Detection of doors in a voxel model, derived from a point cloud and its scanner trajectory, to improve the segmentation of the walkable space

Staats, Bart; Diakite, Abdoulaye; Voûte, Robert; Zlatanova, Sisi

DOI
10.1080/12265934.2018.1553685

Publication date
2018

Document Version
Accepted author manuscript

Published in
International Journal of Urban Sciences

Citation (APA)
Staats, B., Diakite, A., Voûte, R., & Zlatanova, S. (2018). Detection of doors in a voxel model, derived from a point cloud and its scanner trajectory, to improve the segmentation of the walkable space. *International Journal of Urban Sciences*, *23 (2019)*(3), 369-390. https://doi.org/10.1080/12265934.2018.1553685

International Journal of Urban Sciences Special Issue on Geoinformatics in Urban Sciences

B. R. Staats[a*], A. A. Diakité[b], R. L. Voûte[a,c], S. Zlatanova[b]

[a]*Transport, post & logistics, CGI Nederland, Rotterdam, The Netherlands;*

[b]*GRID, UNSW Built Environment, Red Centre Building, UNSW, Sydney, Australia;*

[c]*Department of Urbanism, Faculty of Architecture and the Built Environment, TU Delft, Delft, The Netherlands*

Detection of doors in a voxel model, derived from a point cloud and its scanner trajectory, to improve the segmentation of the walkable space

Generation of indoor networks for navigation will normally be done out of standard floor plans that are only 2D and is more often manual than automatic. These floor plans are drawn at a specific time and do not correspond to the reality, moreover some of those buildings were built already differently than designed. Then in due course the building will change both externally and internally. Also objects like furniture will be moved around in the building. If these changes are not updated in the map of the building, it becomes out of date and cannot be used for the creation of indoor navigable models anymore. To enable correct indoor navigation, we will need to have the current data of the indoor environment. This article concentrates on providing a new approach to generate up to date floor plans by using a mobile (and hand held) laser scanner in the fastest way. This device creates a point cloud and the corresponding trajectory at the same time. Because the mobile laser scanner device is operated by a walking human, the trajectory contains information about the surface the person is walking on. In this article, a method is explained for the detection of walkable spaces based on the analysis of the point cloud and its corresponding trajectory provided by the mobile laser scanner. Three steps will be used: voxelization, trajectory analysis and the identification of floor regions. Dynamic objects, doors, and furniture objects are also used to identify the surfaces which are available for navigation purposes. Three types of surfaces are considered: horizontal, slopes, and stairs.

Keywords: Walkable space; Mobile laser scanner; Indoor geographic; Trajectory; Voxel; Dynamic objects

Introduction

Seamless indoor-outdoor navigation is in great demand in recent years, but it requires a switch between indoor and outdoor environments (Thill, Dao, & Zhou, 2011). While the navigation is well implemented and used in all kinds of outdoor applications, navigation aids for inside buildings (or both indoor and outdoor) are still in development. The complexity of indoor environments is greater, and the indoor aid is specifically important for public areas in larger buildings such as hospitals, airports, railway stations, conference venues, museums, and shopping malls. Similar to outdoor, an indoor navigation system uses an (indoor) positioning system, a navigable map, destinations (points of interest) and directions to follow a path (Brown, Nagel, Zlatanova, & Kolbe, 2013; Boguslawski, Mahdjoubi, Zverovich, & Fadli, 2016). One of the most critical components for this kind of navigation is the indoor map itself, not only for visualisation purposes, but also for the connection information it provides, like destinations, routes, connections, and obstructions. Normally all buildings have 2D floor plans, however many floor plans are outdated since interiors have changed because the walls and doors have been modified (Turner, Cheng, & Zakhor, 2015). In a shorter timeframe, the furniture might have been changed because of the user's preferences or usage. Moreover, it might happen that buildings have not been built according to their blueprints at the first place. Since indoor environments tend to be far more complex than the outdoor world, creating up-to-date indoor maps is time consuming and not easy (Zlatanova, Liu, & Sithole, 2013). This means that the automatic updates of indoor environments are of critical importance.

