3D Path-finding in a voxelized model of indoor environments

Graduation Plan

Martijn Koopman

January 12, 2016

1 Introduction

Path-finding is the core of indoor navigation. Not only for human navigation, but also for navigation of autonomous vehicles like robots and drones.

1.1 Actors

Humans, robots and drones are different kind of actors that each have different requirements on the path that has to be computed. These requirements depend on the properties of the actor. The properties that influence path-finding are listed below.

Size of the actor The size of the actor determines whether or not the actor fits through a passage. The size is expressed in width and height. The width and height can vary between different actors of the same type. For example, an actor of type 'pedestrian' can be a tall person (adult) or short person (child).

Mode of locomotion The mode of locomotion is the method used to move through space. This can for example be walking, driving or flying. Walking and driving actors are bound to a ground surface and can not rise of this ground surface. Next to that, driving actors are bound to a ground surface with a low slope. This limits them from accessing steep surfaces like a staircase.

Notion of best path The notion of best path does not necessarily have be the shortest path. For example, it can also be the path with the least turns or the path with the least height differences. Next to that, human preferences can also influence the path. For example, humans might prefer to visit the central hall every time they change location.

5 example actors are defined in table 1 by their size, mode of locomotion and notion of the best path.

Actor	Size (cm)	Locomotion	Best path
Pedestrian	50 x 190	Walking	Via central hall
Wheelchair	80 x 140	Driving	Least height differences
Cleaning robot	40 x 10	Driving	Total ground coverage
Rotary-wing drone	50 x 40	Flying	Shortest
Fixed-wing drone	100 x 20	Flying	Fixed elevation

Table 1: Example actors

1.2 Voxelized model

One common representation for 3D data is a voxelized model. A voxelized model is a three-dimensional uniform rectilinear grid in which each grid element indicates the occupancy of an object within that space. One grid element in a voxelized model is called a *voxel* which is short for 'volumetric pixel'

A voxelized model has a very simple topology that makes it easily implementable for path-finding. Other 3D representations like the commonly used boundary representation (b-rep) lack this topology and require an additional step before path-finding can be applied. Besides that, certain operations are very easily and efficiently performed on a voxelized model. A path-finding method might utilize these operations for efficient path-finding. 4 of such operations are listed below.

Distance transform A distance field is a derived volumetric model in which each voxel indicates the distance to a closest point. This closest point might for example be a point on a wall.

Skeletonization A skeleton is minimal representation of a model in which the geometry is thinned to a thickness of one voxel. Although the geometry of the model is heavily changed, the internal topology of the model remains the same. This skeleton is very suitable for path-finding for multiple reasons. Firstly, it contains topology. Secondly, it reduces the number of voxels that can be traversed. Thirdly, it guarantees a path through the center of the space.

Dilation Dilation adds a buffer around voxels of an object. Such a buffer can be used in path-finding to consider the size of the actor. For example, a dilation of the model with half the width of the actor will close any passages that are too narrow for the actor to pass through.

Ground, wall and staircase classification Ground, walls and staircases can easily be detected in a voxelized model. The slope of the surface and the presence of empty space voxels above or nearby give an indication for this classification.

1.3 Objective

In this thesis research will be done on a path-finding method for voxelized models of indoor environments. This path-finding method must be applicable for different kind of actors and should consider the geometric, topologic and semantic properties of the model. The actor properties that influence path-finding are parameterized and passed to the path-finding method.

2 Related work

A lot of research has been done on path-finding methods and their applications. Within the domain of geomatics path-finding is mostly used for navigation, but it can also be used for other applications like emergency evacuation simulation [8] and building performance simulation [6].

Shortest path algorithms lie at the basis of path finding. These algorithms are used to compute the shortest path through an environment while avoiding any obstacles. Some common shortest path algorithms are described in subsection 2.1.

Each path-finding method has different characteristics like the model it operates on and its notion of the best path. These characteristics depend on the application of the path-finding method. For example, the fastest path and a graph network would suffice for outdoor car navigation while a shortest path and a grid would suffice for indoor robotic navigation.

There are several papers that describe path-finding methods for indoor environments using a voxelized model or a similar model. Some of these methods also consider the properties of the actor that are described in section 1.1. These path-finding methods are described in subsection 2.2.

2.1 Shortest path algorithms

There are many implementations of shortest path algorithms. One of the simplest implementations is the breadth-first search algorithm which is also called the "Bushfire algorithm" or "Flood fill algorithm". This algorithm works by expanding a region from a starting point until the ending point falls within this region. Each cell in this region is assigned the number of steps from the starting point. The shortest path is found by traversing the cells from ending point to starting point by visiting neighboring cells with the lowest assigned number.

Another common shortest path algorithm is Dijkstra's algorithm [5]. This algorithm is meant for graphs and is technically the same to breadth-first search when applied to a grid.

