Multi-level Indoor Path Planning Method

Qing Xiong[a], Qing Zhu[a,b], Sisi Zlatanova[c],

Zhiqiang Du[a], Yeting Zhang[a], Li Yang Zeng[b]

a State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, P.O. Box C310, 129 Luoyu Road, Wuhan, Hubei, 430079, P.R. China
b Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu, Sichuan, P.R. China
c GISt, OTB, Delft University of Technology, Jaffalaan 9, 2628 BX Delft, The Netherlands, Email: s.zlatanova@tudelft.nl

Key words: indoor, path planning, semantic

Abstract: Indoor navigation is increasingly widespread in complex indoor environments, and indoor path planning is the most important part of indoor navigation. Path planning generally refers to finding the most suitable path connecting two locations, while avoiding collision with obstacles. However, it is a fundamental problem, especially for 3D complex building model. A common way to solve the issue in some applications has been approached in a number of relevant literature, which primarily operates on 2D drawings or building layouts, possibly with few attached attributes for obstacles. Although several digital building models in the format of 3D CAD have been used for path planning, they usually contain only geometric information while losing abundant semantic information of building components (e.g. types and attributes of building components and their simple relationships). Therefore, it becomes important to develop a reliable method that can enhance application of path planning by combining both geometric and semantic information of building components. This paper introduces a method that support 3D indoor path planning with semantic information.

1. Introduce

Indoor navigation is very useful in reality as people spend large parts of their lives in indoor spaces such as office buildings, commercial centers, conference venues, transportation facilities, and so on(Battles and Fu, 2014; Liu and Zlatanova, 2011; Mikkel et al., 2014). Furthermore, Indoor path planning is the core of indoor navigation. Many path planning algorithms have been developed in the fields of computer science and robotic in the past decades years. Today some studies have begun to exploit the path planning methods in various applications due to the significant improved in computer development. One direct application is indoor navigation (e.g. for the blind/wheelchair/pedestrian), in which a fast and efficient path planning algorithm often generates the collision-free paths for providing the navigational assistance. Another application is emergency evacuation simulation (e.g. for fire protection design) in the public buildings, where the planed paths provide the possible evacuation alternatives in different risky occurrence processes. In additional, path planning can help the architectural designer for improving automatic rule-based checking of building designs, for example, the shortest route between a public area and fire exits should be less than a minimum value in many fire protection design checking.

Most of the above applications require 2D drawing or building layouts as input, possibly with few attached attributes for obstacles, in path planning. However, they usually contain only geometric information while losing abundant semantic information. Successful path planning for 3D indoor spaces should depend on the accurate and updated geometry and semantics of building components, which include the correct geometrical positioning of indoor spaces, functions and properties of spaces and obstacles.

For solving this issue, this paper introduce an improved method to cope with path planning in 3D indoor space through complex 3D building models with semantic information. The input data conform to the definition of LOD4 in CityGML. Compared to the traditional CAD data, CityGML is capable of restoring both geometric and rich semantic information of building components, as well as their relationships.

This paper aims to study the particular problem of path planning for 3D indoor spaces, all relying on the building models with semantic information input. The application developed in this paper automatically extracts and manages geometric and semantic information on the building space and the inside components from input file, which enhances application of path planning. After extracting geometric and semantic information of building components for input data, the next step is to fine the most suitable path between two given location in the 3D indoor spaces. Various approaches can be used for the problem of path planning computation. In a general 3D indoor space, the problem becomes much harder, and it has been proved that computing a fine path connection two points considering the obstacles is another hard problem.

The traditional method of determining an optimal path to traverse from one point to another are often based on the graph exploration methods, such as the Dijkstra algorithm, Fast Marching Method, A* algorithm, and the ant colony algorithm. The Dijkstra algorithm could always find the global optimization solution, but it will consume a large amount of time to search. The Fast Marching Method is quite related to the Dijkstra algorithm, and it uses the upwind difference operators to approximate the gradient, which makes it capable of computing a large class of continuous problem. Being different from the Dijkstra algorithm, the A* algorithm does not have to carry out a global search, which uses a heuristic function to search directly towards the goal and find the possible shortest route, so its computational time is much shorter than the Dijkstra algorithm. The ant colony algorithm has a better probability in achieving the globally optimal solution, especially with a larger

This contribution has been peer-reviewed.
doi:10.5194/isprsarchives-XL-4-W5-19-2015

number of iterations, but this will lead to cost of a larger computation time. However, the traditional methods do have some limitations in 3D indoor environment. One is the execution efficiency and precision of the algorithm do not conform the real scene very well, especially at the part of corner. Second is they could not deal with multi-floor issues with foreseeing a wonderful effect.

This paper introduces an improved semantic-oriented path planning method based on A* algorithm. Firstly, the valuable geometric and semantic information is extracted from the input data, where the geometric information will traverse from vector data to grid data. Then, the method will find the most suitable path for a start location to goal destination by evaluating the weight factors of each criterion at grid nodes, which considering the vertical circulations like stairs or elevators between different floors. There will be an experiment for testing the method, and the input data includes two 3D building models containing some obstacles.

2. Coarse-to-fine indoor path planning method

This paper's method is to transform the input geometric data, which is triangularization, into voxelization data. The first process should inherit the semantic information from the input data. Then creating the semantic region based on the voxels. Additionally, we get the navigation meshes based on the contours of the regions. Finally, we use the A* algorithm based on the navigation meshes to find the path with input semantic conditions.

2.1 Voxelization process

Refer to classic voxelization methods(Jones and Satherley, 2000; Oomes et al., 1997; A et al., 1993), we use voxels to present the 3D model's surface information for preparing the data of the next steps. As show in Figure 1, the voxels inherit the triangle's semantic information.

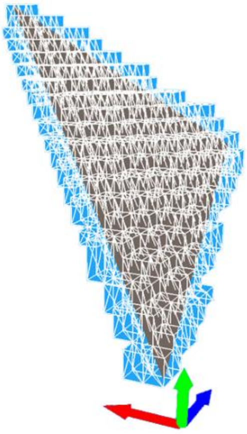

Figure 1 Voxelization process

2.2 Regions generation

After the voxelization process, we now have the voxels set with unrelated relations. This step is to figure out which voxels form potential surfaces of contiguous regions. For this purpose, we introduce the voxel pillar in this paper. We set the voxel pillar contains the same semantic information and the same projected coordinates in x-y plane. As show in Figure 2, for each pillar, its axis-neighbour columns are searched for candidates. A pillar in the neighbour is considered a neighbour pillar if both of the following conditions are met: the up or down between the floors of the two pillars is less than the threshold value. This setting allows for surfaces such as stair steps to be detected as valid neighbours.

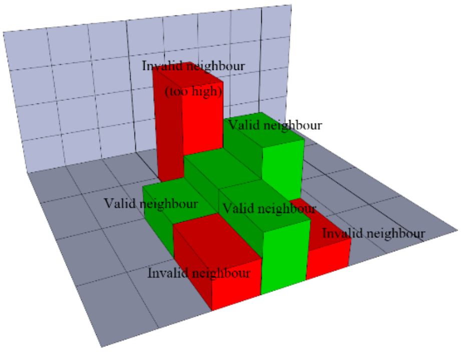

Figure 2 Creating neighbour relations

In this paper, the regions are groups of contiguous pillar that represent walkable surface area with same semantic information. The watershed algorithm is used for initial region creation(Haumont et al., 2003). Using the watershed analogy, the pillars which are furthest from a border represent the lowest points in the watershed. A border pillar represents the highest possible water level. The main loop iterates, starting at the lowest point in the watershed, then increments with each loop until the highest allowed water level is reached. This slowly "floods" the pillars starting at the lowest points. During each iteration of the loop, spans that are below the current water level are located and an attempt is made to either add them to exiting regions or create new regions. During the region expansion phase, if a newly flooded pillar borders an existing region, it is usually added to the region. Any newly flooded span that survives the region expansion phase is used as a seed for new region creation. Regions that are unnecessarily small, which may be caused by the size of input geometric data, are ignored for indoor path planning's presentation.

2.3 Contour generation

For creating contours, we care about the edges of the pillars. There are two types of edges: region and internal. Region edges are pillar edges across which the neighbour is in another region. Internal edges are pillar edges across which the neighbour is in the same region.

The first step is to categorize edges as either region edges or internal edges, as shown in Figure 3. For each pillar, we can check all axis-neighbours. If an axis-neighbour is not in the same region as the current span, the edge is flagged as a region edge.

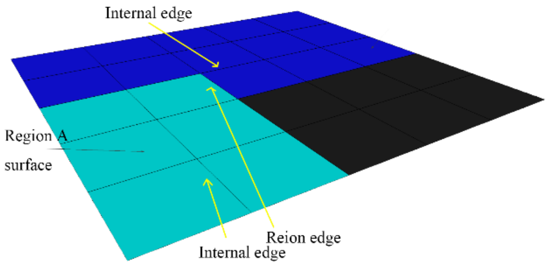

Figure 3 Searching for region edges

Once we have information on which pillar edges are region edges, we can build a contour by walking the edges, as shown in Figure 4. Once again we iterate the pillars. Also, the regions inherit the semantic information of pillars.

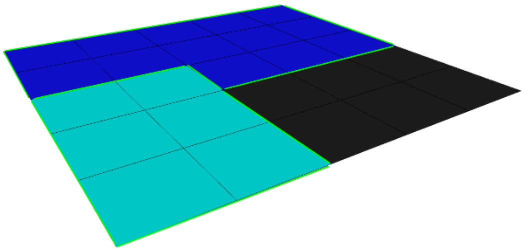

Figure 4 Building the contour of regions

2.4 Navigation meshes generation

The navigation meshed generation is the triangulation process of contours of regions(Oliva and Pelechano, 2013). Triangulation is performed by walking the edges of the contour and, for each group of three vertices, determining whether a valid internal triangles can be formed. From all potential candidates, select the one with the shortest new edge. The process continues with the remaining vertices until triangulation is complete.

Firstly, as shown in Figure 5, finding potential partitions of contour by walking the edges.

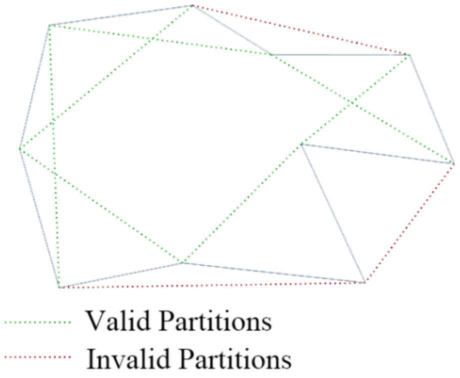

Figure 5 Finding potential partitions

Then selecting the shortest partition and form a triangle. Getting rid of invalidated partitions and generate new potential partitions, as shown in Figure 6.

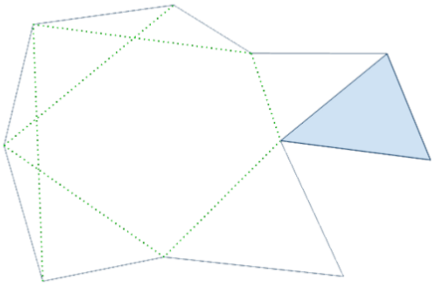

Figure 6 Triangulation of contour

Iterating the partitions, finally, we get the final triangulation, as shown in Figure 7.

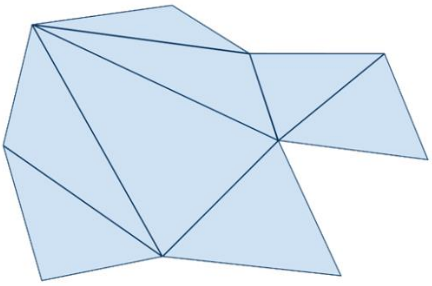

Figure 7 Final triangulation

2.5 Path planning with A* algorithm

After navigation meshes generation, each triangle has its own semantic information. This paper refer to the existing A* algorithm with triangulation (Cui and Shi, 2011; Zhongshan et al., 2011). As shown in , we set a start point A and an end point B, and finally generating a path from A to B.

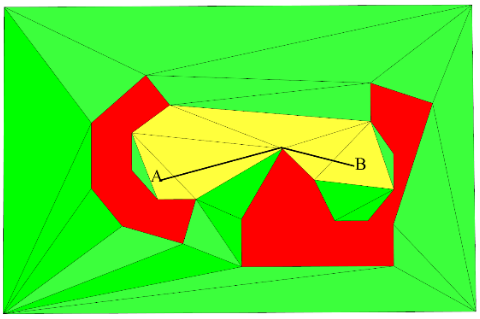

Figure 8 A* algorithm with triangulation

However, the triangulation result contains semantic information, which means any point in the navigation meshes should have a semantic name, or location name. Then, we could find a path from an indoor location to another indoor location.

3. Experiment

According to the indoor path planning method introduced above, this paper using a two floors building for experiment. The input data and voxelization result show in Figure 9.

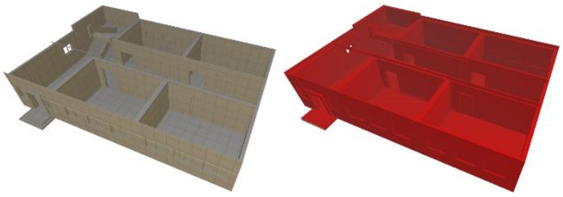

Figure 9 The input data and voxelization result

It contains 2300 vertexes and 3600 triangles as input geometric data. And the voxelization result contains 35752 voxels as the voxel size is 0.3 * 0.3 * 0.3.

The region generation result shows in Figure 10. The reason of top-right region does not have connection relations with other regions is that the room's door is closed (as shown in Figure 9).

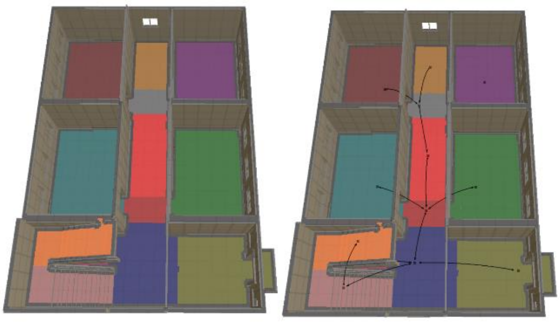

Figure 10 Regions generation result and connection between regions

The contour and navigation meshes generation results show in Figure 11.

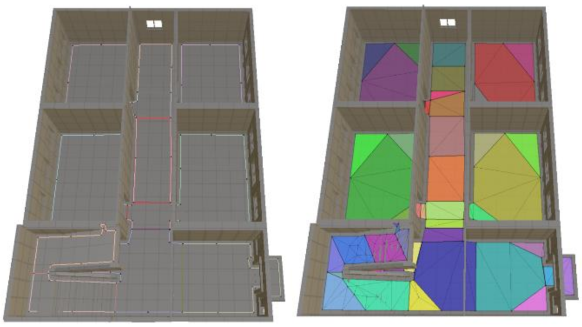

Figure 11 Contour and navigation meshes generation results

Then we set an example the finding an indoor path from "Hall" to "Room 101", the result with A* algorithm shows in Figure 12.