Present research being done on automatic generation of indoor maps has a focus on 2D floor plans. Only part of this research attempts to consider 3D representations (Zlatanova, Liu, Sithole, Zhao, & Mortari, 2014). Floor plans which are in 2D are appropriate for many purposes but have their limitations for user-tailored navigation (Diakité & Zlatanova, 2016). 2D maps always are a simplification of the complex 3D environment. This might lead to wrong representations of the real situation. Also, connectivity between different floor plans can raise problems as the stairs/elevators and lifts cannot be represented (Zlatanova, Liu, Sithole, Zhao, & Mortari, 2014). Moreover, the 2D maps are not suitable to represent furniture and other obstacles that have overhanging parts and therefore limit the space for movement (Díaz Vilariño, Boguslawski, Khoshelham, Lorenzo, & Mahdjoubi, 2016). Thus, new 3D methods for the efficient registering of indoor environments should be looked for.

This paper focusses on the automatic detection of the indoor spaces open for navigation for pedestrians based on the data of a mobile laser scanner. Such a scanner will scan the environment while registering the trajectory of movement of the device at the same time, which makes it more time efficient than a static terrestrial laser scanner (Holenstein, Zlot, & Bosse, 2011; Sirmacek, Shen, Lindenbergh, Zlatanova, & Diakite, 2016). The obtained point cloud should be processed to further identify static objects like floors, stairs, walls, and the more dynamic furniture objects. Many approaches are constrained, like with a Manhattan World approach or with a flat/planar/horizontal surface limitation (Fichtner, Diakité, Zlatanova, & Voûte, 2018; Macher, Landes, & Grussenmeyer, 2016; Khoshelham & Díaz-Vilariño, 2014; Anagnostopoulos, Pătrăucean, Brilakis, & Vela, 2016; Budroni & Boehm, 2010). Our research addresses more complex environments, which is why we envisage a method with less or without any constraints and which is predominantly based on the trajectory of the MLS which is created during the scanning procedure. If the MLS will be operated by a human, the trajectory contains data which can be used to distinguish between different types of surface. The scanned points, which are almost directly below the trajectory, indicate walkable areas for pedestrians (Yan, et al., 2016; Li, 2014). The height differences between neighbouring trajectory points can be used to indicate a stair or a slope, and so also recognizes horizontal surfaces. Apart from this, the trajectory provides connectivity information and represents the complexity of the building when the data of the MLS will be captured according to certain operational rules.

This paper presents an improved method for the detection of walkable surfaces based on the digital analysis of the point cloud in combination with the trajectory of the MLS. It is an extension and improvement of the previously reported work in Staats, Diakité, Voûte, & Zlatanova (2017). By combining the scanned point cloud and the trajectory, it makes it possible to detect the walkable spaces (WS) thanks to the implicit information of the building (e.g. complexity, slopes and connectivity) is already embedded in the trajectory of the MLS. An octree structure is used to organise the point cloud and three types of walkable surfaces are detected: flat, sloped and stairs.

The method is fully detailed and the previous approach is improved by applying a new type of detection of the start and endpoints of stairs or slopes. We also implement door detection based on two different identification methods. The detected doors are used to separate the indoor environment into rooms. Doors (being open, closed or locked) are important components of navigation process by providing connectivity between spaces.

The paper is organised as follows; first, related research is discussed. Second, the method will be introduced, and the implementation will be discussed. Finally, further research will be discussed, and a conclusion will be drawn.

Related work

The related work is primarily focussed on the description of the space in general and identifies different approaches for door detection based on a scanner trajectory.

The definition of space

It is difficult to accurately define the notion of space, as a lot of different definitions exist in literature. Ekholm and Fridqvist (2000) describe that people mostly think of a space as: '… an empty volume, enclosed in some respect - materially or experientially.’

