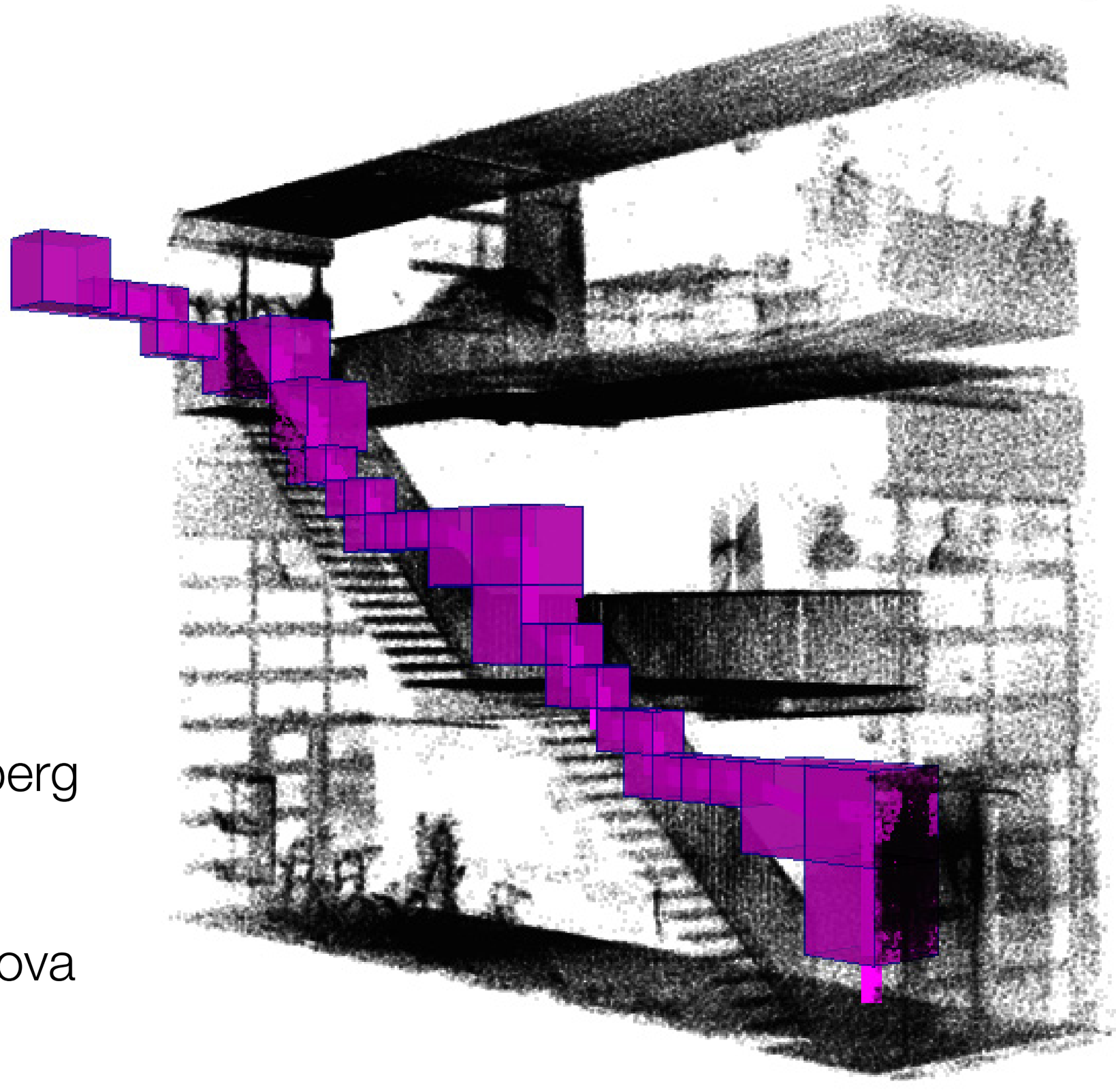
From point cloud to pathfinding



By: Olivier Rodenberg

Supervisors:
Prof.dr. Sisi Zlatanova
ir. Edward Verbree

Research question

What is the relation between the computational effort and path length of A* pathfinding in an octree representation of an indoor point cloud?

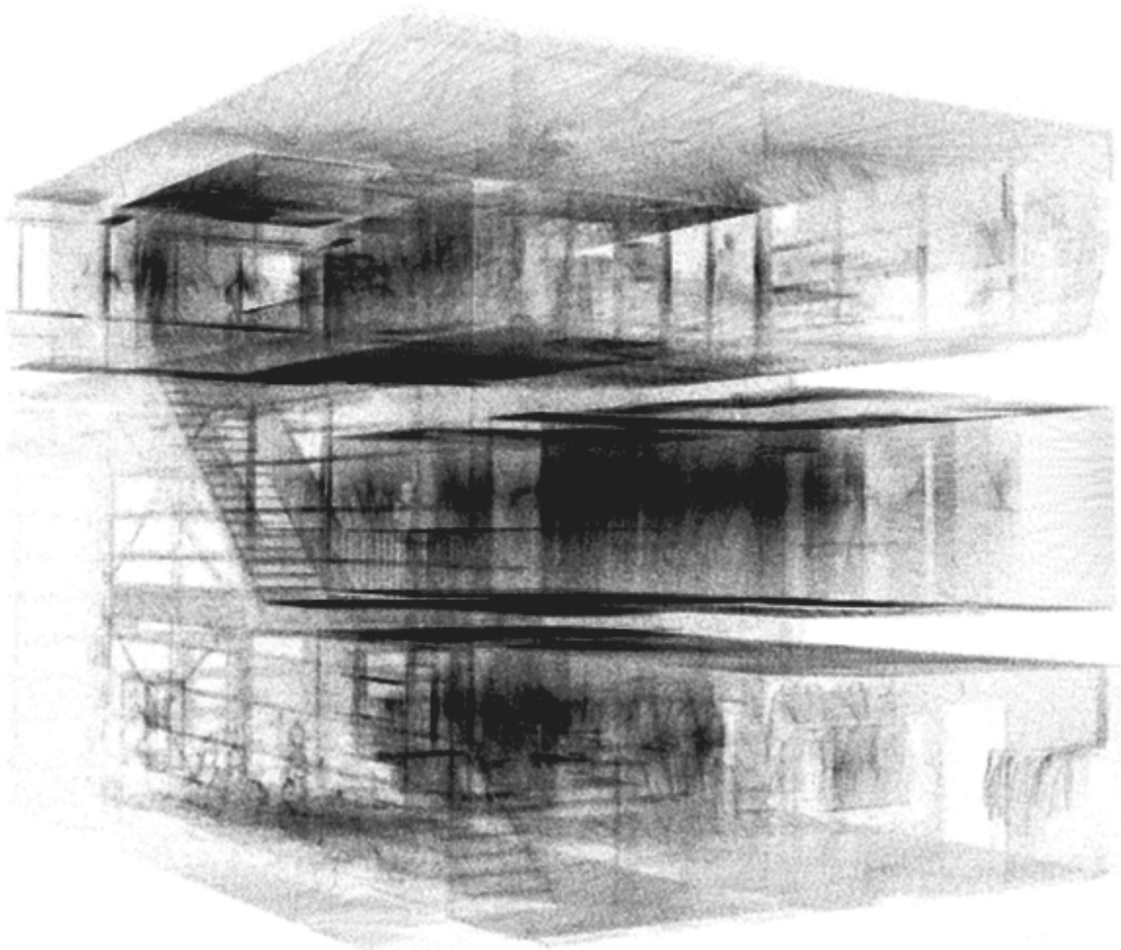
- What point cloud processing operations are important for the generation of the octree and what is their effect?

