Automatic extraction of an IndoorGML navigation graph from an indoor point cloud

P2 Research Proposal for graduation thesis MSc Geomatics

Puck Flikweert

June 6, 2018

Contents

1 Introduction

Since the development of hand-held 3D laser scanners, it has become easier to generate point clouds of indoor environments. Various authors acknowledge the need for indoor point clouds to assess the way a building was built, versus the way a building was planned (Bosché et al., 2014; Alattas et al., 2017; Hong et al., 2015; Volk et al., 2014; Staats et al., 2017). This is because recent indoor point clouds contain up-to-date information on how the interior of a building looks like, as opposed to a floor plan or 3D model that was created during the planning phase. Point clouds do not only include information about placement of structural elements, such as walls, floors, ceilings and stairs, but also about the placement of obstacles. Moreover, this means that they contain information on empty or free space in an indoor environment (Broersen et al., 2015). However, a point cloud can easily contain millions of points, taking up gygabytes of RAM (Schnabel et al., 2007). Although a human can detect various elements in a point cloud by looking at it, a computer does not read more than millions of coordinates; hence the need for abstractions and semantic enrichments in the form of point cloud processing algorithms.

Indoor point cloud data can not only be utilised to define structural elements of the building and create 3D geometric models, but also for purposes such as navigation (Brown et al., 2013) and emergency response (Boguslawski et al., 2016). Besides knowledge about structural elements in a building, these purposes require a sense of space and connectivity relationships in a building. Kwan and Lee (2005) and Rueppel and Stuebbe (2008) draw attention to the need of a 3D network model for indoor navigation, the latter stressing the importance of routing networks for complex buildings in disaster scenarios. A more parametric navigation network can be useful for users that are restricted in their mobility, for example for blind or wheelchair people (Mirza et al., 2012). There already exist methods that can create a navigation network from voxelised point clouds, having classified voxels that are 'walkable' (floor, stairs, ramps) (Staats et al., 2017), or an octree structure in the free space of a point cloud(Fichtner, 2016), both defining connectivity between neighbouring voxels. However, the networks generated are very dense. The octree method even provides voxels in the height of a room, which could be useful for drone routing, but is unnecessary for pedestrian navigation. Pathfinding algorithms such as breadth-first and depth-first search run in an order of time $O(|V| + |E|)$, where $|V|$ corresponds to number of vertices, and $|E|$ to number of edges (Cormen et al., 2009) in a navigation graph. This proves that a less dense navigation network would lead to faster pathfinding; a useful trait in for instance emergency response.

For this reason, this research will be based on a more structured navigation graph. In 2014 the Open Geospatial Consortium (OGC) published IndoorGML, a standard providing a framework for indoor navigation networks (OGC, 2014). IndoorGML is an application schema of the Geography Markup Language (GML), focussing on semantics, geometry and topology of indoor environments. However, it does not extensively cover geometric and semantic properties, to prevent too much overlap with other indoor building modelling standards such as CityGML and Building Information Modeling (BIM). Networks in IndoorGML are based on Poincaré duality (Munkres, 1984). Distinction is made between primal space (2D or 3D cells) and dual space (relationships). The primal space defines the indoor units in an indoor environment, such as rooms and corridors, also called 'cells'. The dual space defines a node in every cell, and edges between nodes that have a relationship. Relationships in IndoorGML can be described as adjacency, connectivity or accessibility, which are saved as a Node-Relation Graph (NRG). These relationships are called the Structured Space Model (SSM), visualised in figure 1. In the NRG the nodes can be called *States*, and edges *Transitions*.

IndoorGML provides support for the decomposition of large indoor spaces into subspaces (called *subspacing* or *subdivision*), which is very useful for the definition of a more complete

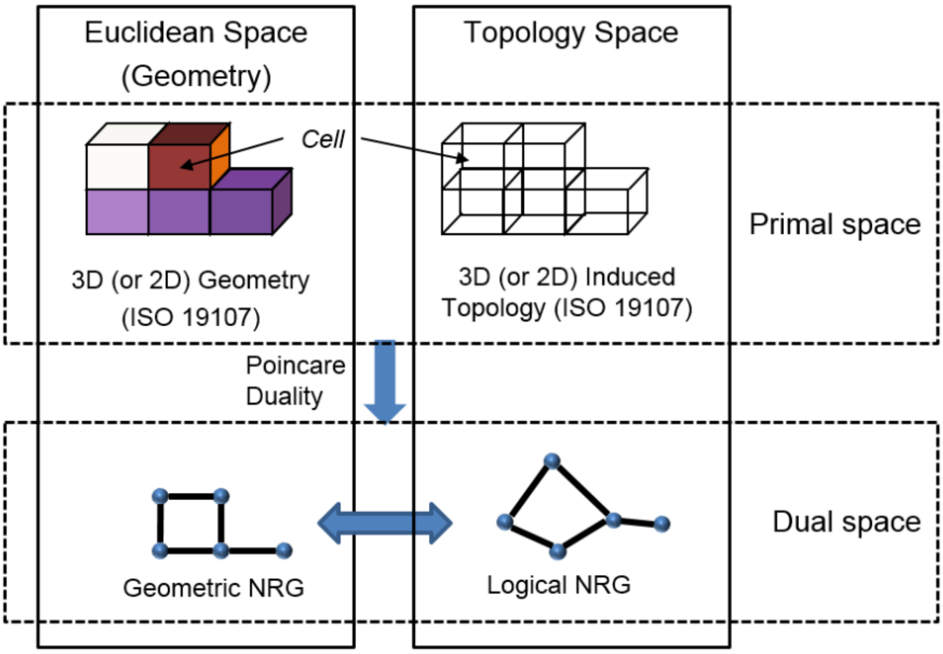


Figure 1: *Structured Space Model as defined in IndoorGML (OGC, 2014)*

navigation network. When subdividing indoor spaces into smaller cells, a better assumption for indoor distances can be made, allowing for better route calculation and visualisation.

In conclusion, up-to-date point clouds are a valuable basis for indoor navigation networks, provided that geometric information and connectivity relationships can be extracted from them. Moreover point clouds can be generated from every possible indoor environment, making them usable in many situations. Connectivity networks could not only assist in indoor navigation, but also in emergency response. By constructing the network as an IndoorGML model, it will be less dense than a network of a voxelised model, and support is given for a fast calculation of routes through the indoor environment. However, methods that automatically construct an IndoorGML model from an indoor point cloud are still lacking, which provides an interesting opportunity for more research.

2 Related work

2.1 Indoor navigation networks

Duality is a reoccurring theme in indoor navigation networks. Yang and Worboys (2015) describe how to obtain a navigation graph from existing 2D building plans using dual graph to model relationships, just as Boguslawski et al. (2011) do for 3D buildings. Jamali et al. (2015) present a way to obtain a navigation network of an indoor environment by using a laser rangefinder, where dual nodes are placed at the locations of the rangefinder. Of every room all corners are measured as x,y and z coordinates. The connection between the dual nodes is obtained by Delaunay triangulation.

A more recent development introduces the use of indoor point clouds as an input for a navigation network. Díaz-Vilariño et al. (2016) use region growing for the segmentation of a point cloud, and later classification of the segments. In this method obstacles are taken into account in the network. A Variable Density Network (VDN) is created, of which the dual is used as a navigation graph. However, the density of this graph is too high for the purposes of this thesis. VDN, based on the Voronoi Diagram, was first introduced by Boguslawski et al.