Breadth-first search and Dijkstra both guarantee to give the shortest path, but they are not computationally efficient because they only consider the distance to the starting point and not the distance to the ending point.

A more efficient algorithm is the A* algorithm [7] (pronounced A-star). This algorithm also considers the distance to the ending point by using a heuristic function. This heuristic is an estimation of the distance and can for example be the Euclidean distance or Manhattan distance. The sum of the travelled distance and the heuristic function is used to determine which neighboring cell should be visited. This sum should be minimum. A* greatlty reduces the number of cells to visit making it much more efficient than breadth-first search and Dijkstra's algorithm.

There are many variants of the A* algorithm. Some of these make it computationally more efficient while others make it suitable for dynamic environments. For example, Hierarchical Path-Finding A* [3] uses a hierarchical approach for reducing problem complexity in path-finding. This speeds up the path-finding by 10 times, but does not always result in the optimal path.

Another variant is the θ* algorithm [4] (pronounced Theta-star). This algorithms allows to take more possible directions into consideration. As result less steps have to be taken to go from starting point to ending point. To implement θ* more information about the environment has to be known to avoid any obstacles. A visibility graph could be used for this.

The D* algorithm [9] is suitable for partially unknown environments. It uses a network graph that gets updated when new information is available. This algorithm is mostly used for robotic navigation.

2.2 3D indoor path-finding

Voxelized models are used in multiple researches on indoor path-finding. In some of these researches path-finding is actually performed on the voxelized model. In other researches the voxelized model is just an intermediate model from which a graph is derived. In this subsection several path-finding papers are describe that use a graph-based or grid-based approach.

2.2.1 Graph-based

To perform path-finding on a graph two things have to be realized. First, the graph has to be derived from a geometrical model. Secondly, a path has to be computed.

Graph generation Graph generation is a two-step process. First, empty regions of space have to be identified which act as nodes in the graph. Second, shared faces between regions have to be identified which act as edges in the graph. In this process the geometrical properties (width and height) of the regions and shared faces can be assigned to the nodes and edges in the graph. This makes it possible to take geometrical properties into account when performing path-finding. The same is true for semantical properties.

1. Region of empty space identification In [10, 1] a distance field is used for region creation. From this distance field a spherical [10] or cubical [1] region is derived. The center of such a region has the maximum distance value. In [11, 12] a region growing algorithm is used for region creation. In this process empty blocks of space are added to the region if they fulfil certain geometrical or semantical criteria. These geometrical criteria can be the height of the space or the slope of the floor underneath. The semantical criteria can be the name of the room or the purpose of that space.

2. Shared face identification In [10, 12] shared faces are identified by intersecting overlapping spheres [10] or cuboids [12] that were derived from the distance field. The size of this intersection can then be used to consider the size of the actor in the path-finding. If the actor is smaller than this intersection, than it can go through the passage. In [11] connected edges are identified by performing a triangulation on the contours of the regions. A disadvantage of this approach is that size of the actor can not be taken into account.

Path generation In [10, 1, 11] the A* algorithm is used to compute the shortest path. This gives a coarse path from region to region that is not conform human walking. In [1] a finer path is computed using the voxels that make up each region. This path can be conform human walking.

2.2.2 Grid-based

In [2] no graph is derived from the voxelized model. The voxelized model itself is used for path-finding. As a result a lot of geometrical properties of the actor and the environment are taken into account. For example the *horizontal footspan* which is the number of voxels a human could cross in one step. A breadth-first search algorithm is used for path generation. This is not very efficient in a voxelized model when many voxels are traversable. Efficiency could be improved by using A* and a hierarchical data structure.

3 Research objectives

3.1 Objectives

The goal of this research is to develop a path-finding method that is applicable for different kinds of actors. This path-finding method has to operate on a voxelized model which is a geometrical, topological and semantical representation of an indoor environment. The characteristics of such a model should be effectively utilized by the path-finding method. The properties that describe an actor should be parameterized and passed to the path-finding method.

The main research question of this thesis is:

Is it possible to implement one path-finding method that is applicable for different kind of actors?

To achieve this, the following sub-questions will be important:

- What kind of actors exist?
- What requirements does each kind of actor have on the computed path?
- What parameters can describe the required path of an actor?
- In what data structure should the voxels be stored to facilitate path-finding?
- What is the influence of the model's resolution on the path-finding?
- What implementations could improve the perfomance of the path-finding method?

3.2 Scope of the research

This research will focus only on indoor environments, but a generic approach will be used that might make it suitable for outdoor environments also. Next to that, research will only focus on voxelized models. Other 3D representations like boundary representation (b-rep) are not considered.

4 Methodology

Research will be done to come up with the best path-finding method that is applicable for different actors.

4.1 Steps

The development of the path-finding method will be done in 3 consecutive steps. Information gained from literature study will be included in this path-finding method.

1. Selection navigable space The first step is to select only those voxels that are navigable for a certain actor. This selection depends on the mode of locomotion and size of the actor.

Locomotion	Voxels
Flying	Any empty space voxels
Walking	Empty space voxels with a ground voxel underneath
Driving	Empty space voxels with a ground voxel underneath with a low slope

Table 2: Selection of voxels based on locomotion

Of these empty space voxels only those are selected which have enough clearance nearby to hold the size of the actor. Dilation of the model could be used as operation for this selection.

2. Implemenation of best path The second step is to implement the best path for a certain actor. This step is subdivided in two smaller steps.

2.1. Shortest path A shortest path algorithm will be implemented. This will be the A* algorithm.

2.2. Actor specific path An algorithm will be implemented that gives the best path for a certain actor. For example, a wheelchair and flying drone might require a path with the least height differences because change in height costs energy.

3. Performance The third step is to consider any type of performance enhancements. This could for example be a hierarchical path-finding approach.

5 Schedule

5.1 Activities

The following schedule has been set up for the activities which are needed to meet the research objectives.

Start	End	Activity
9 nov	10 nov	**P1 - Progress review Graduation**
10 nov	11 jan	Literature study
11 jan	22 jan	**P2 - Formal Assessment Graduation Plan**
22 jan	5 feb	Implement shortest path algorithm
5 feb	19 feb	Implement selection of navigable space
19 feb	1 april	Implement best path for actor
end of	march	**P3 - Colloquium midterm**
1 april	20 may	Thesis writing
1 april	15 april	Implement performance enhancement
9 may	20 may	**P4 - Formal process assessment**
20 may	13 jun	Finalize thesis
6 jun	19 jun	Prepare final presentation
20 jun	7 jul	**P5 - Public presentation and final assessment**

5.2 Meetings

Meetings will be held once a week with the two supervisors S. Zlatanova and B.G.H. Gorte.

6 Tools and Data

6.1 Tools

ParaView will be used for research and development. ParaView is a software application for scientific analysis and visualization and is highly extensible through the Python scripting language and the C++ programming language. Prototyping will be done in Python and if performance is an issue than C++ can be used.

6.2 Data

Multiple Datasets will be used for research and development. These datasets are voxelized models or true volumetric models.

OTB Building This dataset is a representation of the OTB building on the TU Delft campus. It is converted from a CityGML LoD 4 building and contains semantics like floor, ceiling, walls and open space. This dataset is scaled up. The scaled up version has a resolution of 1 meter and covers a space of 1550 by 1320 by 470 meter.

Bentley House This dataset is a representation of a small environment including windmills, trees and a house with furniture. The different objects are semantically labelled to distinguish them. This dataset has a resolution of 50 cm and has an axis-aligned bounding box of 84 by 88 by 25 meter.

Other datasets Besides these datasets any other voxelized models can be used for research and development. Two very important characteristic are the resolution and the semantics included in the model.

References

[1] C. Andújar, P. Vázquez, and M. Fairén. Way-finder: Guided tours through complex walkthrough models. *Computer Graphics Forum*, 23(3 SPEC. ISS.):499–508, 2004.

[2] Srikanth Bandi and Daniel Thalmann. Path finding for human motion in virtual environments. *Computational Geometry*, 15(1-3):103–127, 2000.

[3] Adi Botea, M Müller, and Jonathan Schaeffer. Near optimal hierarchical path-finding. *Journal of game development*, pages 1–30, 2004.

[4] Kenny Daniel, Alex Nash, Sven Koenig, and Ariel Felner. Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, 39:533–579, 2010.

[5] E. W. Dijkstra. A Note on Two Probles in Connexion with Graphs. *Numerische Mathematik*, 1(l 959):269–271, 1959.

[6] Rhys Goldstein, Simon Breslav, and Azam Khan. Towards Voxel-Based Algorithms for Building Performance Simulation. *eSim*, (Molloy), 2014.

[7] P.E. Hart, N.J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths, 1968.

[8] M Meijers, S Zlatanova, and N Pfeifer. 3D geoinformation indoors: structuring for evacuation. *Proceedings of Next generation 3D city models*, pages 21–22, 2005.

[9] Anthony Stentz and Anthony Stentz. Optimal and efficient path planning for partially-knownenvironments. *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, 3(1):3310 – 3317, 1994.

[10] Nicolas Vandapel, James Kuffner, and Omead Amidi. Planning 3-D path networks in unstructured environments. *Proceedings - IEEE International Conference on Robotics and Automation*, 2005(April):4624–4629, 2005.

[11] Q. Xiong, Q. Zhu, S. Zlatanova, Z. Du, Y. Zhang, and L. Zeng. Multi-Level Indoor Path Planning Method. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-4/W5(May):19–23, 2015.

[12] Wenjie Yuan and Markus Schneider. Supporting 3D Route Planning in Indoor Space Based on the LEGO Representation. *Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, (November):16–23, 2010.