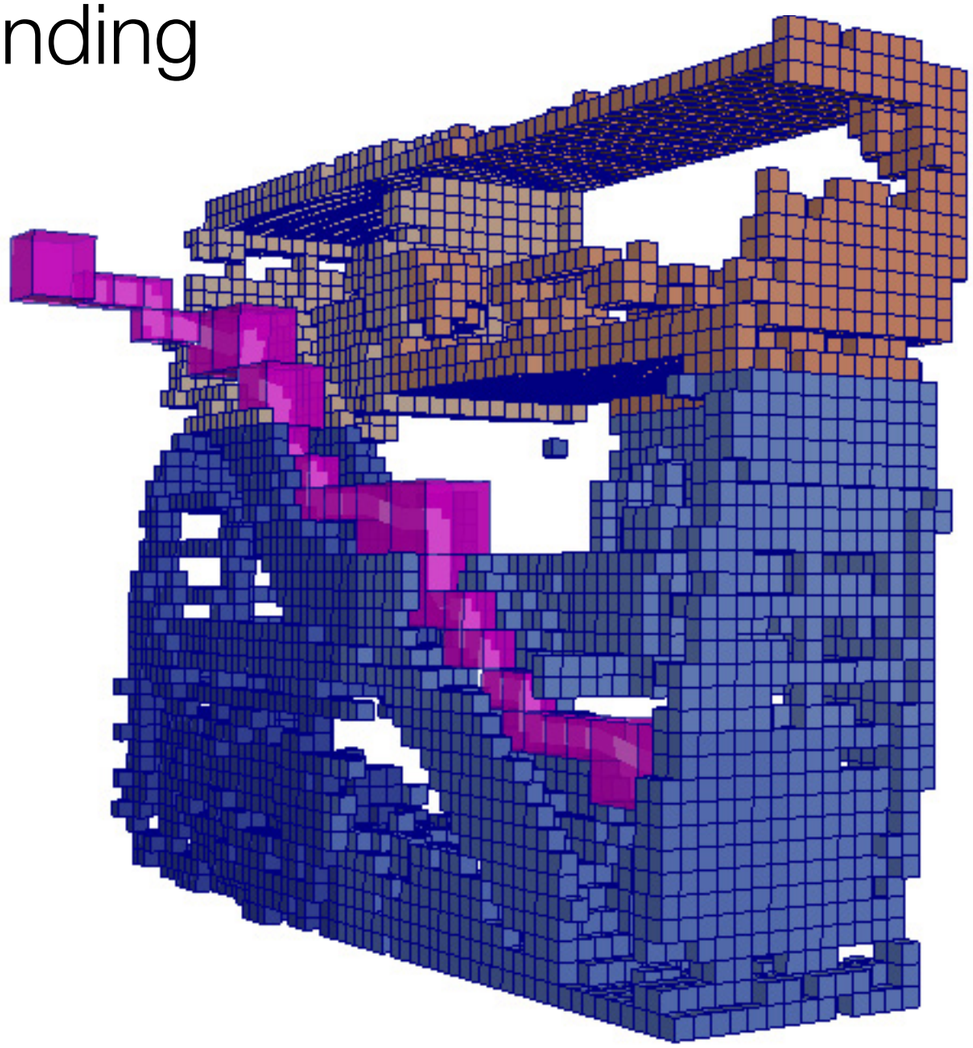
- What octree properties influence the computational effort and path length and what is their effect?

- What components in the A* algorithm influence the computational ef fort and path length and what is their effect?

Research

What steps need to be taken to create a data set from a point cloud which can be used for pathfinding





point cloud → pathfinding

Point cloud

(large) collection of points having an x, y and z coordinate and optional colour value.

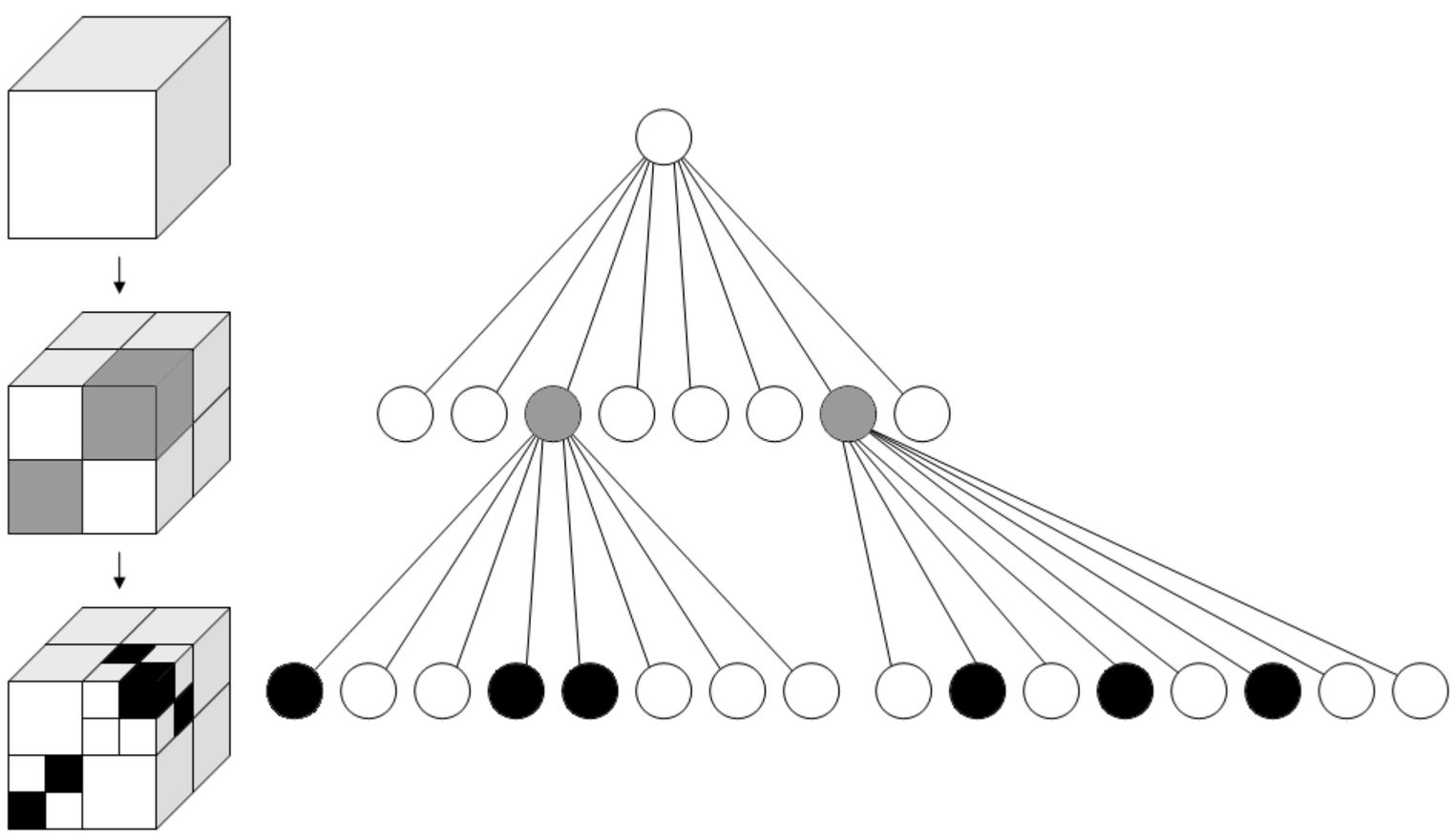
Point cloud for navigation

Problems:

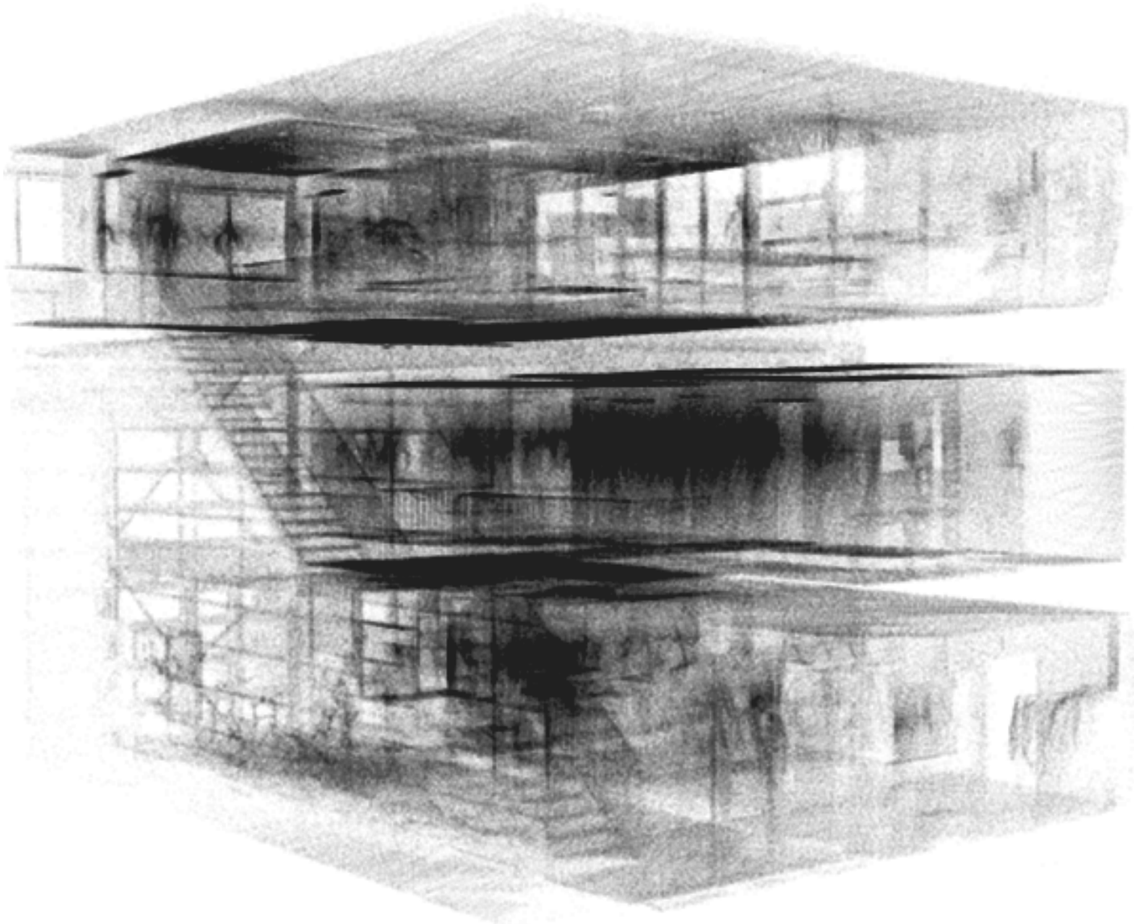
- A point cloud on its own is unstructured
- No idea which places are free space
- How is the empty space connected
- How to avoid collisions

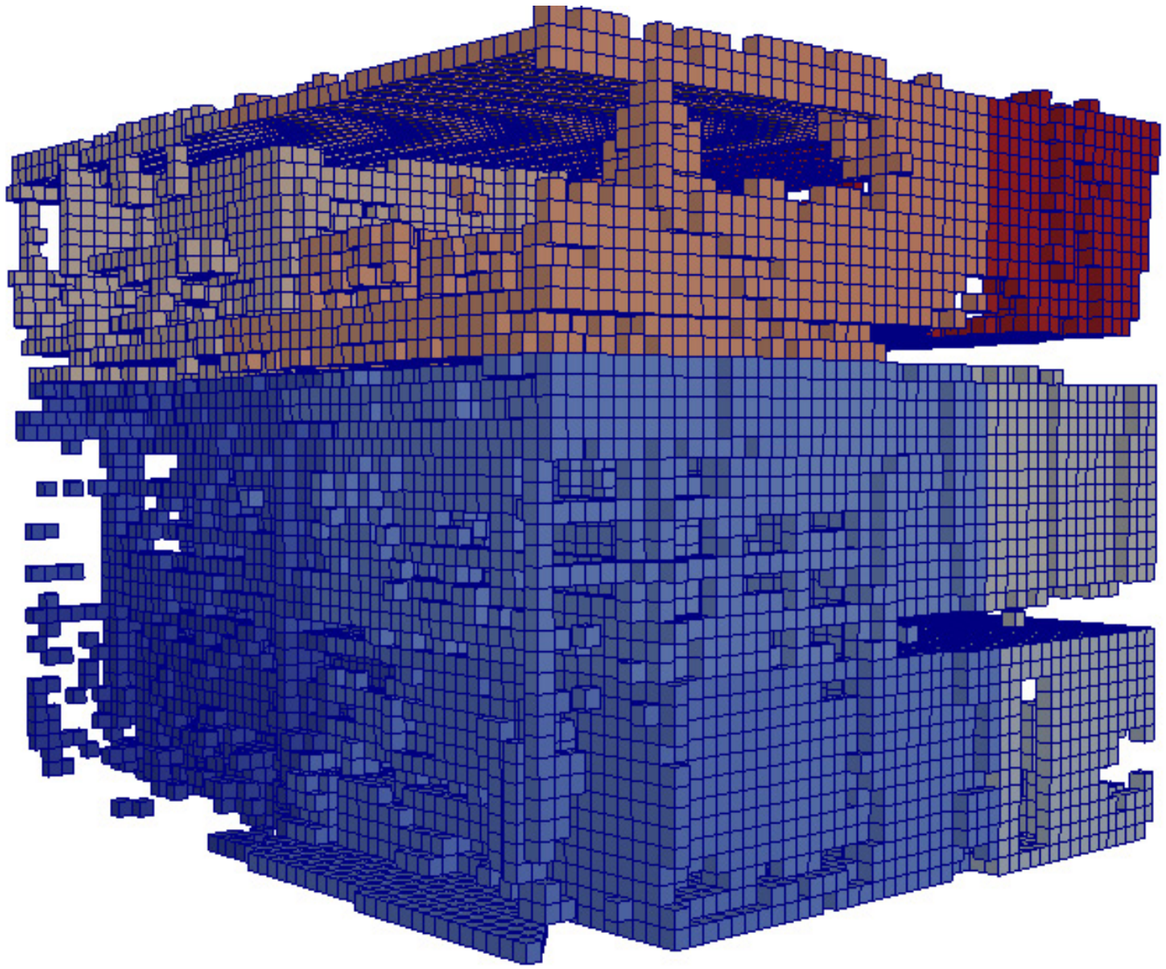
Octree

An octree is a three dimensional extension of a region quadtree data structure. It consist out of a cubical volume and is recursively subdivided into eight congruent disjoint cubes (called octants) until blocks of a uniform colour are obtained, or a predetermined level of decomposition is reached (Samet, 1988).



