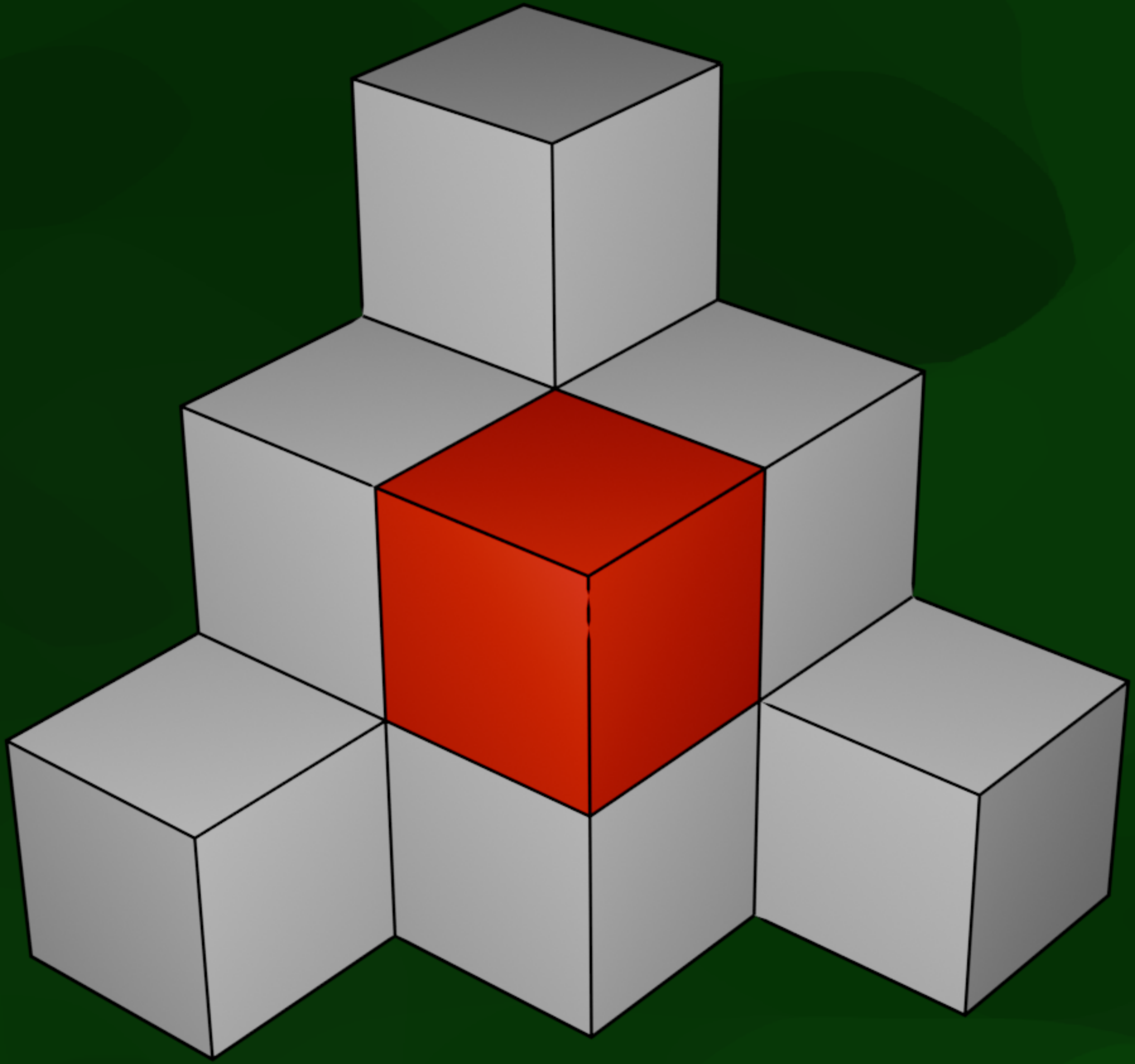
Voxels and voxelisation

GEO1004:
3D modelling of the built environment

https://3d.bk.tudelft.nl/courses/geo1004

3D geoinformation
Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology

Voxels: 3D pixels



Voxel models

Why?

- Simplicity: simple storage, simple processing methods, simple to use in applications
- Robustness: many issues solved automatically or at least more easily, few degenerate cases
- Speed: easy to do parallel processing by processing voxels independently

Nomenclature

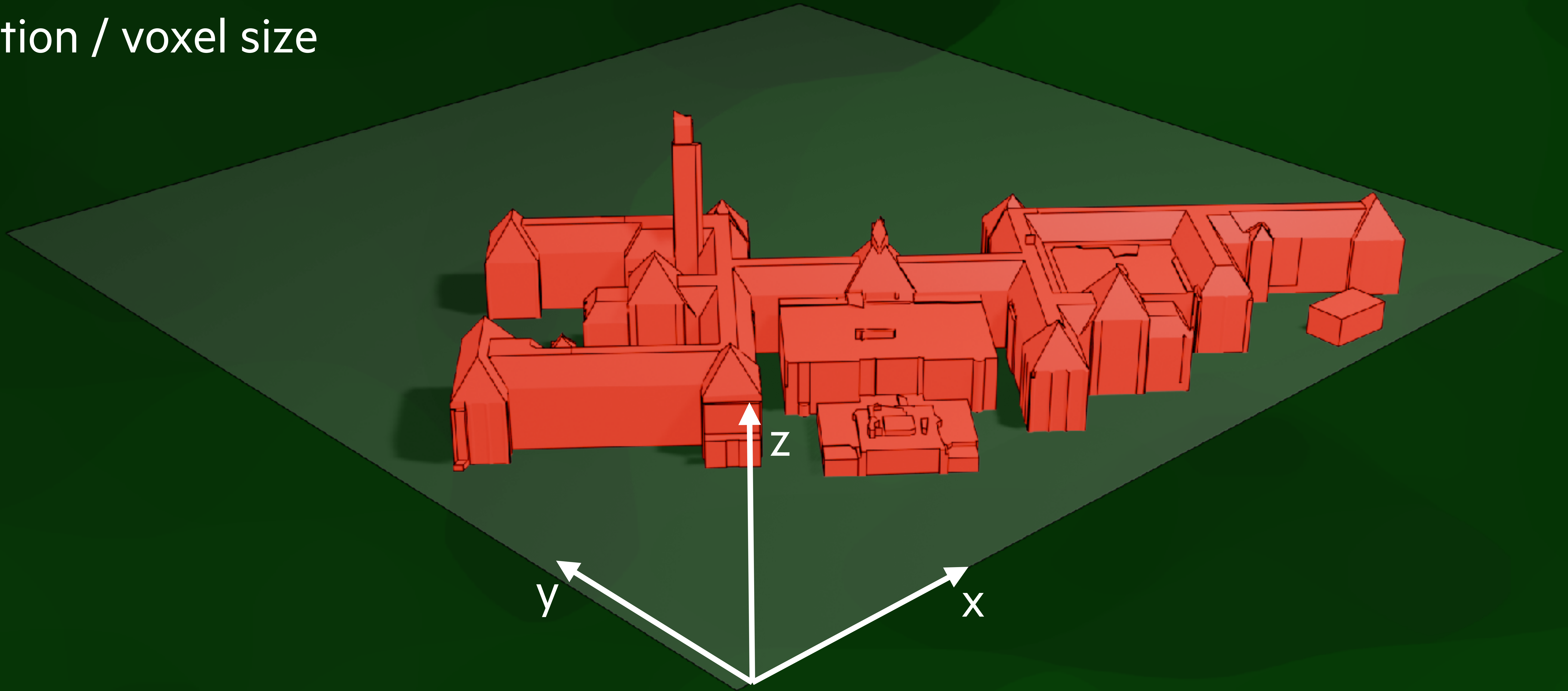
- 2D:
  - pixel
  - image / 2D raster
  - (2D) rasterisation

- 3D:
  - voxel
  - voxel model / 3D raster
  - voxelisation / 3D rasterisation

Setup

1. Define domain (origin and orientation / axes)

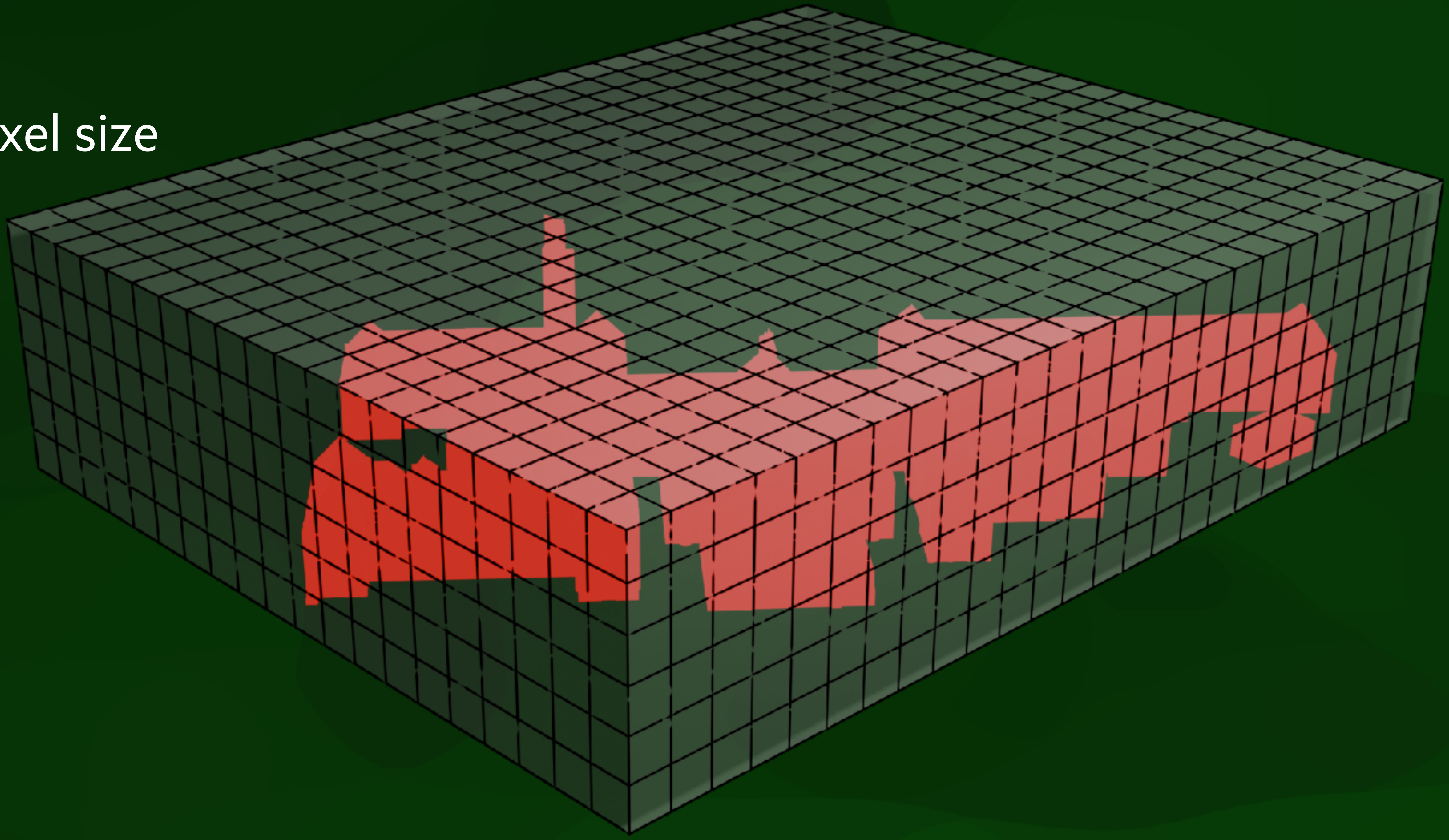
2. Define resolution / voxel size



Setup

1. Define domain (origin and orientation / axes)

2. Define resolution / voxel size



Usual (dense) encoding

1. Define order for storage / traversal (first x 0...i, then y 0...j, then z 0...k)
2. Following the order: specify what is in each voxel, e.g. occupied or not, object IDs, field values, etc.