(2016), who use it to generate a navigation graph from a 3D model of a building without obstacles. For this method corners of cells should be detected, as an input for the graph.

Another method that has been discussed is cuboid reconstruction and merging in a point cloud (Tran et al., 2017), which creates a more general graph, having a node in every cell and connectivity and adjacency relationships similar to IndoorGML. Limitations of this method are that it is based on a Manhattan world assumption, and connectivity relationships are defined by manual insertion of doors.

A recent development in the generation of navigation graphs from indoor point clouds using hand-held (mobile) laser scanners is making use of the scanner trajectory. This is for instance applied in Staats et al. (2017), where both point cloud and trajectory are voxelised, after which the trajectory voxels are projected down to identify floor voxels. Trajectory information can also be used in door detection (Díaz-Vilariño et al., 2017), which is extended on in section 2.3. By segmenting the trajectory at the locations of doors it can be identified which points belong to a certain indoor space. From this a more general navigation network can be generated, as described in IndoorGML.

2.2 IndoorGML

IndoorGML is based on the notion that an indoor environment can be divided into organisational or structural cells, which represent for instance a room or corridor. These cells are non-overlapping and each has a unique identifier. Relationships between cells can be represented by the Node-Relation Graph (NRG), of which there are three types: Adjacency NRG, Connectivity NRG and Accessibility NRG (figure 2). In the latter more user specific information can be saved, such as the width of a door, or whether a certain edge is traversable by a wheelchair user. The graph (V, E) consists of nodes (V), representing the cells (*States*), and edges (E) indicating the relationships (*Transitions*). The NRG can be logical; when nodes and edges do not refer to a position in space, or geometric; when they contain geometry information.

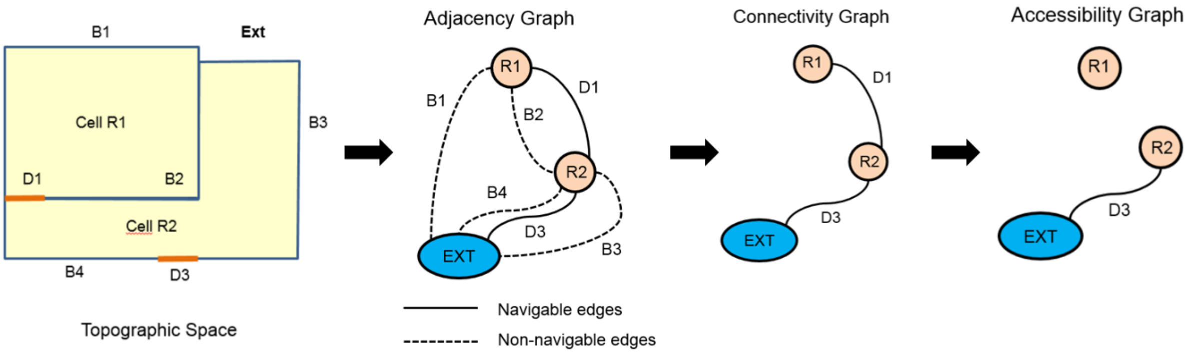


Figure 2: *The different Node-Relation Graphs (NRG) in IndoorGML (OGC, 2014)*

One important thing to note is that in IndoorGML the inclusion of geometry in the model is not compulsory. A valid IndoorGML model can be a logical NRG, without any geometric information. However it is possible to save basic geometry, both in the geometric NRG and in the primal space as 3D or 2D geometric information about the cell. This idea is represented in the SSM, as shown earlier in figure 1.

In practice IndoorGML has already been implemented by various authors, including a method to go from CityGML to IndoorGML (Kim et al., 2014), and from BIM to IndoorGML (Teo and Yu, 2017). A general study on the implementation of IndoorGML is described by Kang and Li (2017). In addition to this, criteria for when to partition a cell into subspaces are discussed, as are the concepts of horizontal and vertical distance and *State* and *Transition* semantics.

2.3 Door detection in point clouds

One of the most crucial steps needed to create a connectivity graph from an indoor point cloud is the detection of traversable doors or openings. When this information is known, connectivity relationships can be defined between different indoor spaces. Díaz-Vilariño et al. (2014) establish door detection in a point cloud by using Generalized Hough Transform (GHT) (Ballard, 1981) to detect edges in orthoimages generated from a point cloud. By assuming general parameters for these edges, doors can be classified. Edge detection is also implemented by Díaz-Vilariño et al. (2016). Instead of using colour information, wall planes are rotated in such a way that they can be read as a binary image. Pixels are assigned a value based on whether they contain points or not. These edge detection methods highly depend on the parameters that are defined for the size of doors. Doors that have non-standard sizes are ruled out for detection.

Besides edge detection, there are methods that use the trajectory of the mobile laser scanner. One is described by Díaz-Vilariño et al. (2017), in which the height of the point cloud along the trajectory is analysed. When seeing a decrease in height, a door is probable to be at that location. This method only works in buildings where door frames are clearly lower than the ceiling. In this research the point cloud is labelled into different regions, each containing all points that were scanned between two doors in the trajectory. A ray-casting method is carried out to evaluate the completeness of these regions. A ray-casting algorithm is also used by Nikoohemat et al. (2017) to detect openings in a voxelised point cloud. In this method distinction can be made between real openings and false ones caused by occlusions. Searching for voxels nearby trajectory points that represent door centres, doors can be detected.

2.4 Division of subspaces

To provide for a more accurate measure of distance in a navigation network, and thus better route calculation and visualisation, some cells should be subdivided into smaller subspaces. There are two main questions that come forth from literature: which cells should be subdivided, and how should the subdivision be done?

Jung and Lee (2015) define a set of rules in a flowchart as to when an indoor space should be subdivided. The space should be both navigable and accessible, and moreover a transition space, connecting at least two spaces in a navigation graph. In an indoor environment these kinds of spaces are usually squares or hallways. Kang and Li (2017) also discuss about when to apply subspacing, and mention size of the space and the presence of obstacles.

Some authors define semantic criteria on how to subspace a indoor cell, such as subdivisions based on "functional area" or "functional space" around objects (Krūminaitė and Zlatanova, 2014; Diakité and Zlatanova, 2017). However, from a point cloud it is difficult to get semantic information on the use of cells and objects they contain. Information like functional space around objects would be difficult to automatically extract. This means that the subdivision should be based on geometric elements. There is a distinction in literature between the subdivision of an area to provide more nodes, and the addition of extra nodes in a navigation network at convenient locations.

Methods based on the subdivision of areas geometrically include ones based on Voronoi diagrams (Wallgrün, 2005; Boguslawski et al., 2016), triangulation (Lamarche and Donikian, 2004), or the complete subdivision of a floor plan to a grid (Afyouni et al., 2012). However the subdivision of space in a grid is very coarse and will slow down calculation times for path-finding algorithms. Diakité et al. (2017) discuss criteria for how to subdivide indoor environments for IndoorGML. Distinction is made between types of criteria, the first one being geometry-driven criteria, such as the shape of the cell. Topology-semantic-driven criteria take into account the placement of doors, and subdivide a corridor such that each subspace contains a door. The last one are navigation-driven criteria, which are based on the walkable surface in the cell. The visualisation of *Transitions* in a navigation graph is also discussed. Instead of a straight line, a *Transition* can be visualised as a gml:LinearString, which means that intermediate points on the line can be defined. This means that in some cells it is not necessary to add extra nodes, but just to edit the geometry of the Transition.