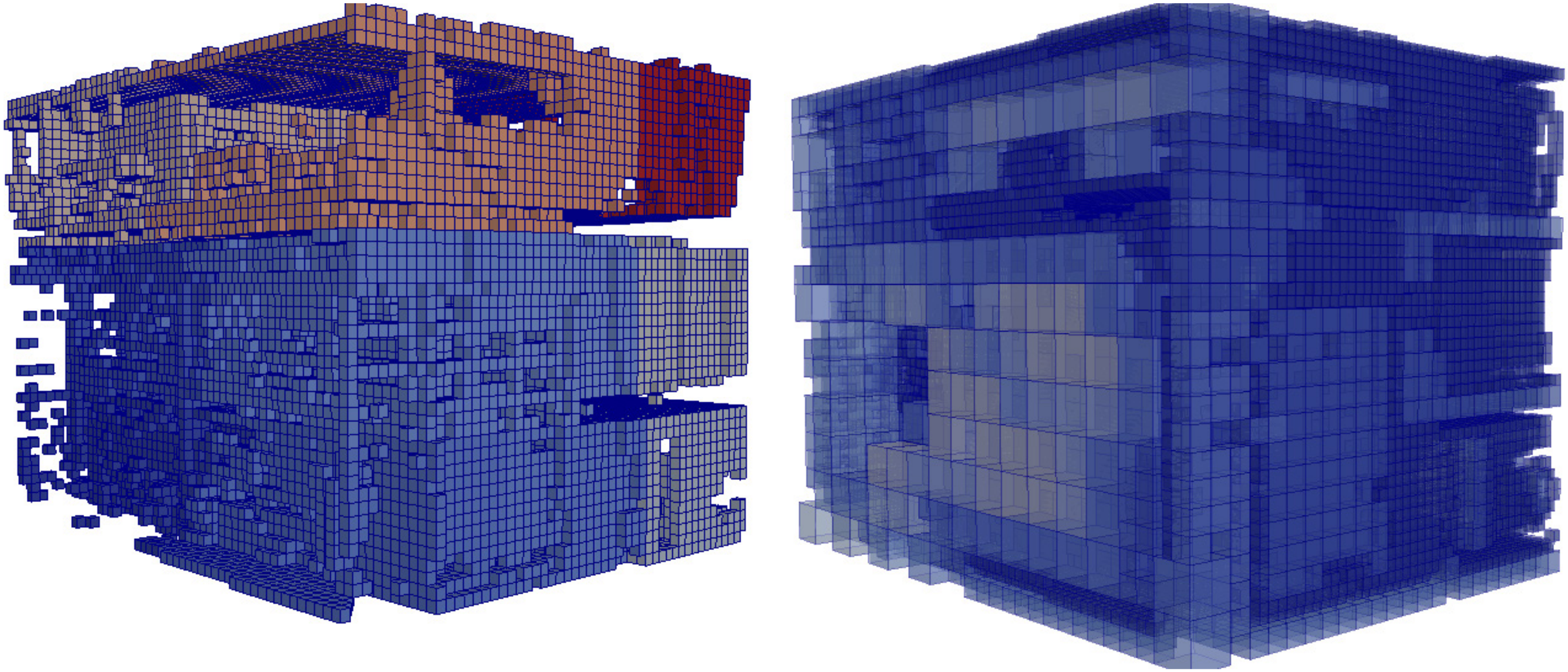
Octree





point cloud → Non empty space

Octree



Non empty space → empty space

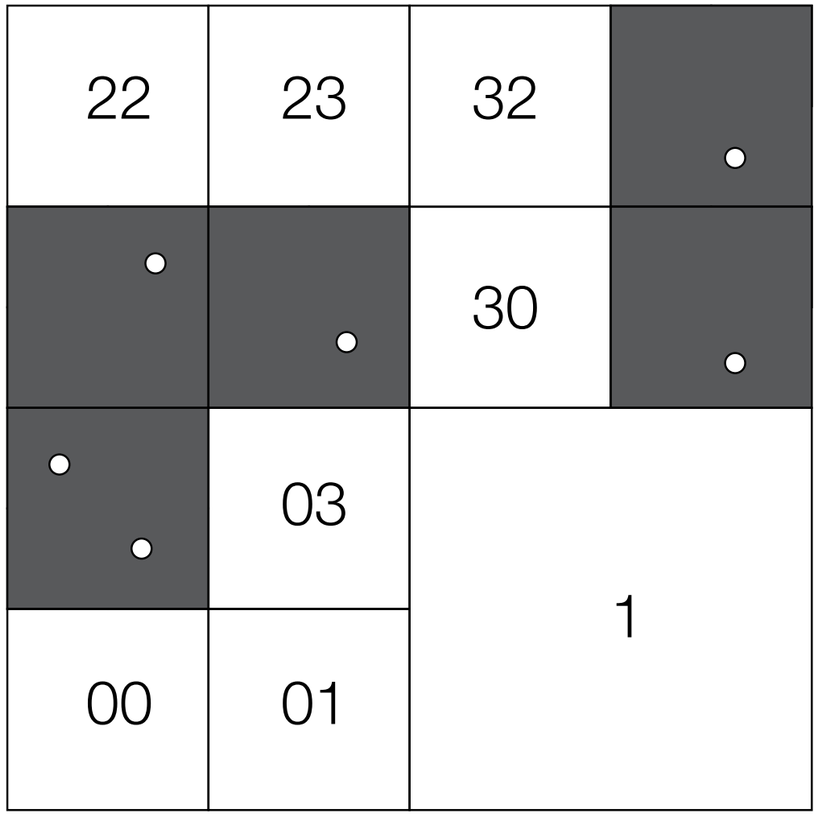
Octree

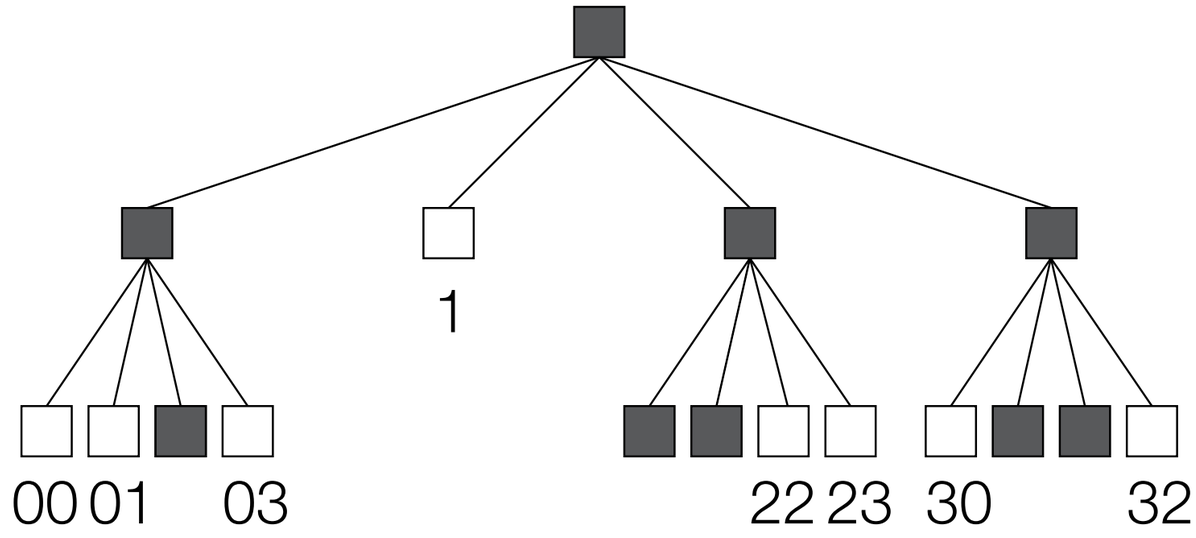
Goal of creating the octree:

- Structured dataset
- Identify the empty space
- Can derive the connectivity between the empty space
- A large empty area can be represented by a single node high in the octree reducing the amount of octants
    - Reducing the storage space
- Minimal use of geomatrical operations

Locational code

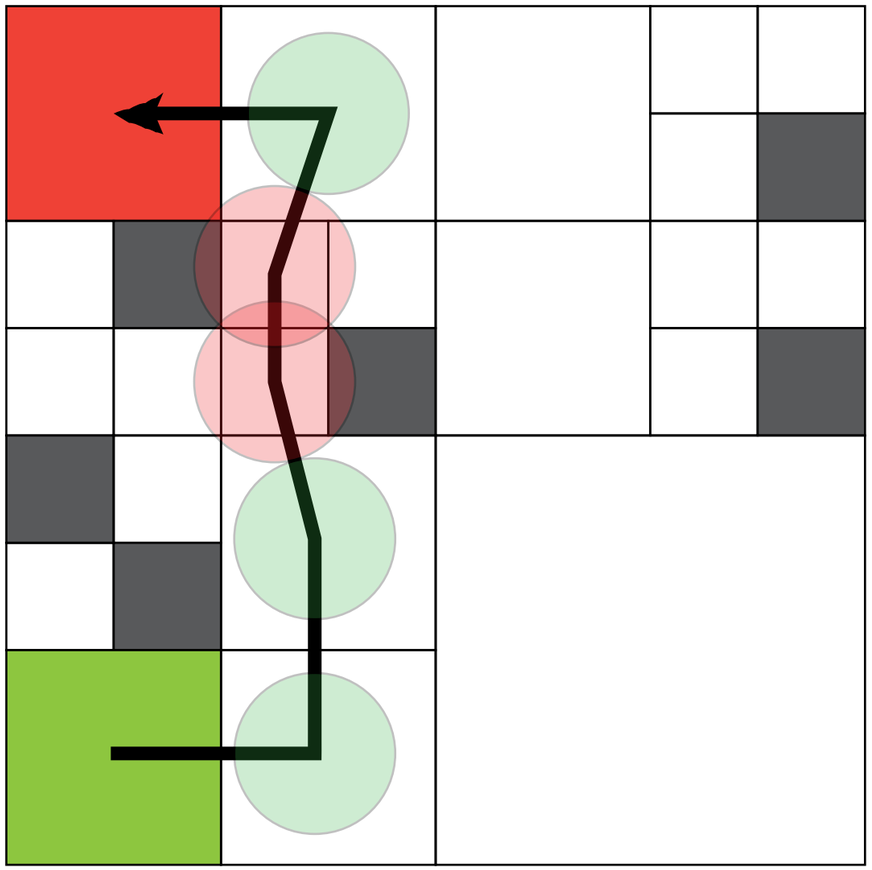
The route in the octree from the root node to a leaf node.

