells

20	23	25	26	32	46
21	20	24	28	41	46
24	21	23	31	3?	36
23	22	24	27	33	34
32	22	29	30	35	34
29	30	33	34	36	37

LOS to two nearby points intersect a lot of the same cells

new

common

old

How to express that a cell intersects/does not intersect the LOS?

It's the azimuth!!

cell intersects LOS ==> azimuth(LOS) between a1 and a2

Towards a faster algorithm

- **Idea 1**: The lines-of-sight to two nearby points intersect a lot of same cells.

- **Idea 2**: Assume that the vertical angle for a cell is the same throughout the cell (i.e. nearest neighbor interpolation instead of linear)

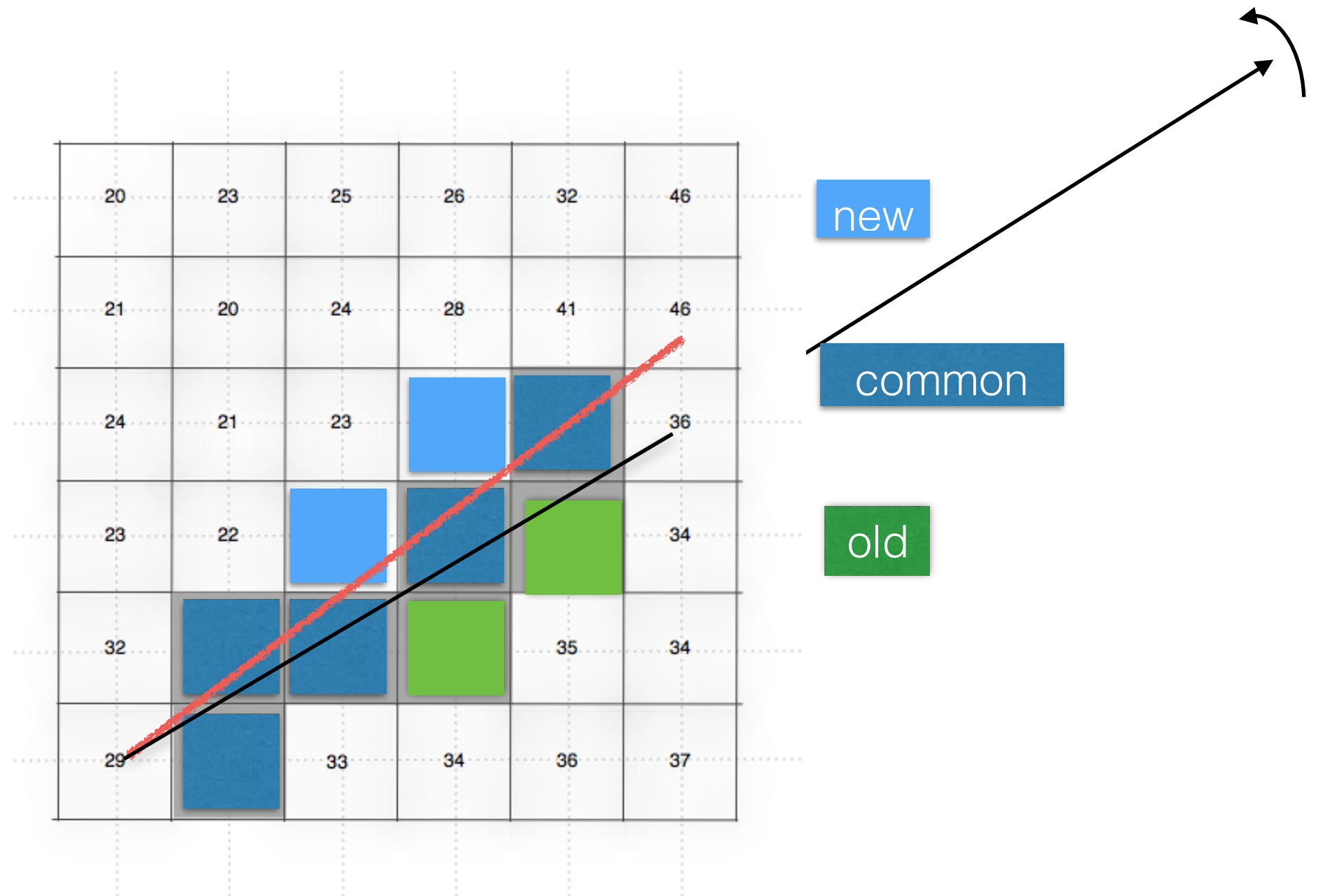- **Idea 3**: Compute visibility of points in radial order around v

Viewshed in row-column order

- for i = 0; i< rows; i++
  - for j=0; j< cols; j++
    - find if (i,j) is visible from v

Viewshed in radial order

- sort points (i,j) by radial angle of (i,j)
- for each point (i,j) in order:
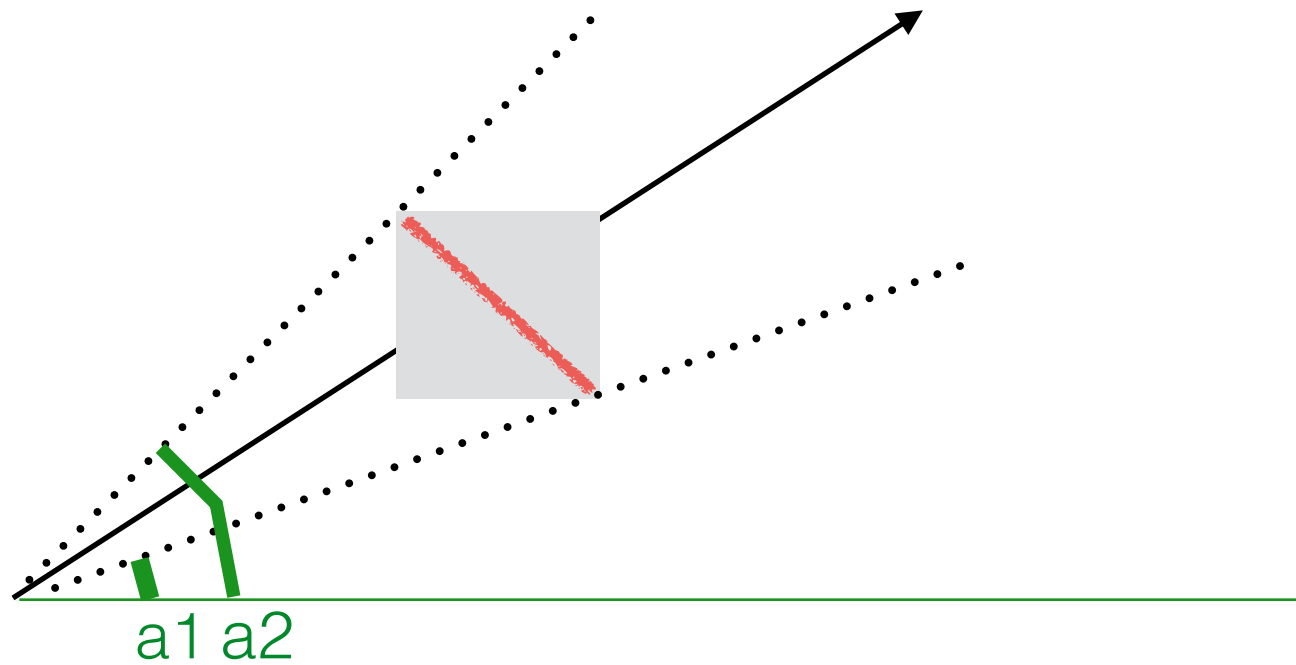 - find if (i,j) is visible from v

Some of the cells that intersected the previous point will also intersect the current point; re-use them. HOW???
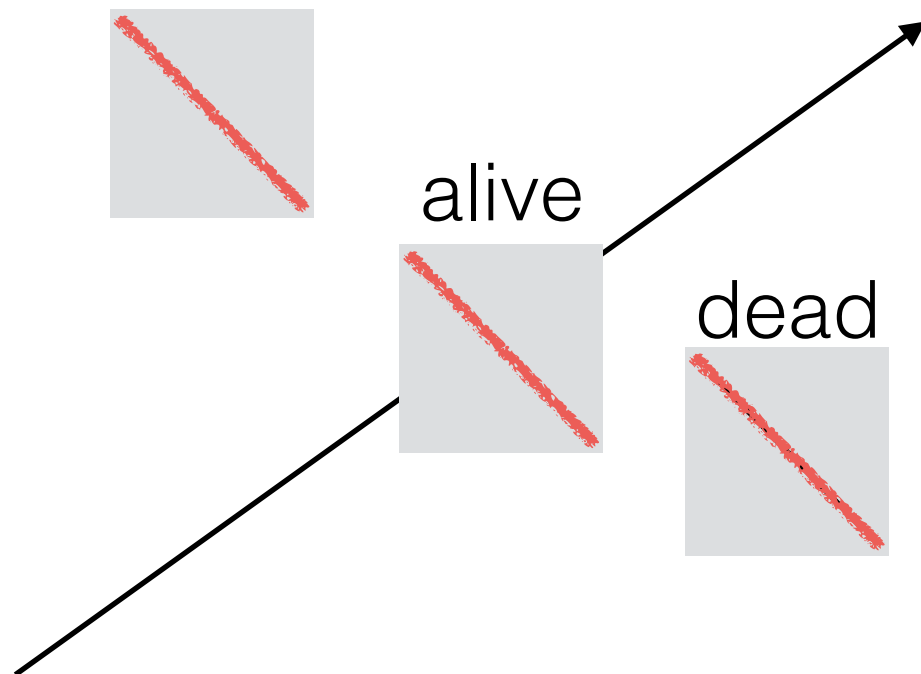
We'll compute visibility of points in radial order around v

- **Radial sweep**

  - rotate a ray radially around v
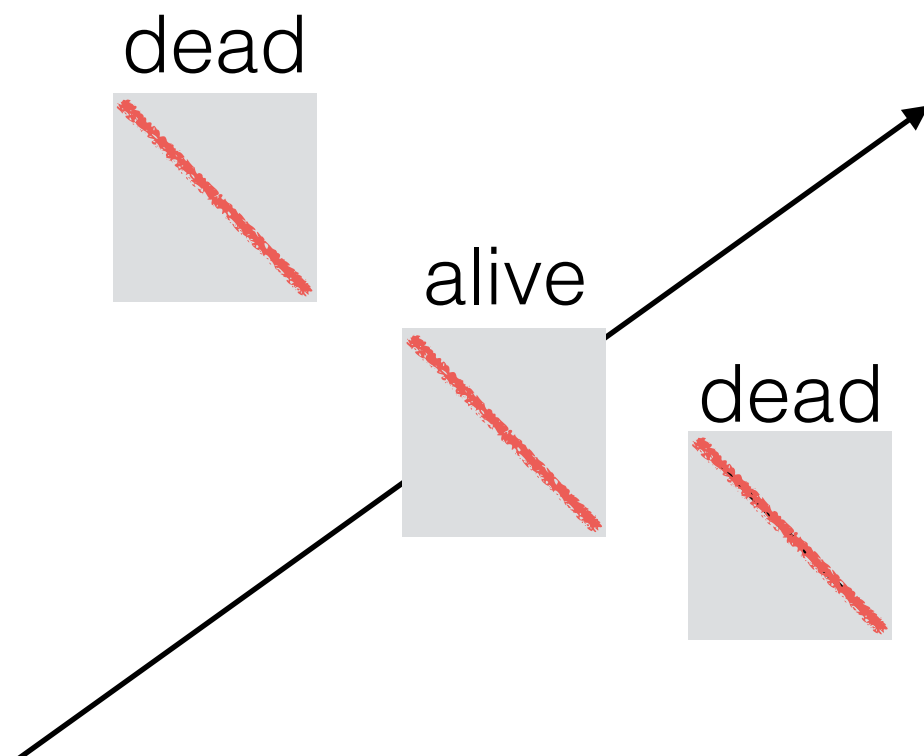
  - respond to "interesting" events

The events

- Ray hits a1: cell alive
- Ray hits a2: cell dies
- Ray between a1 and a2: cell intersects ray

a1 a2

alive

dead

The events

- when ray hits ENTER(i,j):   cell (i,j) becomes active

- when ray hits a grid point (i,j):   determine if (i,j) is visible

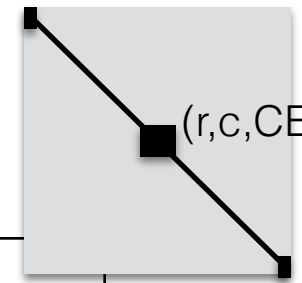- when ray hits EXIT(i,j):   cell (i,j) becomes inactive

dead

alive

dead

between ENTER(i,j) and EXIT(i,j):

cell (i,j) will be active

Van Kreveld's radial sweep algorithm

(r-.5,c-.5, EXIT)

(r,c,CENTER

(r+.5,c+.5, ENTER)

- For each point (i,j): compute its ENTER, CENTER, EXIT events

- Sort all events by radial angle wrt v

- initialize AS to contain all cells that are active at angle=0

- For next event (r,c, type) in radial order

  - if type is ENTER: //cell becomes active

    - insert cell(r,c) in AS

  - if event is EXIT: //cell stops being active

    - delete cell(r,c) from AS

  - if event is CENTER:

    //CLAIM: all cells that intersect the los from v to (r,c) must be in the AS

    - use AS to find maximum verticalAngle of all cells between v and cell(r,c)