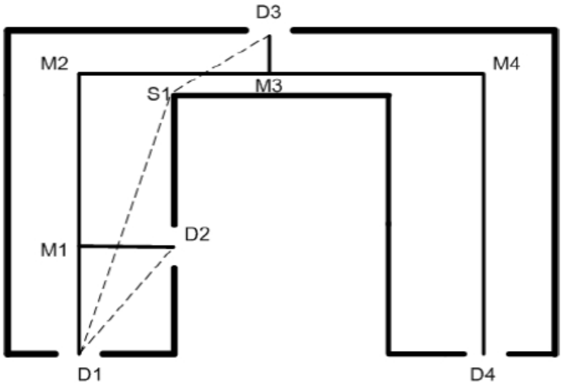


Figure 3: *Door-to-door (D1-S1-D3) vs. S-MAT (D1-M1-M2-M3-D3) route (Liu and Zlatanova, 2011)*

A method that focuses more on the addition of nodes in a navigation network is the Straight-Medial Axis Transformation (S-MAT) (Eppstein and Erickson, 1999). It abstracts a skeleton of a polygon, that represents a route in the middle of the cell. A method based on this is the "door-to-door" approach (Liu and Zlatanova, 2011). Instead of creating a route in the middle of a polygon, routes are created between doors, and corners of concave polygons. Both methods are shown in figure 3. The "door-to-door" approach argues that routes can be calculated and visualised in a more natural way, especially in situations like D1-D2 ("door-to-door") versus D1-M1-D2 (S-MAT) in figure 3.

3 Research objectives

3.1 Research questions

Based on the problem statement in the introduction and the related work described in section 2, the main research question is defined as:

How can an IndoorGML connectivity graph between indoor spaces automatically be extracted from a cluttered point cloud, and to what extent can it be utilised for navigation?

In this context a cluttered point cloud means that there are obstacles present.

The end goal of this graduation thesis will be an IndoorGML model that was automatically generated from an indoor point cloud. The following sub-questions to make the aims more specific:

1. How can the connectivity graph of a building be extracted from an indoor point cloud?
2. To what extent can geometry information be extracted from the point cloud, and added to the IndoorGML primal and dual space?

3. In what circumstances should indoor spaces be subdivided to allow for a more complete navigation network?
4. How can complex indoor spaces automatically be subdivided in a way that keeps the number of nodes in the navigation graph to a minimum?
5. To what extent can the connectivity graph be extended to an accessibility graph based on features in the point cloud?

3.2 Research scope

The focus of this thesis will be on the generation of a connectivity graph for IndoorGML, with extension to an accessibility graph, based on geometric properties in the point cloud. The other type of graph, Adjacency NRG, will not be considered in this research, because it is the least relevant for navigation purposes. Moreover 3D geometry modelling of indoor space will only be done as far as deemed necessary for the IndoorGML model. Multiple floors of buildings will be considered in the method, but as the use case of the topic concerns pedestrian navigation, only navigable floor space will be regarded. For the subdivision of certain cells in the graph, only geometric or topologic characteristics will be taken into account, as semantics are difficult to extract from an indoor point cloud.

Additionally this research will be an exploration of whether the methodology is effective, and will not consider the search for the fastest method to accomplish the results. It will however focus on being as generic as possible, which excludes for instance Manhattan world solutions. Finally the application of the IndoorGML model in a navigation system will not be researched, as it is out of scope of the problem statement.

4 Methodology and preliminary results

In order to answer the defined research questions accordingly, the steps in figure 4 will be followed.

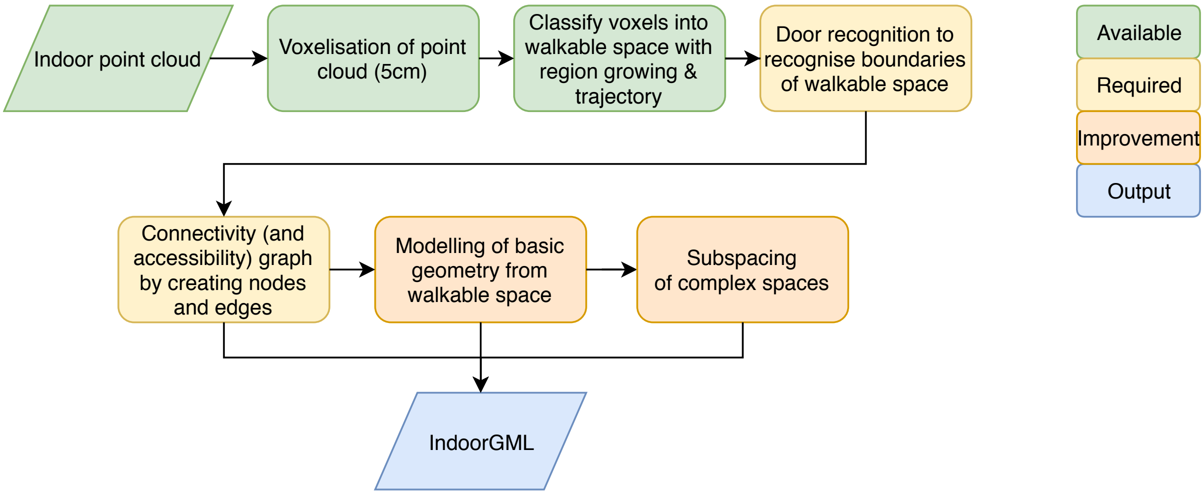


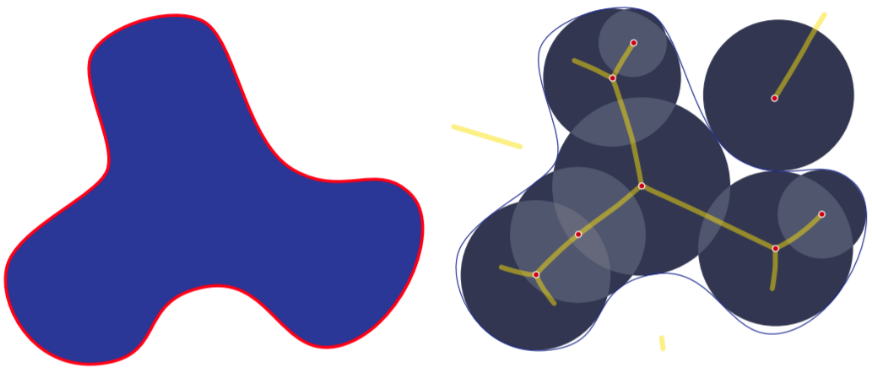
Figure 4: *Flowchart representing methodology to go from point cloud to IndoorGML model (own work)*

The starting point of this research will be a voxelised point cloud of which the walkable space (floors, stairs and ramps) is known, based on the work of Staats et al. (2017). In the methodology flowchart, as shown in figure 4, these steps are summarised as the green boxes. In order to get an IndoorGML model from walkable voxels, more information should be extracted from the point cloud. First of all separate spaces should be identified from the walkable space by clipping them at the locations of doors. Then connectivity relationships should be defined. These steps are absolutely crucial in the creation of an IndoorGML model. Then to answer further research questions, the two steps shown in orange should be implemented. Geometry of the nodes of a navigation graph can be extracted from the point cloud, and it will be researched whether an approximation of 2D or 3D geometry of cells can also be defined. Then, lastly, complex cells in the navigation network will be subdivided.