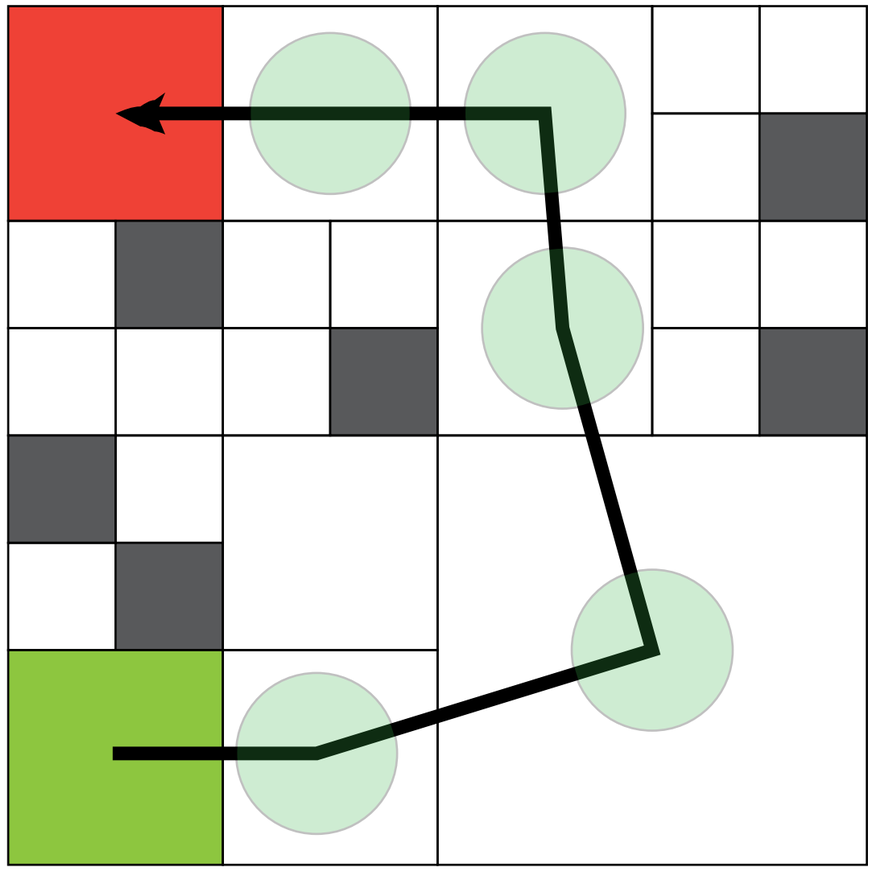




Collision free path

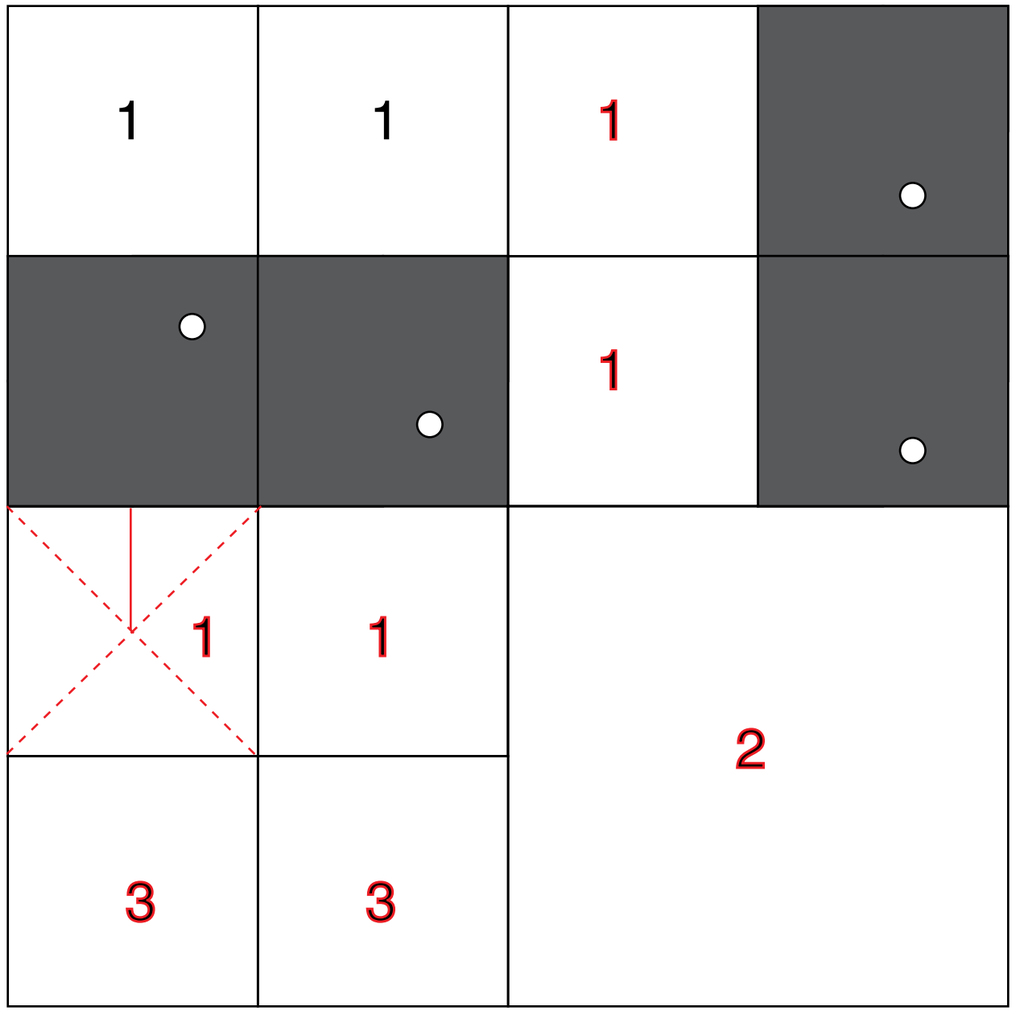
A collision free path occurs when a node in the octree is chosen which is closer than half the object size away from a black node.