Materialised enclosed spaces have physical borders like a wall or a door. Experientially enclosed spaces have no physical boundaries at all. However, experientially spaces can be perceived differently depending on the circumstances. The Industry Foundation Classes (IFC) standard describes the following: ’A space represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building.’ (SMART). Earlier, Ekholm (1996) described a space from a construction sector’s point of view: ’A room in a building is a space with a free void that is large enough to accommodate users and equipment; the building parts that make up the space are enclosing, e.g., to climate, light, sound, or fire.’

A space is described as a result of the enclosing elements in the definitions mentioned previously. Zlatanova, Liu, and Sithole (2013) describe indoor navigation – focused space as: 'the environment in which humans store resources (items of interest) and engage in navigation activities.’ The authors added further that a logical grouping of resources and navigation activities requires the creation of sub-spaces. This definition of a space does not focus on the building components that form a space but the other way around.

Finding a definition of space where all the different views can be merged is not easy. In our case, the best way to define the word space is to rely on a more general definition, such as the one proposed by Ekholm and Fridqvist (2000). However, that definition still leaves room for different interpretations. Therefore, it is important that the definition is clearly specified.

Regarding the WS, we could not find any formal definition provided to it in the existing literature, but it is assumed to be the space where pedestrians can move freely. It is generally related to the floor and all surfaces related to it (stairs, slopes, etc).

Door detection

The detection of doors is a difficult task and is the subject of lots of researches. According to Quintana, Prieto, Adán, & Bosché (2018), this research can be divided in two groups based on the type of data capture: 2D approaches which uses colored images and 3D approaches which use laser scanners or photographic systems.

Quintana, Prieto, Adán, & Bosché (2018) are detecting doors based on the existence of data holes in wall planes. Closed doors are detected by investigating rectangular wall areas that, after subsequent processing, does not belong to that wall. Díaz Vilariño et al. (2016) developed a method to detect windows and doors by finding consistent shapes based on the Generalized Hough Transform (GHT) in combination with binary images of the point cloud.

Besides these two approaches there are many more to discuss but most approaches don’t use the trajectory of a mobile laser scanner. The trajectory is used to detect doors and subdivide spaces by Zheng, Peter, Zhong, Oude Elberink, & Zhou (2018). Their method analyses the different scan lines of a mobile laser scanner. If a gap in a scanline is detected, a possible door of window could be found. Opening candidates are defined by combining multiple scanline analysis. An opening is classified as a door if an opening candidate is crossed by the trajectory because the data collector has walked there. This method doesn’t reconstruct a whole building to detect doors but uses only the original point cloud and the trajectory. Díaz-Vilariño, Verbree, Zlatanova, & Diakité (2017) also don’t make a model of the indoor environment. They detect doors by extracting a vertical profile along the trajectory. By analysing the height changes at a specific level, doors can be detected. The detection of doors works well if the height difference between the ceiling and the door are large enough. If the distance between the ceiling and the trajectory is smaller, the detection of doors will be harder if not possible at all. After the doors are identified the trajectory is split and the point cloud is divided into rooms based on the scan time of the point cloud and the trajectory using a ray tracing method. The ray tracing is also used by the approach of Nikoohemat, Peter, Elberink, & Vosselman (2017). They first construct surfaces by growing segments. After this, they create a voxel space and use ray tracing between the time of the point in the trajectory and the corresponding time in the point cloud. By checking the intersection of a ray with the identified surfaces, the voxels are marked as occluded, opened and occupied. In this step some doors are detected. The real door identification detects the centre of doors if:

(1) There are some voxels above the door centre
(2) There are within 15 cm other trajectory points
(3) There are within 50 cm of a door voxels empty voxels and occupied voxels.

Unlike most of the approaches that have been previously discussed, our focus is exclusively on features critical to pedestrian navigation. Our approach structures the point cloud into a voxel space, performs filtering operations to identify moving and static obstacles, and relies directly on the information embedded in the trajectory to identify walkable surfaces and categorize them into flat areas, sloped ones or stairs. Furthermore, we propose door identification methods that also take advantage of the trajectory and the voxel structure. Our approach to detect doors relies on the combination of a horizontal and vertical checks of the voxel structures and the trajectory, which allows to detect them more precisely.