4.1 Door detection

4.1.1 3D Medial Axis Transform for door detection

A novel method for door detection will be introduced in this research. It is making use of the 3D Medial Axis Transform (MAT) as developed by Peters (2018), which is based on the shrinking ball algorithm as introduced by Ma et al. (2011). This method gives the ”skeleton” of a point cloud, as shown schematically in a 2D way in figure 5.



(a) A object or shape $\mathcal{O}$.

(b) The corresponding medial axis and some of its medial balls.

Figure 5: *2D schematic visualisation of MAT representing skeleton of an object (Peters, 2018)*

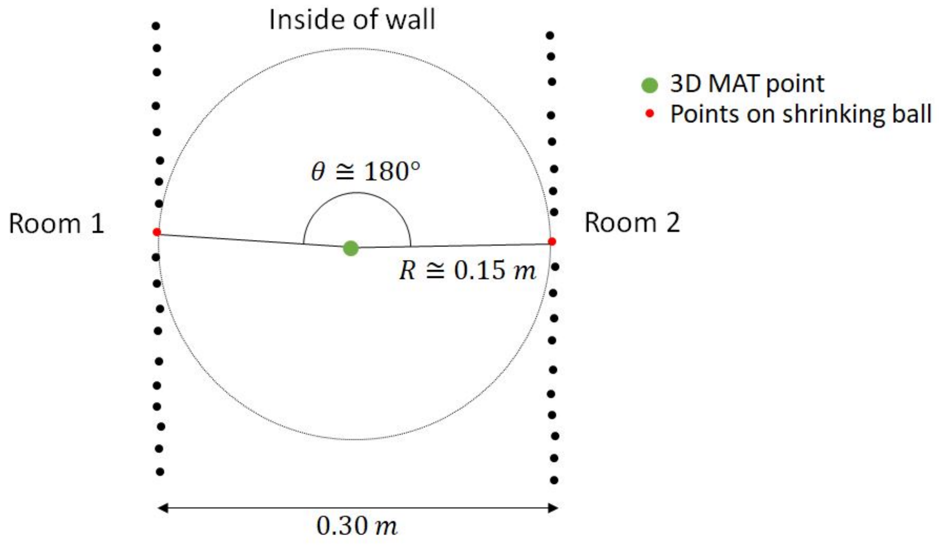


Figure 6: *Side view of 3D MAT in an indoor wall (own work)*

In a 3D situation, the medial axis will be represented by sheets of points, instead of 2D lines.

When the method is applied to a cluttered indoor point cloud, the first results are very noisy. However, interesting results occur inside walls that were scanned from both sides. Perfectly straight sheets are formed in the middle of these walls, that could be used for door detection. Although the rest of the results are noisy, medial balls have useful characteristics on which filtering can be applied. As can be seen in figure 6 the medial balls inside a wall have a large separation angle θ between the two connected points. Moreover the radius, R, is generally quite small; half the width of an indoor wall. Another advantage of this method is that inside walls no noise will occur, because the laser scanner has not been there.

When this filtering is applied to 3D MAT sheets generated from a small part of a two-storey indoor point cloud, the result is as shown in 7. Not only sheets of points generated inside walls are kept, but also those between the ceiling and floor.

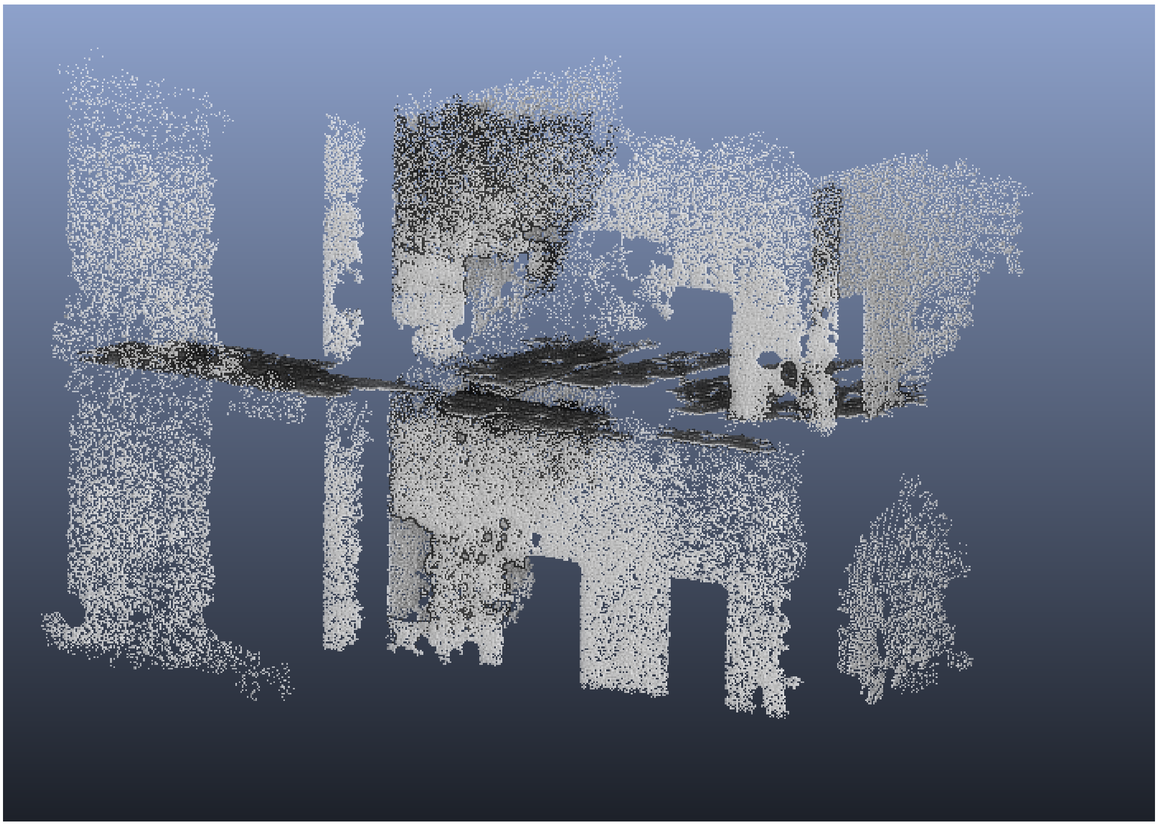


Figure 7: *Result of filtering 3D MAT sheets based on θ and R (own work)*

It can be argued that these sheets are incomplete, because they are only generated in walls that were scanned from both sides. However, these are the only relevant walls in which doors for a navigation network need to be detected. Doors that lead to a place that was not scanned do not provide useful information for such a network. There are multiple ways in which doors could be detected in these sheets. One option is to apply an edge detection algorithm to the sheets, and search for shapes that could represent a door. This method would be a good basis for not only detecting open doors, but also closed ones, as can be seen in figure 8. This phenomenon occurs because doors are usually much thinner than walls and, when scanned from both sides, points will lie too close together to create a 3D MAT sheet through them. However, it can not be said with 100% certainty that this closed door is really traversable. The closed door in the original point cloud in figure 8 does not seem have a door handle, so it could be a false door.