    - if this angle is below verticalAngle(r,c) then (r,c) is visible; otherwise (r,c) is invisible
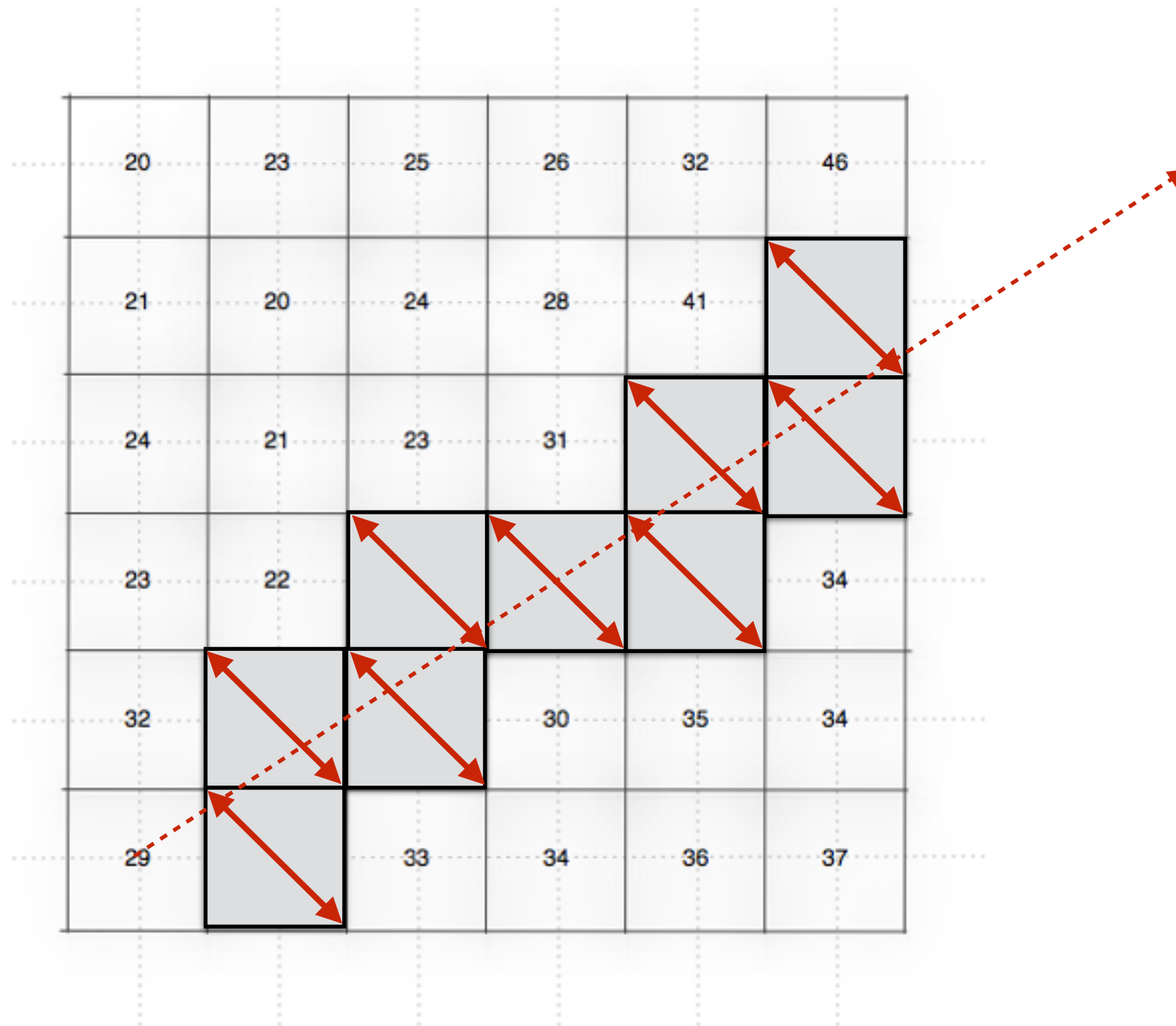
The 3 events corresponding to each cell
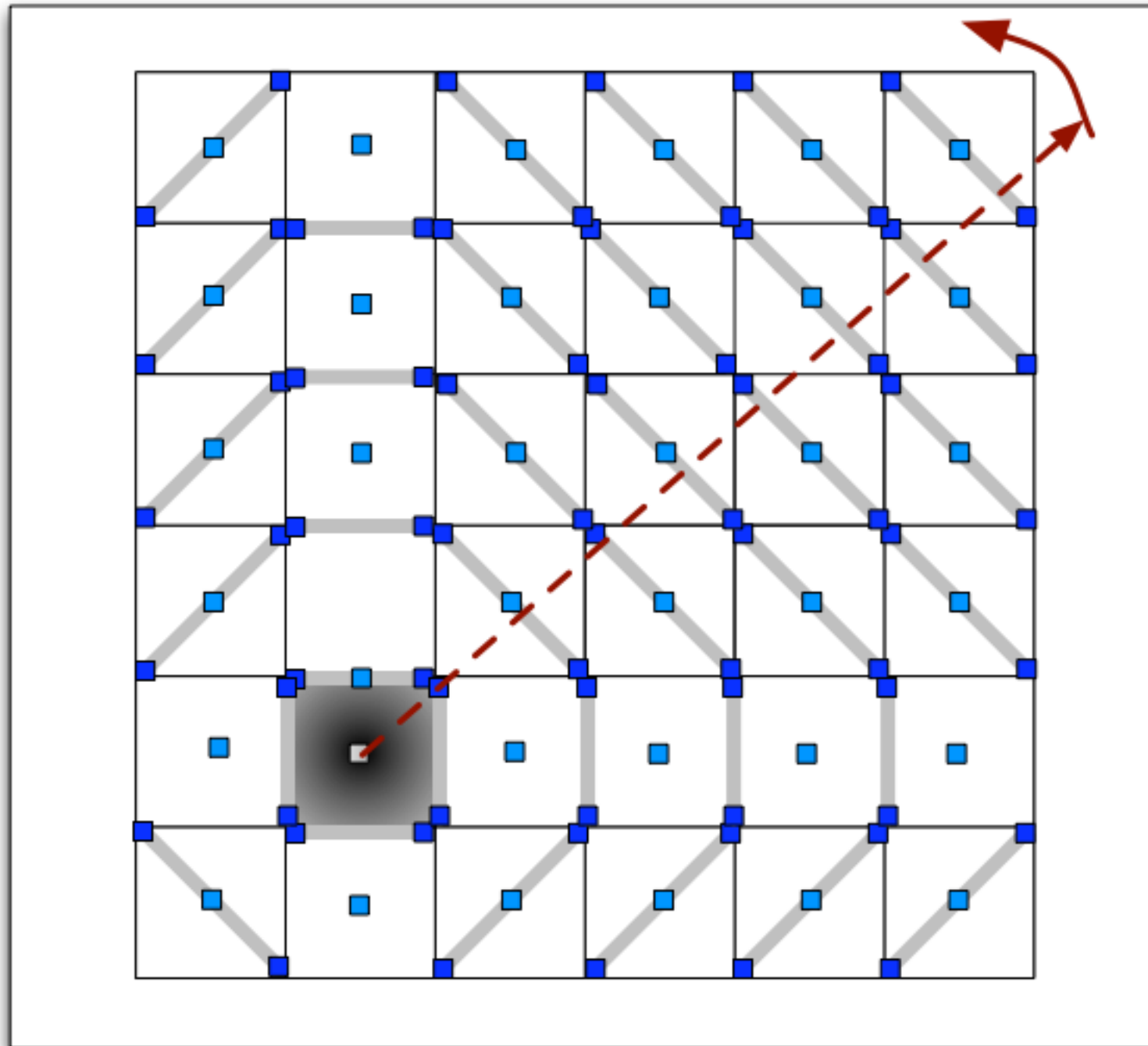
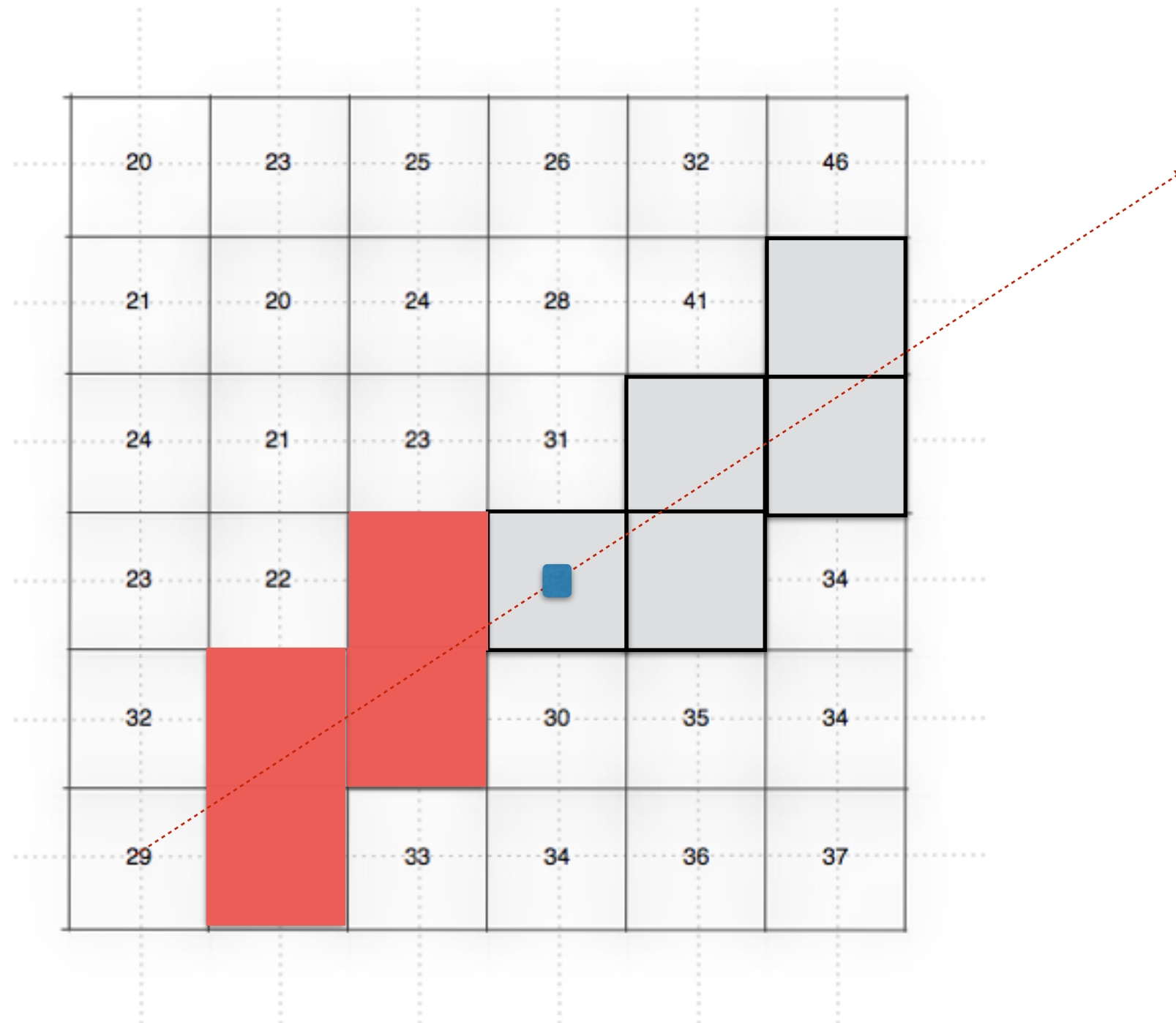For an arbitrary position of the ray, all cells that it intersects will be in AS
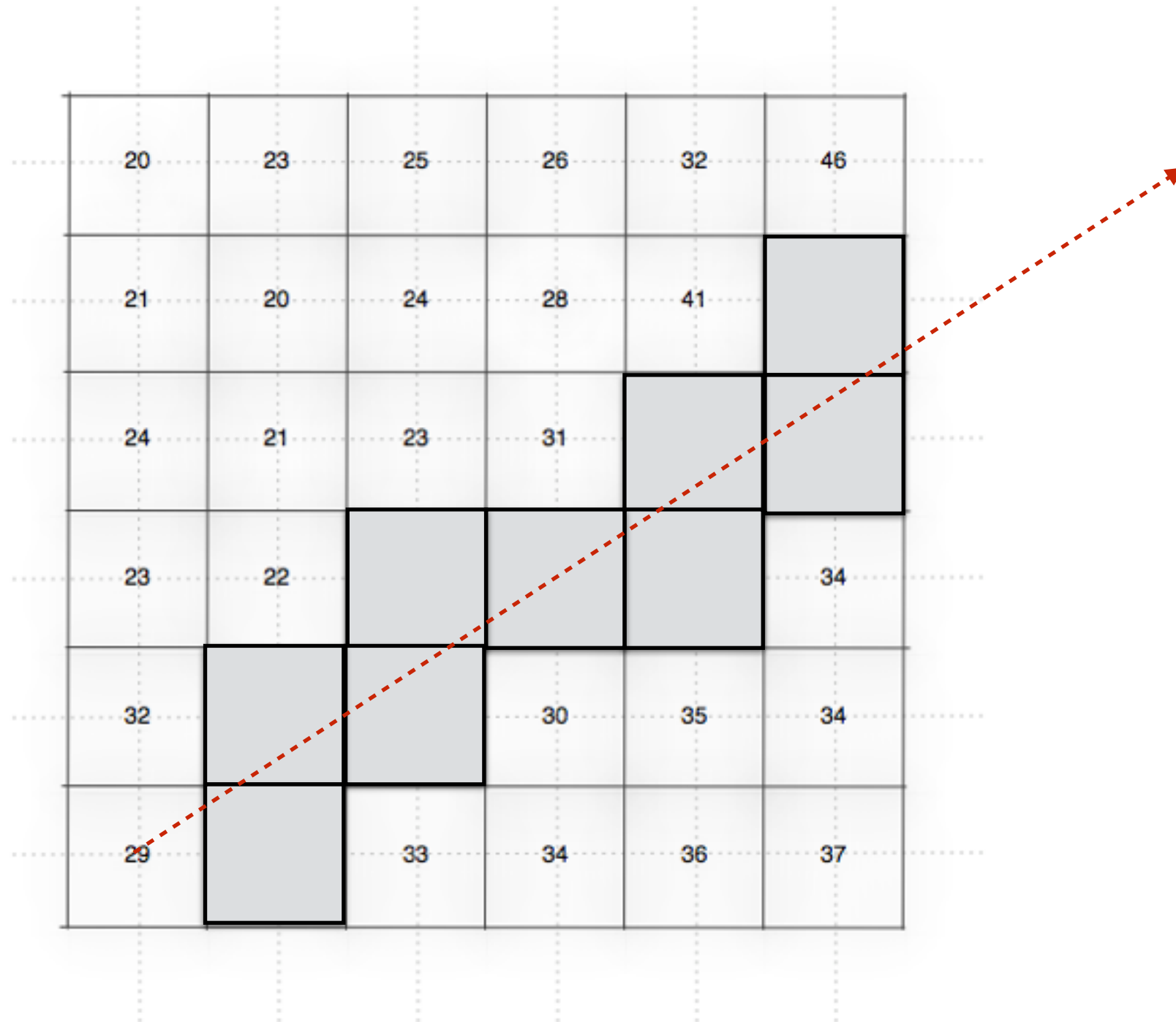
For an arbitrary position of the ray, all cells that it intersects will be in AS

When process CENTER(r,c): , all cells that intersect ray will be in AS

When process CENTER(r,c): , all cells that intersect ray will be in AS

20	23	25	26	32	46
21	20	24	28	41	
24	21	23	31		
23	22				34
32			30	35	34
29		33	34	36	37
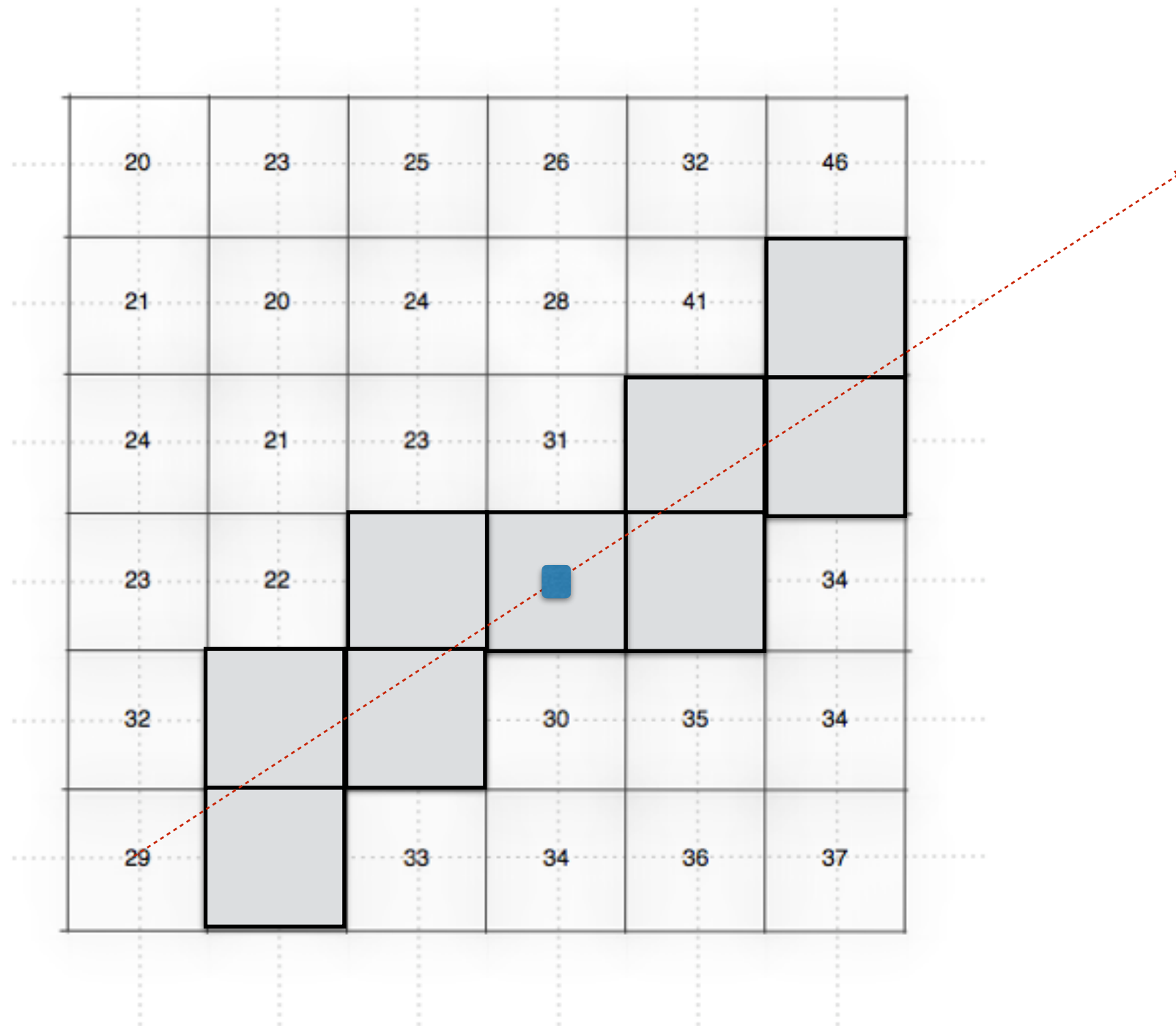
Want only the cells that are in between v and (r,c)

Analysis

- What's a good data structure for the AS?

  - Needs to be able to insert and delete cells

  - Find vertical angles of all active cells between v and given (r,c)

Class work

Consider the following grid, with viewpoint in red
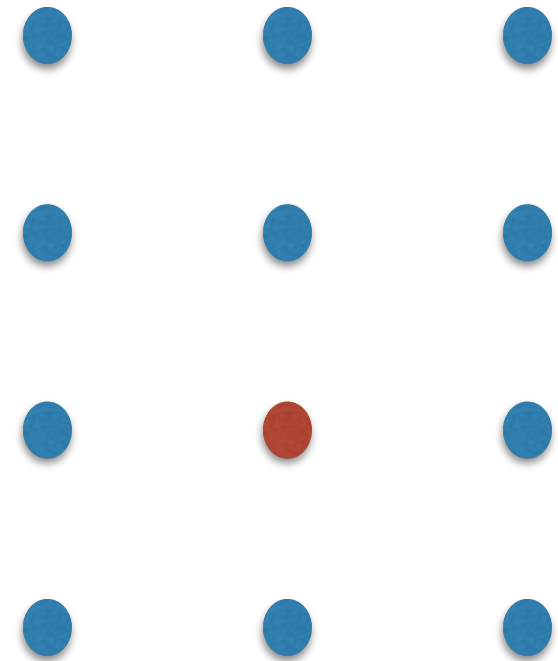
1. Number the points in order of their azimuth from v=(2,1)

2. Make a list of the cells, and their spans

3. For each cell, write down its 3 events. Pause to think about what you want to store in an "event". How many events in total?