Method

As discussed in the related work section, the definition of a space can be perceived in different ways. In this research we want to identify the WS for pedestrians with a specific height. Therefore, the definition of WS can be defined as follows; a space is the free space that is used for human navigation inside a building without colliding with any obstacles considering the actors height and size. A section of the free space is the indoor environment, which can be accessed through an entryway and is enclosed by walls, entryways, windows, ceilings or the actors height, see Figure 1.

To be able to plan a path through a building for different types of actors like pedestrians, people in a wheelchair or people with a walker, it is important to classify stairs and slopes which then can or cannot be used in the path planning algorithm. In this method horizontal, stair, and sloped surfaces will be identified.

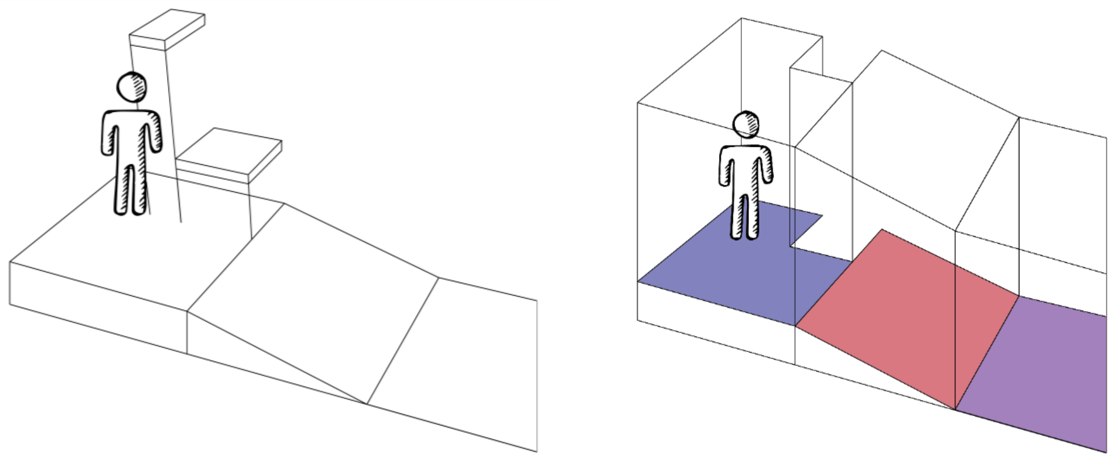

Figure 1: Indoor space with objects (left), three walkable spaces (right)

As explained in Staats, Diakité, Voûte, & Zlatanova (2017), the method rely on the two datasets produced by the MLS device during data acquisition, which are the point cloud on one hand, and the trajectory of the MLS on the other hand. The method is divided in different steps to derive the WS, see figure 2.

