The effect of A* path-finding characteristics on the path length and performance in an octree representation of an indoor point cloud

Olivier Rodenberg
Delft University of Technology

Abstract:

This work presents a new workflow for path finding through an octree representation of an indoor point cloud. I applied the following steps: 1) the point cloud is processed so it fits best in an octree; 2) during the octree generation the interior empty nodes are identified and further processed; 3) for each interior empty node the distance to the closest non empty node directly under it is computed, this can be used as constraint in path finding; 4) a network graph is computed with the connectivity of all interior empty nodes; 5) a collision avoidance system is pre-processed in two steps: firstly the clearance of each empty node is computed, and secondly the maximal crossing value between two empty neighbouring nodes is computed. 6) the A* path finding algorithm is conducted.

The A* uses the interior empty nodes and the network graph to find a path. Finally, benchmark test were conducted to identify the effect of octree operators and A* operations on the path length and computation time. The A* path finding computation time is positively affected by: firstly, pre-processing a network graph, using a Manhattan distance and by keeping the octree depth to a minimum. The path length is positively influenced by: extending the path connectivity, by using an Euclidean distance type and by increasing the octree depth. Although the octree depth depends on the minimal and maximal spatial resolution needed for path finding. And this is dependant of the size and orientation of the point cloud. By pre processing a point cloud the spatial resolution can be increased.

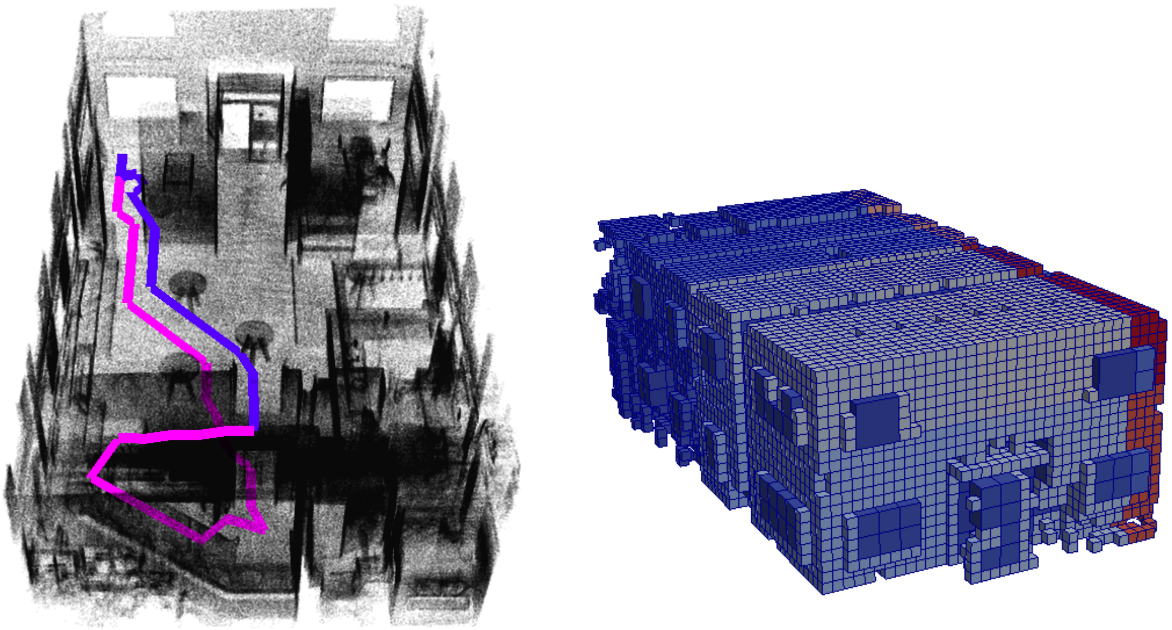

Figure 1: Example of a A* path-finding on an indoor point cloud (left) relying on the octree structure of its empty space (right).