- [..., false, true, true, false, ...]
- [..., 1, 1, 3, 3, 1, ...]
- [..., 3.45, 3.56, 3.78, 3.99, ...]

Sparse encoding

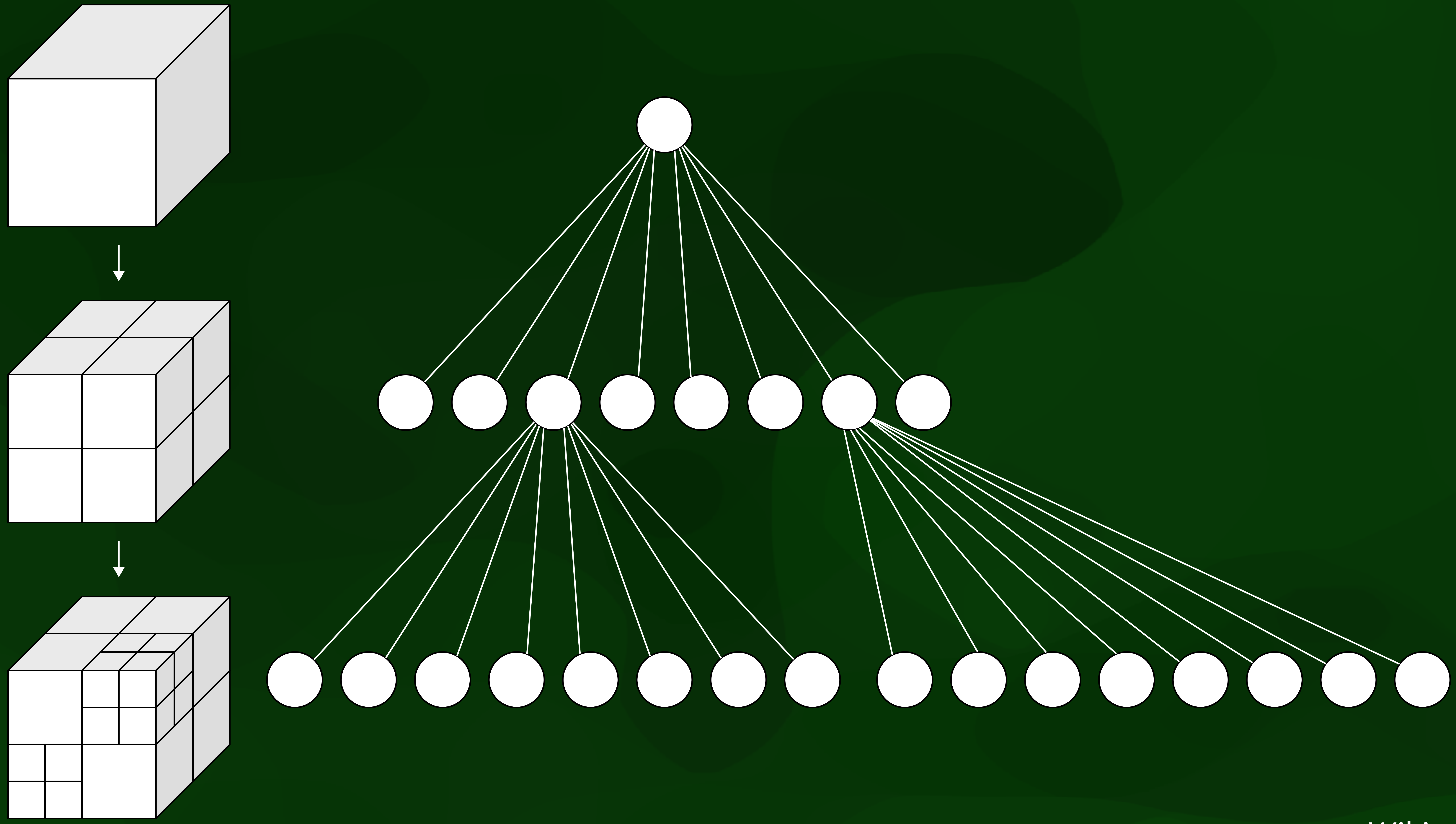
- List of entries containing for each voxel:
  - location (along x, y, z)
  - contents
- e.g. (x, y, z, ID) = ..., [100, 105, 0, 3], [101, 105, 0, 3], ...

What encoding is better?

Variations

- Varying sizes, e.g.
    - different size along z dimension
    - smaller size around an area of interest
- Differently shaped cells, e.g. octahedra

Hierarchical subdivisions



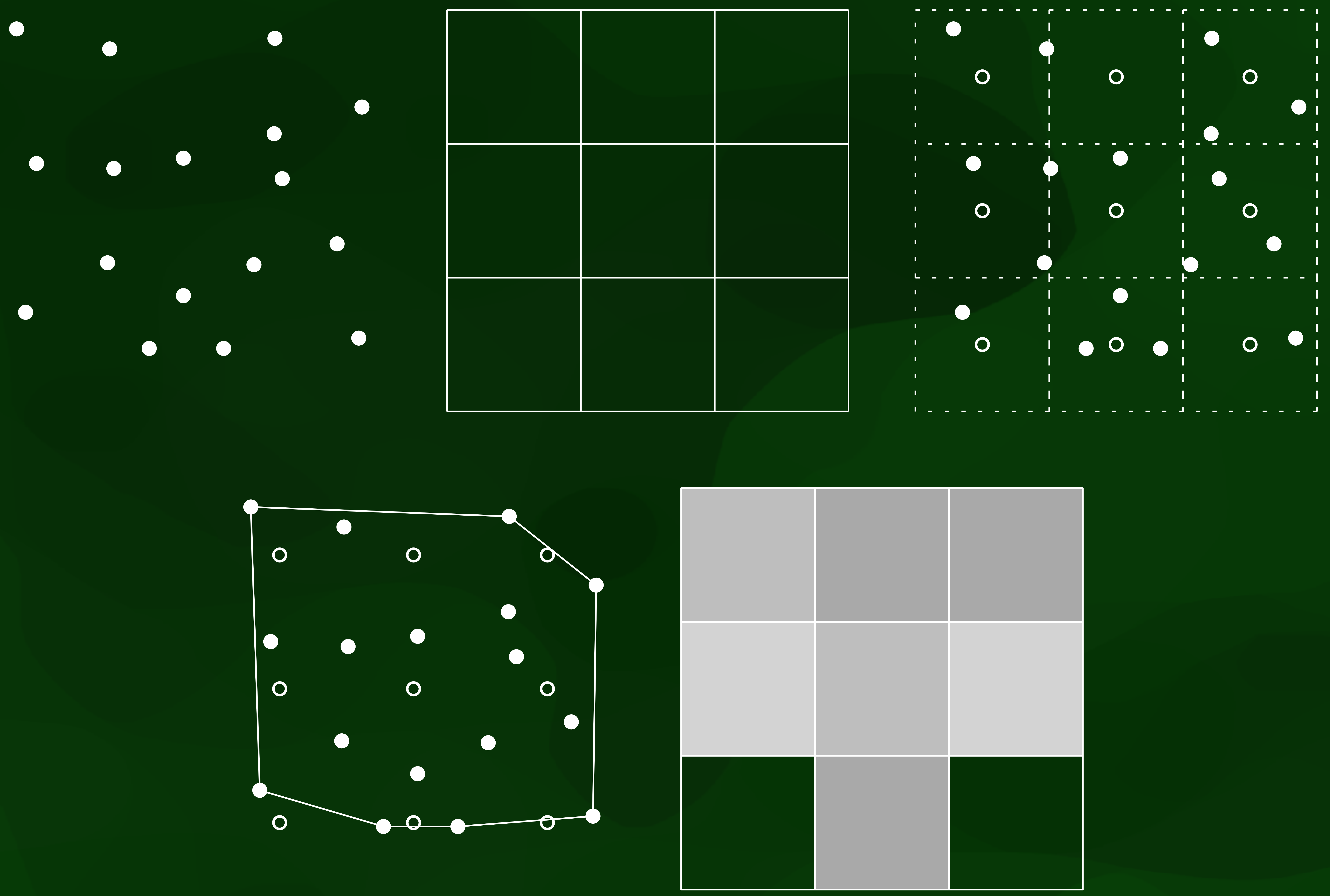
Wikimedia Commons

Hierarchical subdivisions

- Octrees: evenly along every axis
- Bintrees: evenly along alternating axes
- k-d trees: using a value along alternating axes

Voxelisation

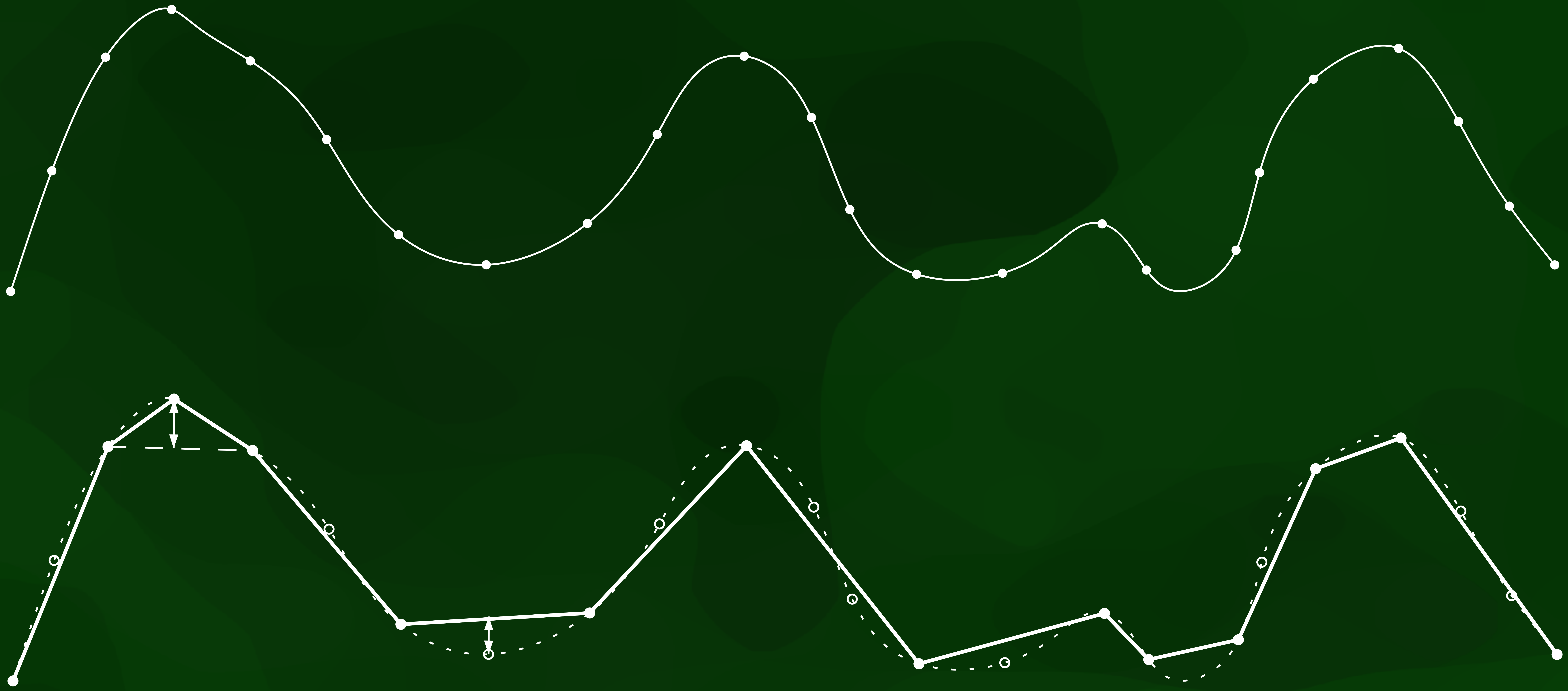
Voxelisation of fields



Voxelisation of fields

- Nearest neighbour
- IDW
- Linear in tetrahedra
- Natural neighbour
- Kriging
- etc.

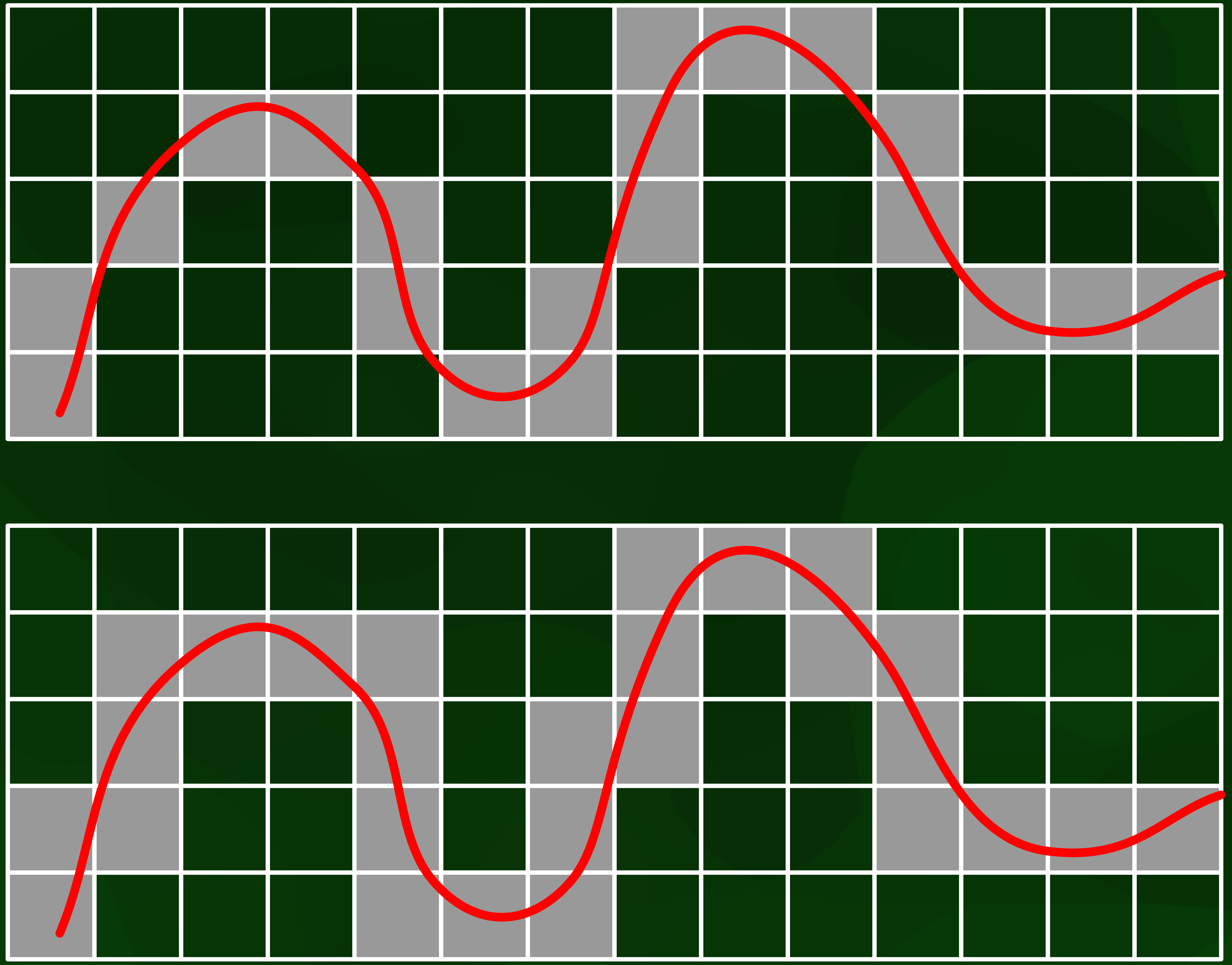
Voxels to points



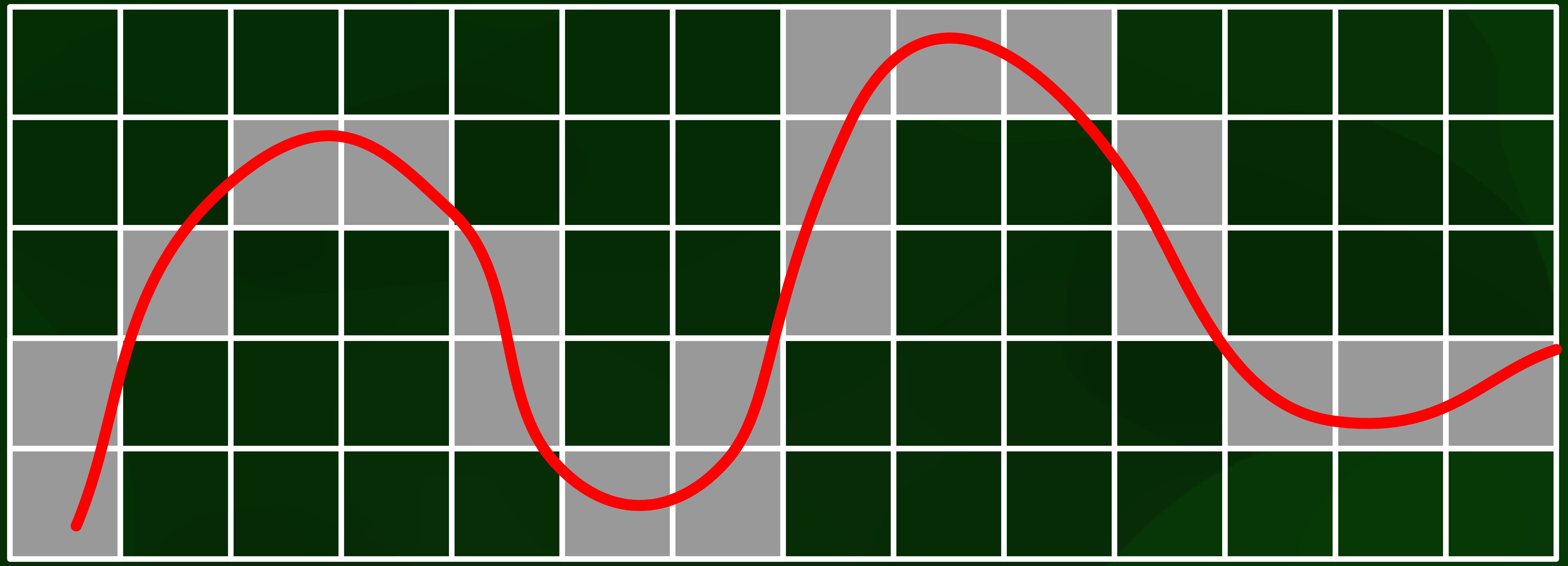
Voxelisation of objects

Starting with 2D: rasterisation

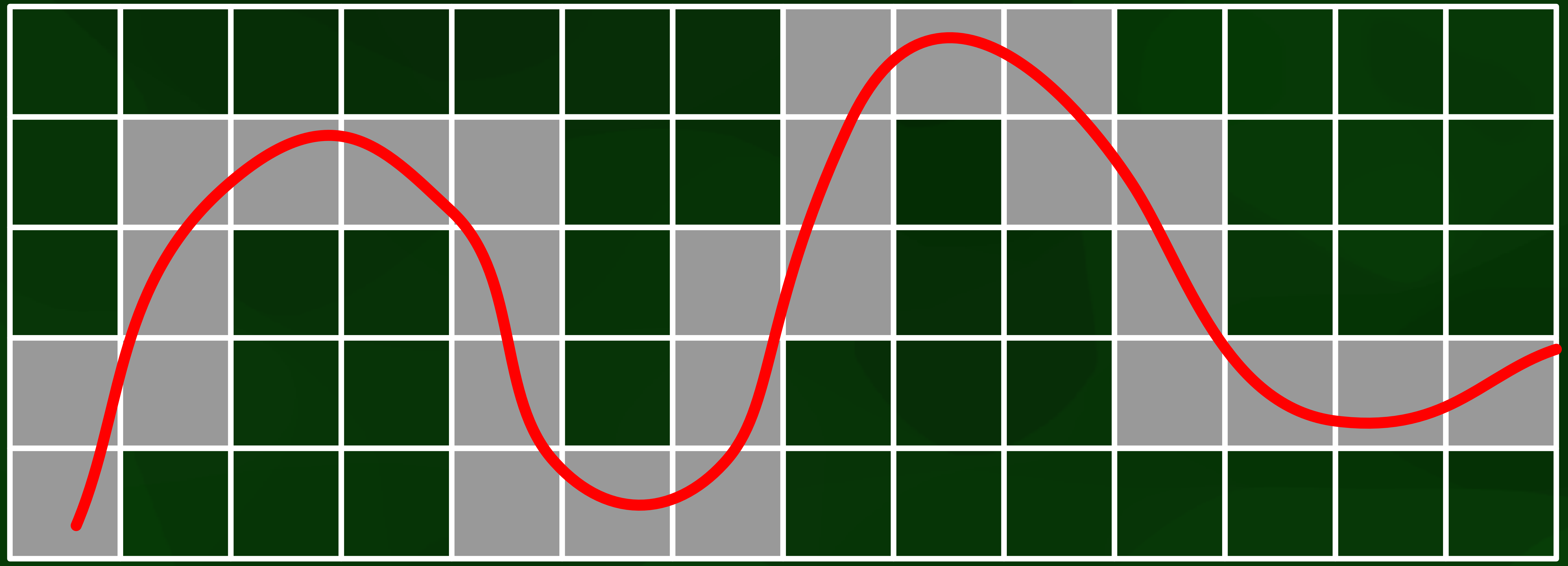
In 2D: what is the difference?



In 2D: what is the difference?

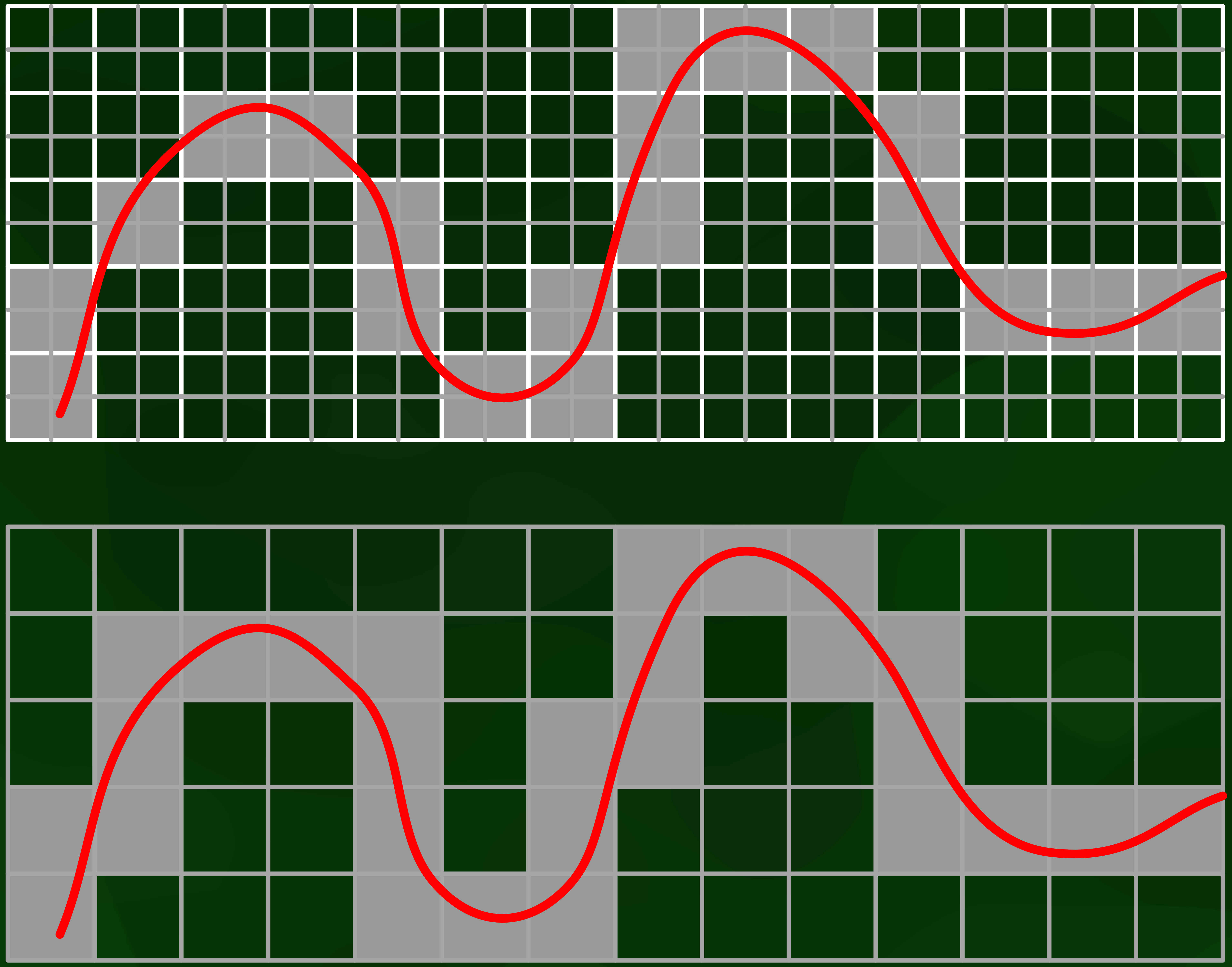


4-connected

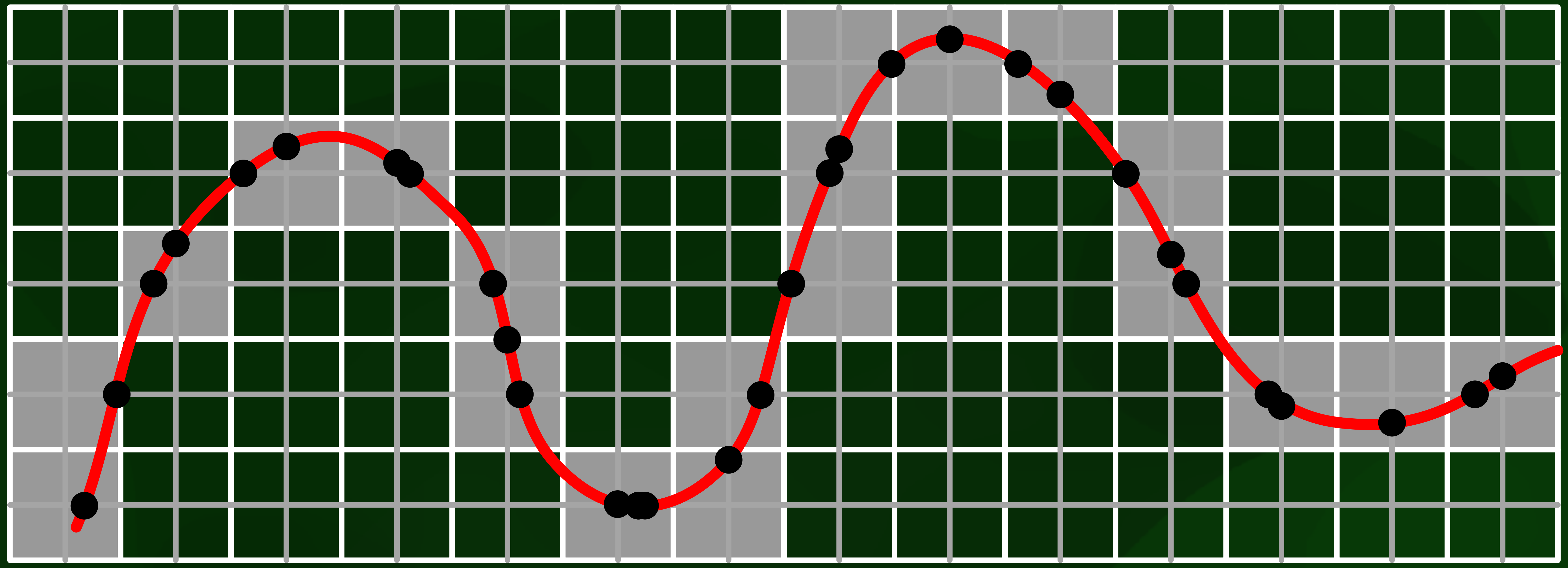


8-connected

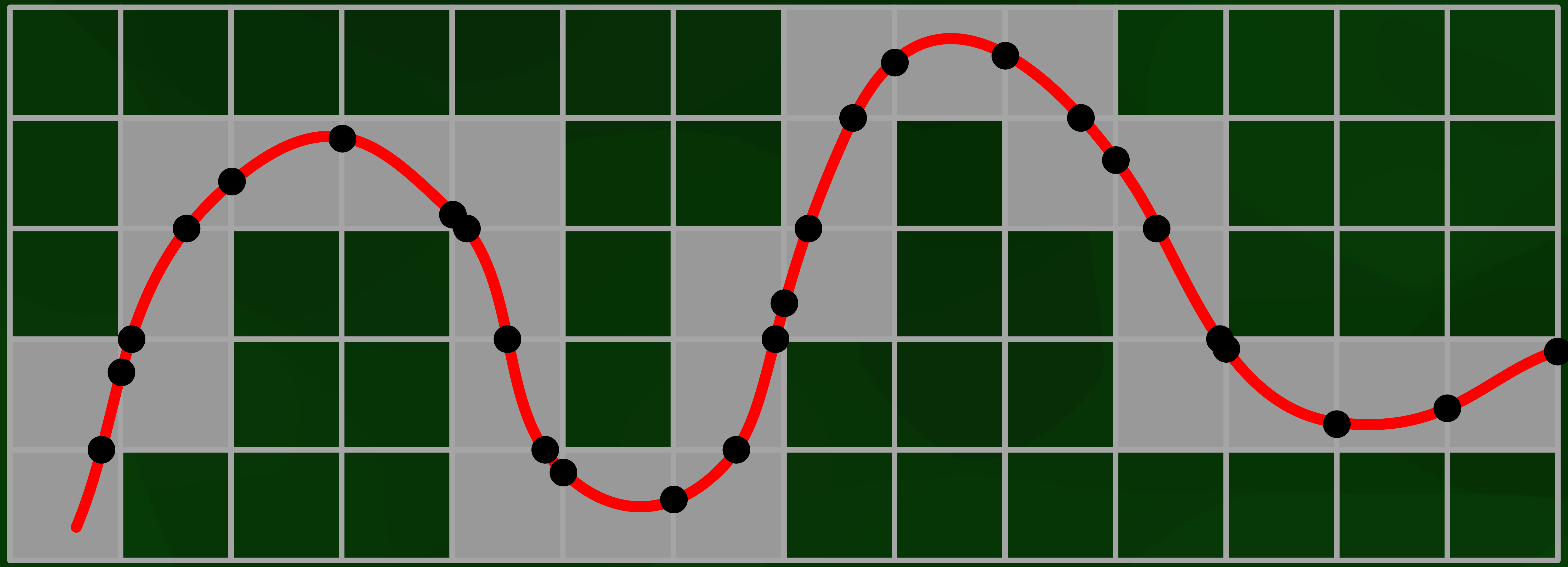
2D targets



2D targets

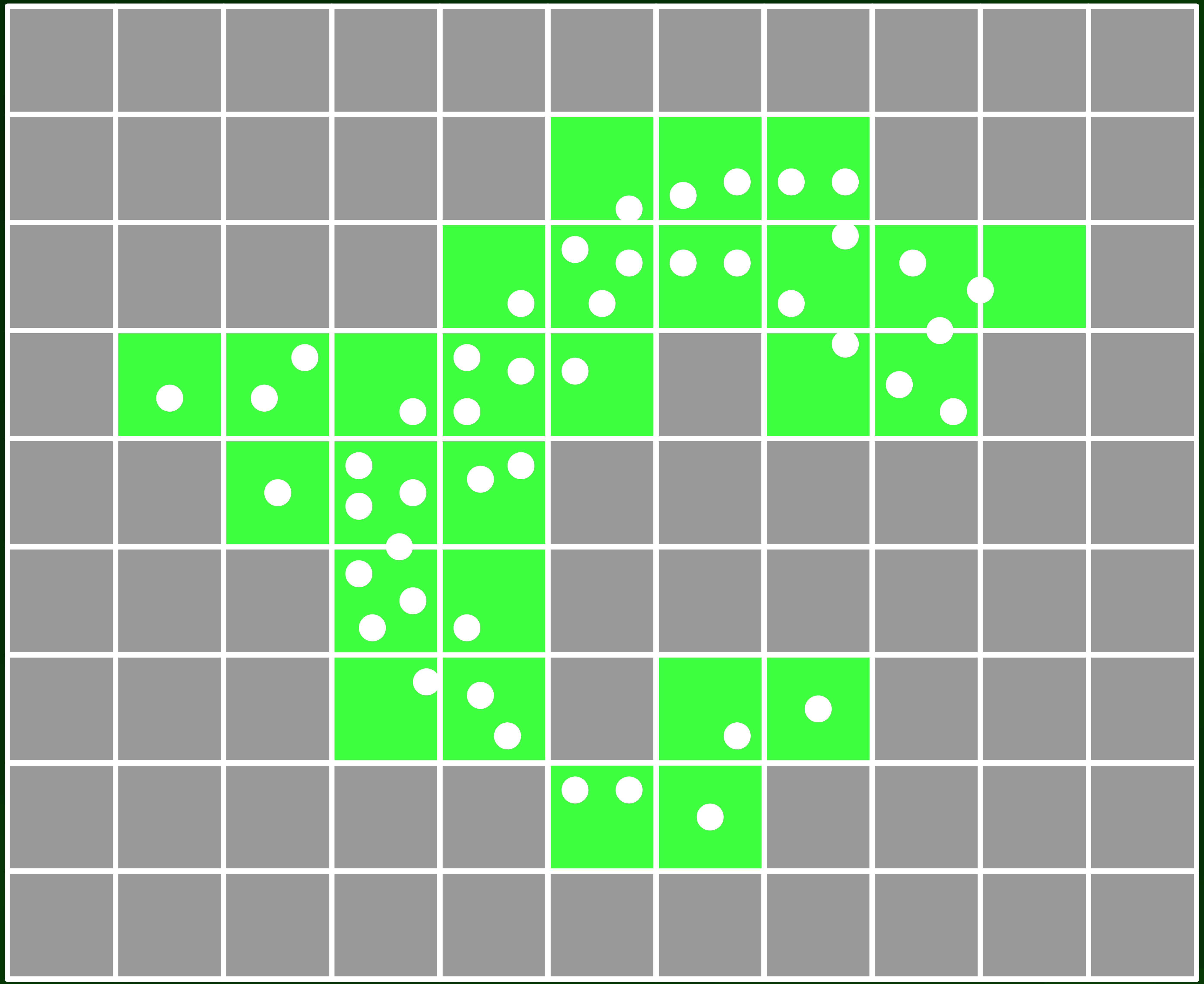


4-connectivity

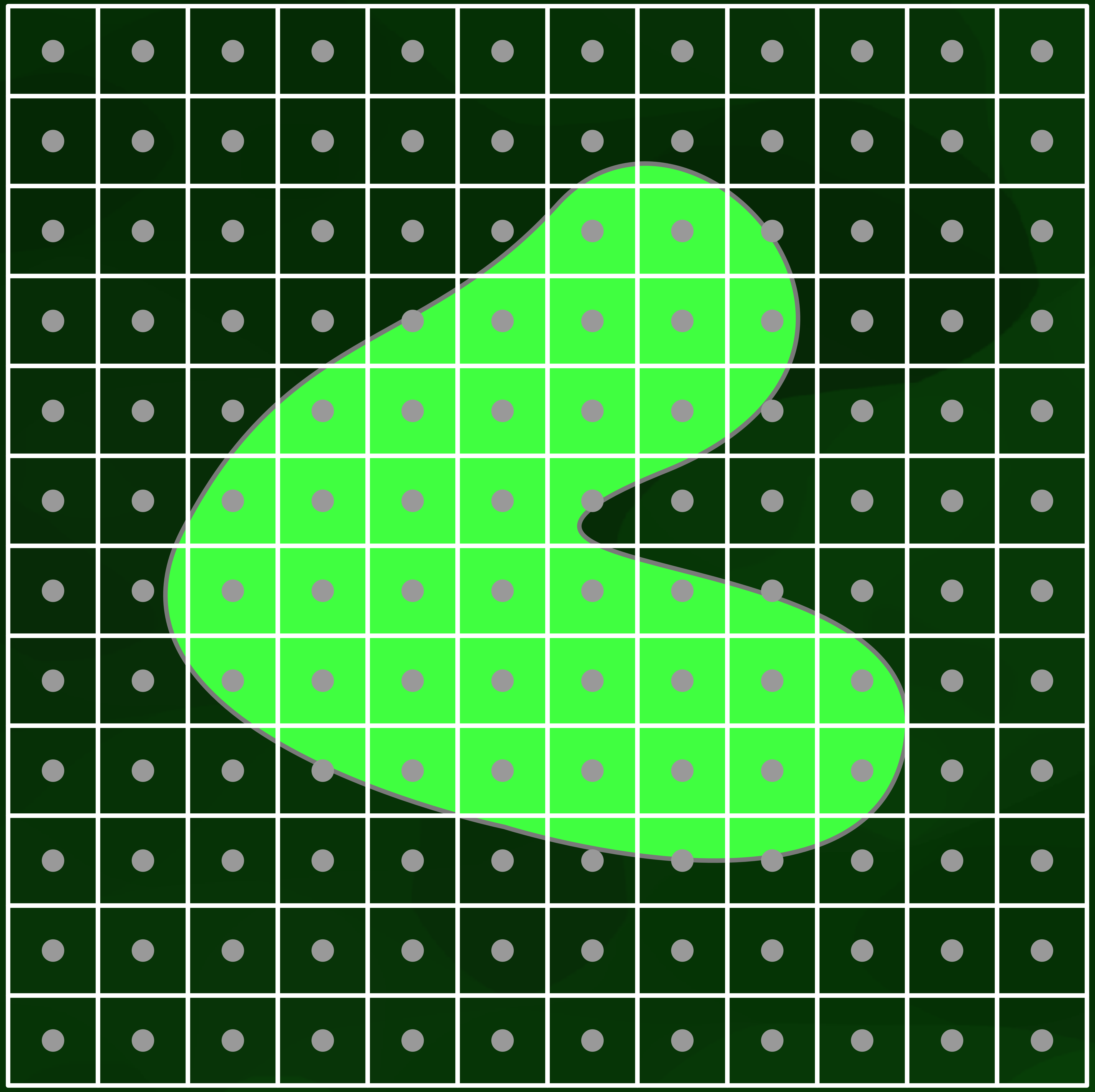


8-connectivity

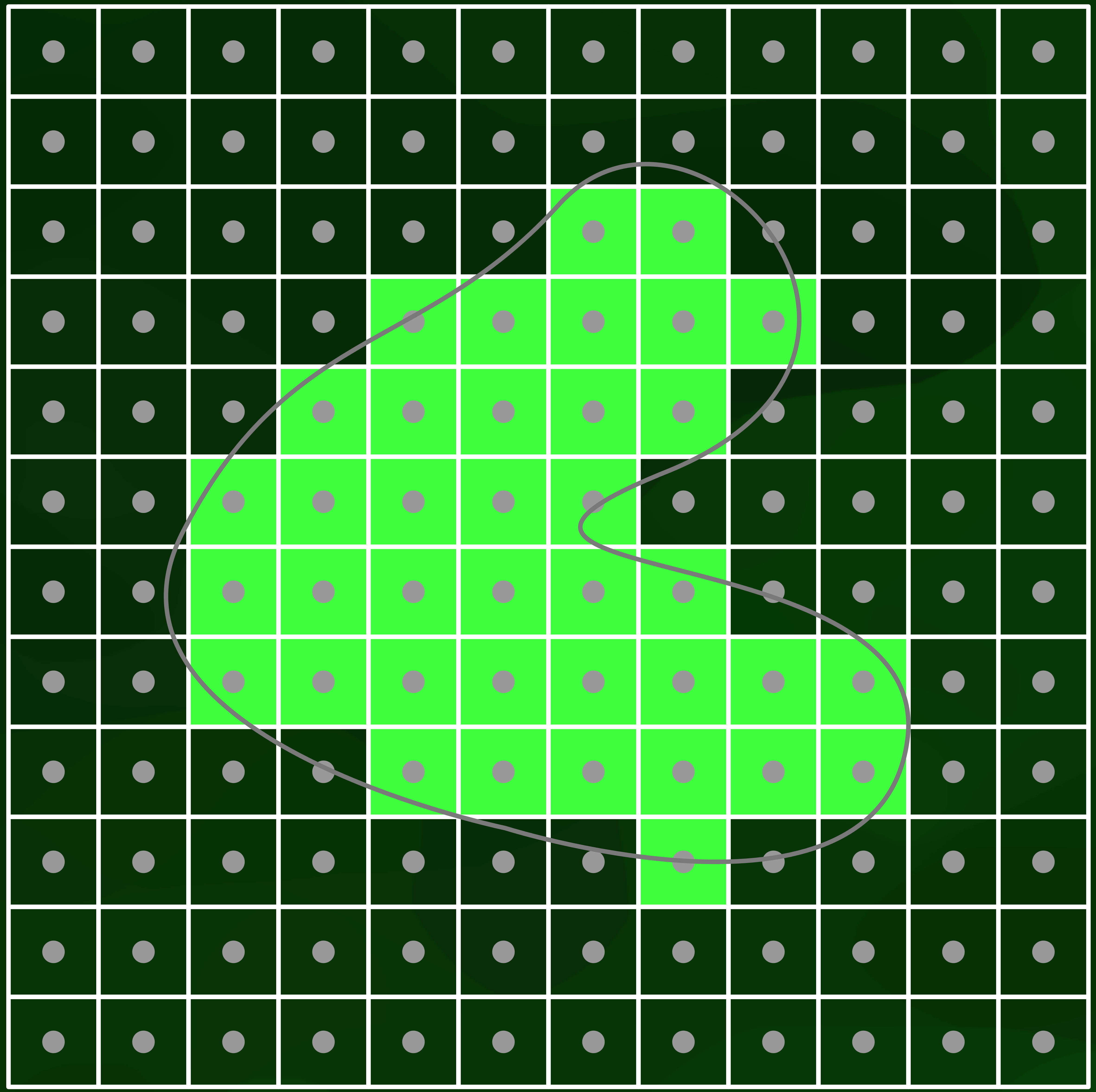
Targets for points



Targets for areas



Targets for areas



Duality of targets in 2D

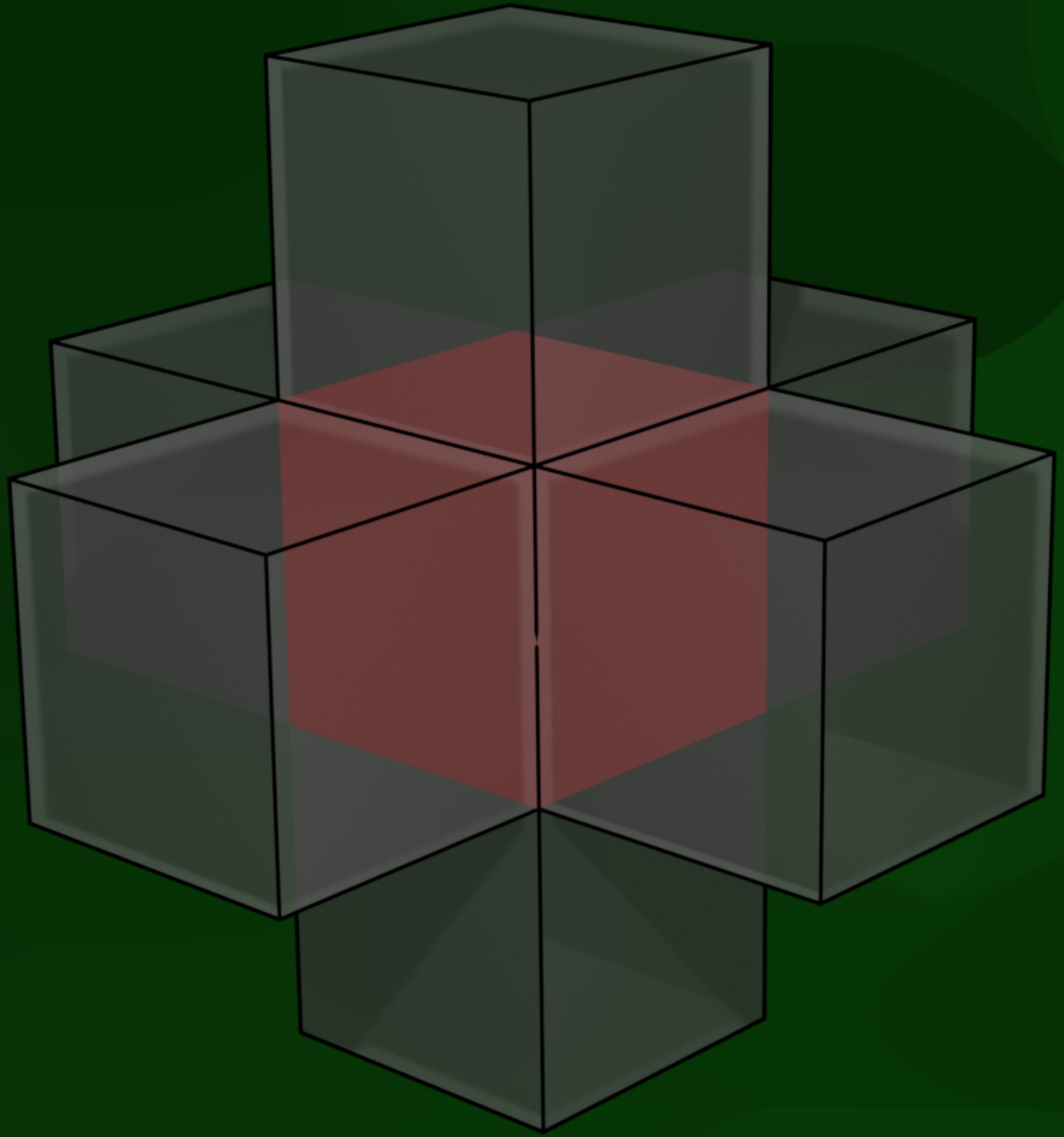
- To detect:
  - points (0D) -> squares (2D)
  - lines (1D) -> line segments (1D)
  - areas (2D) -> centre points (0D)

Duality of targets in 3D

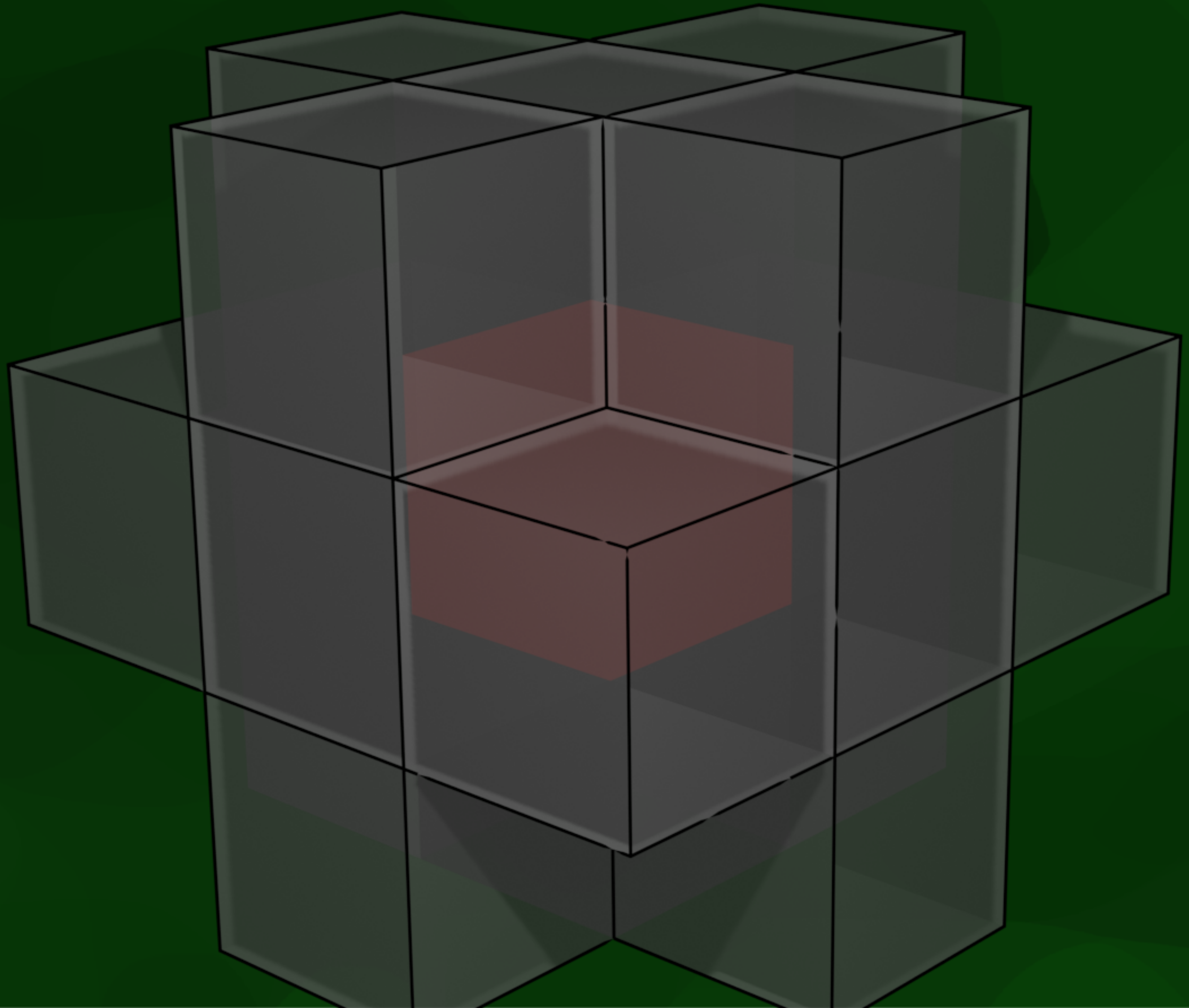
- To detect:
    - points (0D) -> cubes (3D)
    - lines (1D) -> surfaces (2D)
    - surfaces (2D) -> line segments (1D)
    - volumes (3D) -> centre points (0D)

Connectivity in 3D

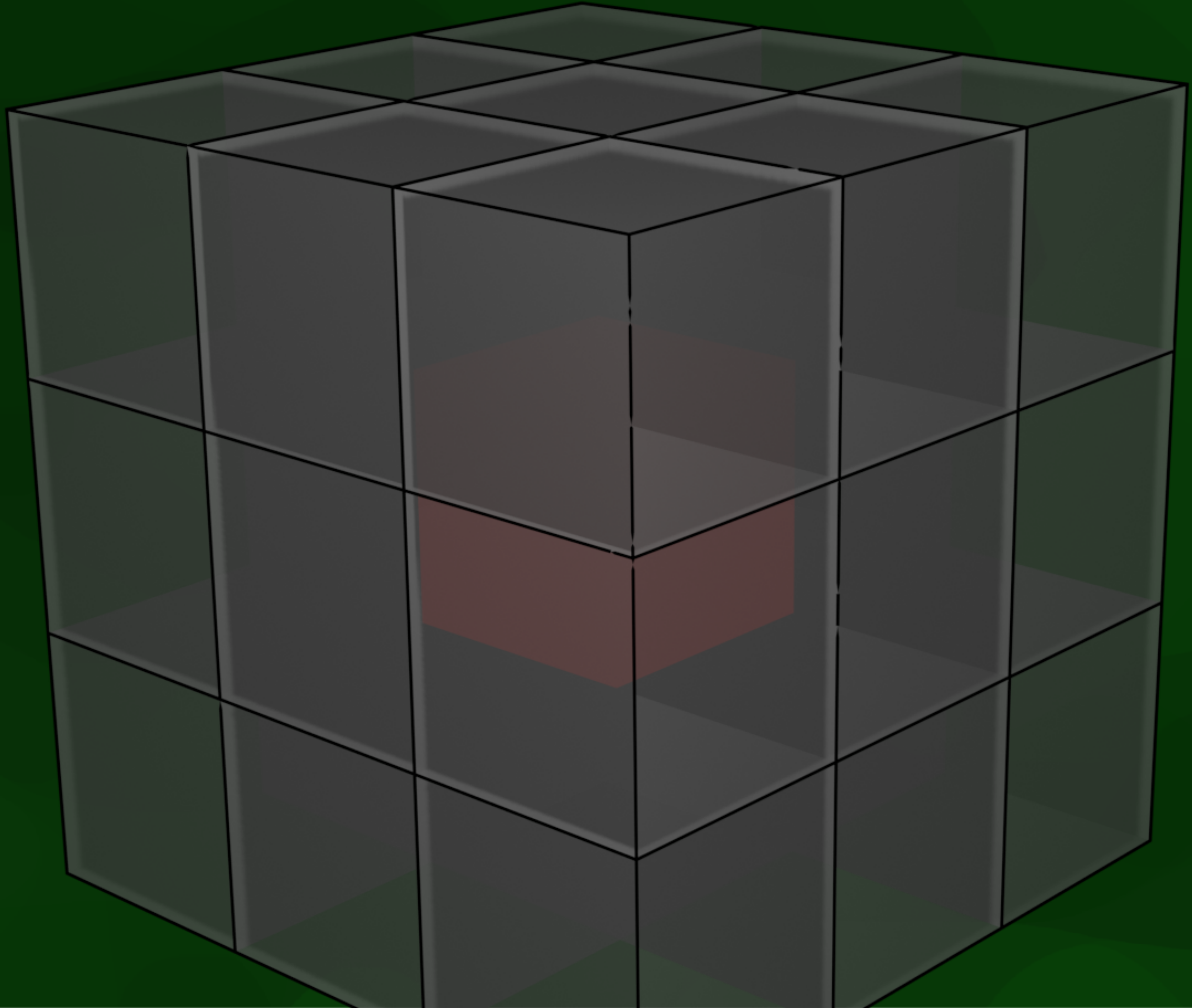
6-connectivity



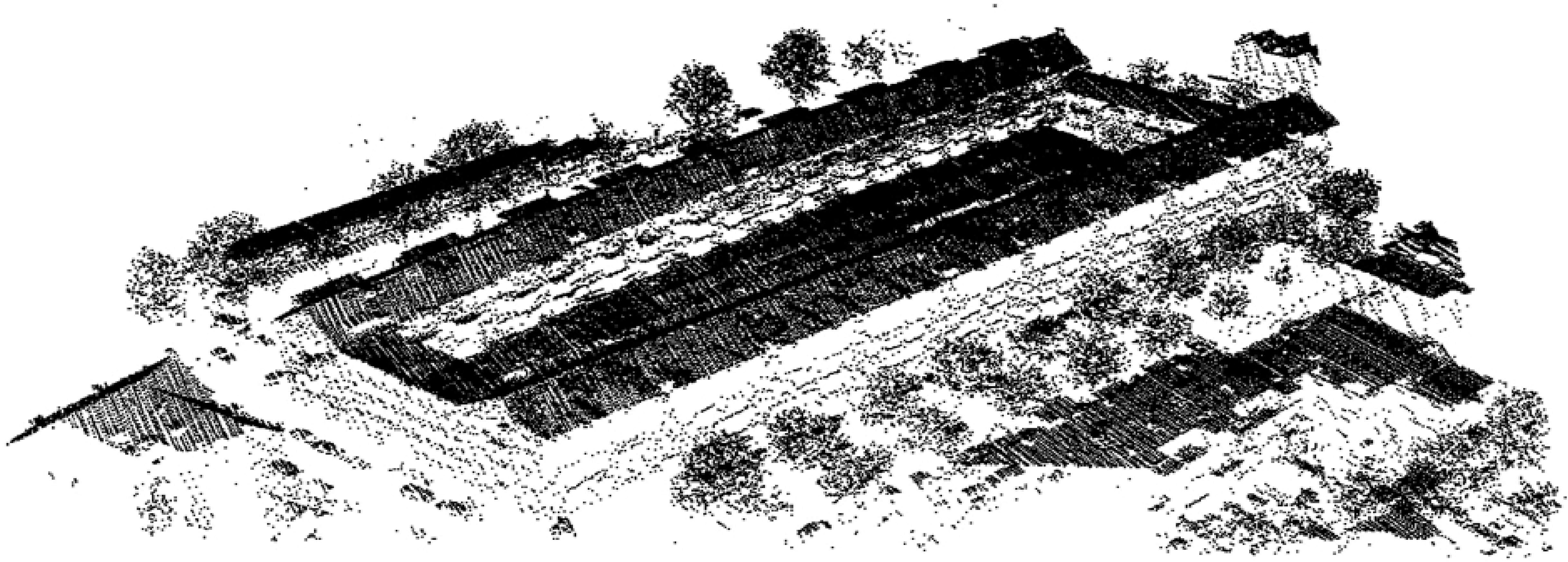
18-connectivity

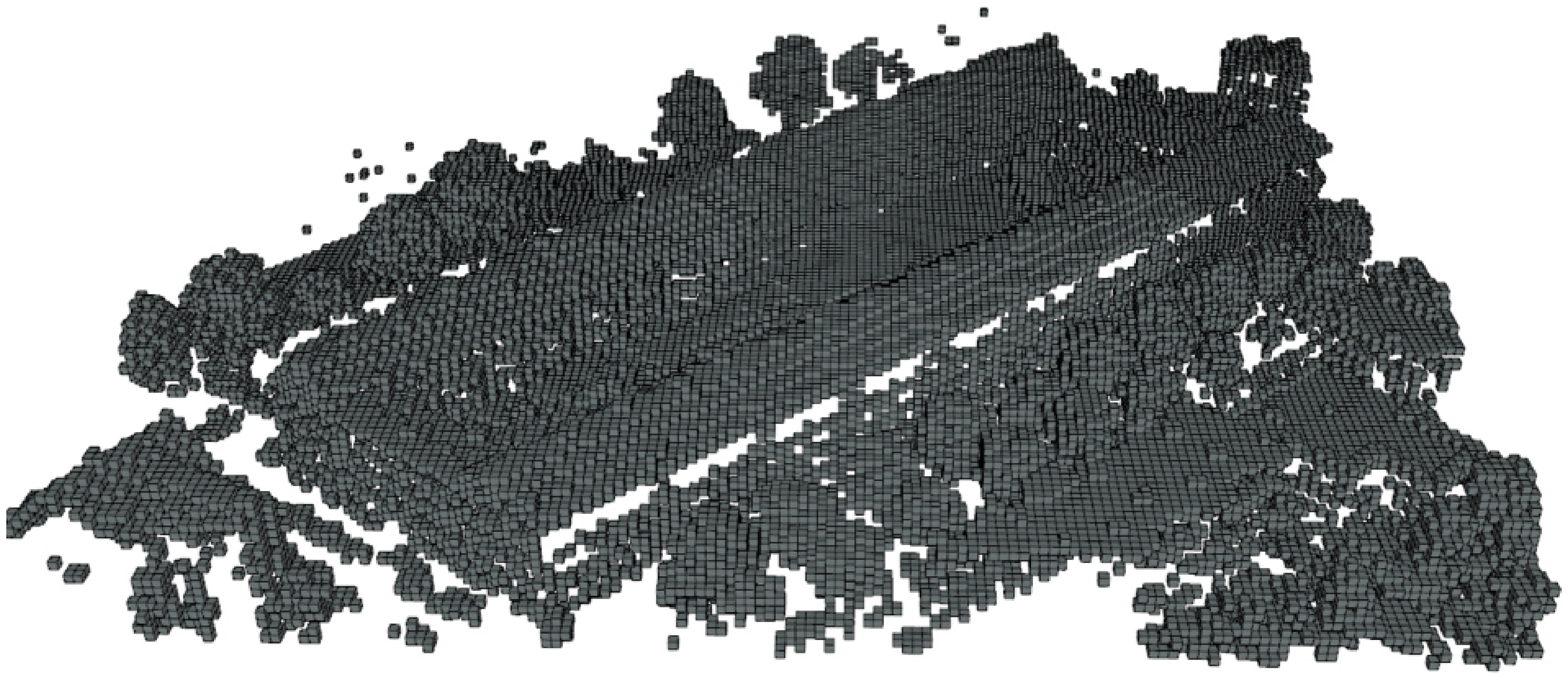


26-connectivity

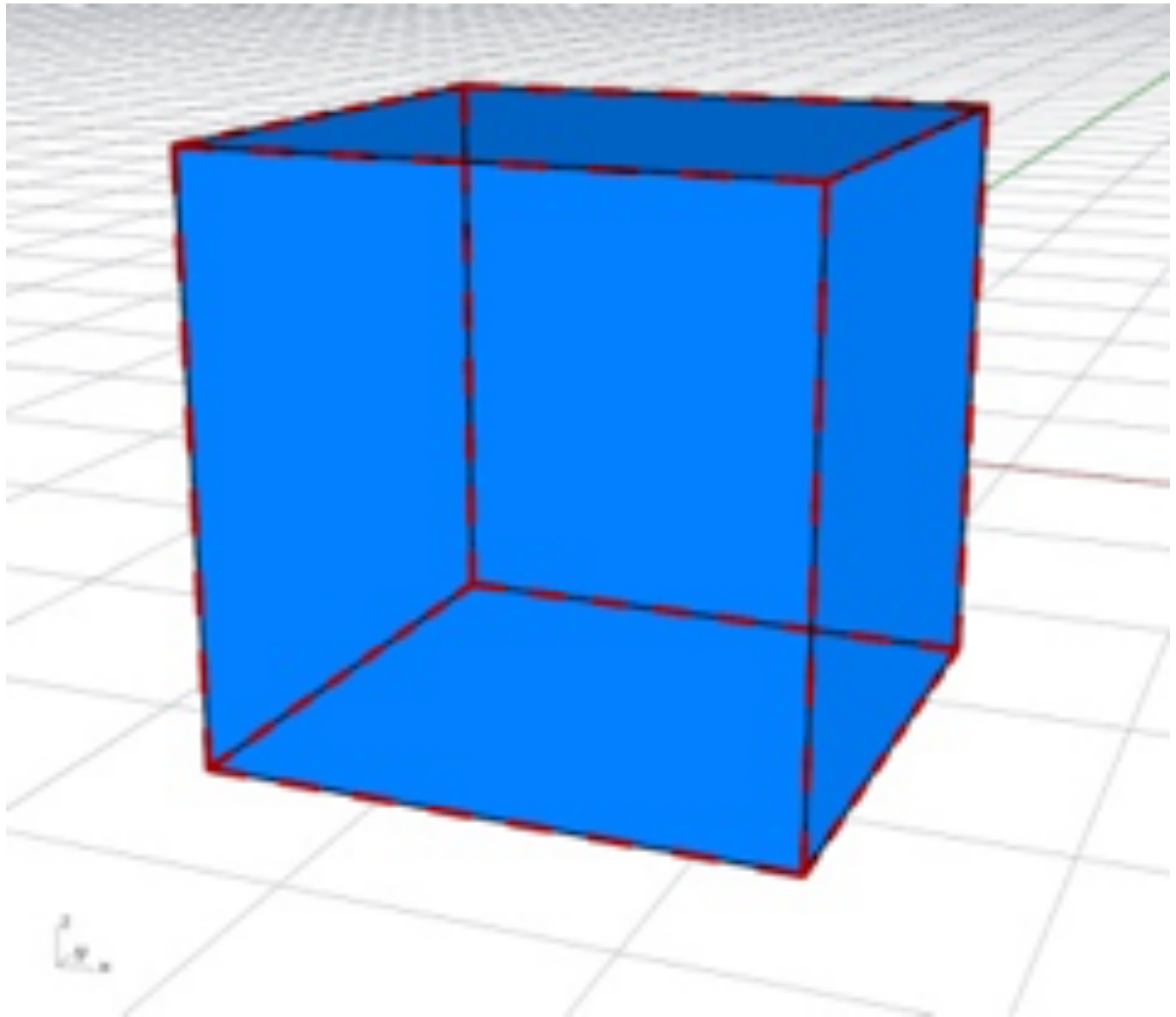


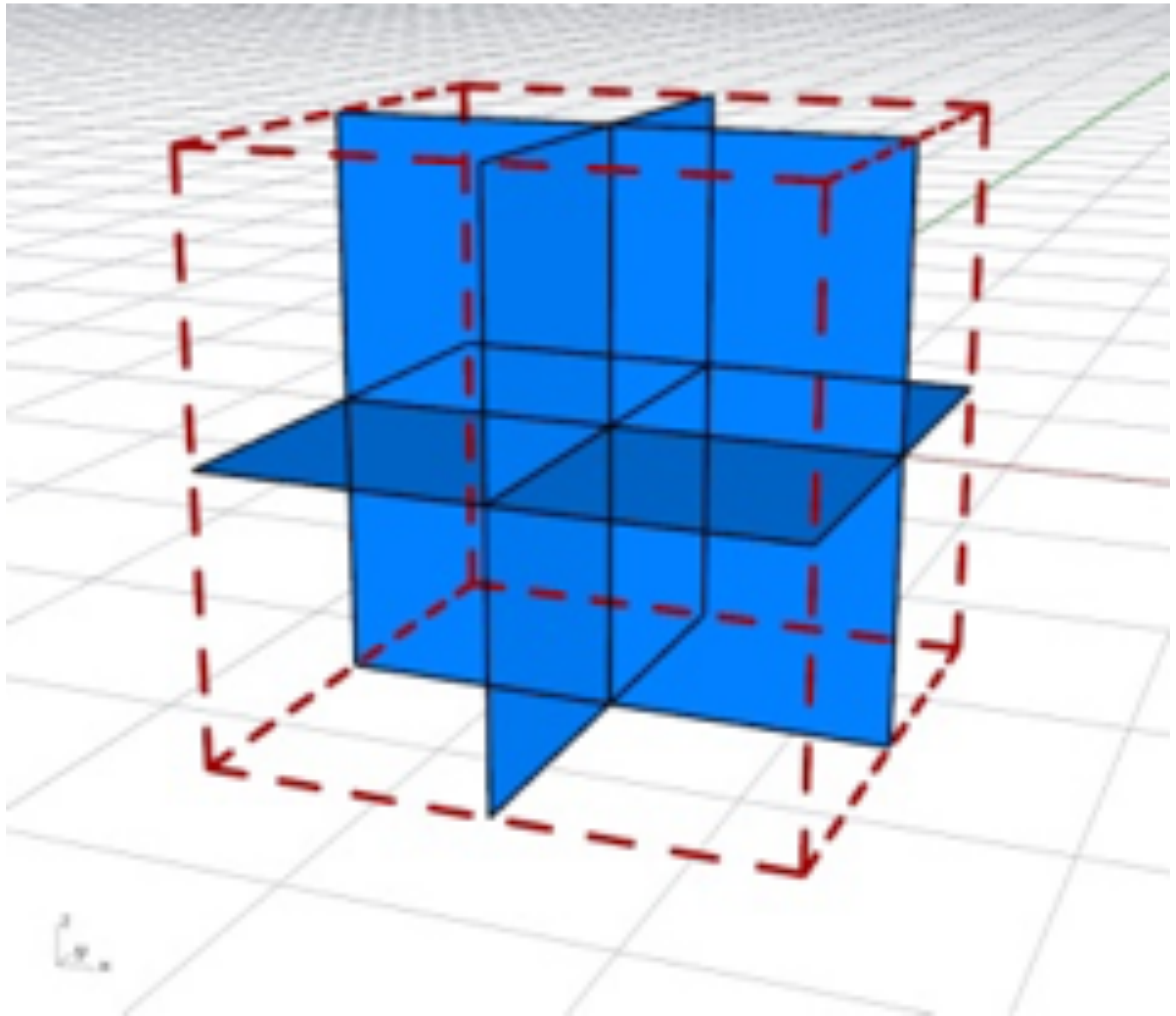
Voxelising points



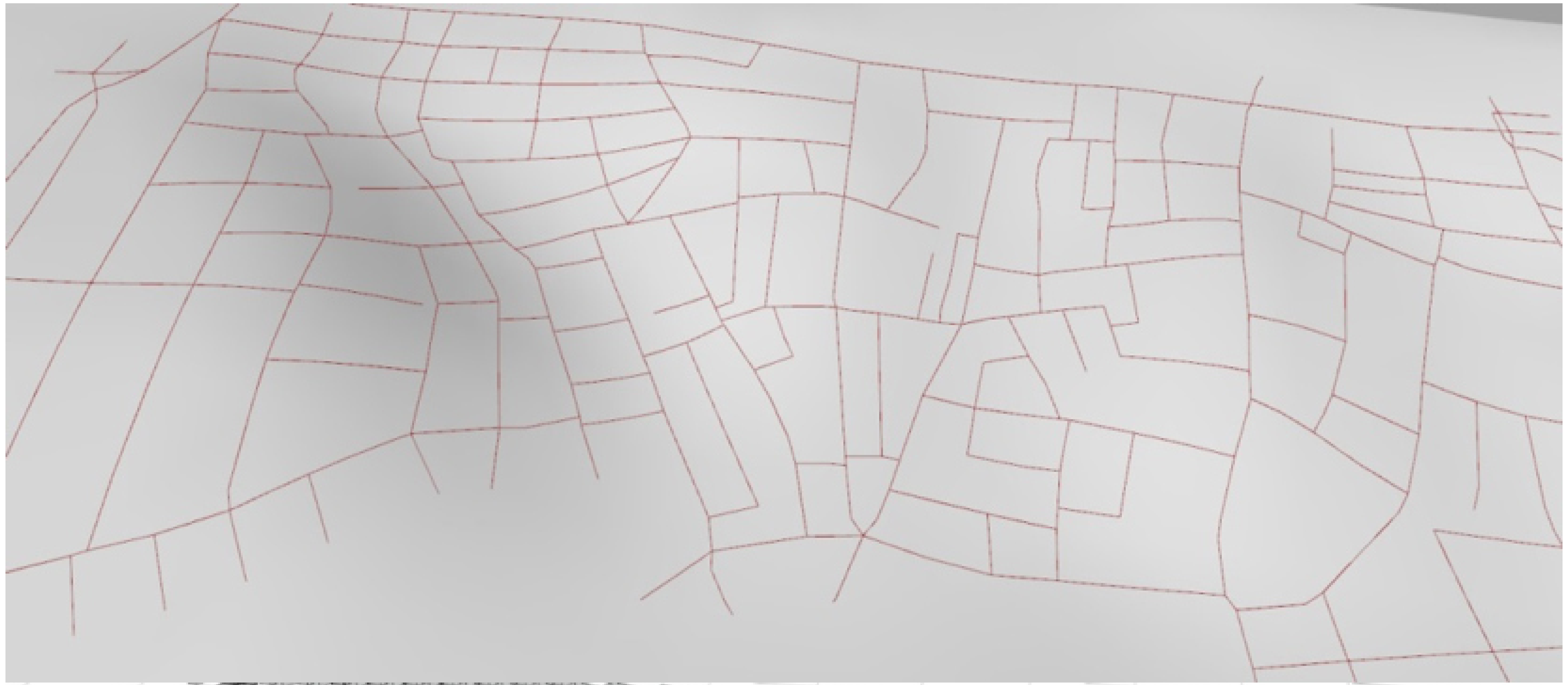


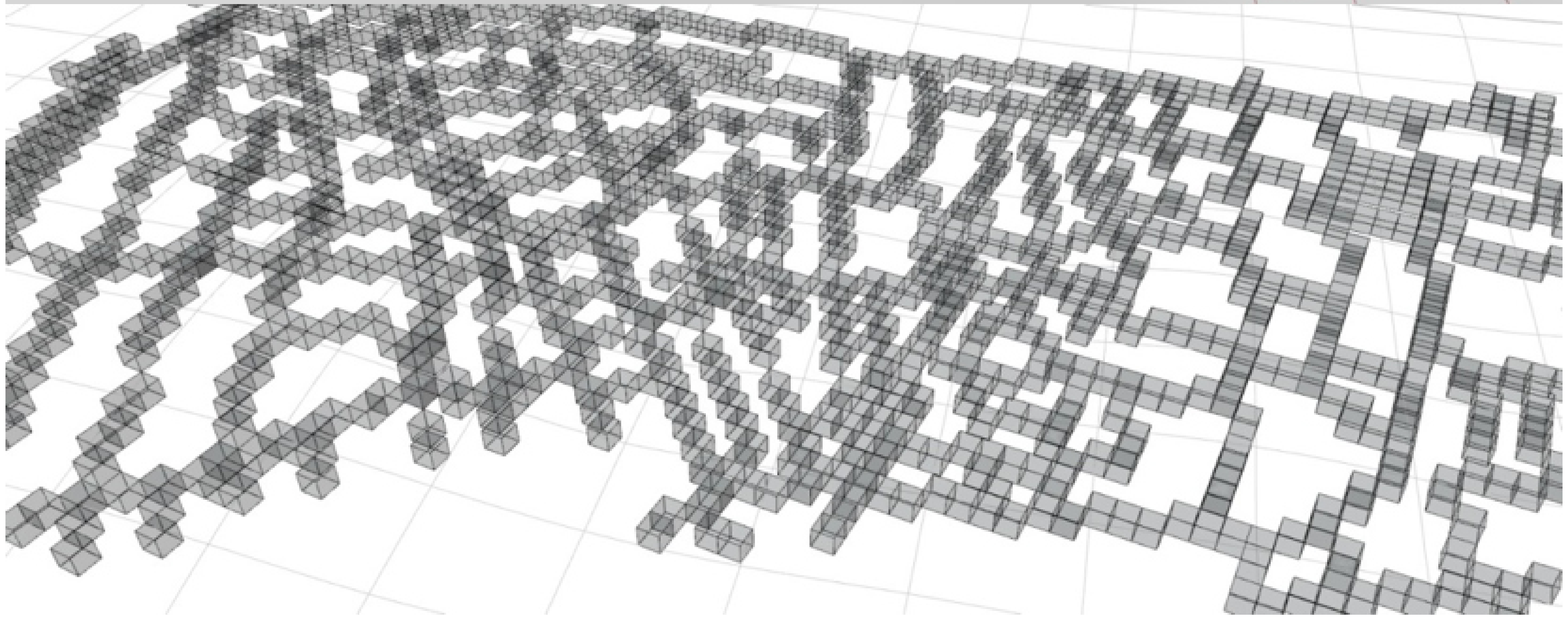
Voxelising lines



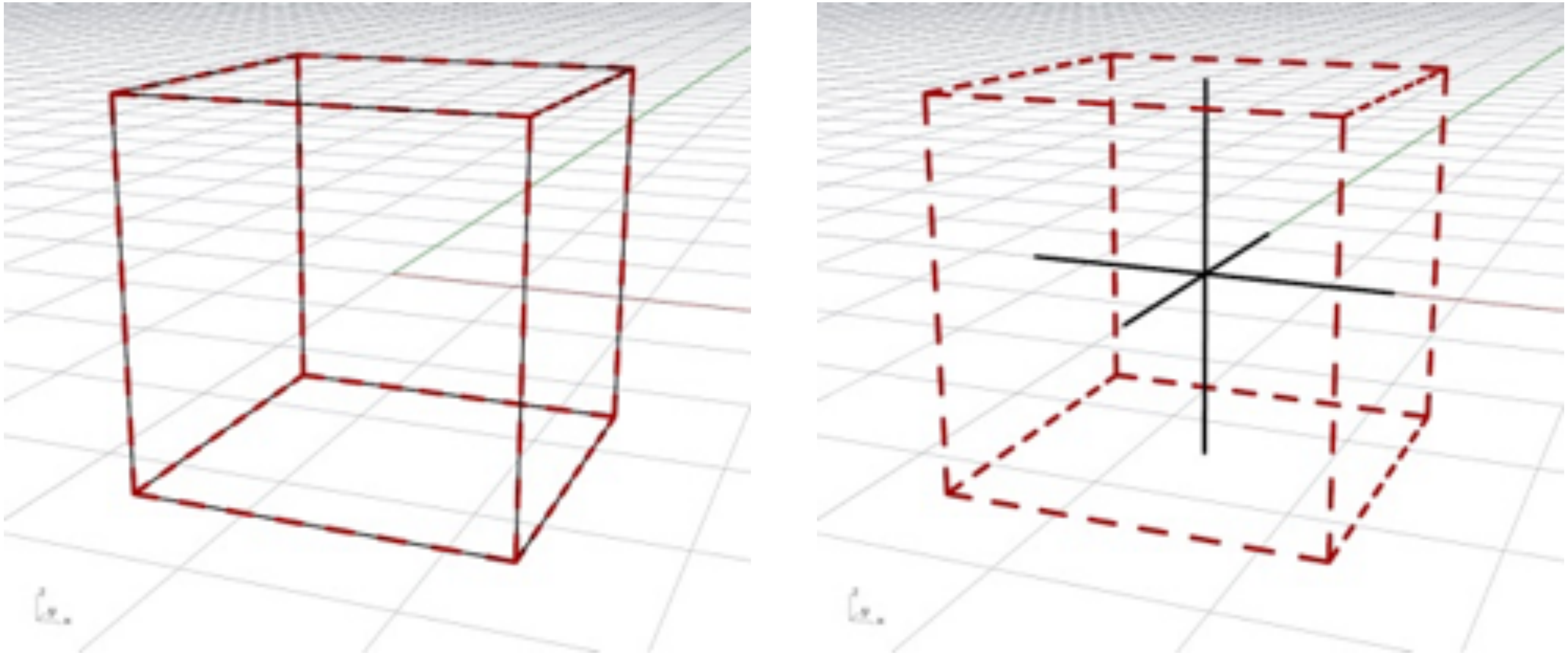


Voxelising lines

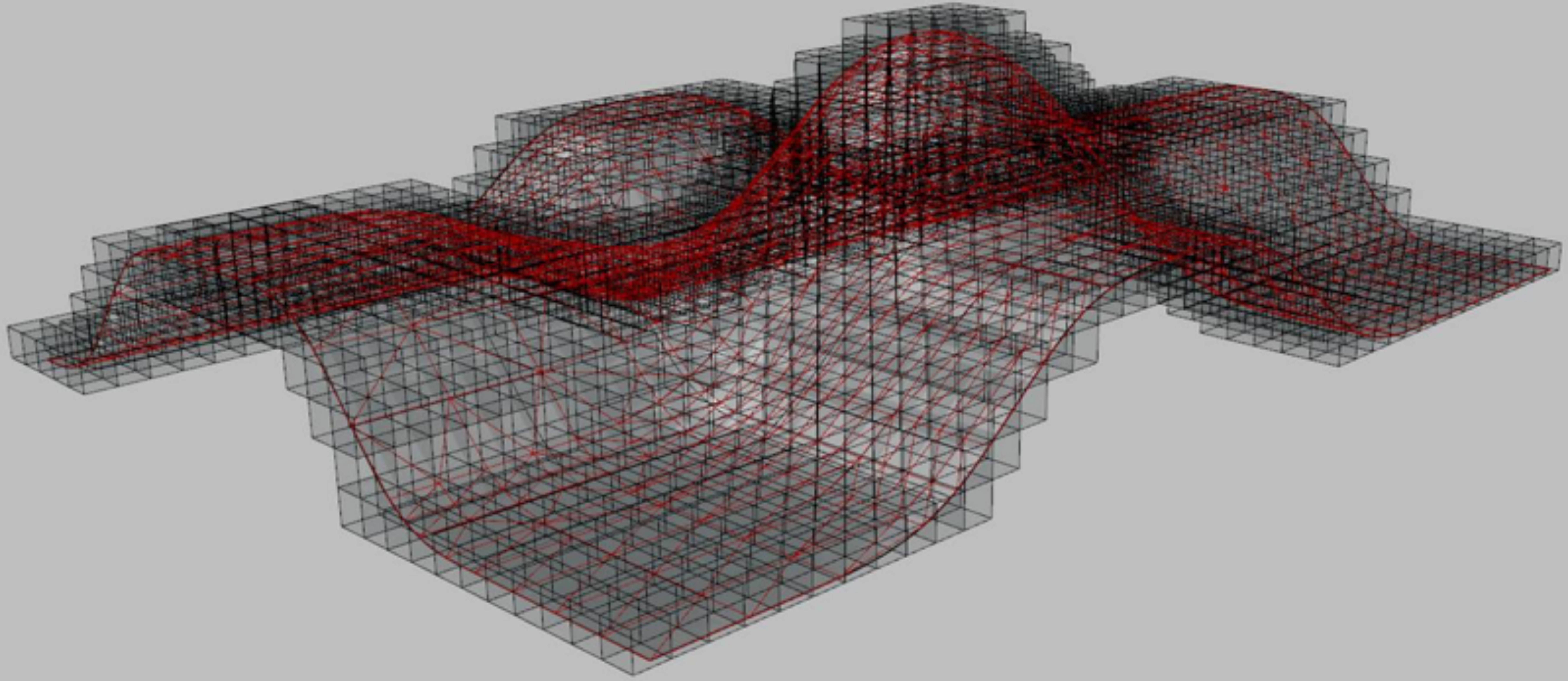




Voxelising surfaces



Voxelising surfaces



Voxelising volumes?

What to do next?

1. Today:
   - Start with Homework 1 (triangulation of faces)
   - Go to geo1004 website and study today's lesson (3D book Chapter 4)
   - If you have extra time: study Wednesday's lesson (3D book Chapter 3)
2. Wednesday: demo on 3D Voronoi and help with lessons or Hw 1

https://3d.bk.tudelft.nl/courses/geo1004

3D geoinformation

Department of Urbanism
Faculty of Architecture and the Built Environment
Delft University of Technology