4. Sort the events radially, and for equal azimuth, increasingly by distance from v.

5. Sketch out the comparison function that you'll pass to quicksort.

6. What events will initialize the PQ?

7. Simulate the sweep.

Computing viewsheds

1. Straightforward algorithm

- O(n$\sqrt{n}$)

- Uses linear interpolation

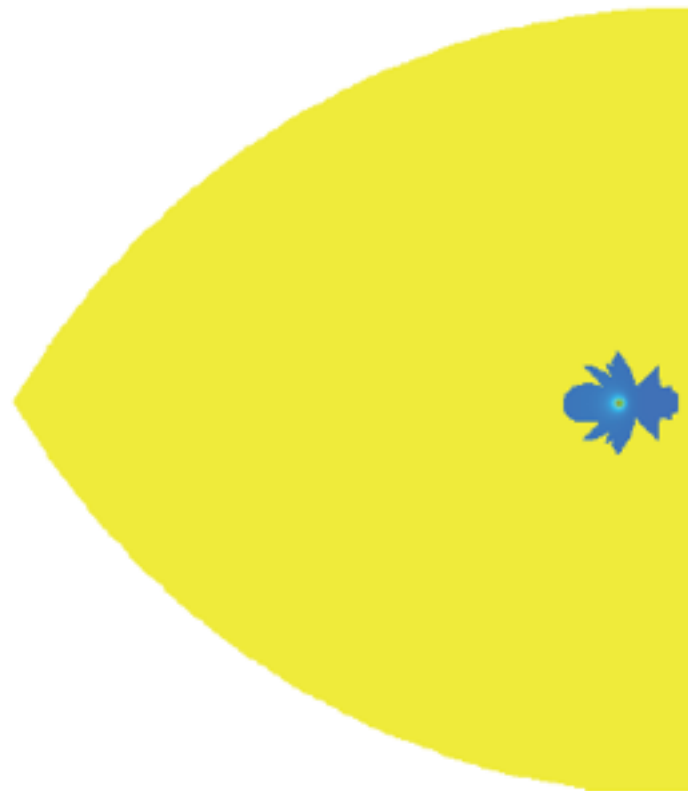- Can be adapted to other interpolations

2. Radial sweep approach

- O(n lg n)

- Uses nearest neighbor interpolation

- Not easy to extend: crucially exploits that cells are "flat"

- Nearest neighbor produces some artifacts

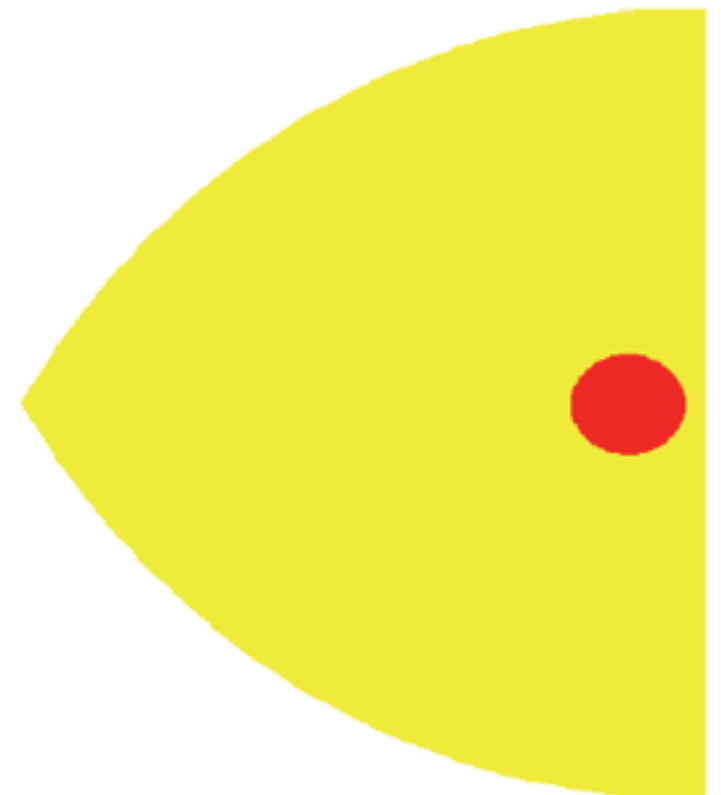3. NEXT: Concentric sweep and horizons

Accuracy!!

test grid: hemisphere   viewshed with NN interpolation   viewshed with linear interpolation
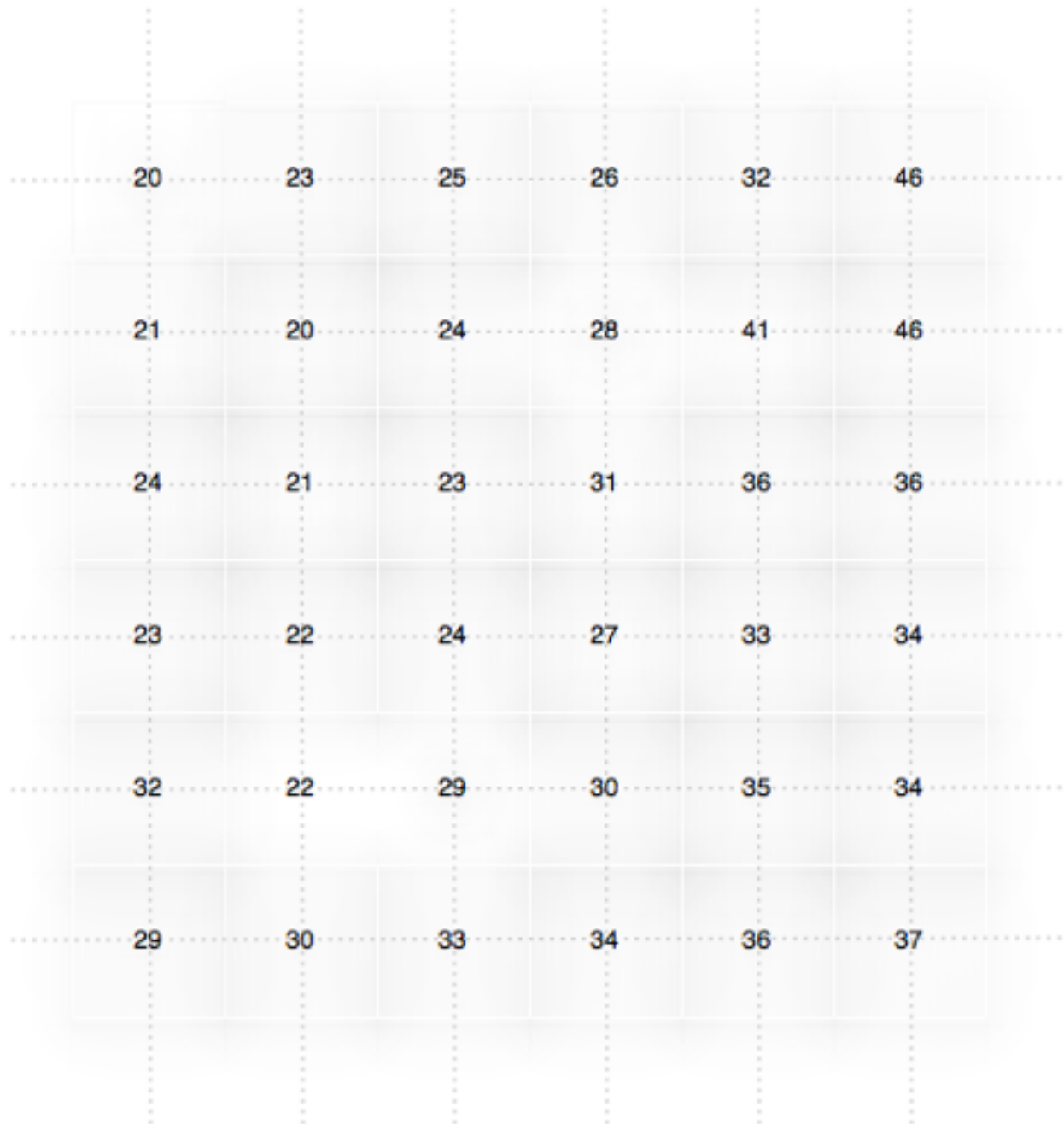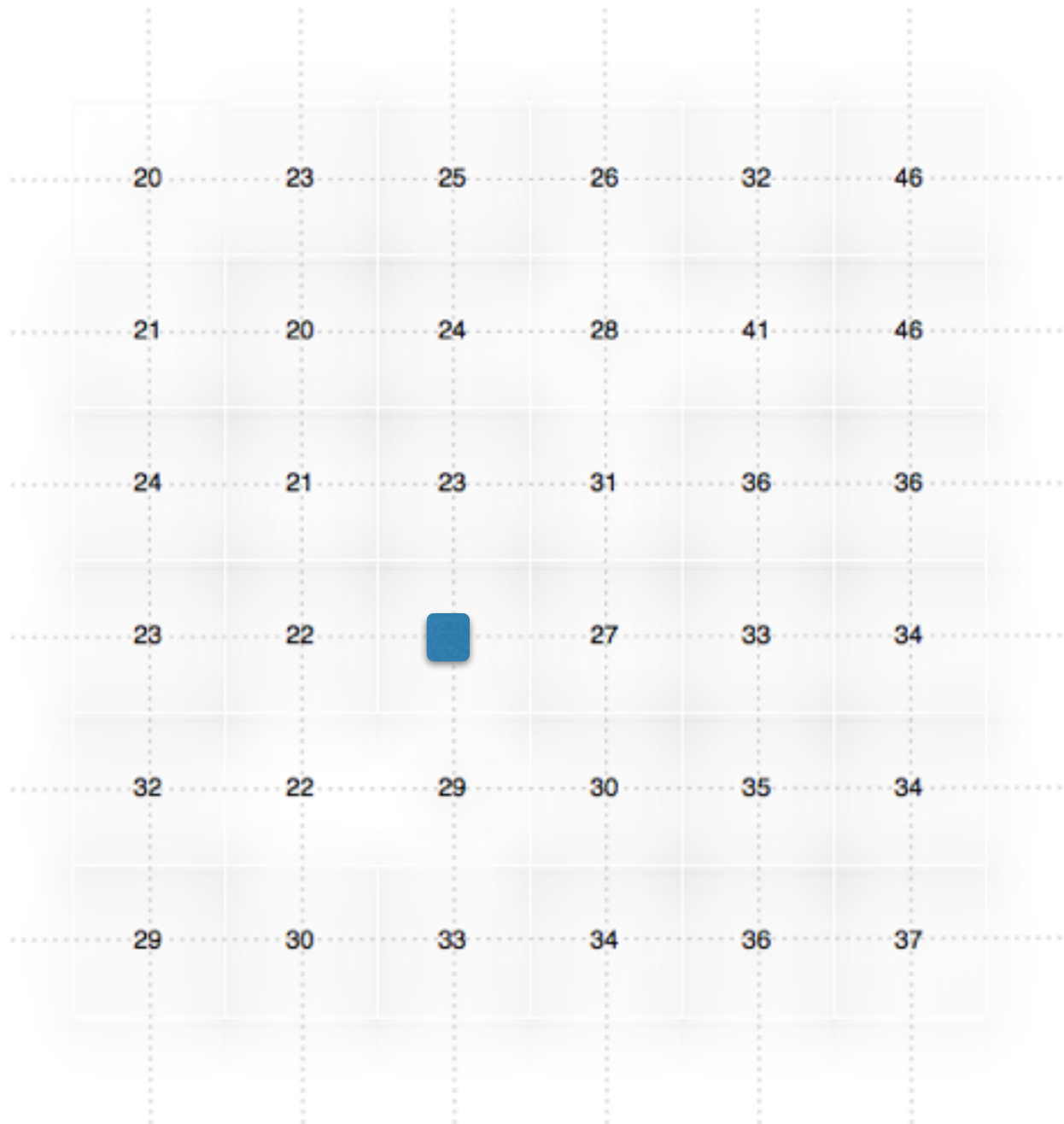
Viewsheds on grids:

3. Viewsheds and horizons

Ingredient 1:   concentric sweep
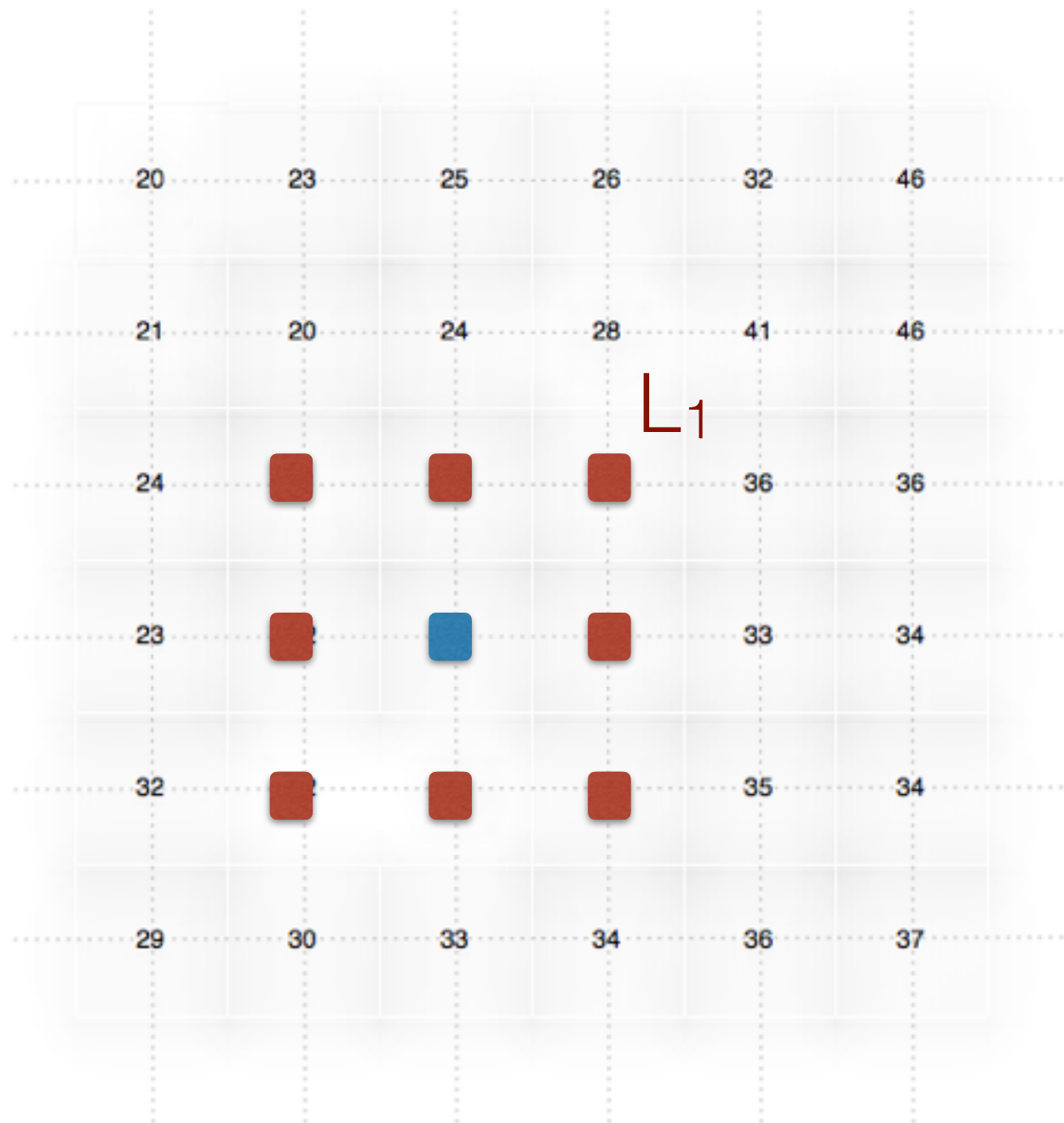
Ingredient 2:   horizons