To prevent false doors from occurring in the navigation graph, the chosen method will only regard doors that were walked through. This will ensure that false openings such as glass

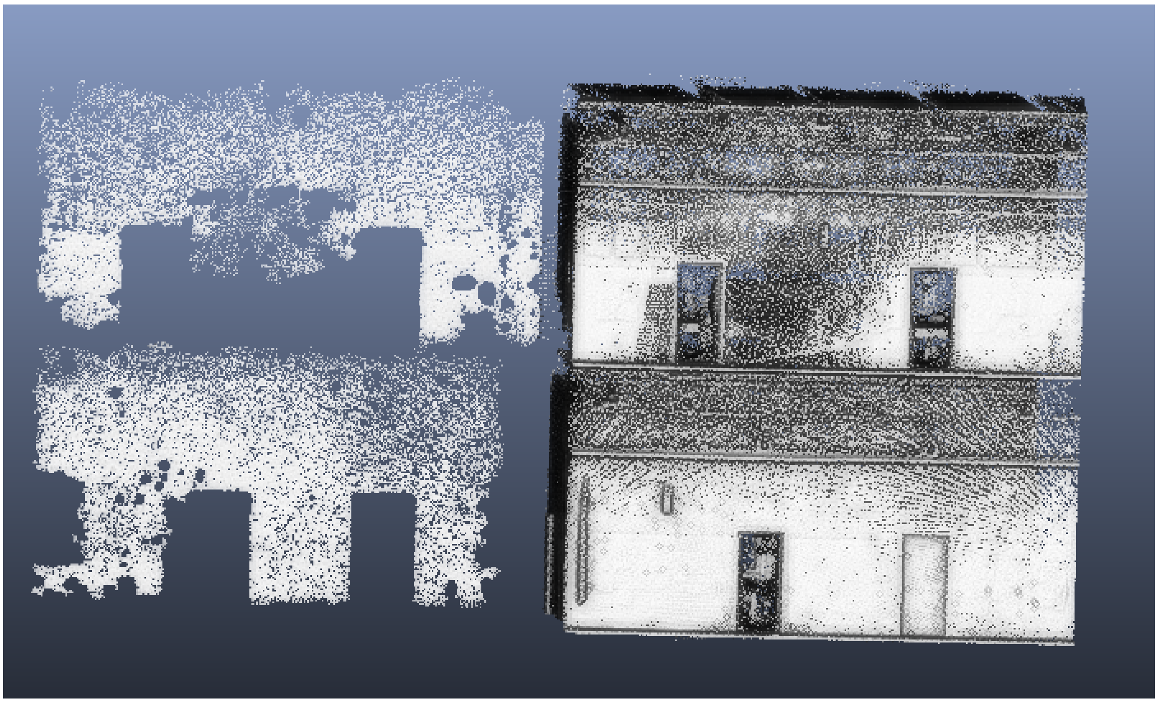


Figure 8: *One 3D MAT sheet spanning 2 floors (left) compared to original point cloud (right). A closed door will lead to a hole. (own work)*

panes will not be regarded as doors. On the other hand, this method will also leave out doors that are truly traversable, but not walked through during the scanning period. This is a limitation that has to be taken into account in the scanning process. In order to ensure that a correct navigation network is constructed from the point cloud, every traversable door should be walked through.

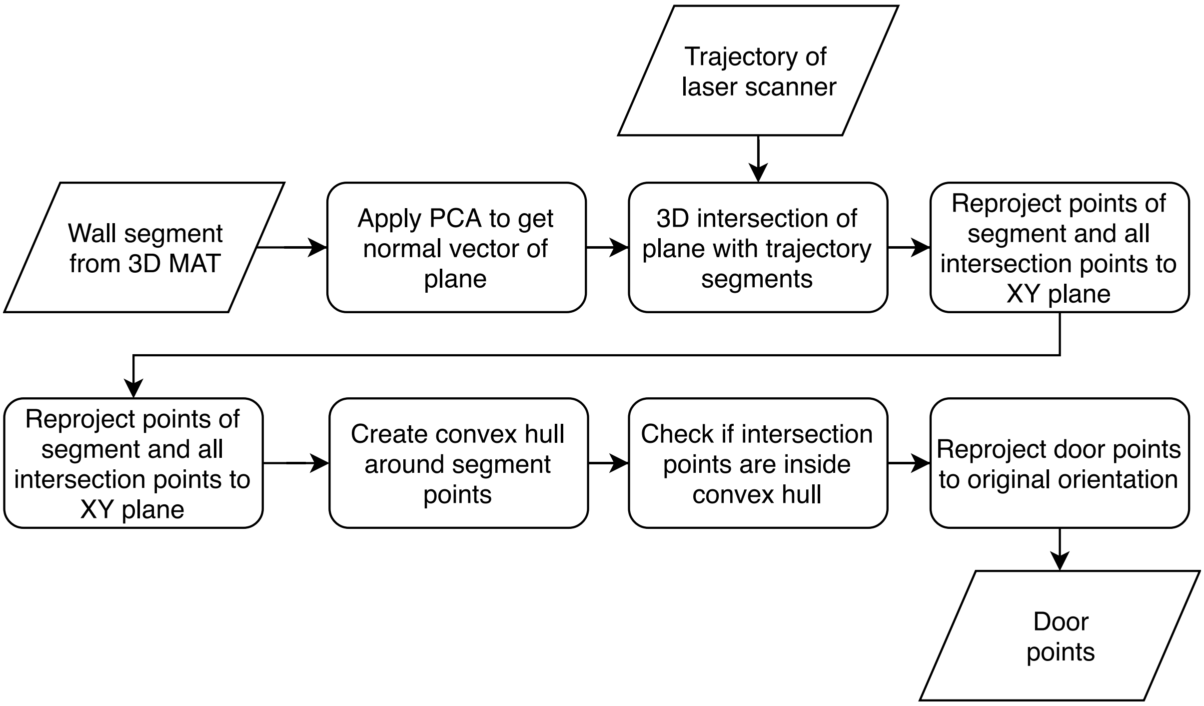


Figure 9: *Flowchart of door detection in point cloud using 3D MAT sheets and trajectory (own work)*

4.1.2 Door detection from 3D MAT sheets and trajectory

The flowchart in figure 9 represents the steps taken to get locations of doors in a point cloud from 3D MAT sheets. At first Principal Component Analysis (PCA) (Jolliffe, 1986) is applied to all points in one 3D MAT sheet. The covariance matrix of these points is computed, from which eigenvectors and eigenvalues are calculated. The eigenvector belonging to the smallest eigenvalue represents the normal vector of the plane that the points lie in. The other two vectors are orthogonal to each other, both lying on the plane.

As the points in the trajectory are saved in order of time, a line segment is generated between each point (i) and the next ($i+1$). The line equation of these line segments is intersected with the plane equation, after which it is calculated whether the intersection point lies on the line segment. The intersection points that apply, are kept for the next check.

A rotation matrix is applied on the 3D MAT sheet points and intersection points, to project them parallel to the XY plane. The normal vector to the original plane is $\vec{v} = (a, b, c)^T$, and the normal vector to the XY plane is $\vec{w} = (0, 0, 1)^T$.

The rotation matrix (R) is defined based on the work of Cole (2015).

$$R = \begin{bmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z \sin\theta & u_x u_z(1-\cos\theta) + u_y \sin\theta \\ u_y u_x(1-\cos\theta) + u_z \sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x \sin\theta \\ u_z u_x(1-\cos\theta) - u_y \sin\theta & u_z u_y(1-\cos\theta) + u_x \sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{bmatrix}$$

In which $\vec{u}$ = (u_x, u_y, u_z) is the unit vector, where $u_x^2 + u_y^2 + u_z^2 = 1$, and θ the rotation angle about an axis in the direction of $\vec{u}$. $\vec{u}$ is defined as:

$$\vec{u} = \frac{\vec{v} \times \vec{w}}{|\vec{v}|} = \frac{(b, -a, 0)^T}{\sqrt{a^2 + b^2 + c^2}}$$

And θ is defined as:

$$\theta = \frac{(\vec{v}, \vec{w})}{|\vec{v}|} = \frac{c}{\sqrt{a^2 + b^2 + c^2}}$$

In this case u_z is always 0, so the rotation matrix can be rewritten as:

$$R = \begin{bmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) & u_y \sin\theta \\ u_y u_x(1-\cos\theta) & \cos\theta + u_y^2(1-\cos\theta) & -u_x \sin\theta \\ -u_y \sin\theta & u_x \sin\theta & \cos\theta \end{bmatrix}$$

After applying the rotation matrix, the Z component of the transformed points is temporarily removed, after which a 2D convex hull is created around the sheet points. For each intersection point it is then checked whether they are inside this convex hull, meaning that the trajectory crosses the wall segment there. This is done by reconstructing the convex hull for each intersection point with the sheet points. If the convex hull stays the same, it means that the point is inside. If the convex hull changes, it means that the point is outside.

The points that are proven to be inside are rotated back to the original orientation of the plane, showing the location of the doors. For this the inverse of the rotation matrix is used. The results of this method are shown in figure 10, together with the 3D MAT sheet points and the convex hull around them. The opening on the lower right, which was a closed door in the original point cloud, is not recognised as a door, while the other three doors are.

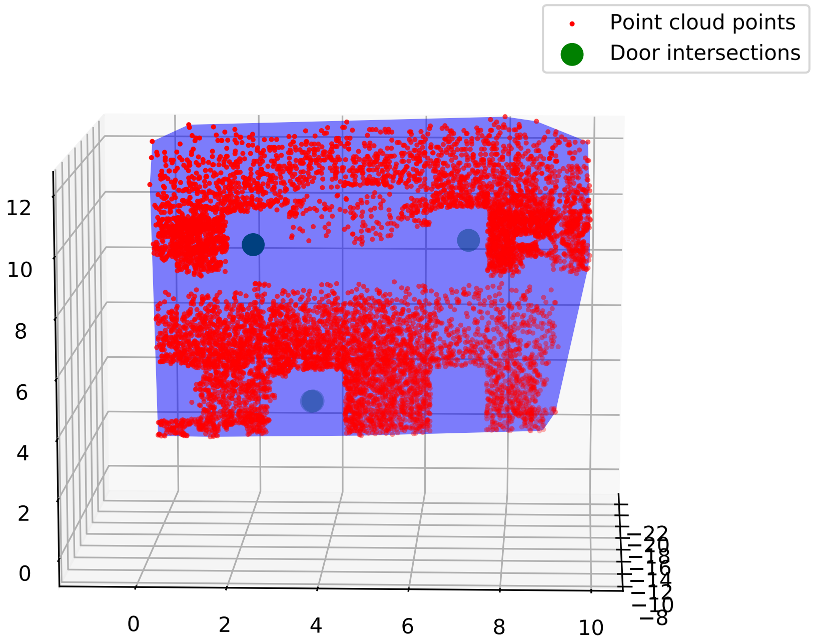


Figure 10: *Intersection points representing doors that were crossed by trajectory (own work)*

4.2 Geometry for IndoorGML

After doors are identified in the point cloud, the walkable space can be separated into different parts that represent cells such as rooms or corridors. One of the goals of this research is to include basic geometry in the final model. From the outline of voxels belonging to a cell a 2D floor outline can be defined as a polygon. This will be added to the model as geometry belonging to the primal space. For the dual space a position should be defined for each of the nodes in the navigation graph. The position of doors are known from the door recognition method, and the position of nodes in cells can be extracted from the floor outline. Taking the centroid of this polygon is not always the best method, because it can lie outside of a concave polygon, so a test should be done to verify this. When the centroid is outside the floor outline, a point can be defined inside the polygon by triangulating it first. Because triangles are always convex, any centroid of any of the triangles will be inside the floor outline, thus could define the coordinate of the node.

4.3 Subdivision of complex cells

As discussed in the introduction and related literature, subspacing in IndoorGML can be used to create a more complete navigation network, that provides for better calculation and visualisation of routes. Jung and Lee (2015) argue that spaces (nodes) that have only one connection (edge) do not need to be subdivided. A filtering will be applied such that these spaces are not taken into account for subdivision.

As for the other spaces the network, an analysis should be done on the floor outline of the space, and the placement of doors. An illustration is given in figure 11 of two situations where subspacing would be desirable. The route (dotted line) is crossing the wall in the left situation, whereas it is crossing a hole in the right situation. These floor plans both represent concave polygons, which should be filtered on. Another situation in which subspacing would be required is for long hallways with multiple doors. These cases can be detected by analysing the number of connections that a node makes, in combination with the floor outline.

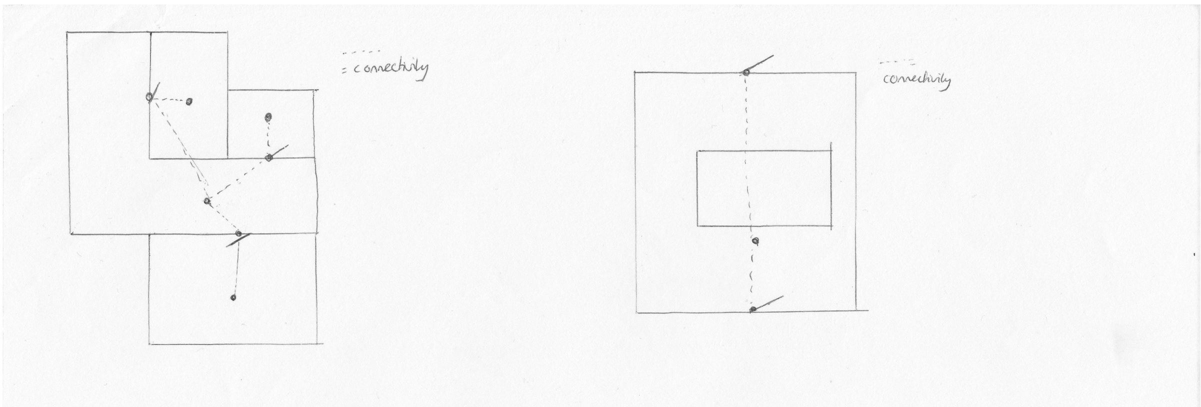


Figure 11: *Two situations where subspacing would be desirable (own work)*

In section 2.4 various ways of subdividing indoor spaces and routes are discussed. However, the results of extracting the navigable space from a point cloud, as done by Staats et al. (2017), are not as straightforward as a 2D floor plan would be (see figure 12). To approximate a floor plan as a polygon from the navigable space, a concave hull could be drawn around the centre points of the voxels. Concave hulls are alpha shapes (Edelsbrunner et al., 1983), and are very dependent on the setting of parameter $alpha$, which determines how connected the output polygon is. The smaller the value, the less connected the polygon. Testing should be done to determine the best value.

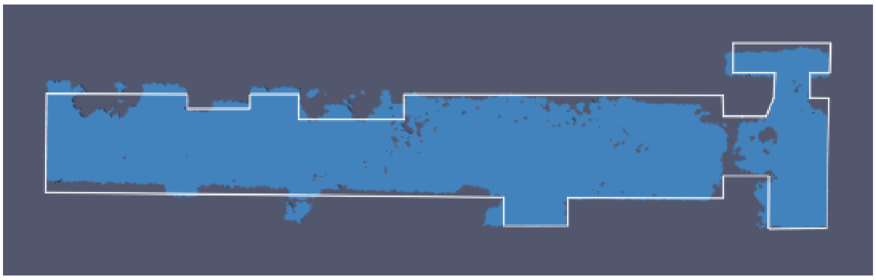


Figure 12: *Comparing the navigable space generated from a point cloud to a floor plan (Staats et al., 2017)*

After the concave hull is obtained, a subdivision method should be applied to extend the navigation network. A type of S-MAT will be applied to get a ”skeleton” of the polygon. After this a connection from every door node will be made to this skeletal line. For a polygon that has many vertices, the S-MAT will be noisy. In order to prevent this, the MAT should be filtered. For this the Scale Axis Transform (SAT) will be applied (Giesen et al., 2009), see figure 13. The parameters of this method should be researched, in order to see what gives the best results. After applying the SAT, connecting orthogonal edges are added from door nodes to the SAT. Extra nodes are added to the IndoorGML model at the locations of these intersections.

4.4 Accessibility information

Finally geometric accessibility information can be extracted from an indoor point cloud. Amongst others both the occurrence of stairs and the dimensions of door frames will be taken into account. This can be done by analysing geometry around nodes. These will be saved as extra attributes to *States* and *Transitions* in the IndoorGML model.

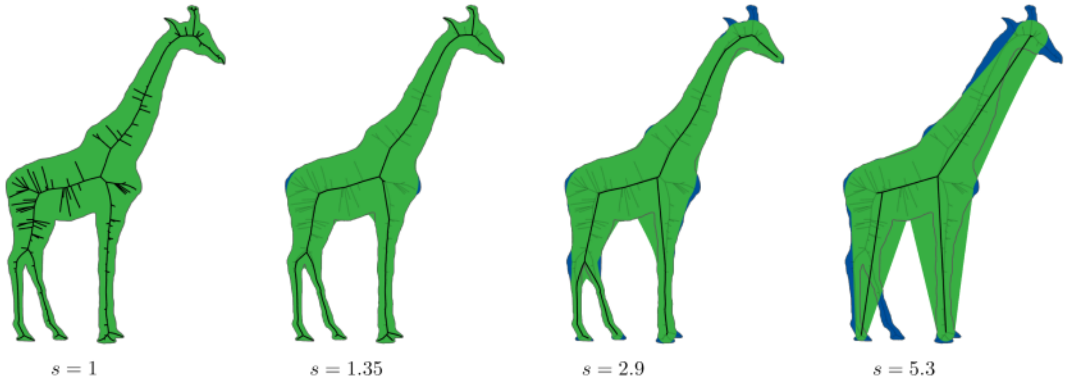


Figure 13: *Example of SAT applied to a polygon (Giesen et al., 2009)*

5 Time planning

5.1 Planning

A Gantt chart is shown in figure 14 with the planning from P2 until P5. The current date of P2 is shown with the orange vertical line.

5.2 Meetings

Every two weeks a meeting with the first supervisor R. Peters is planned, and Skype meetings with the second supervisor L. Díaz-Vilariño are planned when necessary. Meetings with internship supervisor R. Voûte are planned every two weeks, with a possibility for more meetings when necessary.

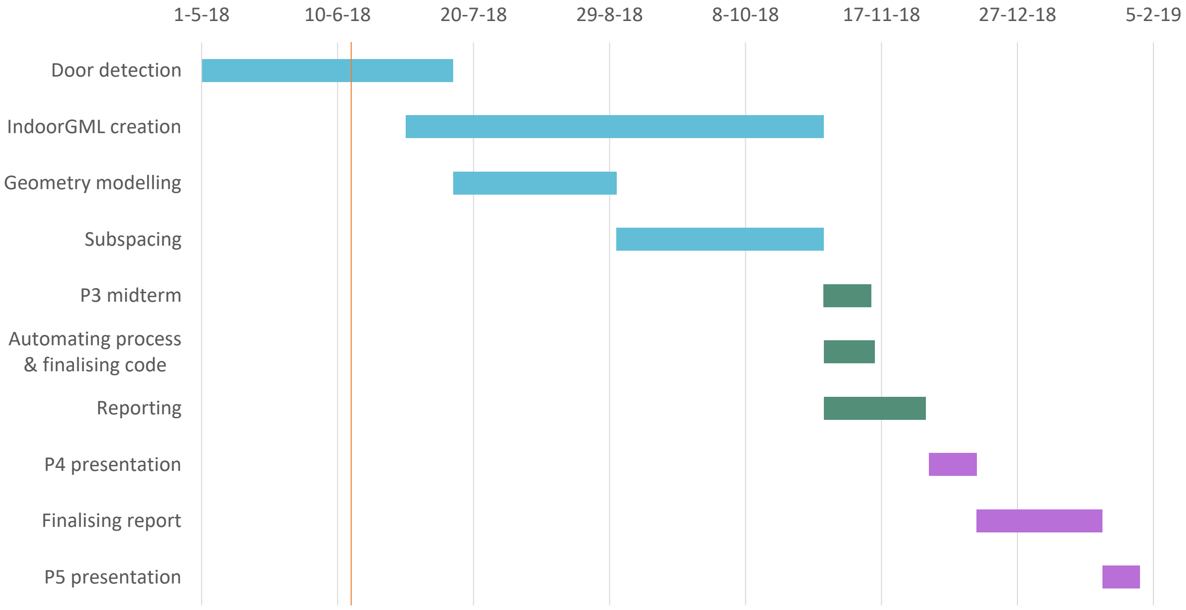


Figure 14: *Time planning of thesis (own work)*

6 Tools and datasets used

6.1 Tools

The processing of the point cloud and the testing of the methods will be done using Python, and various libraries, such as numpy, scipy, cgal, sympy, matplotlib, skel3d (Peters, 2018), and others. Also functions of PostGIS will be looked into. The final IndoorGML model will be checked on validity with the GML 2.1.2 Validation Tool of OGC (`http://cite.opengeospatial.org/test_engine/gml/2.1.2`). CloudCompare will be used to visualise point clouds, and Paraview to visualise the voxelised point cloud.

6.2 Data

For testing the methods an indoor point cloud (and trajectory) of the east wing of the Architecture faculty in Delft is used.

References

Afyouni, I., Ray, C., and Claramunt, C. (2012). Spatial models for context-aware indoor navigation systems: A survey. *Journal of Spatial Information Science*, (4).

Alattas, A., Zlatanova, S., Oosterom, P. V., Chatzinikolaou, E., Lemmen, C., and Li, K.-J. (2017). Supporting indoor navigation using access rights to spaces based on combined use of IndoorGML and LADM models. *ISPRS International Journal of Geo-Information*, 6(12):384.

Ballard, D. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122.

Boguslawski, P., Gold, C. M., and Ledoux, H. (2011). Modelling and analysing 3d buildings with a primal/dual data structure. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2):188–197.

Boguslawski, P., Mahdjoubi, L., Zverovich, V., and Fadli, F. (2016). Automated construction of variable density navigable networks in a 3d indoor environment for emergency response. *Automation in Construction*, 72:115–128.

Bosché, F., Guillemet, A., Turkan, Y., Haas, C. T., and Haas, R. (2014). Tracking the built status of MEP works: Assessing the value of a scan-vs-BIM system. *Journal of Computing in Civil Engineering*, 28(4):05014004.

Broersen, T., Fichtner, F., Heeres, E., De LIefde, I., Rodenberg, O., Meijers, B., Verbree, E., Van der Spek, S., and Ten Napel, D. (2015). Project pointless: Identifying, visualising and pathfinding through empty space in interior point clouds using an octree approach. *Geomatics Synthesis Project 2015/16*.

Brown, G., Nagel, C., Zlatanova, S., and Kolbe, T. H. (2013). Modelling 3d topographic space against indoor navigation requirements. In *Progress and new trends in 3D geoinformation sciences*, pages 1–22. Springer.

Cole, I. R. (2015). *Modelling CPV*. PhD thesis, Loughborough University.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to algorithms*. MIT press.

Diakité, A. A. and Zlatanova, S. (2017). Spatial subdivision of complex indoor environments for 3d indoor navigation. *International Journal of Geographical Information Science*, 32(2):213–235.

Diakité, A. A., Zlatanova, S., and Li, K.-J. (2017). ABOUT THE SUBDIVISION OF INDOOR SPACES IN INDOORGML. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W5:41–48.

Díaz-Vilariño, L., Boguslawski, P., Khoshelham, K., Lorenzo, H., and Mahdjoubi, L. (2016). INDOOR NAVIGATION FROM POINT CLOUDS: 3d MODELLING AND OBSTACLE DETECTION. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B4:275–281.

Díaz-Vilariño, L., Martínez-Sánchez, J., Lagüela, S., Armesto, J., and Khoshelham, K. (2014). Door recognition in cluttered building interiors using imagery and lidar data. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-5:203–209.

Díaz-Vilariño, L., Verbree, E., Zlatanova, S., and Diakité, A. (2017). INDOOR MODELLING FROM SLAM-BASED LASER SCANNER: DOOR DETECTION TO ENVELOPE RECONSTRUCTION. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W7:345–352.

Edelsbrunner, H., Kirkpatrick, D., and Seidel, R. (1983). On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559.

Eppstein, D. and Erickson, J. (1999). Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions. *Discrete & Computational Geometry*, 22(4):569–592.

Fichtner, F. (2016). Semantic enrichment of a point cloud based on an octree for multi-storey pathfinding. Master's thesis, TU Delft.

Giesen, J., Miklos, B., Pauly, M., and Wormser, C. (2009). The scale axis transform. In *Proceedings of the 25th annual symposium on Computational geometry - SCG 09*. ACM Press.

Hong, S., Jung, J., Kim, S., Cho, H., Lee, J., and Heo, J. (2015). Semi-automated approach to indoor mapping for 3d as-built building information modeling. *Computers, Environment and Urban Systems*, 51:34–46.

Jamali, A., Rahman, A. A., Boguslawski, P., Kumar, P., and Gold, C. M. (2015). An automated 3d modeling of topological indoor navigation network. *GeoJournal*, 82(1):157–170.

Jolliffe, I. T. (1986). Principal component analysis and factor analysis. In *Principal Component Analysis*, pages 115–128. Springer New York.

Jung, H. and Lee, J. (2015). INDOOR SUBSPACING TO IMPLEMENT INDOORGML FOR INDOOR NAVIGATION. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-2/W4:25–27.

Kang, H.-K. and Li, K.-J. (2017). A standard indoor spatial data model—OGC IndoorGML and implementation approaches. *ISPRS International Journal of Geo-Information*, 6(12):116.

Kim, J.-S., Yoo, S.-J., and Li, K.-J. (2014). Integrating IndoorGML and CityGML for indoor space. In *Web and Wireless Geographical Information Systems*, pages 184–196. Springer Berlin Heidelberg.

Krūminaitė, M. and Zlatanova, S. (2014). Indoor space subdivision for indoor navigation. In *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness - ISA 14*. ACM Press.

Kwan, M.-P. and Lee, J. (2005). Emergency response after 9/11: the potential of real-time 3d GIS for quick emergency response in micro-spatial environments. *Computers, Environment and Urban Systems*, 29(2):93–113.

Lamarche, F. and Donikian, S. (2004). Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. *Computer Graphics Forum*, 23(3):509–518.

Liu, L. and Zlatanova, S. (2011). A "door-to-door" path-finding approach for indoor navigation. *Proceedings Gi4DM 2011: GeoInformation for Disaster Management, Antalya, Turkey, 3-8 May 2011*.

Ma, J., Bae, S. W., and Choi, S. (2011). 3d medial axis point approximation using nearest neighbors and the normal field. *The Visual Computer*, 28(1):7–19.

Mirza, R., Tehseen, A., and Kumar, A. V. J. (2012). An indoor navigation approach to aid the physically disabled people. In *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*. IEEE.

Munkres, J. R. (1984). *Elements of Algebraic Topology*. Addison-Wesley, Menlo Park, CA.

Nikoohemat, S., Peter, M., Elberink, S. O., and Vosselman, G. (2017). EXPLOITING INDOOR MOBILE LASER SCANNER TRAJECTORIES FOR SEMANTIC INTERPRETATION OF POINT CLOUDS. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W4:355–362.

OGC (2014). OGC Indoor GML. (14-005r5).

Peters, R. (2018). *Geographical point cloud modelling with the 3D medial axis transform*. PhD thesis, TU Delft.

Rueppel, U. and Stuebbe, K. M. (2008). BIM-based indoor-emergency-navigation-system for complex buildings. *Tsinghua Science and Technology*, 13(S1):362–367.

Schnabel, R., Wahl, R., and Klein, R. (2007). Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226.

Staats, B. R., Diakité, A. A., Voûte, R. L., and Zlatanova, S. (2017). AUTOMATIC GENERATION OF INDOOR NAVIGABLE SPACE USING a POINT CLOUD AND ITS SCANNER TRAJECTORY. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W4:393–400.

Teo, T.-A. and Yu, S.-C. (2017). THE EXTRACTION OF INDOOR BUILDING INFORMATION FROM BIM TO OGC INDOORGML. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W2:167–170.

Tran, H., Khoshelham, K., Kealy, A., and Díaz-Vilariño, L. (2017). EXTRACTING TOPOLOGICAL RELATIONS BETWEEN INDOOR SPACES FROM POINT CLOUDS. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W4:401–406.

Volk, R., Stengel, J., and Schultmann, F. (2014). Building information modeling (BIM) for existing buildings — literature review and future needs. *Automation in Construction*, 38:109–127.

Wallgrün, J. O. (2005). Autonomous construction of hierarchical voronoi-based route graph representations. In *Spatial Cognition IV. Reasoning, Action, Interaction*, pages 413–433. Springer Berlin Heidelberg.

Yang, L. and Worboys, M. (2015). Generation of navigation graphs for indoor space. *International Journal of Geographical Information Science*, 29(10):1737–1756.