Clearance map

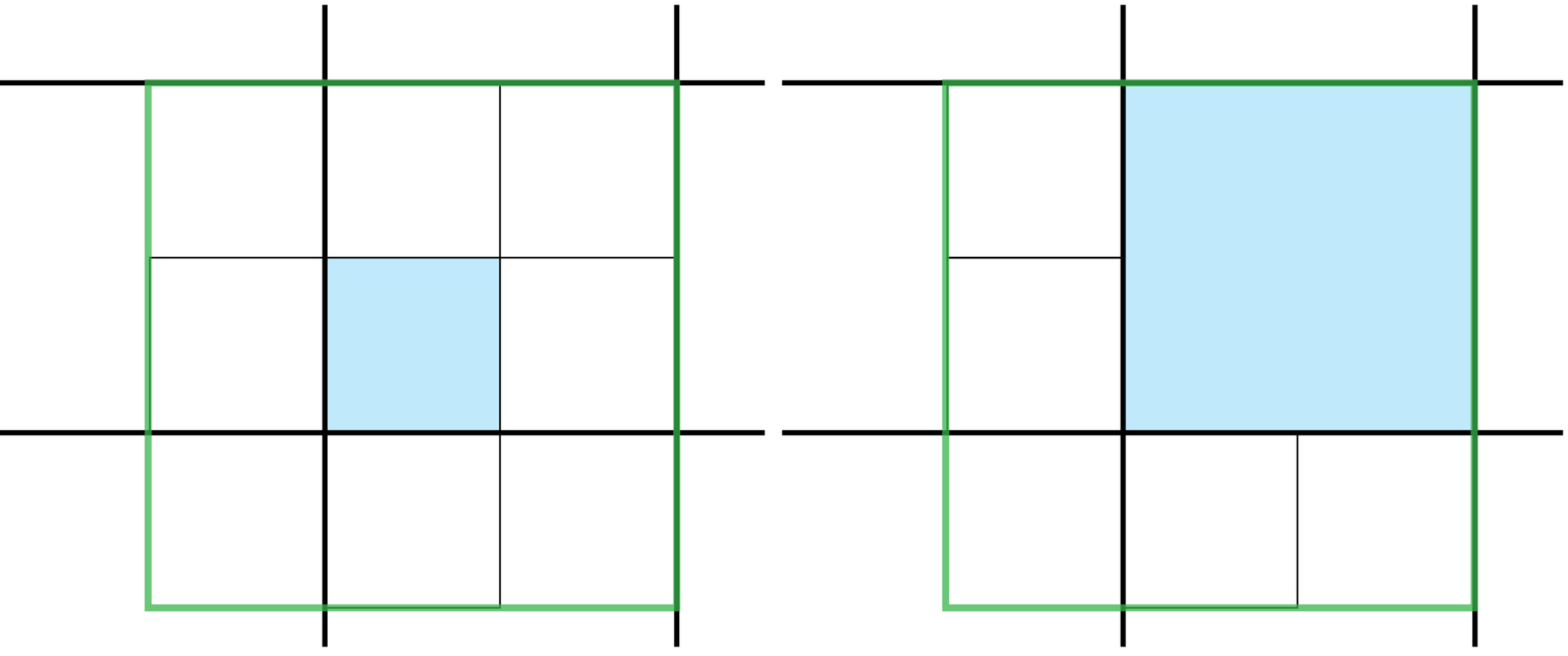
Calculate for each white node the minimal distance to a border with an black node



Clearance map

Distance transform

Calculate for each white node the minimal distance to a border with an black node

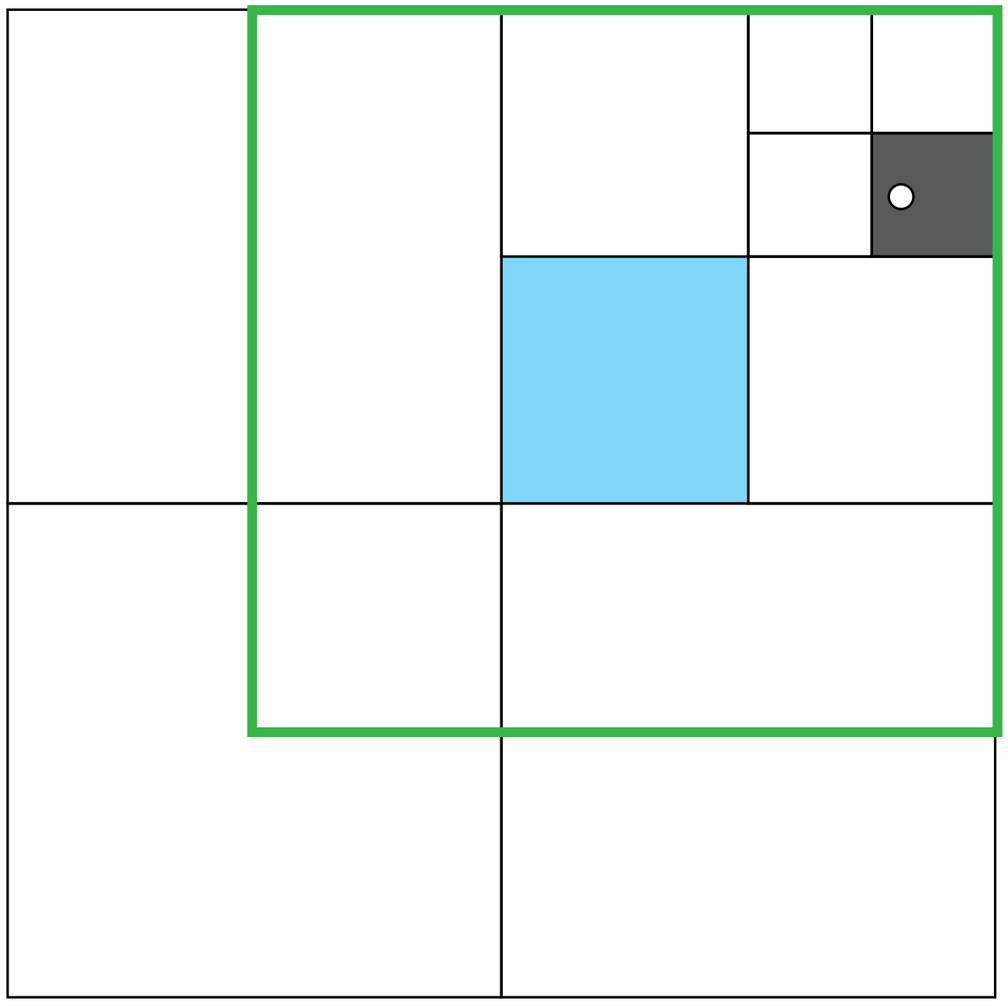


For any white node, its equal sized neighbours cannot all be black.

Otherwise merging would take place. (Samet, 1982)

Clearance map

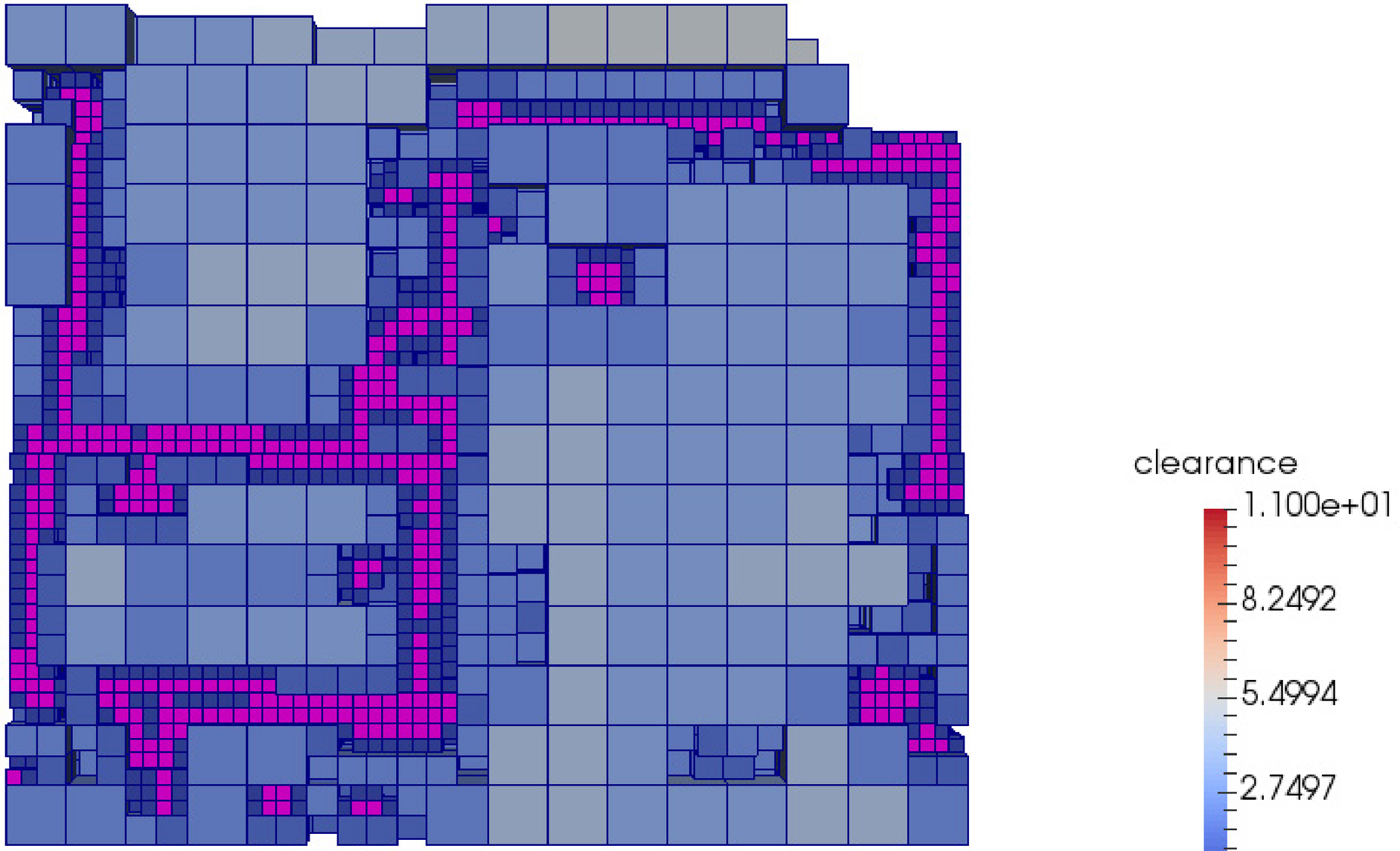
Distance transform



So closest black node needs to be a direct equal sized neighbours.

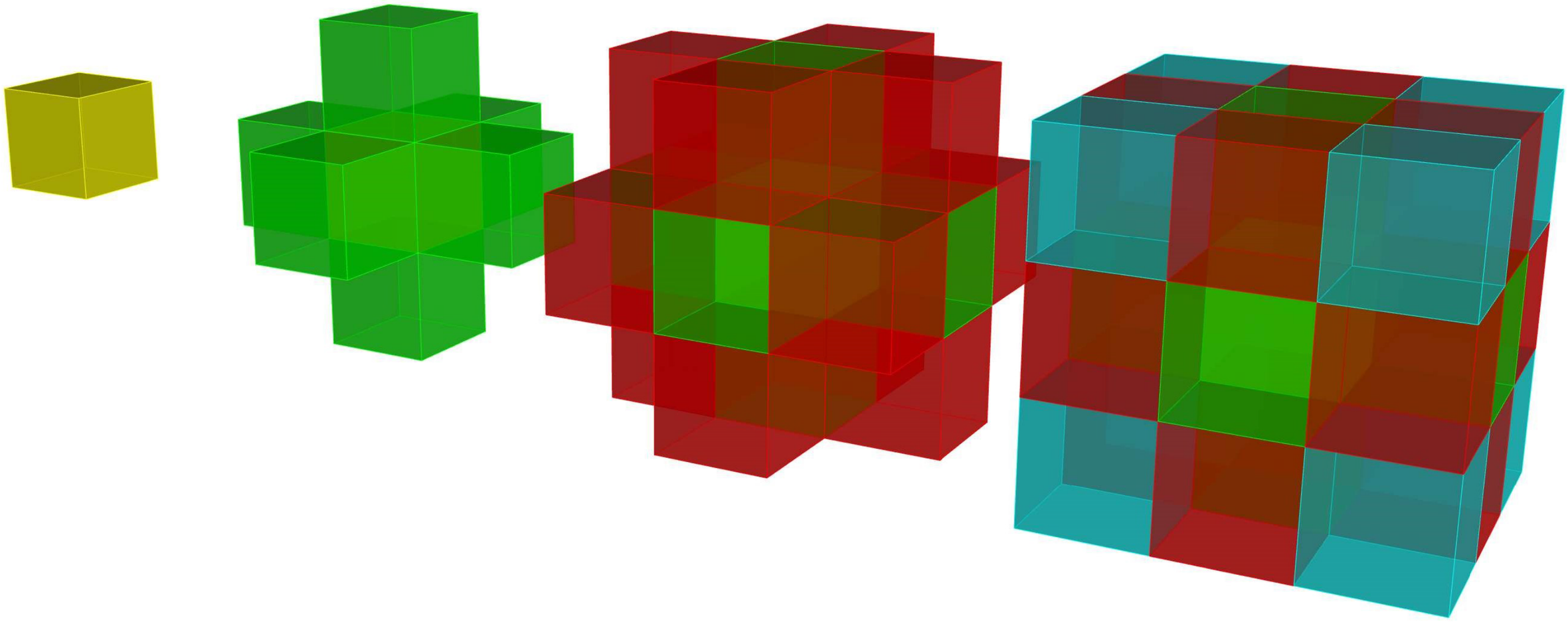
8 equal sized neighbours need to be checked in an quadtree (26 in oc-tree).

Clearance map



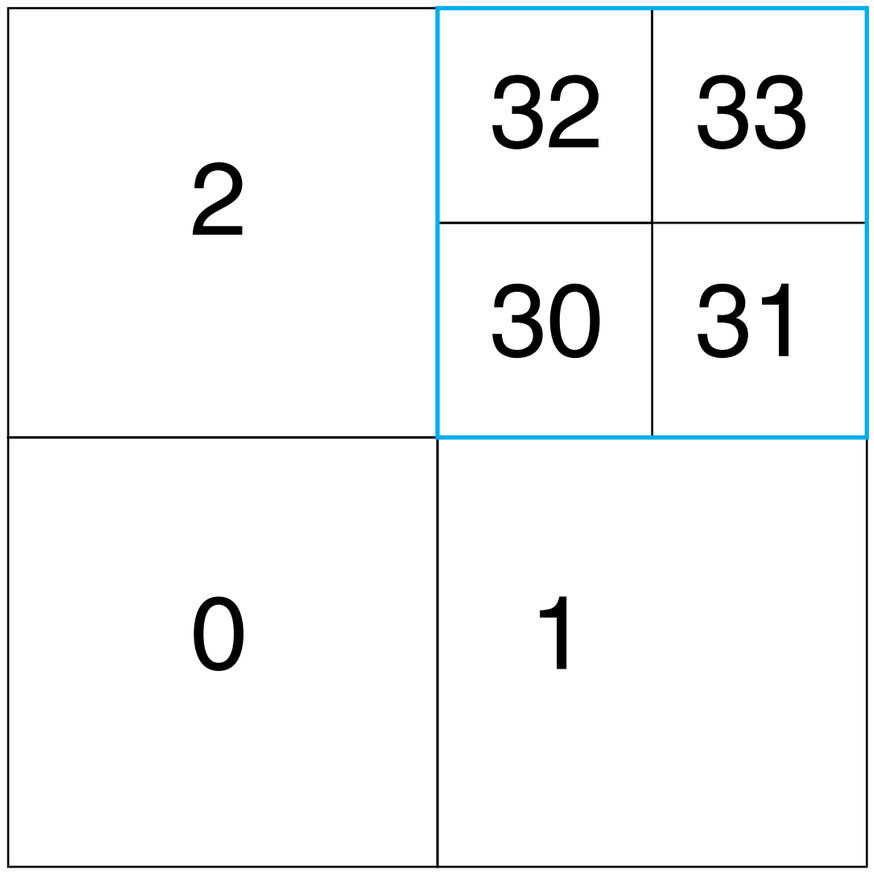
Connectivity between nodes

Node sharing a common: face, edge, vertex



Neighbour finding

Finding neighbours using the location code

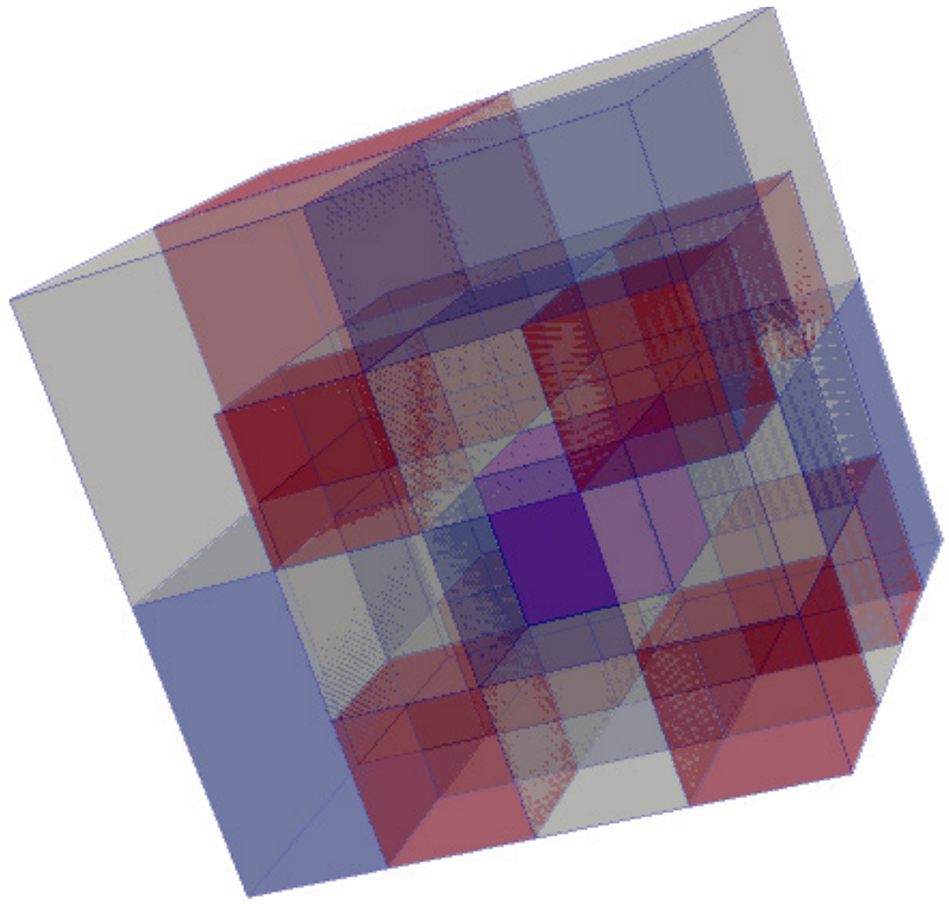


Neighbours of node 30:

30
31
32
33

Neighbour finding

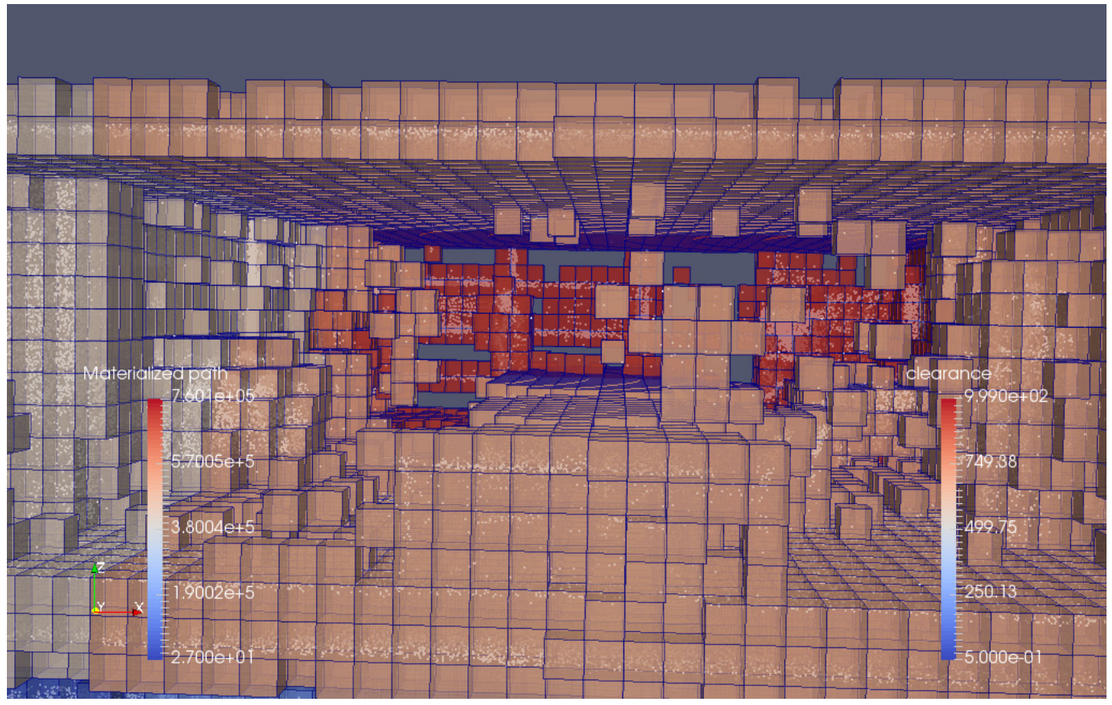
Neighbours during octree construction from top to down
Store neighbour connection in both nodes
Only equal sized or larger sized neighbours need to be stored

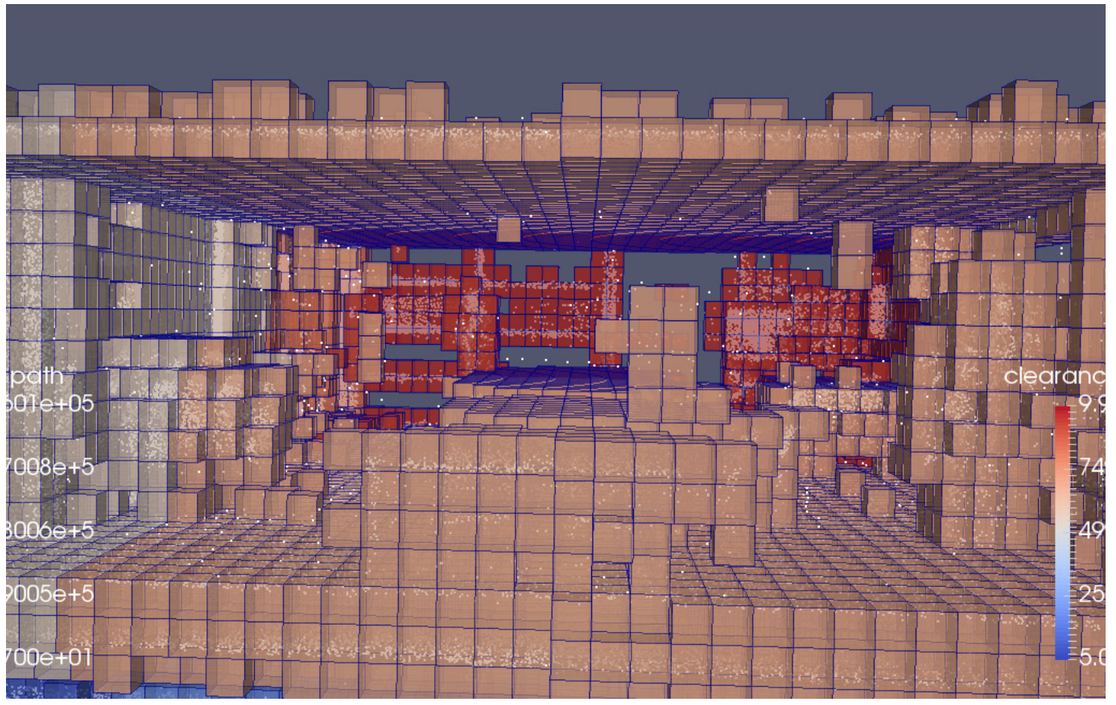


Neighbour finding

- Store explicit or derive

- filtering operation
    - Node with no neighbours is probably noise.





Next

- Testing the effects of the components
- Research the possibility to separate the interior and exterior empty space