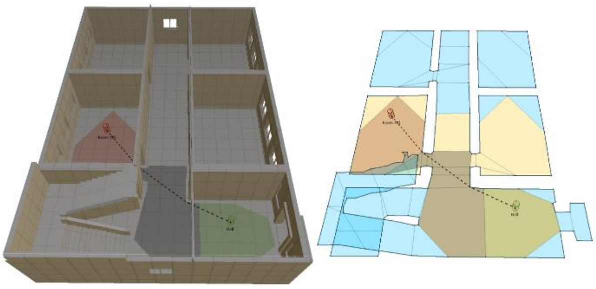

Figure 12 Indoor path planning from "Hall" to "Room 101"

As the same above, shows the path from "Room 102" to "Staircase".

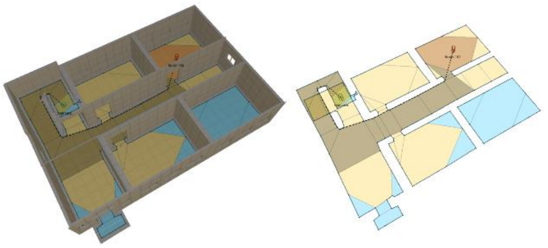

Figure 13 Indoor path planning from "Room 102" to "Staircase"

4. Conclusion

As the experiment shows, this paper's indoor path planning method could generate a fine rout from one semantic location to another semantic location. And this process does not break the integrity of original geometric data, and also inherit the semantic information. This should perform nice in indoor navigation application. However, the finally path did not conform to the human's walking habit in indoor, as some part of the path is too close to the wall or staircase. This defect will be improved in the next research.

5. Acknowledgement

This work is supported by National Science and Technology Support Program of China (No. 2012BAH35B03), and the National Natural Science Foundation of China (No. 41171311).

References:

A, K., D, C. and R, Y., 1993. Volume graphics. IEEE Computer, 51 - 64(26).

Battles, A. and Fu, W., 2014. Navigating Indoor with Maps: Representations and Processes.

Cui, X. and Shi, H., 2011. A*-based pathfinding in modern computer games. International Journal of Computer Science and Network Security, 11(1): 125--130.

Haumont, D., Debeir, O. and Sillion, F., 2003. Volumetric cell-and-portal generation. Blackwell Publishing, Inc, pp. 303--312.

Jones, M.W. and Satherley, R., 2000. Voxelisation: Modelling for volume graphics. Press, pp. 319--326.

Liu, L. and Zlatanova, S., 2011. Towards a 3D network model for indoor navigation. CRC Press, pp. 79-94.

Mikkel, B., de Haas, C., Lu, H., Xie, X. and Pilvinyte, A., 2014. Constructing indoor navigation systems from digital building information. Data Engineering (ICDE), 2014 IEEE 30th International Conference on, pp. 1194-1197.

Oliva, R. and Pelechano, N., 2013. NEOGEN: Near optimal generator of navigation meshes for 3D multi-layered environments. Computers & Graphics, 37(5): 403 - 412.

Oomes, S., Snoeren, P. and Dijkstra, T., 1997. 3D shape representation: Transforming polygons into voxels. In: B. ter Haar Romeny, L. Florack, J. Koenderink and M. Viergever (B. ter Haar Romeny, L. Florack, J. Koenderink and M. Viergever)^(B. ter Haar Romeny, L. Florack, J. Koenderink and M. Viergevers)|,*Scale-Space Theory in Computer Vision. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 349-352.

Zhongshan, X.X.D.O., Zhongshan, P.R.C.K. and China, P.R., 2011. Pathfinding of 3D Game with Navigation

Mesh, Pathfinding of 3D Game with Navigation Mesh. Proceedings of 2011 3rd IEEE International Conference on Information Management and Engineering(ICIME 2011) VOL.04, Zhengzhou,China, pp. 5.