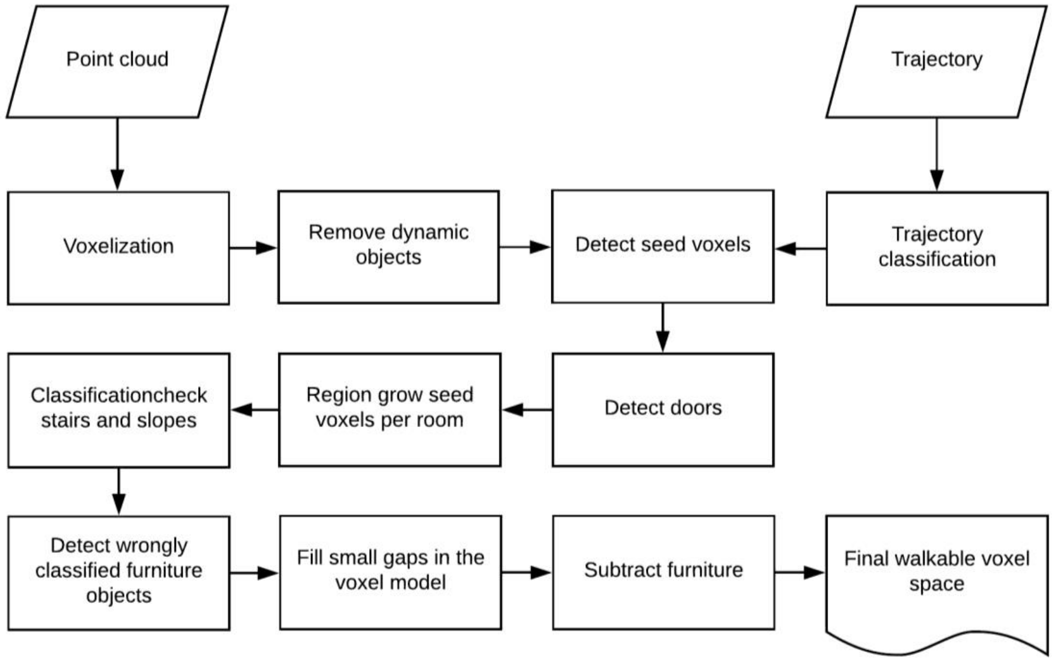

Figure 2: Steps of the proposed method

The first step of the approach consists in voxelizing of the point cloud. As described in Vo, Truong-Hong, Laefer, & Bertolotto (2015), this process has two advantages. It first introduces a spatial structure to the data, and secondly it reduces its amount, which improves the processing time. The size of the voxel has a large impact on the representation of the point cloud, but also on the final WS, and therefore needs to be chosen with care.

The step following the voxelization is dedicated to the detection and removal of dynamic objects present during the data capture, on the basis of their capturing time which is stored in the point cloud. This time component is added by the Simultaneous Localization And Mapping (SLAM) algorithm, which stitches all the different scans along the trajectory together. As the amount of time spent by a dynamic object at a specific position is short, the points that represent, for example, a walking pedestrian form a long-drawn shadow and not a dense representation of a human shape (Józsa, 2012). By calculating the unique scanning seconds of the points contained in a given voxel, dynamic objects can be detected and then will be removed. If a pedestrian stands still during the data capturing, a human will be visible as a recognizable shape, see figure 3. In this case, a pedestrian was scanned during the red and the blue time slot.

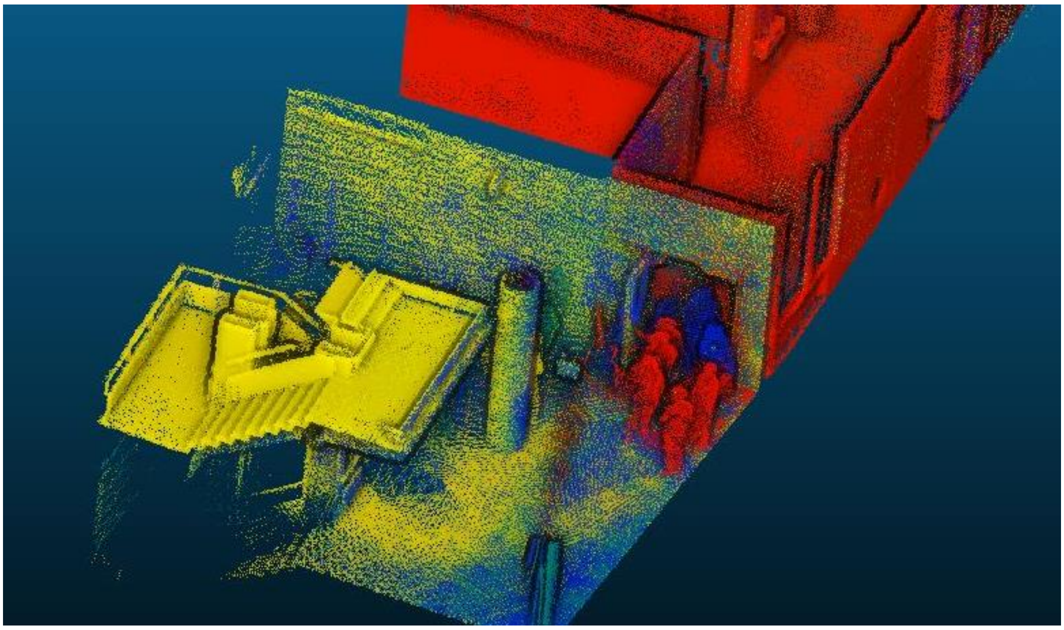

Figure 3: Dynamic objects staying at a specific location during the data capture (one colour per scanning time).

The following step of the approach consists in classifying the trajectory into: stairs, flat, and sloped regions. This is done based on angular parameters analysis between the successive trajectory points. There are different angle parameters for each structural component (stairs, slopes, etc.), and for each of them, there are several successive points that need to have the same characteristics before they can be classified into one or another. The output of this classification is therefore sensitive to the changes in height of the MLS device during the data capture.

Similarly to the point cloud, the trajectory is also voxelized using the same spatial structure. As described in the introduction, the point cloud is captured by an operator. Therefore, the trajectory is only present in places where the operator has walked and the voxels below the trajectory represent walkable voxels or seed voxels. This is done by projecting the trajectory's voxels on top of the point cloud's ones.

The next step consists of the identification of doors. This is an improved method in comparison with the previous paper. These latter divide the indoor space in confined smaller spaces; this information will be used as one of the two stopping criteria of the region growing process. The identification of these smaller spaces is important because the user navigates to a specific location in the building rather than an entire floor. The detection of doors is based on two processes. First, a vertical-check is performed where the variation of height change between the seed voxels and the ceiling is identified. This approach was also used by Díaz-Vilariño, Verbree, Zlatanova, & Diakité (2017), as illustrated in the left part of figure 4. This process works when the difference in height between the top of an entryway and the ceiling is large enough. When this is too small, using this approach during the detection of doors is very difficult. Second, the horizontal-check which looks at specific fluctuations in the horizontal plane, see figure 4. This check is implemented at a height of 1,60 meters, because there are less furniture objects at that height.

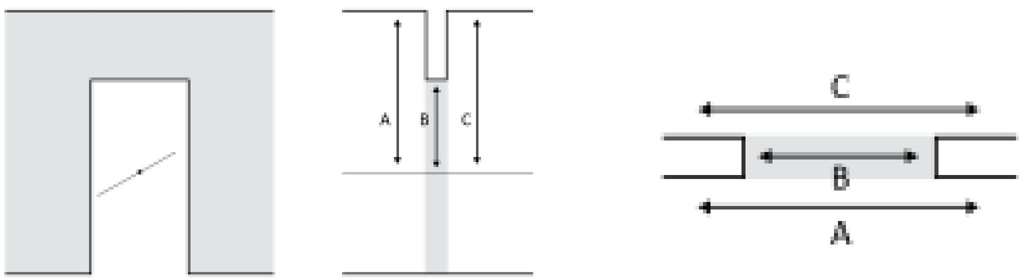

Figure 4: Vertical check (left and middle). The height is first large (A), then small (B), and then large again (C). Horizontal check (right). The free distance is first large (A), then small (B), and then large again (C).

To find the WS, a region growing algorithm is used to further process the seed voxels and identified doors; the ST_ClusterDBSCAN algorithm in a PostgreSQL database (PostgresSQL, 2017). Because the algorithm only processes 2D clusters, the voxels are region grown per height level. Only the regions containing seed voxels are saved.

The type of floor regions is based on the seed voxels it contains. Because the MLS is held in front of the operator, it can already be above the second riser of a stair before the operator even enters the stair. The produced effect is visible in figure 5 (see the green box), where the trajectory (red) is increasing in height after the geometry model (blue) has increased in height. At this moment, the geometry indicates a riser of a stair, but the trajectory is not ascending the stair at all. