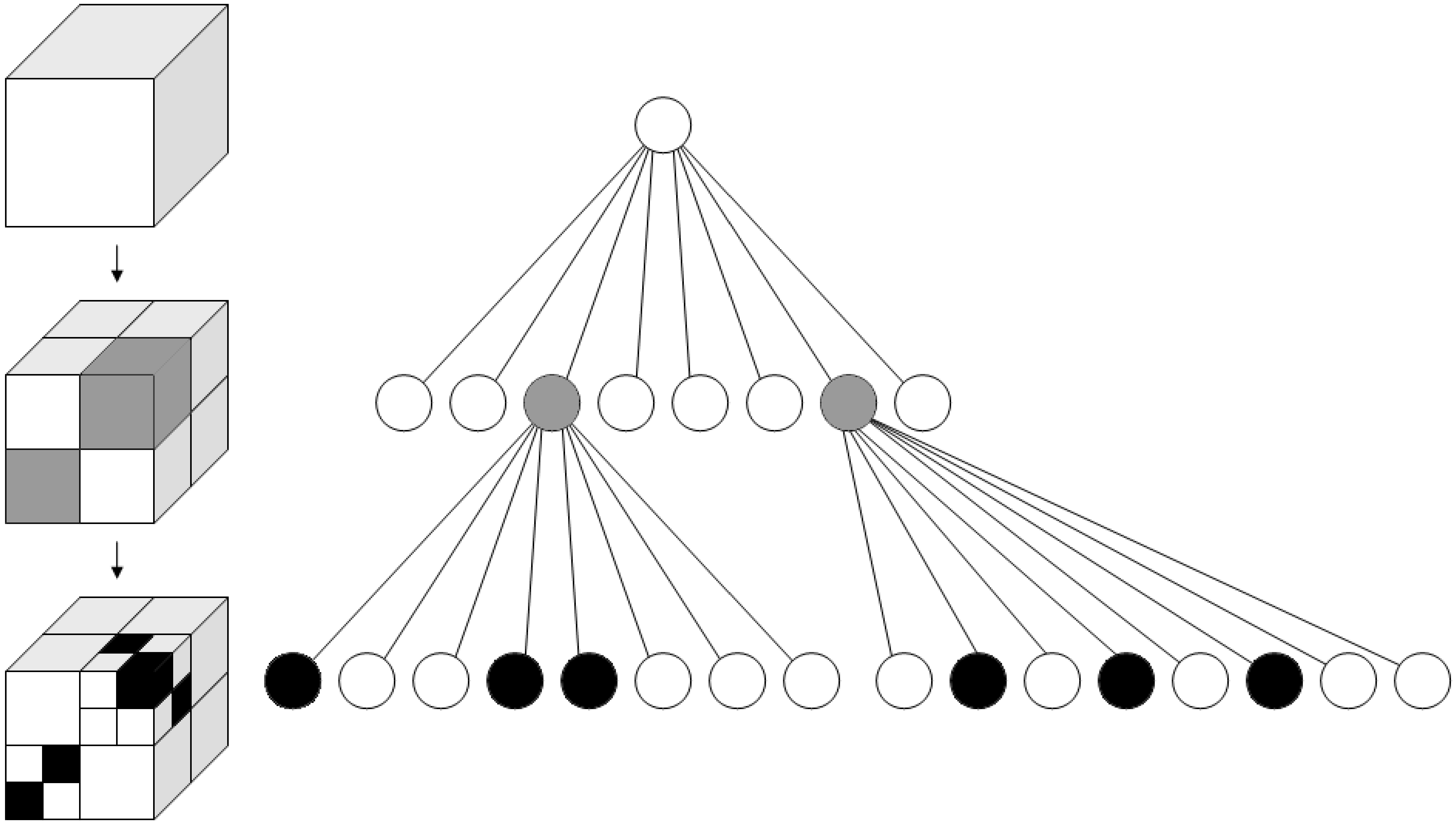
The relation between the computational effort and path length in A* pathfinding in an octree representation of an indoor point cloud.

By: Olivier Rodenberg

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

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).



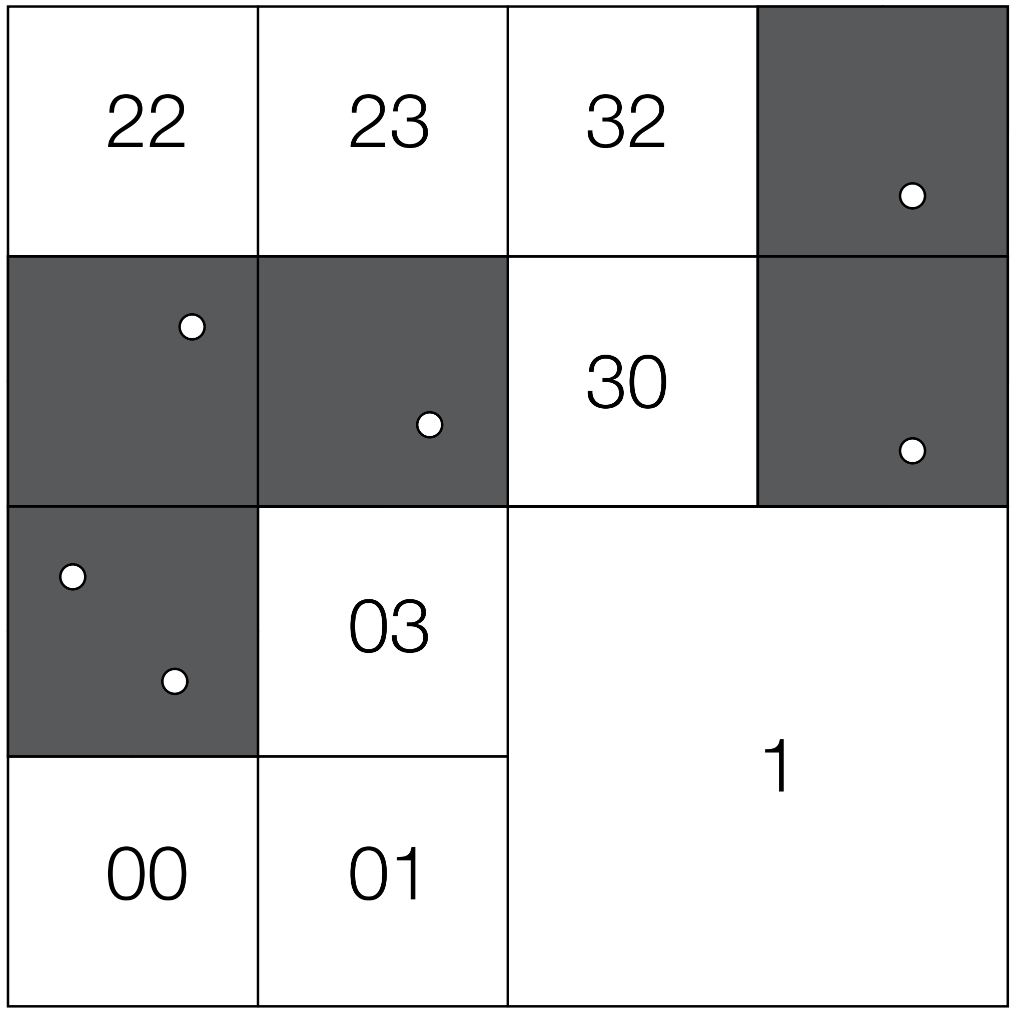
Octree & Naviagtion

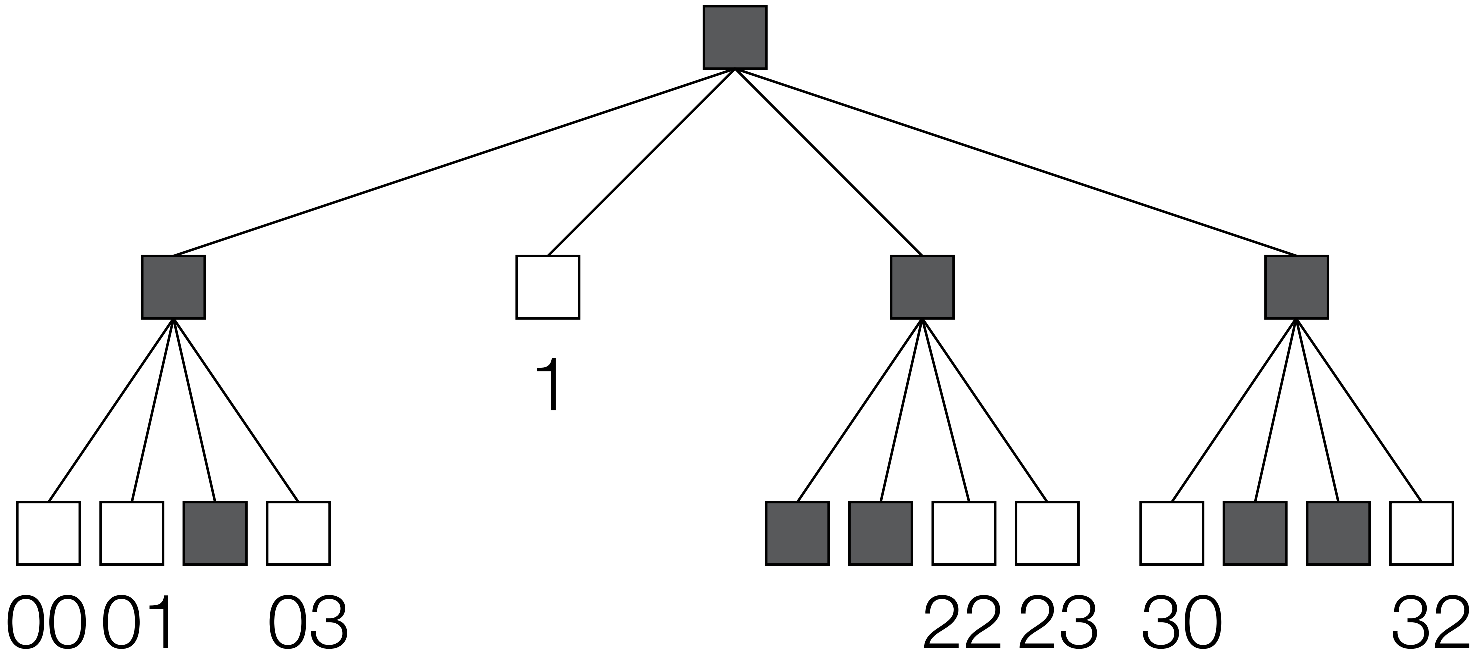
Advantages:

- A large empty area can be represented by a single node high in the octree reducing the amount of octants
- (empty) Space is stored in an hierarchical structure

Location code

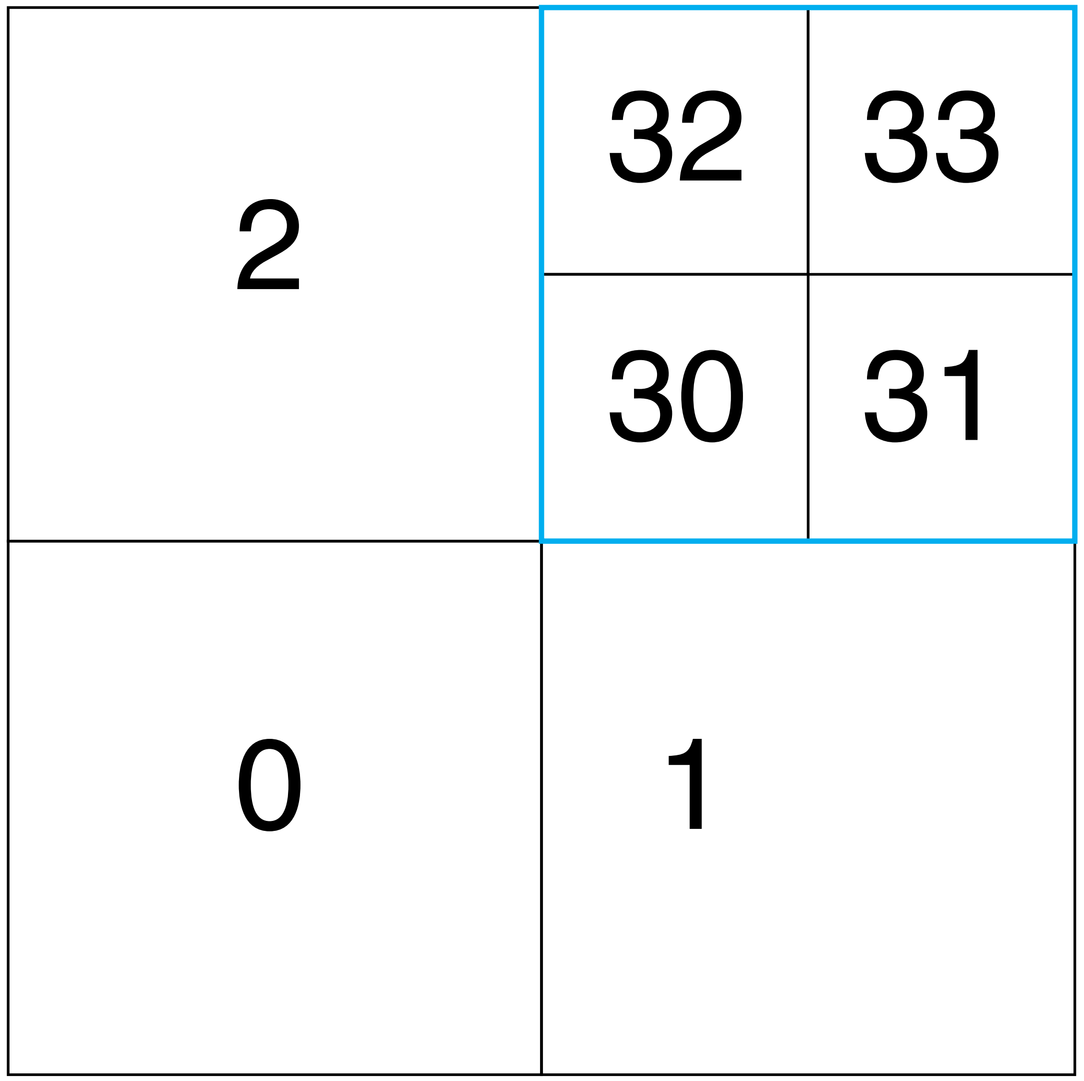
The route in the octree from the root node to a leaf node.





Neighbour finding

Finding neighbours using the location code



Neighbours of node 30:

30
31
32
33

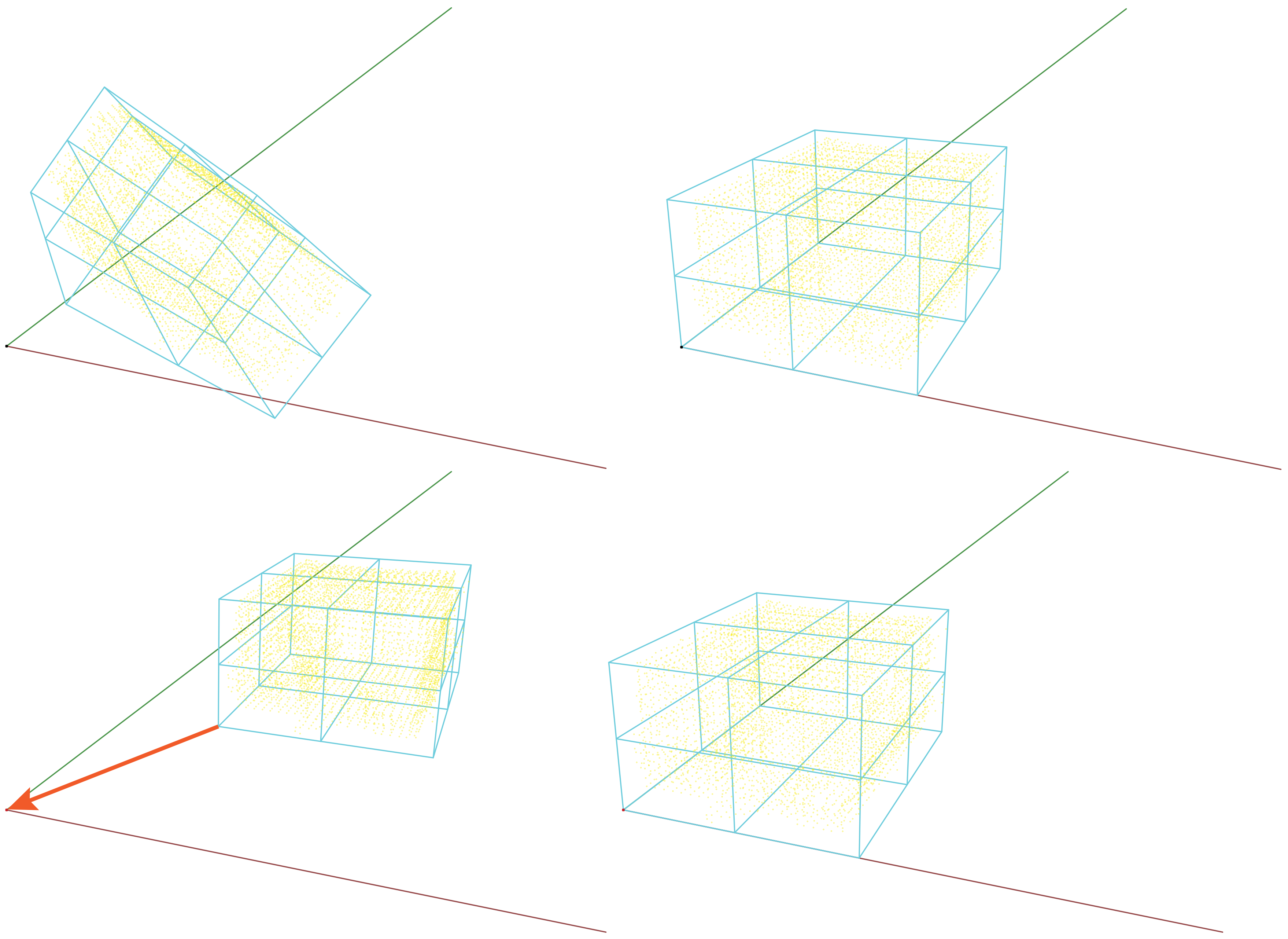
Research question

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 effort and path length and what is their effect?

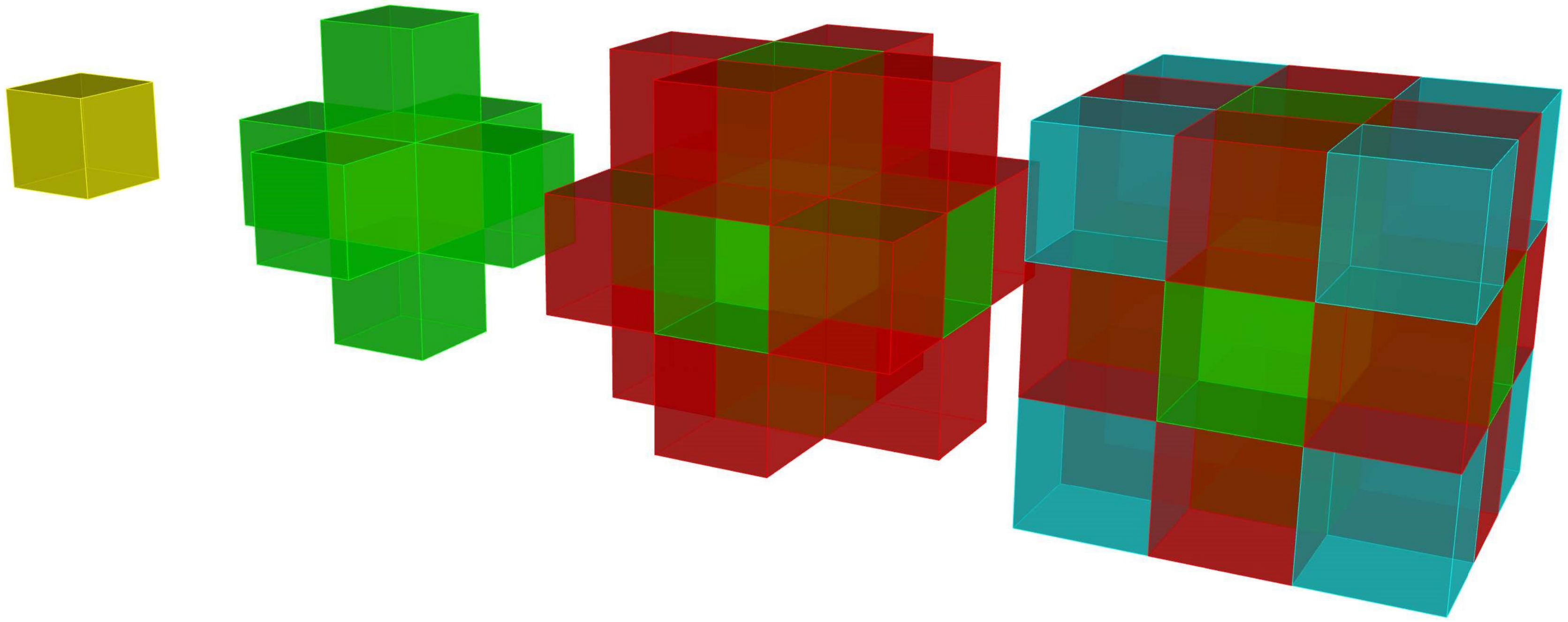
Research question

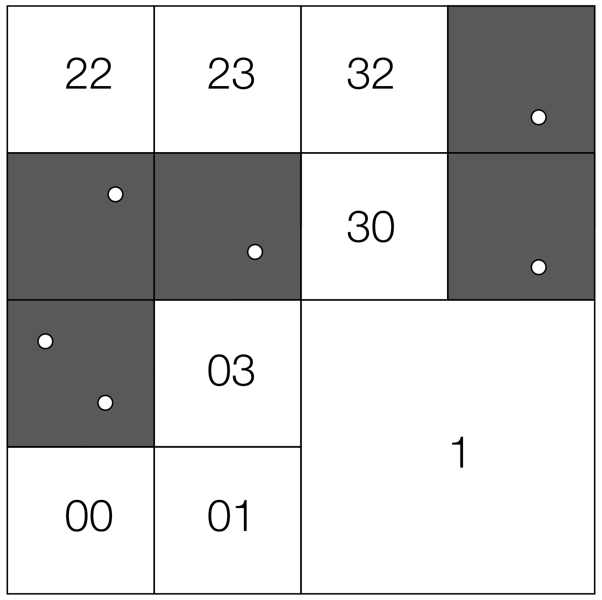
What point cloud processing operations are important for the generation of the octree and what is their effect?

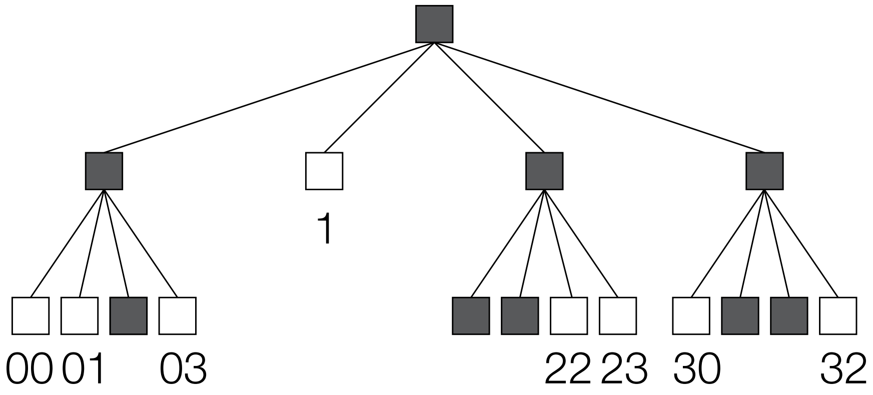


Research question

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







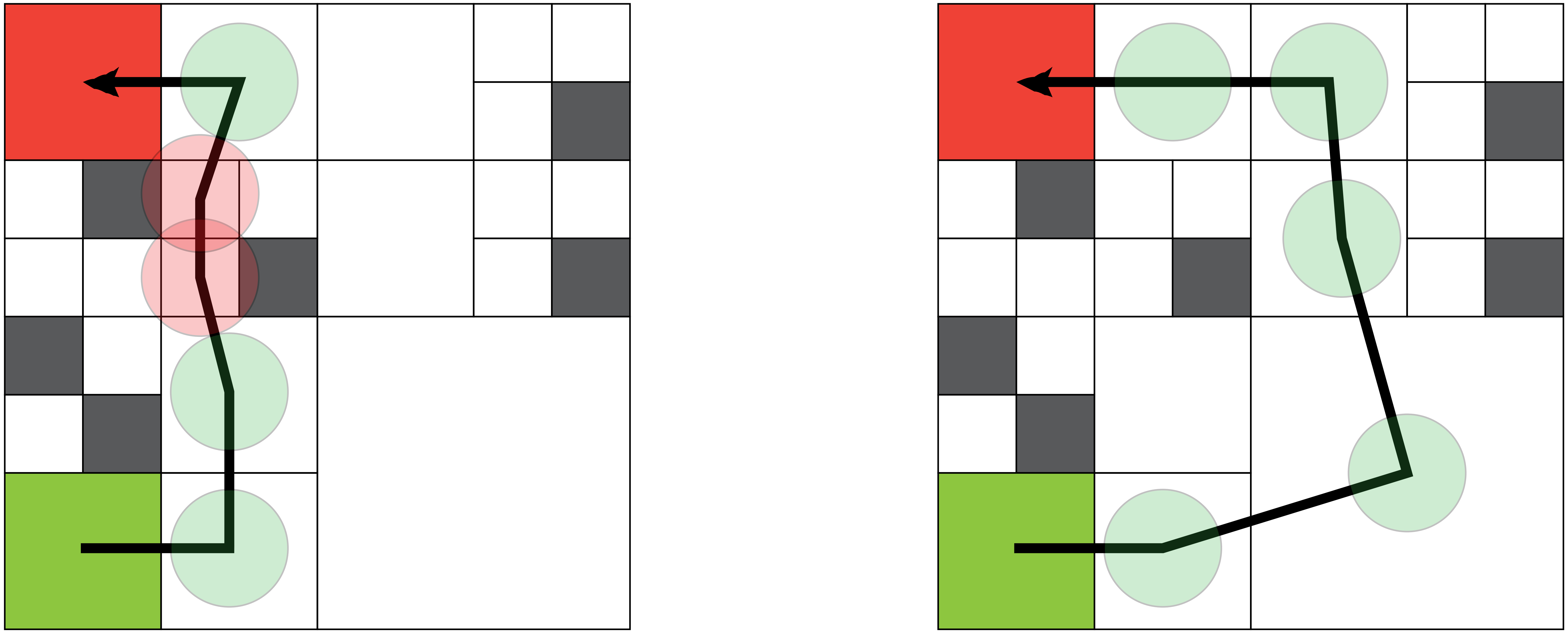
Research question

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

- Heuristic cost in A* path finding

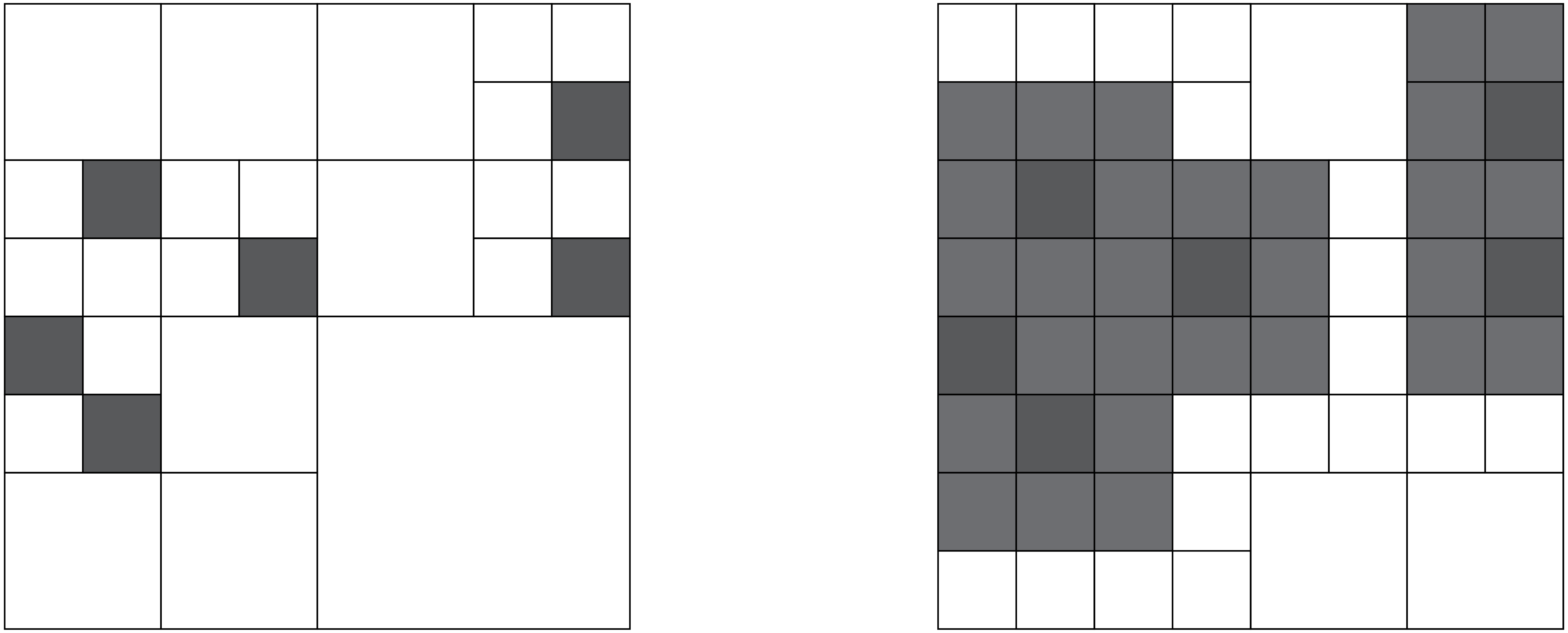
Object size

A collision free path considering the object size



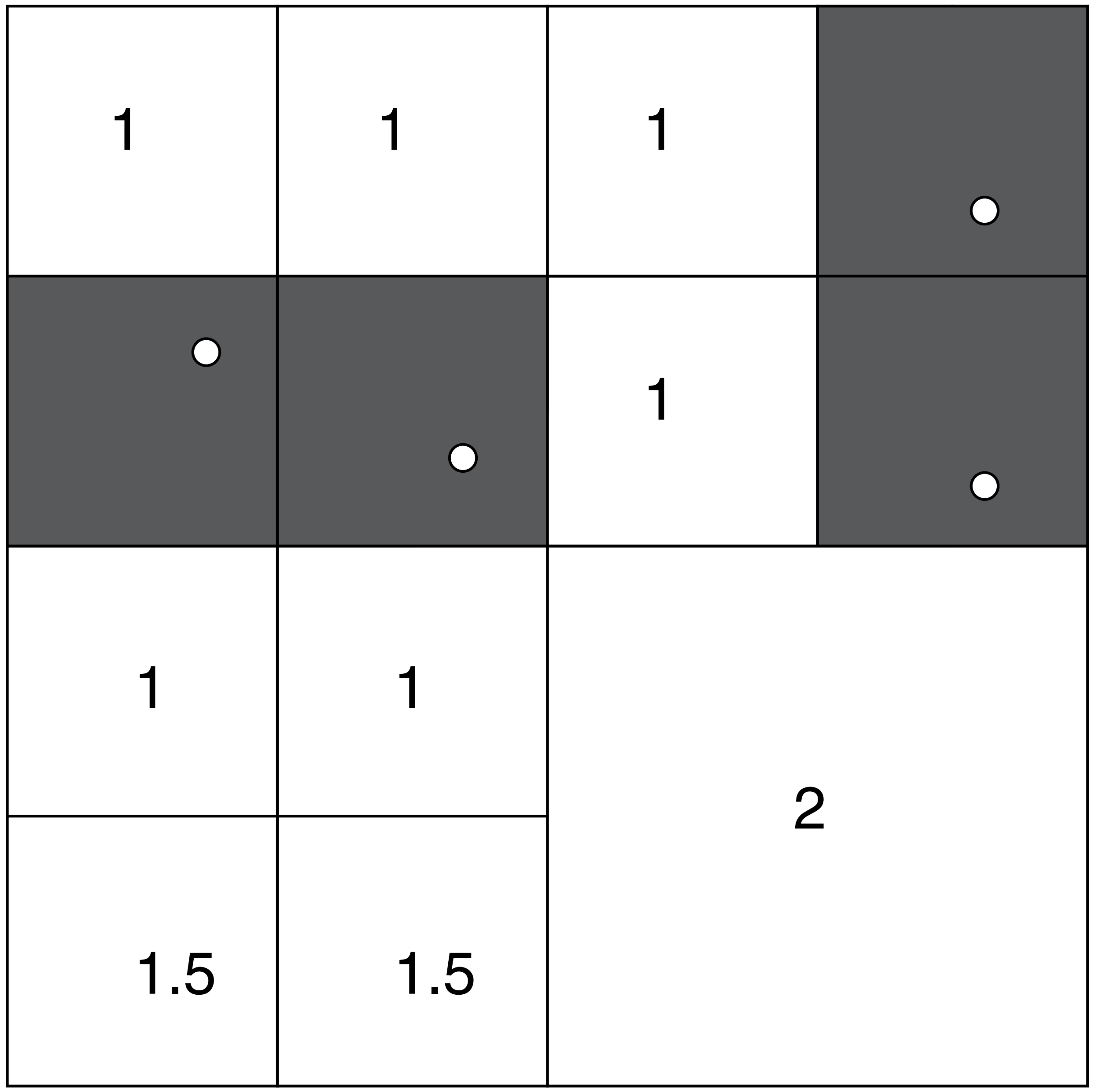
Object size

Buffer of at least half the size of the object around each black node



Object size

Distance transform/ map

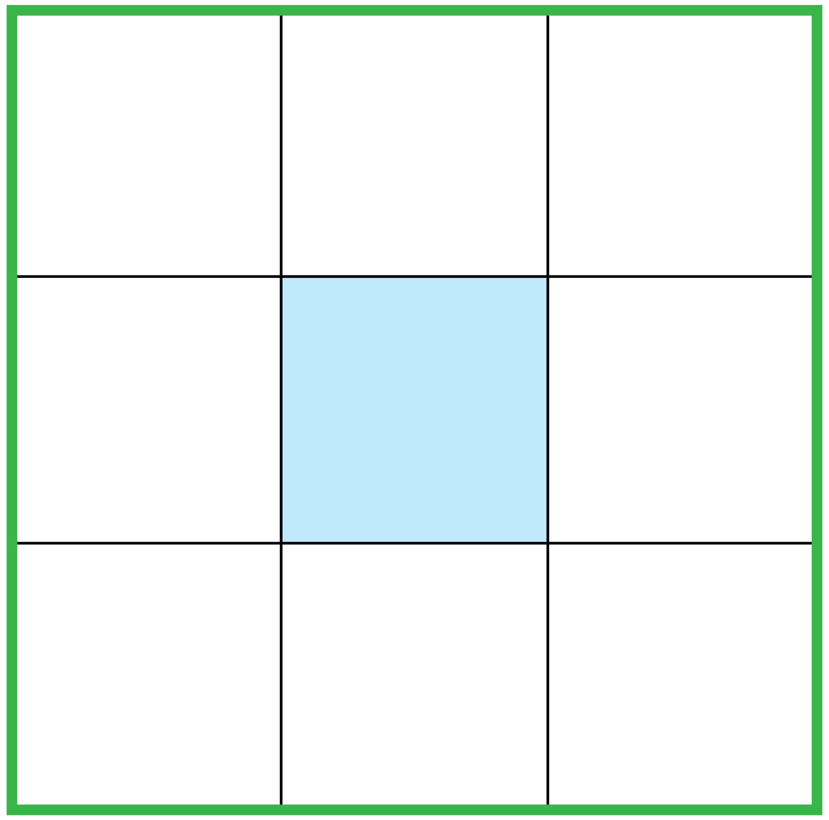


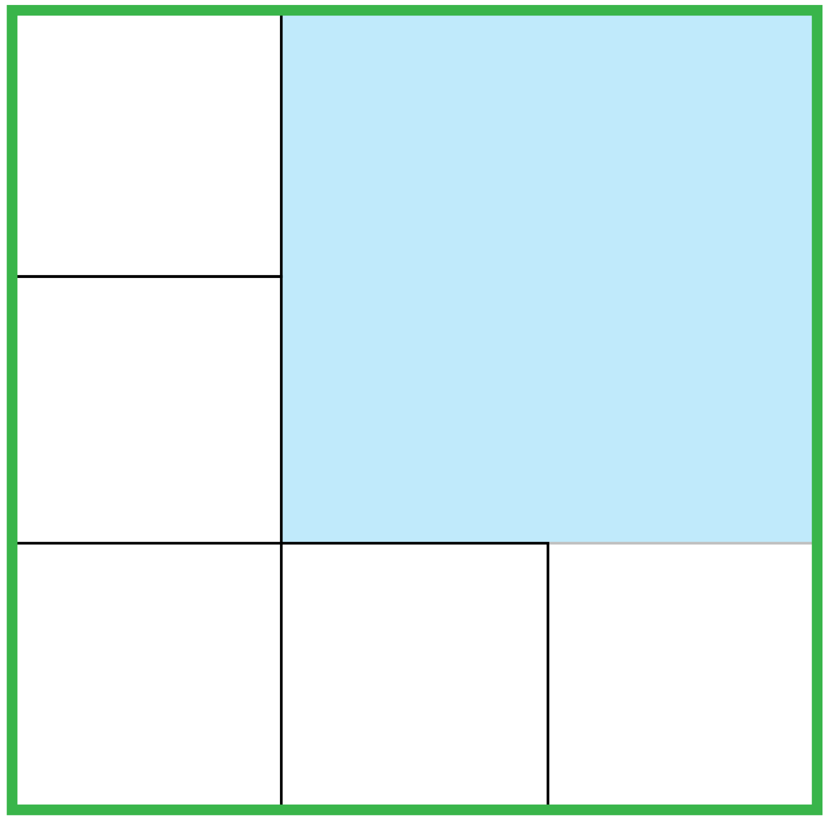
Calculate for each white node the minimal distance to a border with an black node

Object size

Distance transform

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



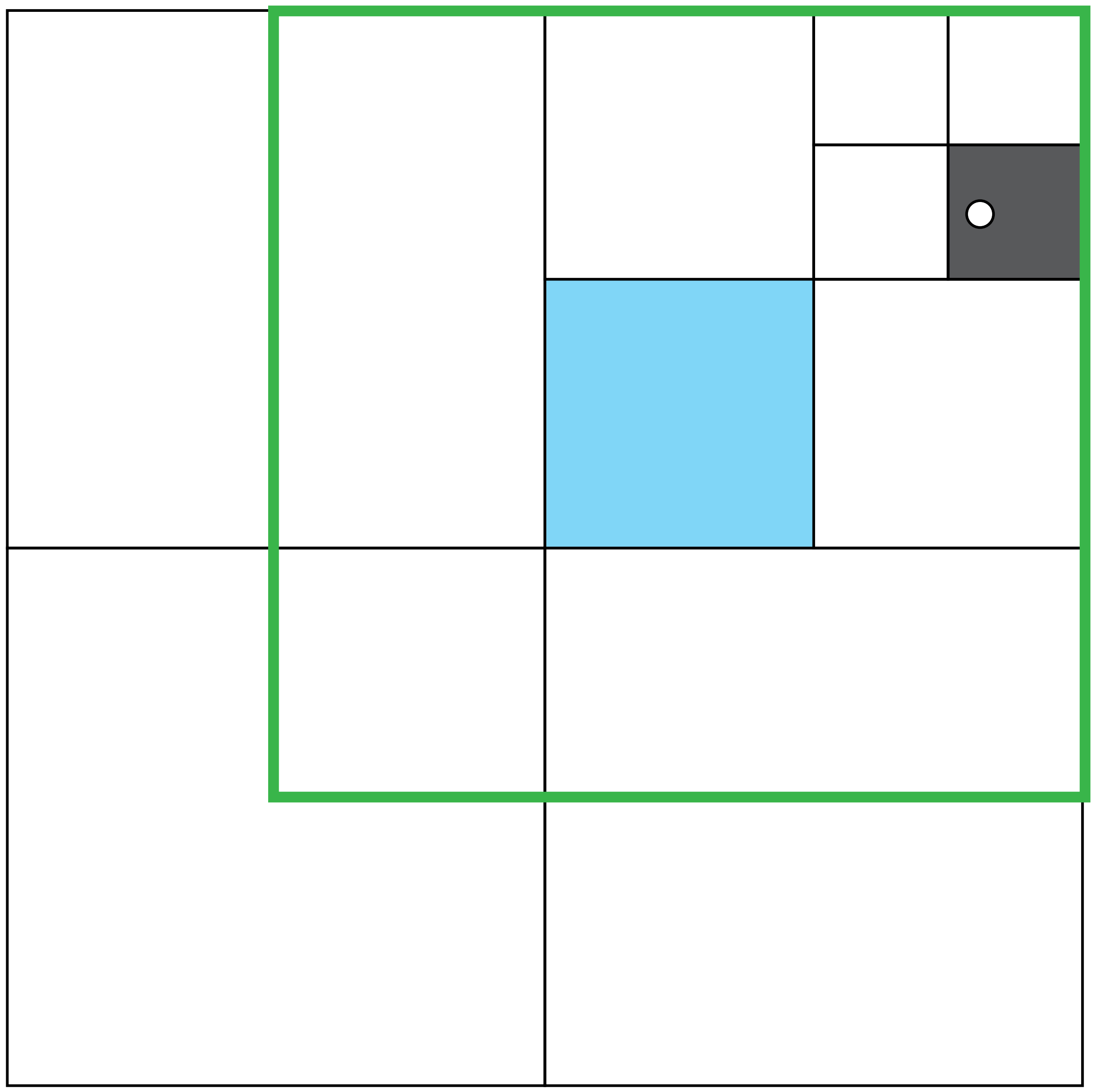


For any white node, its equal sized neighbours cannot all be black.

Otherwise merging would take place. (Samet, 1982)

Object size

Distance transform



So closest black node needs to be a direct neighbours.

8 eqeal sized neighbours need to be checked in an quadtree (26 in octree).

Next

- Implement distance transform
- Find all 26 neighbours in an octree