Concentric sweep

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

Concentric sweep

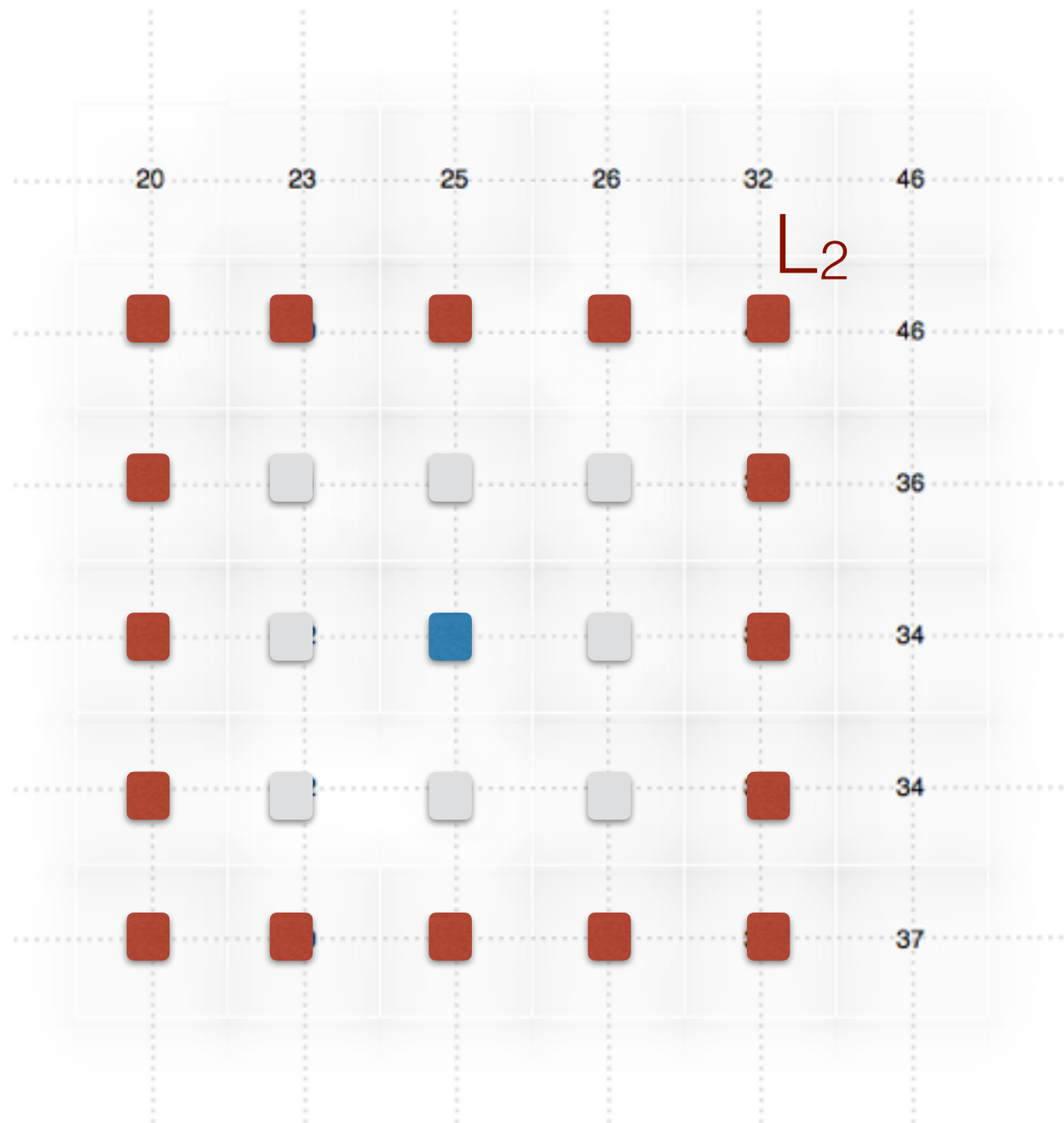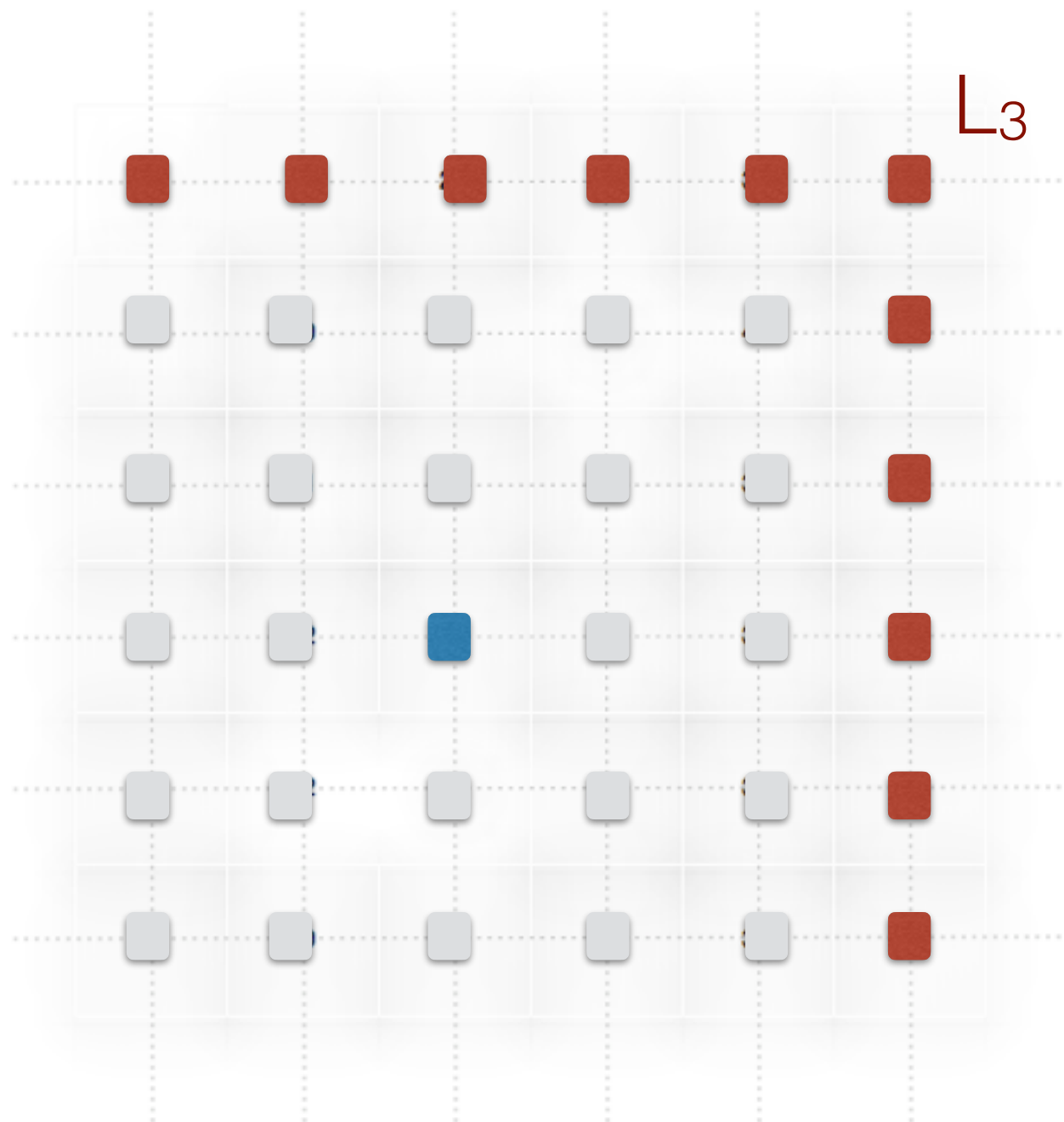| 20 | 23 | 25 | 26 | 32 | 46 |
21	20	24	28	41	46
24	21	23	31	36	36
23	22	▣	27	33	34
32	22	29	30	35	34
29	30	33	34	36	37

Concentric sweep

Concentric sweep

Concentric sweep

Horizons

- **Merriam Webster:**

    - the line where the terrain and the sky seem to meet

Horizons

- Merriam Webster:

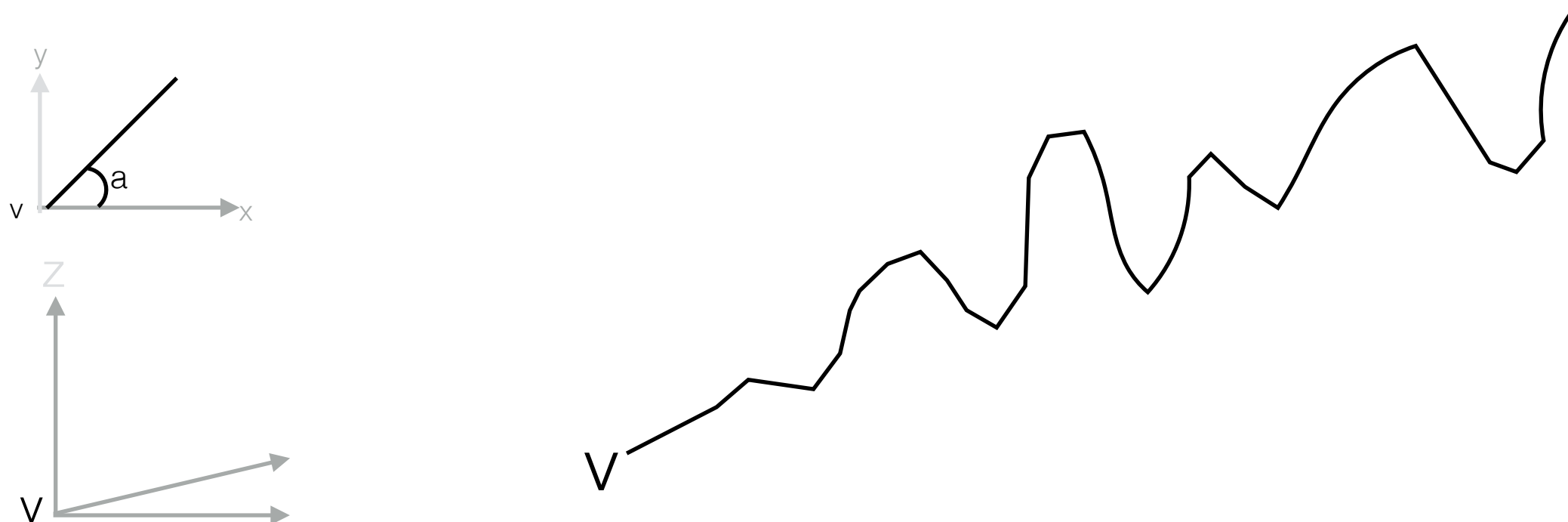 - the line where the terrain and the sky seem to meet

Horizon

$H_v : [0, 2PI) \longrightarrow R$

$H_v(a)$: horizon (with respect to v) in direction a

- cut the terrain with a vertical plane through ray from v of azimuth a

- $H_v(a)$ is the maximum vertical angle (zenith) of all points intersected by this plane (all the points on T whose projection on the xy-plane has azimuth a)
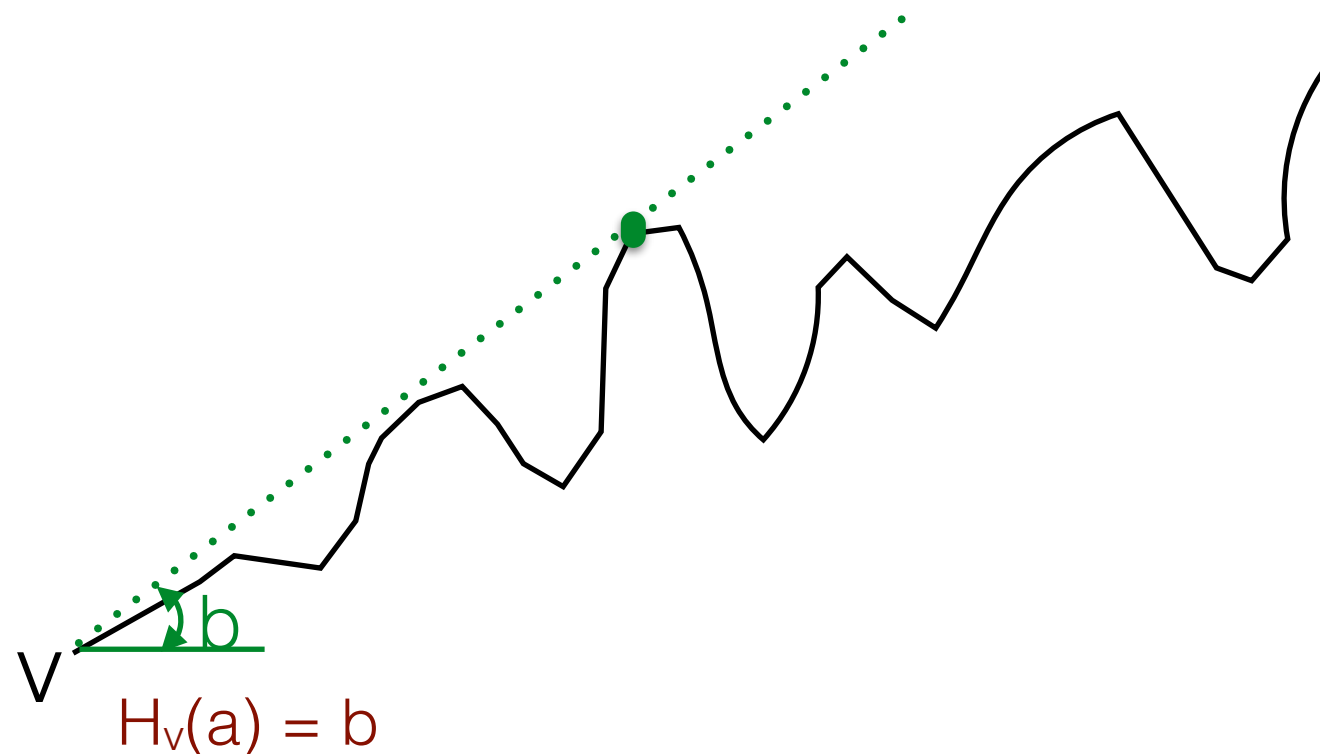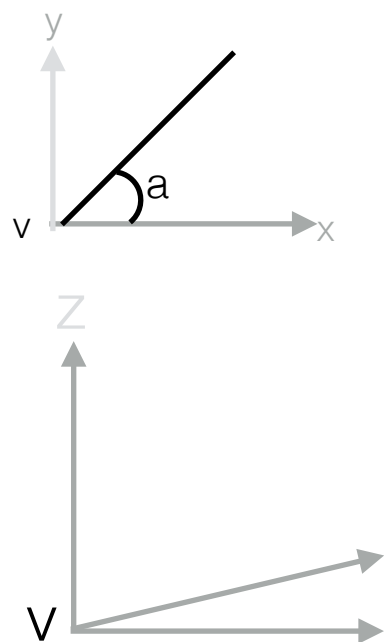
Horizon

$H_v : [0, 2\pi) \longrightarrow R$

$H_v(a)$: horizon (with respect to v) in direction a

- cut the terrain with a vertical plane through ray from v of azimuth a

- $H_v(a)$ is the maximum vertical angle (zenith) of all points intersected by this plane (all the points on T whose projection on the xy-plane has azimuth a)
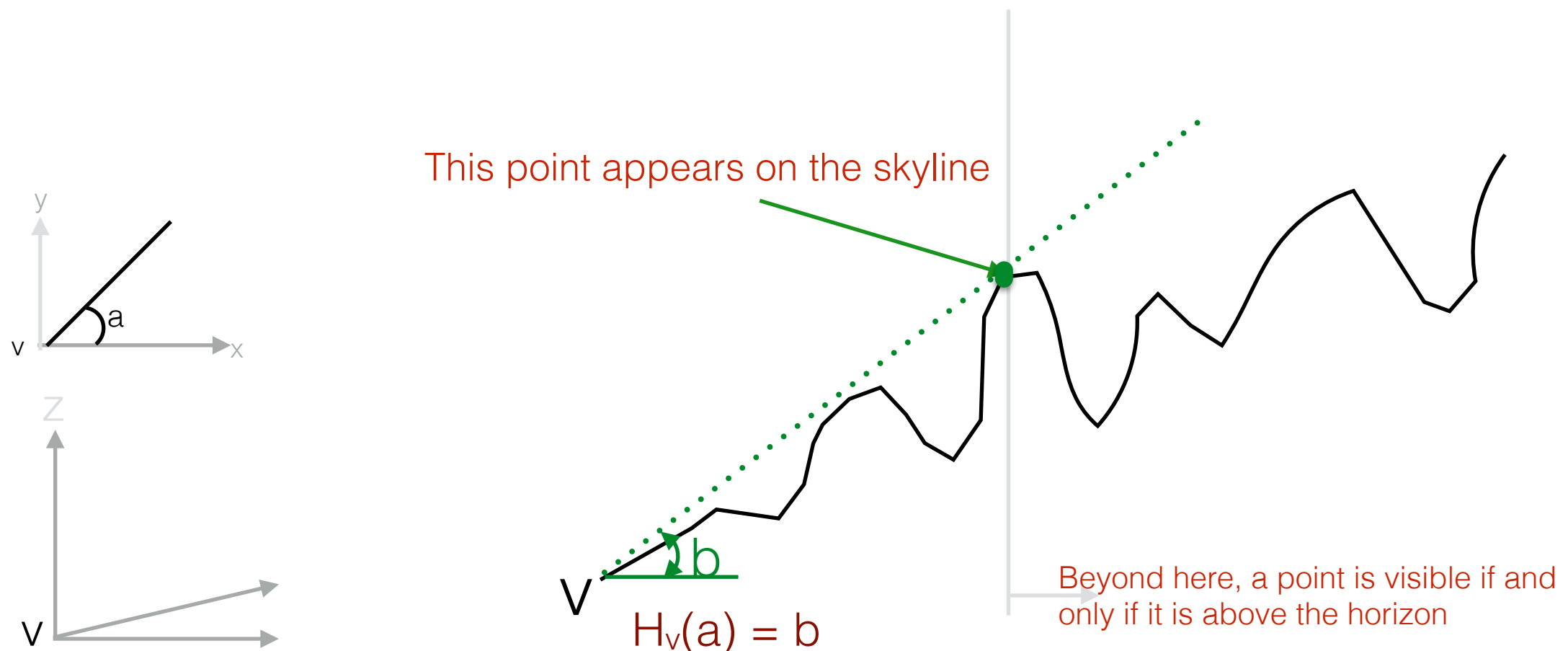

$H_v(a) = b$

Horizon

$H_v : [0, 2\text{PI}) \longrightarrow R$

$H_v(a)$: horizon (with respect to v) in direction a

- cut the terrain with a vertical plane through a ray from v of azimuth a

- $H_v(a)$ is the maximum vertical angle (zenith) of all points intersected by this plane (all the points on T whose projection on the xy-plane has azimuth a)
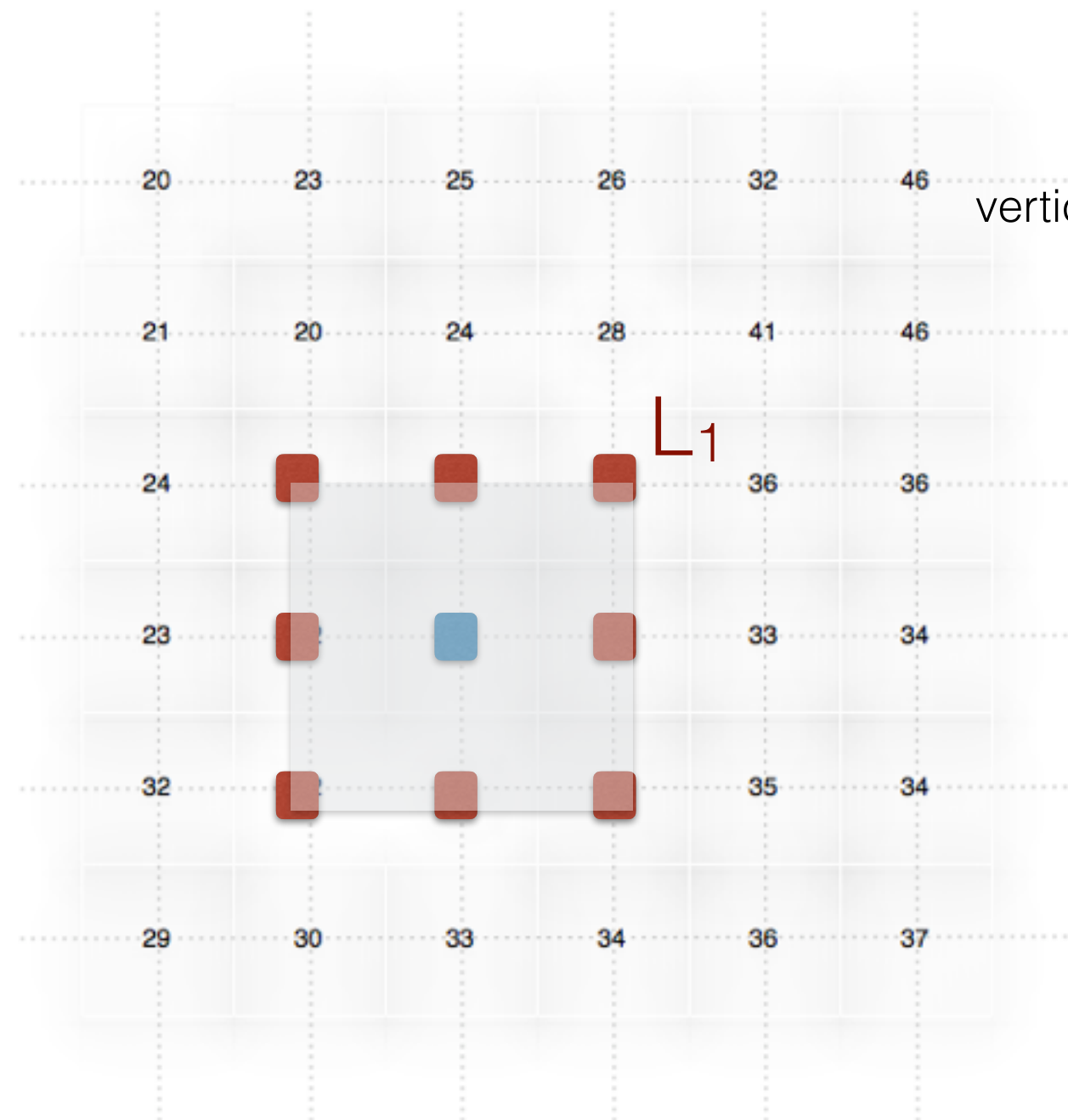
This point appears on the skyline

Beyond here, a point is visible if and only if it is above the horizon
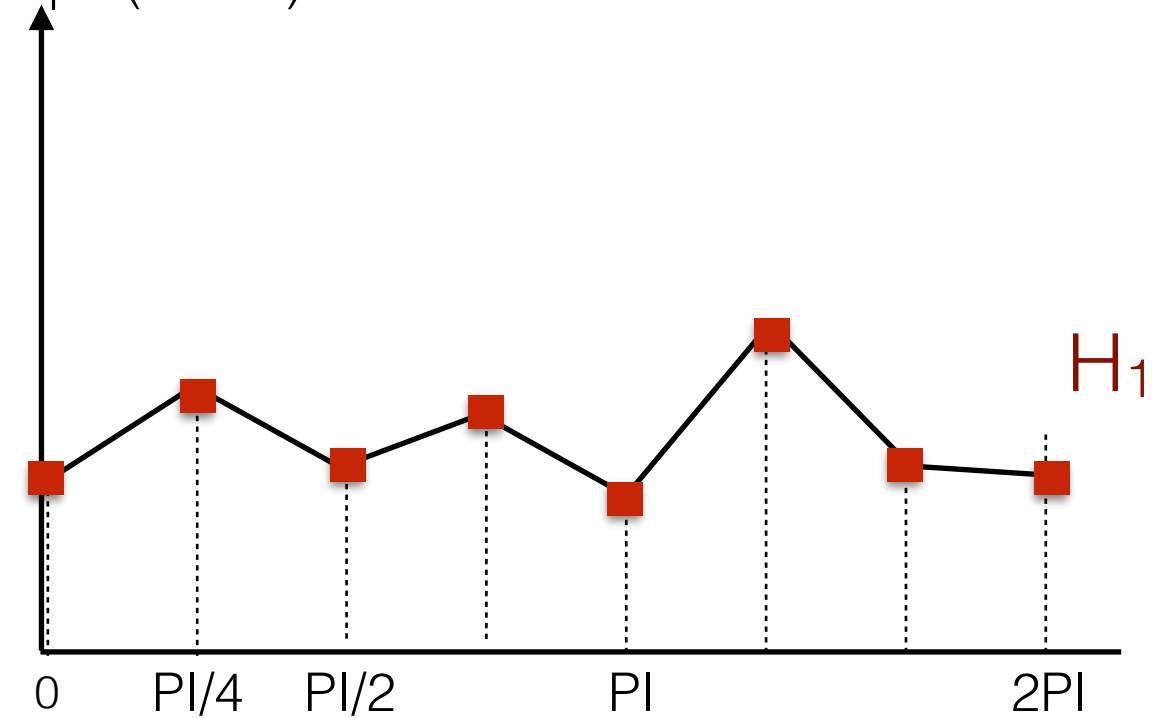
$H_v(a) = b$

A point is visible if it is above the horizon.

Compute the viewshed by computing horizons.

Viewshed and horizons
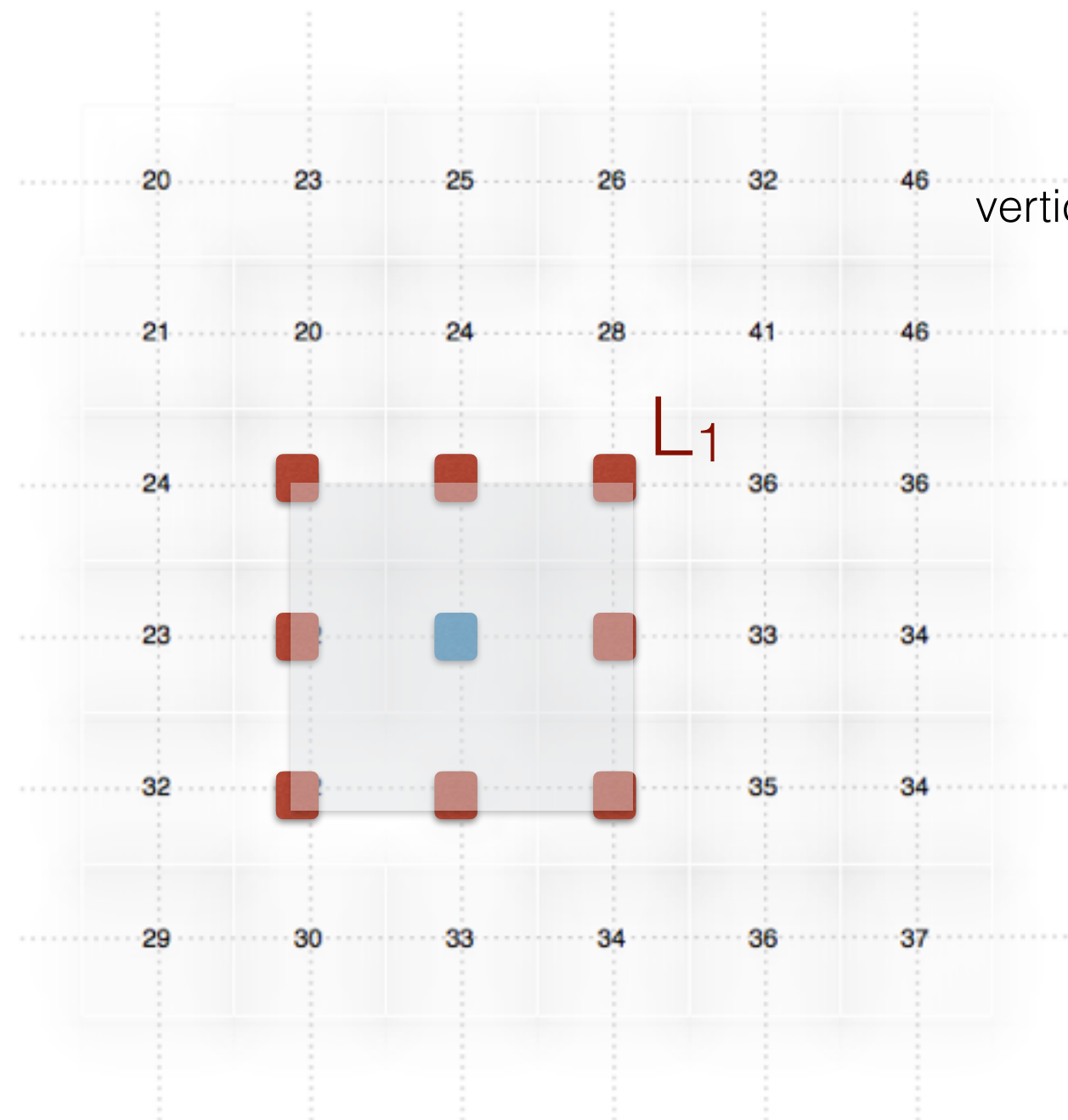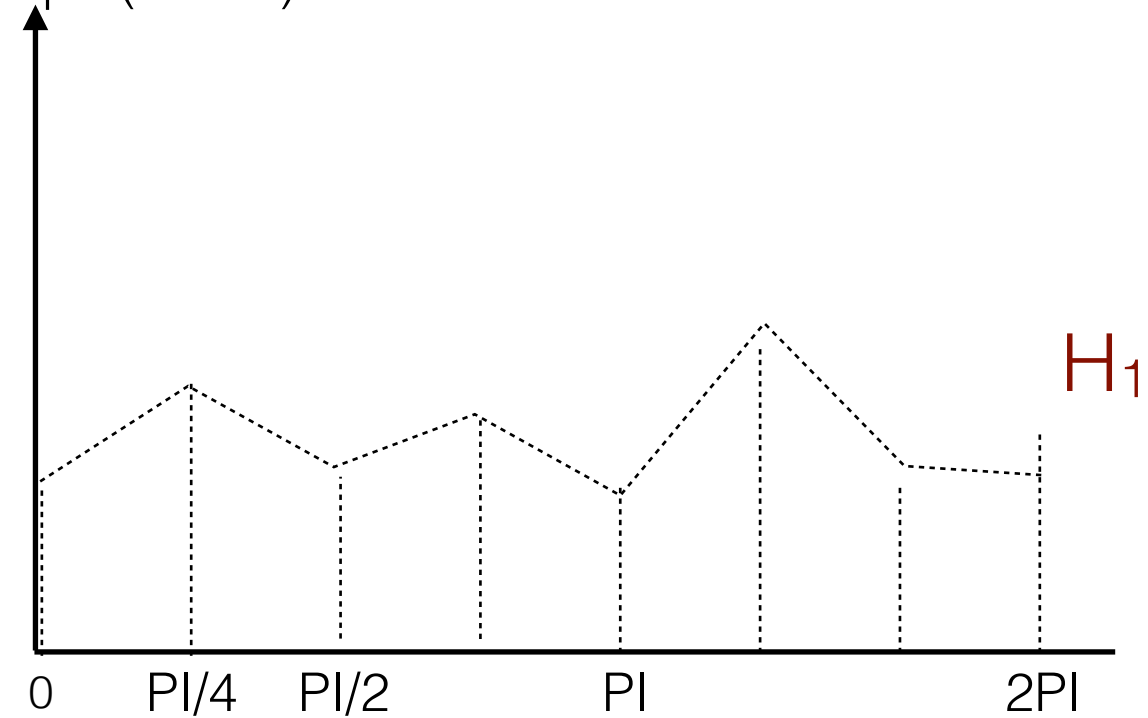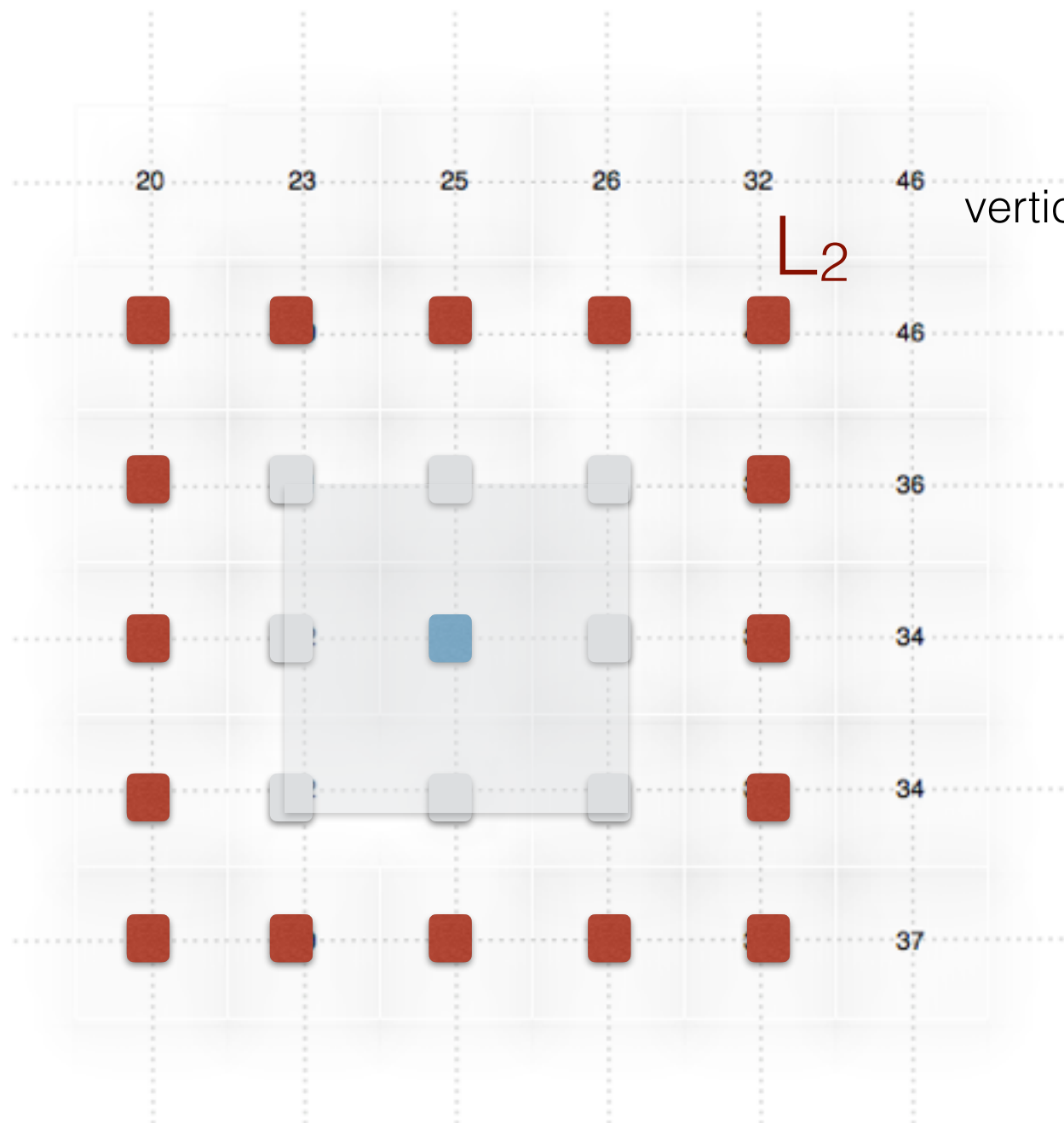
Viewshed and horizons

Viewshed and horizons
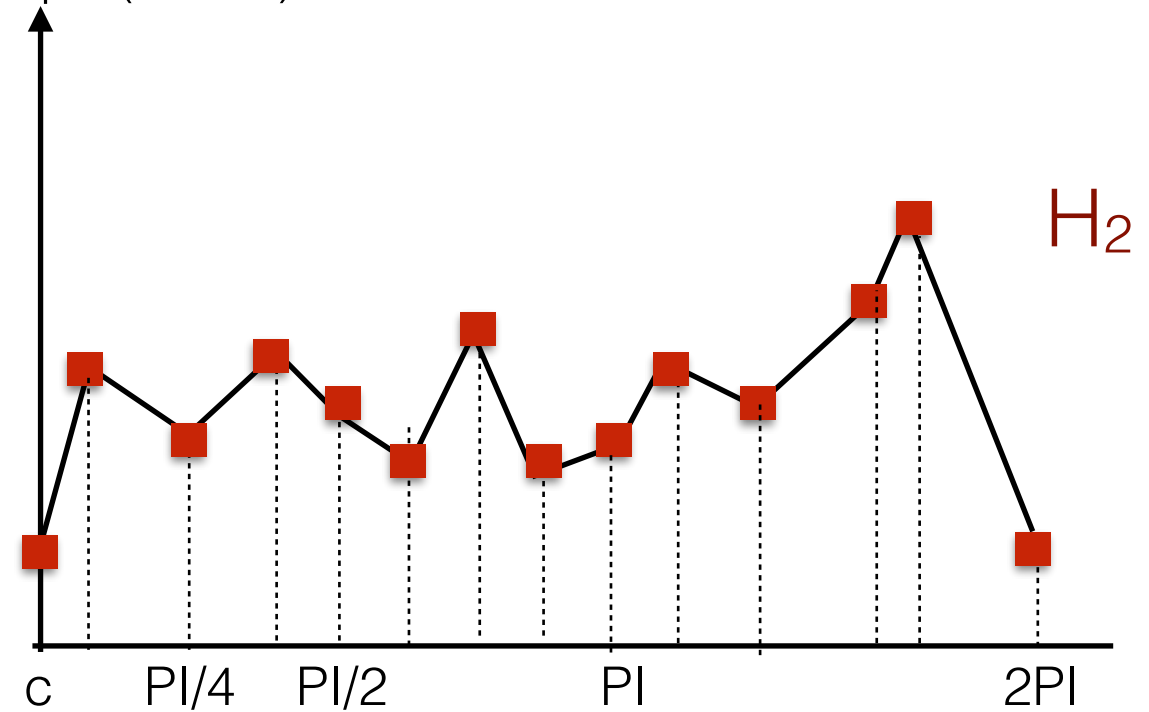
Viewshed and horizons

Viewshed and horizons

Viewshed and horizons

Viewshed and horizons

Viewshed and horizons

L_3

H_{12}
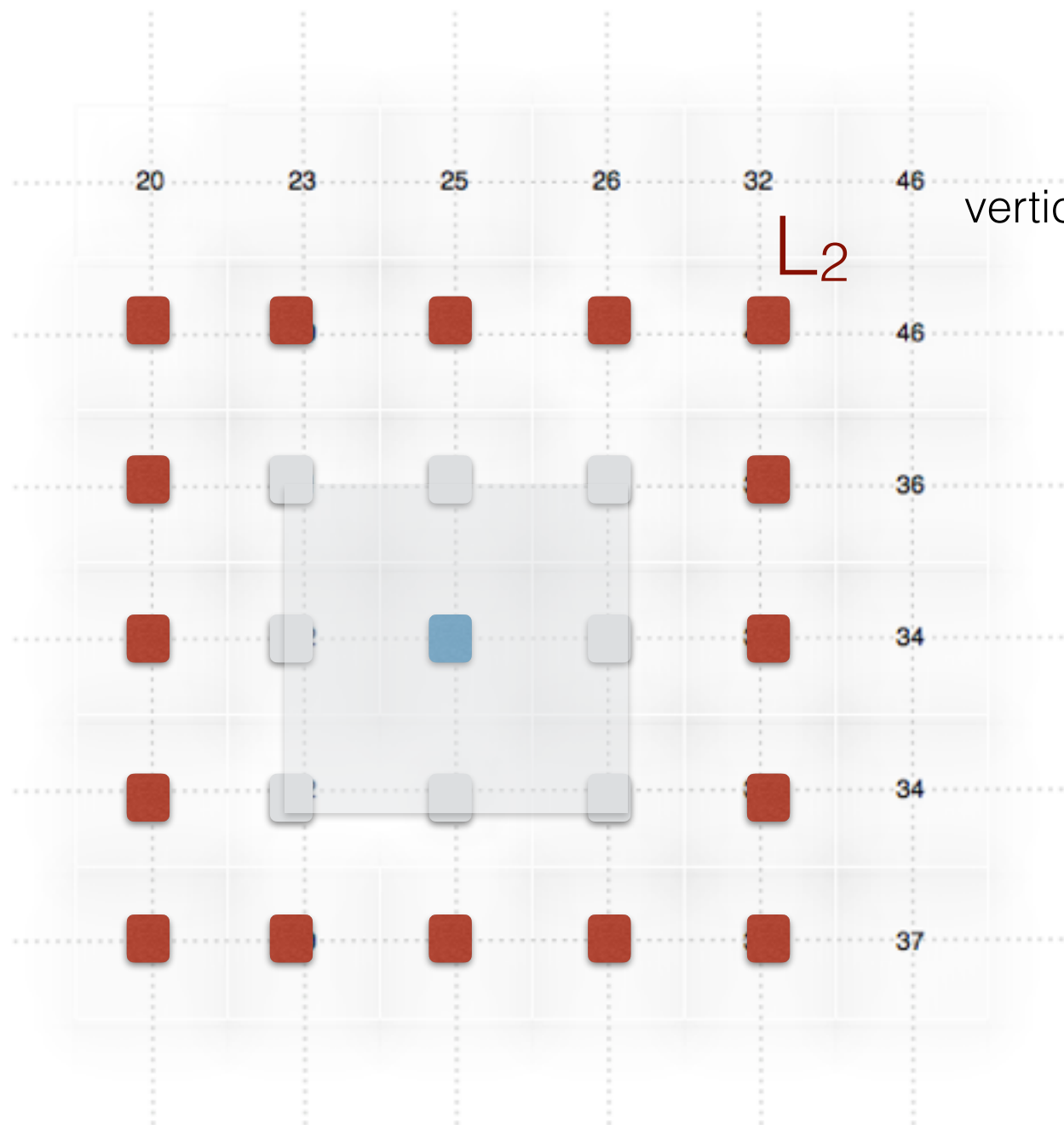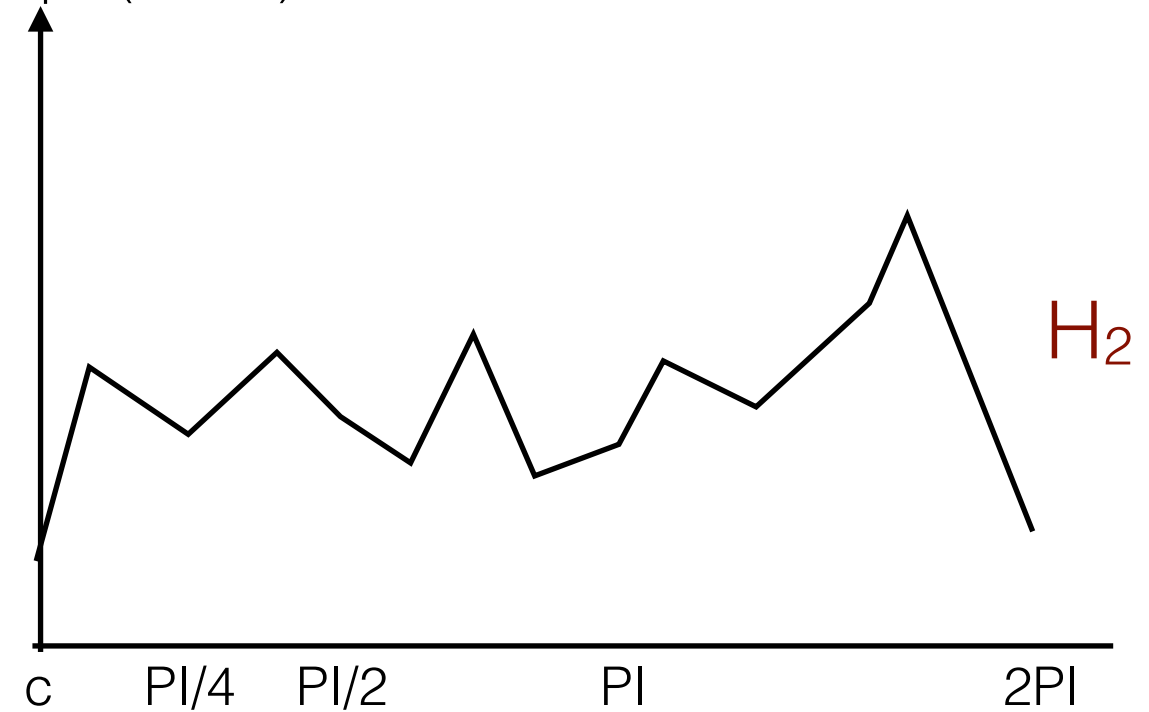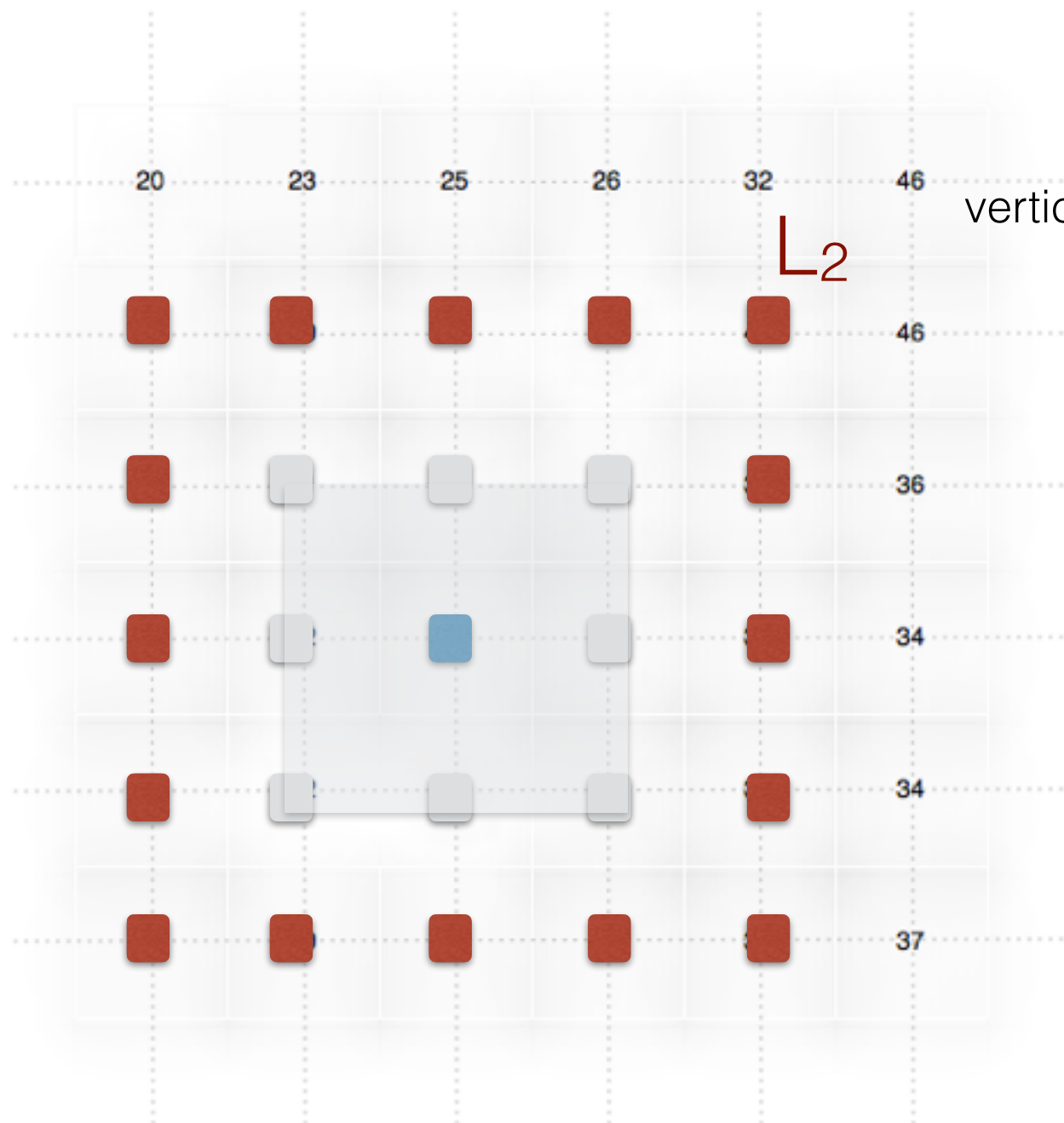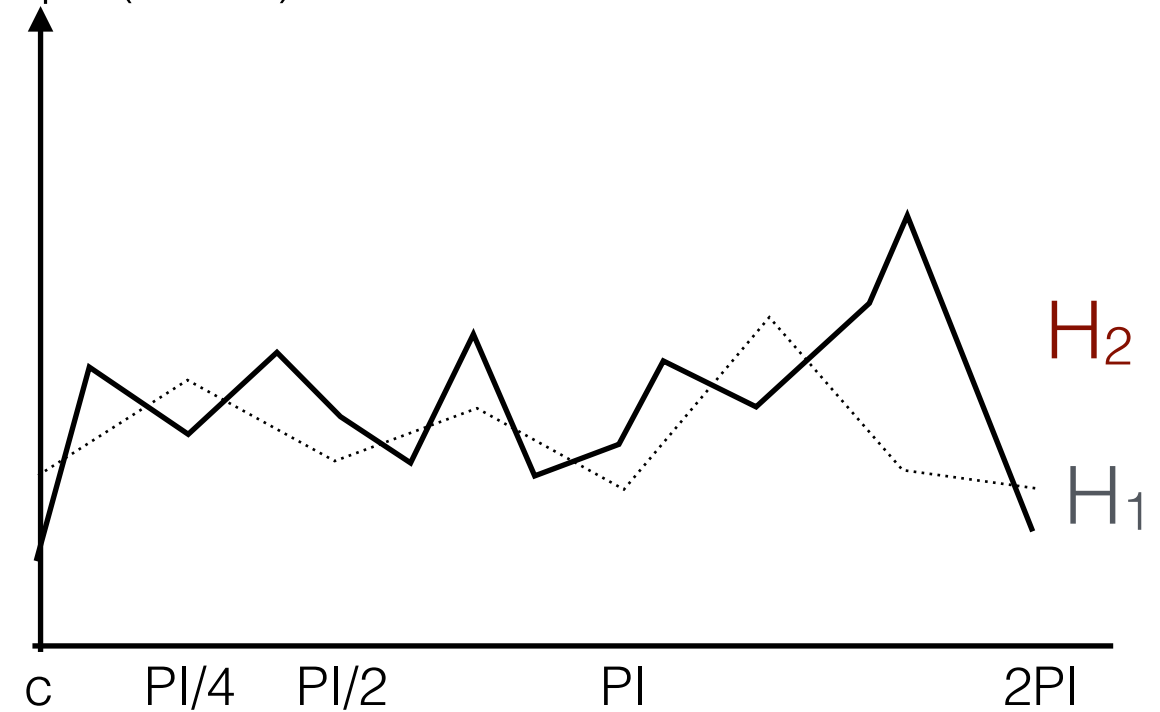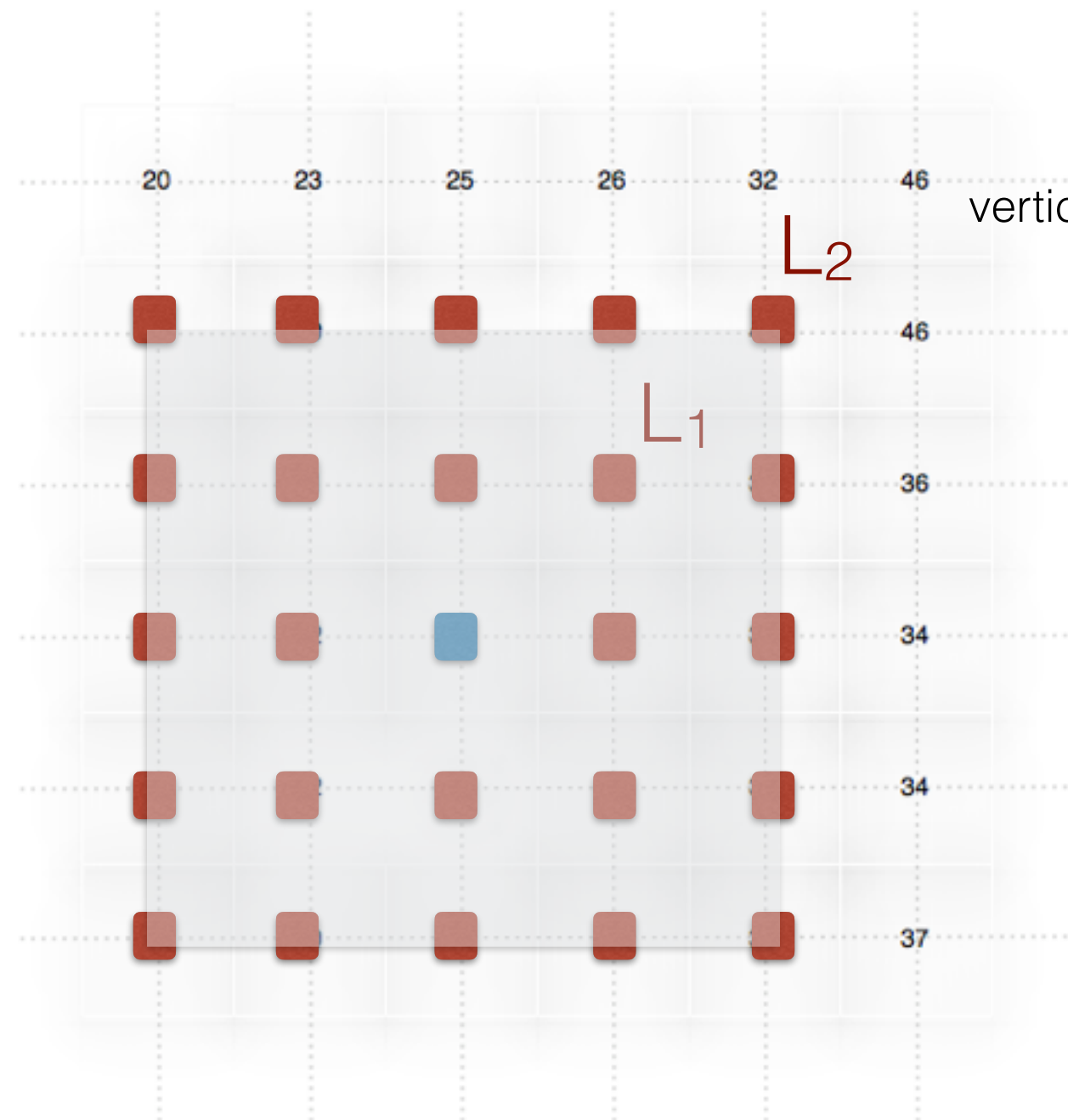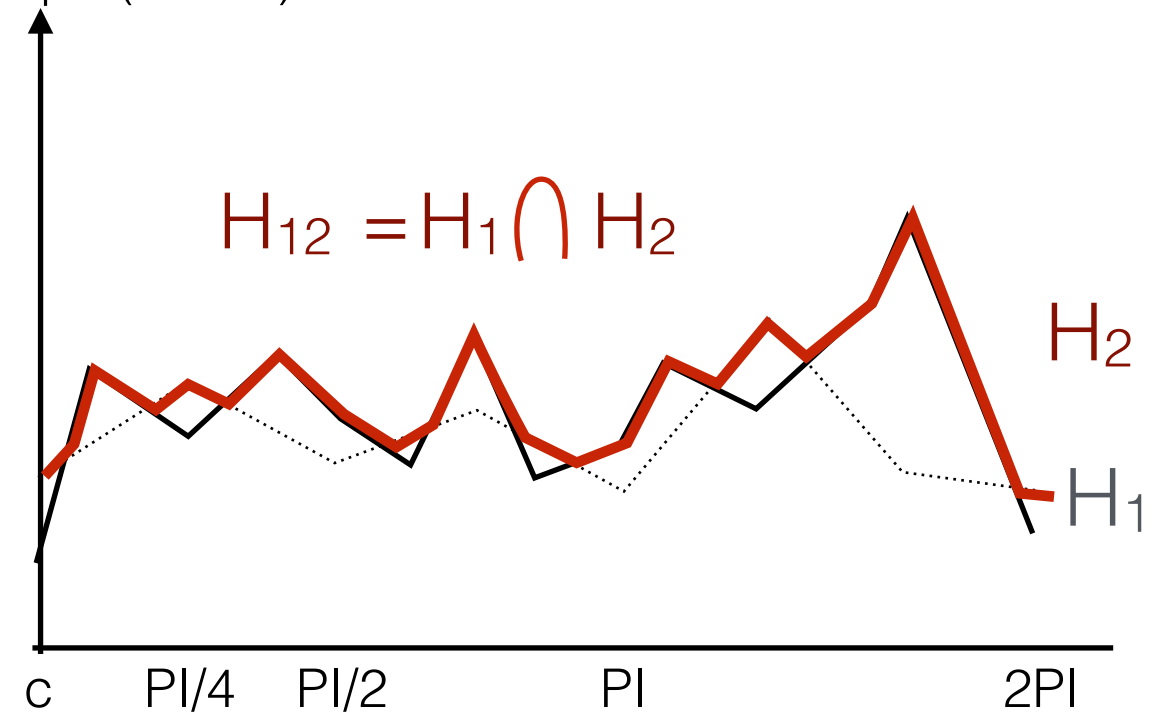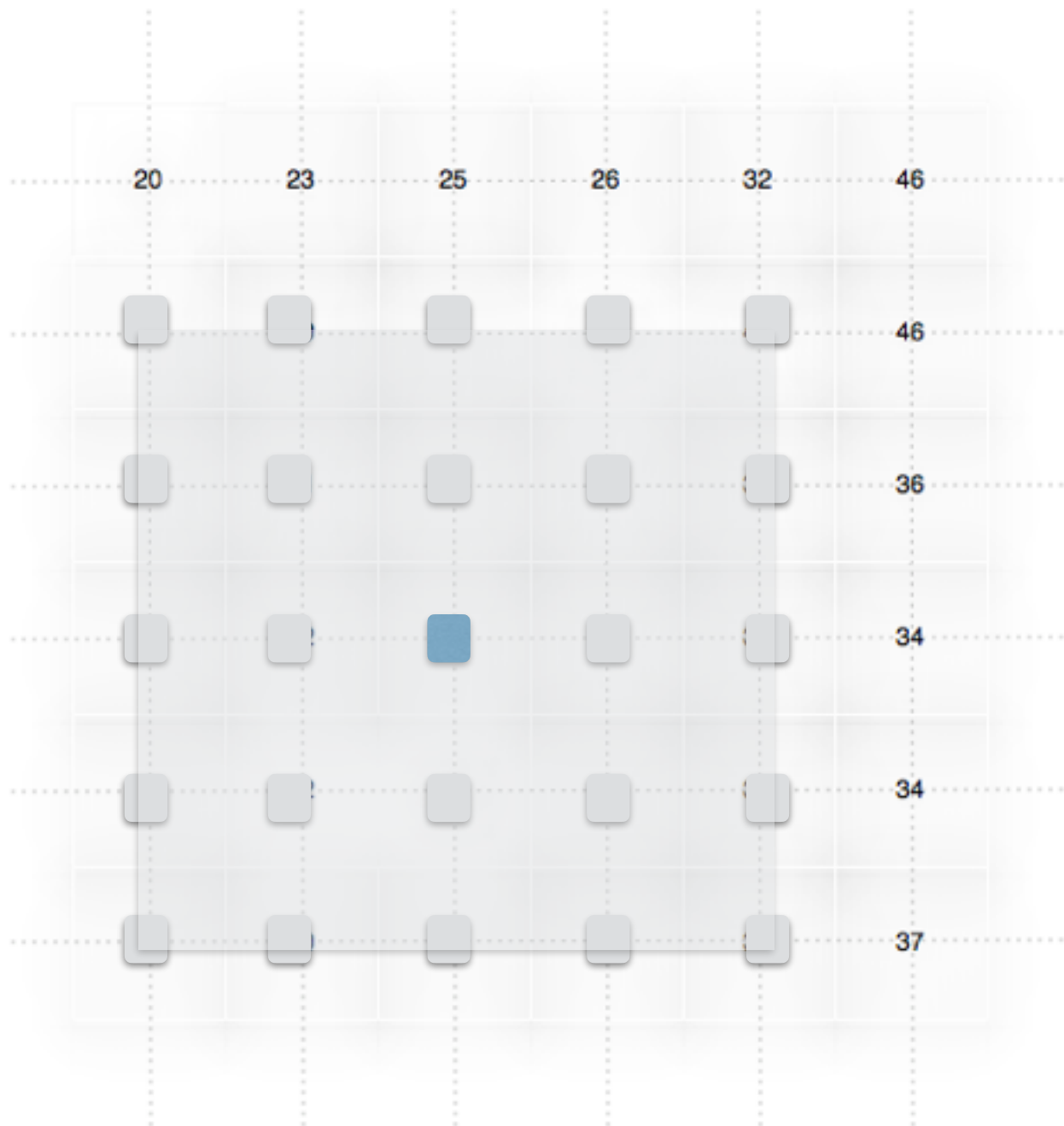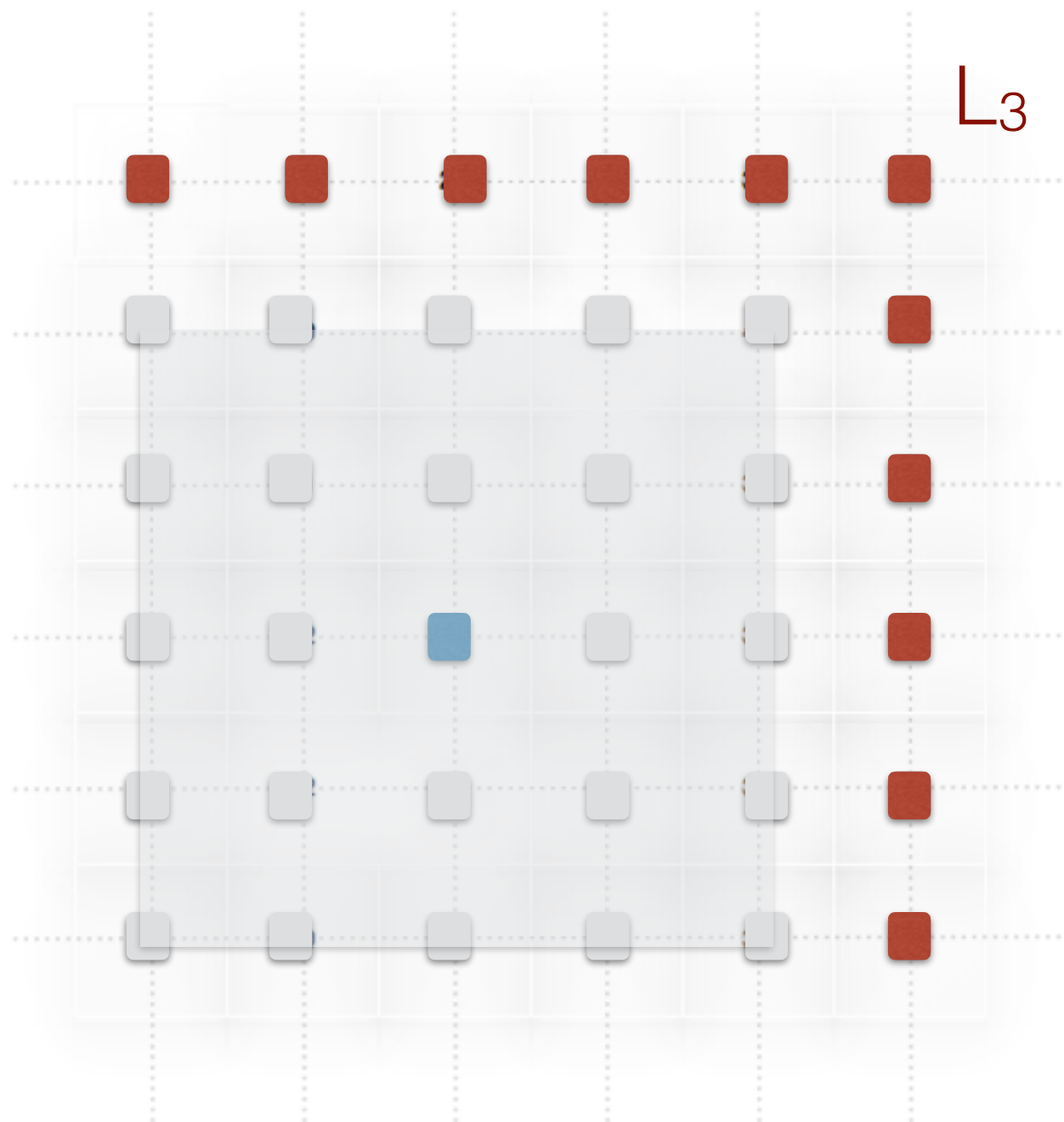
We walk along L3, computing the horizon of L3 and determining if points on L3 are visible or not

Viewshed and horizons

p

L_3

a
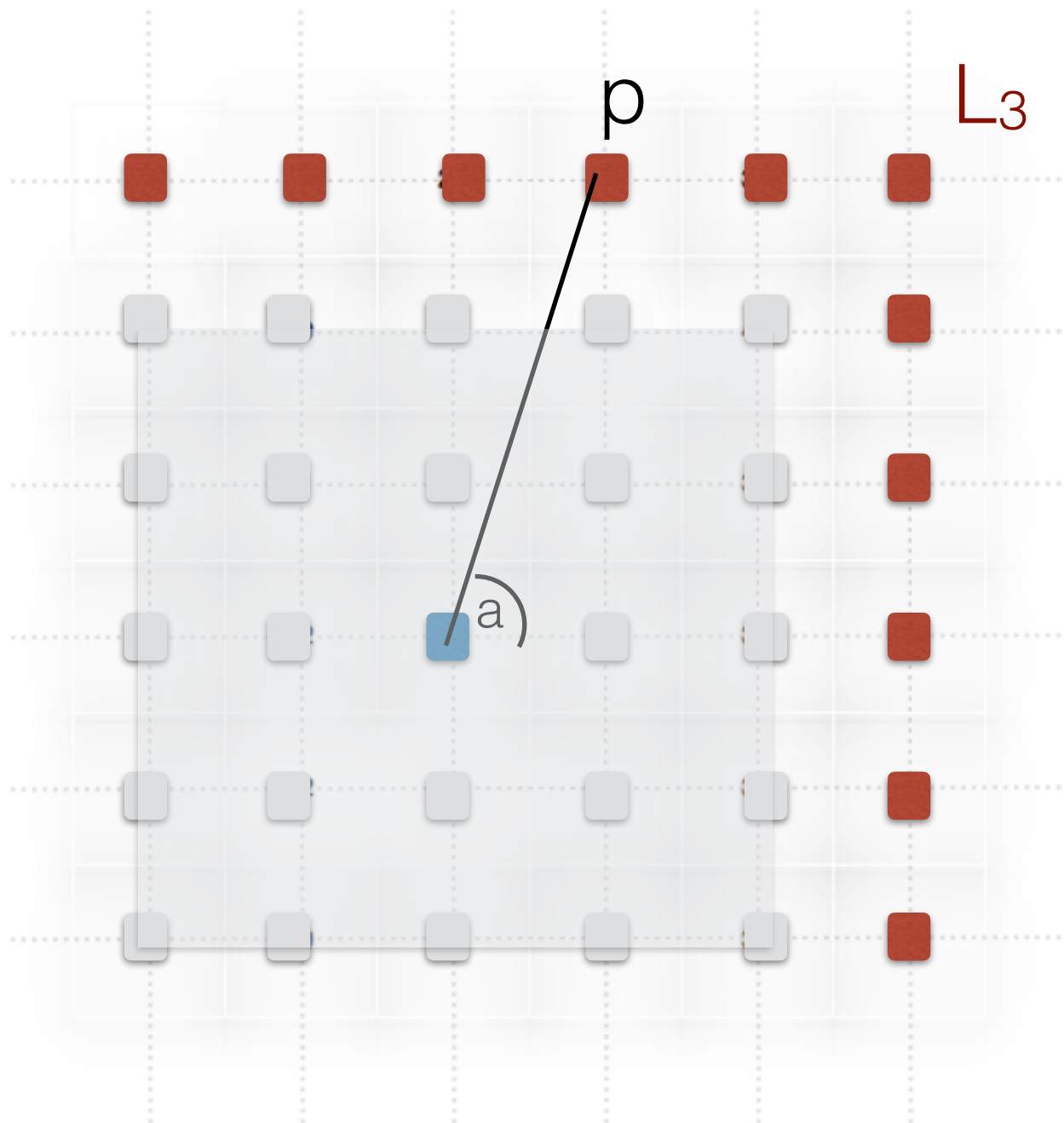
Is point p in L3 visible ?

p is visible if
slope(vp) < H_{12}(a)

H_{12}

a
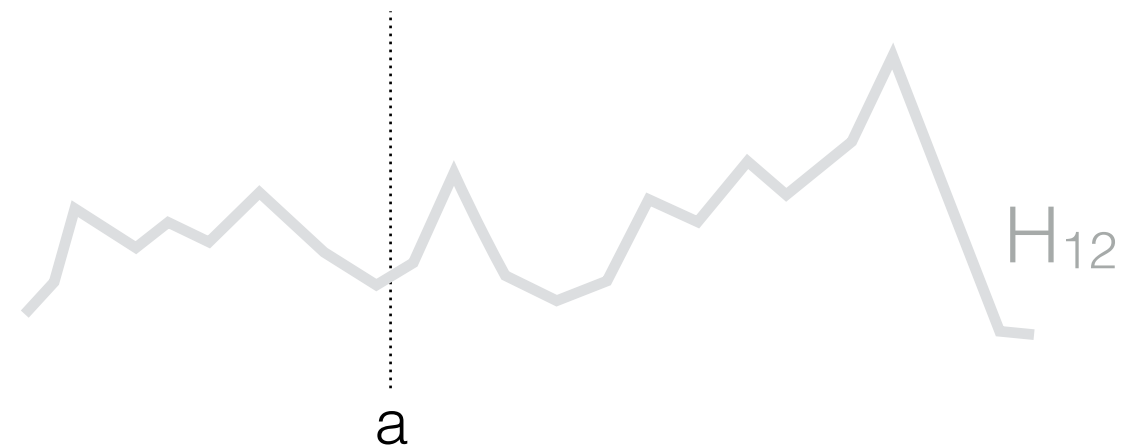
Viewshed and horizons

- Elegant techniques that can be extended

  - Linear interpolation or nearest neighbor,…

  - Starting point for triangulated terrains

- Worst-case bounds not great

  - fast in practice because horizons stay very small

Viewsheds on triangulated terrains

Viewsheds on triangulated terrains

- viewshed(p) contains all points of the terrain that are visible from p

from: http://arxiv.org/pdf/1309.4323.pdf
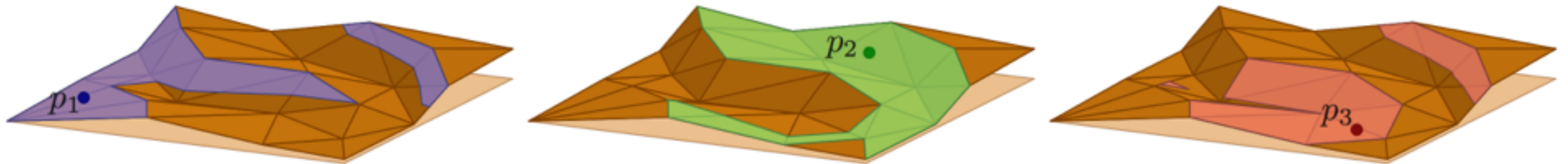
- viewshed(p) may intersect a triangle multiple times.

Viewsheds on triangulated terrains

- viewshed(p) contains all points of the terrain that are visible from p

from: http://arxiv.org/pdf/1309.4323.pdf
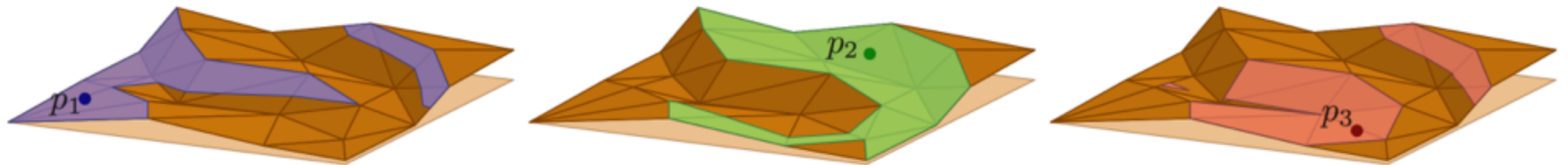
- viewshed(p) may intersect a triangle multiple times.

- How big can viewshed(p) be?

  - Space complexity = number of edges on boundary of viewshed(p)

  - It is known that the complexity of a viewshed on a triangulated terrain can be $O(n^2)$. On a triangulated grid, the complexity of a viewshed is $O(n\sqrt{n})$

  - These worst-case cases exist, but are contrived/not realistic.

  - In practice, on realistic terrains, viewsheds are small. Proving realistic upper bounds still open problem.
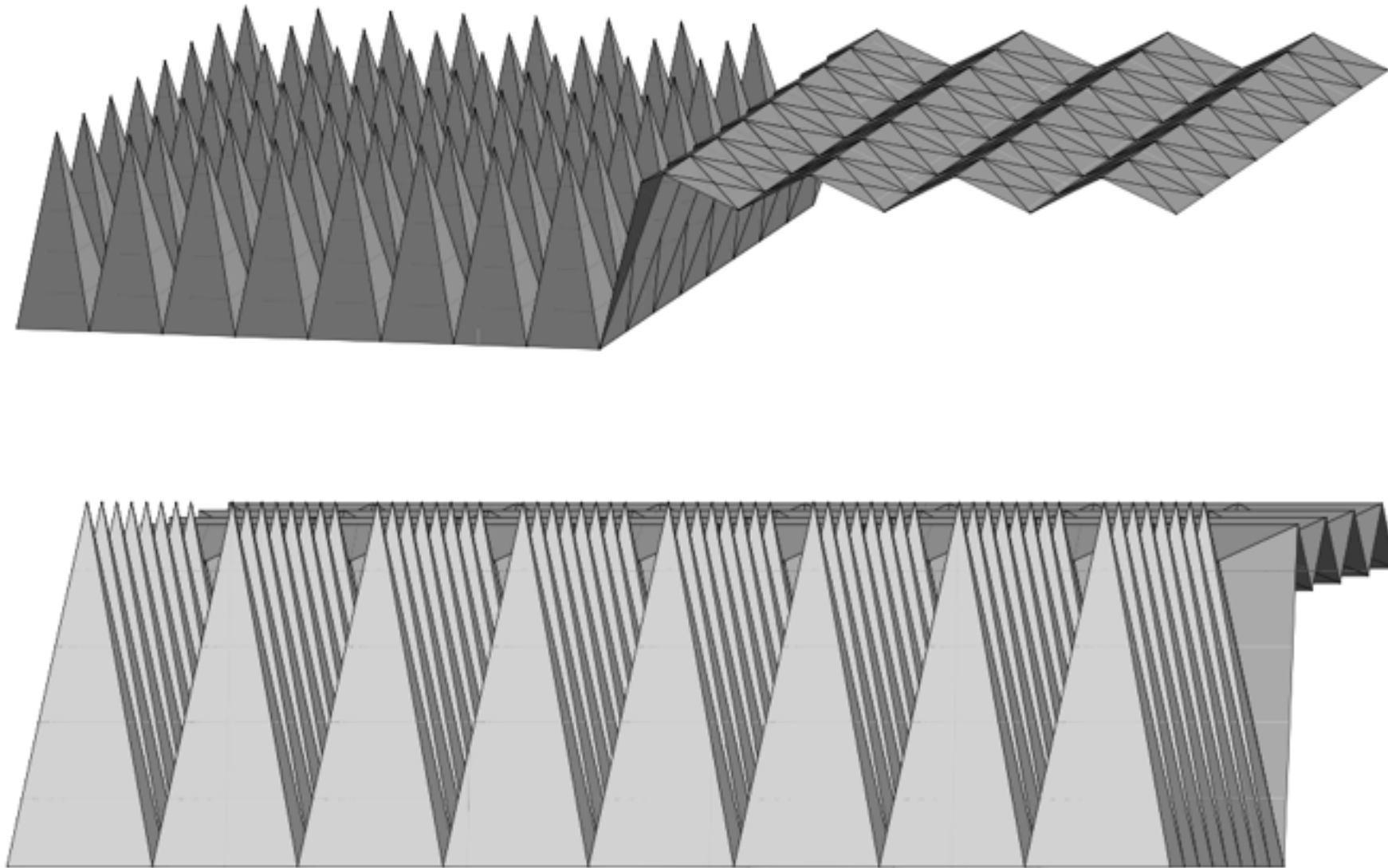
Figure 1: Two views of the same terrain defined by a regular grid. The second view gives a visibility map of complexity $\Theta(n\sqrt{n})$. Note that the terrain can be flattened further without changing the view combinatorially.

Viewsheds on triangulated terrains

- Several algorithms are known

- Based on horizons

  - Idea: traverse triangles in order of increasing distance form viewpoint, and update horizon.

  - Bootstrap with divide-and-conquer

Outline

- Viewsheds on grid terrains

    1. straightforward algorithm

    2. radial sweep algorithm

    3. viewshed via horizon

- Viewsheds on TIN terrains

    - quadratic size, worst-case construction

- Cumulative viewshed

- Total viewshed

- Find point of maximum/minimum visibility

- Find optimal paths

Beyond viewsheds

Improving Methods for Viewshed Studies in Archaeology:
The Vertical Angle

Mar Zamora

Universidad Autónoma de Madrid, Spain
mar.zamora@uam.es

Abstract

As many authors have observed, binary viewsheds are too simplistic a way to represent visibility around a particular viewpoint. Several deficiencies, very well summarized in Wheatley and Gillings (2000), must be corrected in order to make computer-generated viewsheds more realistic and geared to archaeological purposes. One of those required improvements relates to the vertical angle of vision. Viewing from a low angle gives less perception of detail than viewing from a high angle. Standard viewsheds do not allow the identification of this kind of perceptual issue. In the real world, visible areas at eye level are seen as a narrow strip; however, on the ground they can extend for many kilometres. The map thus gives a false representation of visibility. Dividing the viewshed calculation into several vertical angles helps to analyze the result in a more realistic way than is customary, especially in warlike contexts where dominant visibility could have been important for military purposes.

1. The importance of visual control during the Late Iron Age in Spain

During the Roman conquest period in Spain (II-I centuries BC), in several areas of Andalusia, some new settlements were located on hilltops, with extensive visibility.

This fact has been interpreted in two ways:
- as a wish to visually supervise indigenous settlements;
- as a way to show Rome's presence in the area, and to reinforce its power.

In particular, in the Guadalquivir River Valley, this fact has been observed in two adjoining areas. Romo *et al.* detected two new settlements on hilltops during the Roman republican period in Gilena. The authors think that these sites were established in order to (visually) control indigenous settlements, in the unstable context of the beginning of the roman conquest of Hispania (Romo *et*

similar phenomenon has been observed east of these zones, in an area shared by the current provinces of Seville and Cordoba, in the Genil River Valley (Zamora in press).

1.1. The Priego-Alcaudete basin

The Priego-Alcaudete basin is located east of the Sierras Subbéticas, in the southern part of the provinces of Cordoba and Jaen (Andalusia, Spain). The area is adjacent to the Genil River Valley (*Fig. 1*).

During the Middle Iberian period (prior to the beginning of the Roman conquest) there was no

Total viewshed

- Input: elevation grid G

- Output: TV grid

    - TV(i,j) = nb. visible points in viewshed(i,j)

- Algorithm?

- Running time?
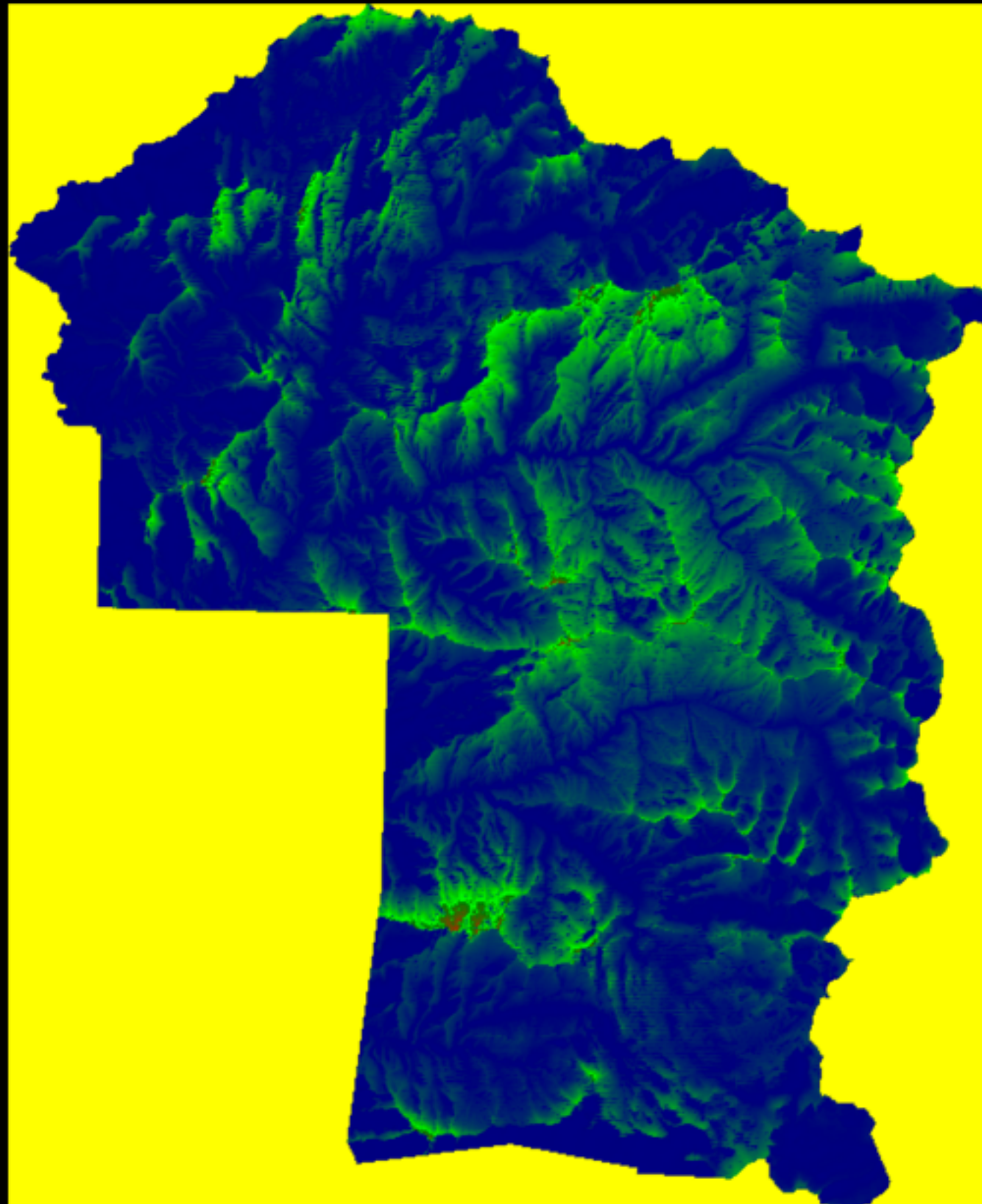
total viewshed on kaweah (1M points)
time: 42.6 hrs

Summary

- Viewshed

    - Straightforward solution

        - Reasonably fast even for very large terrains (as long as they fit in memory)

    - Refined solutions expose elegant ideas

        - Radial sweep + augmented RB-trees

        - Horizons

        - Carry on to triangulated terrains

    - Accuracy

        - Interpolation is important

- Total viewshed

    - Straightforward solution: $O(n^2\sqrt{n})$

    - Refined: $O(n^2 \lg n)$

    - Too slow…

Need better algorithms!
Need parallel algorithms!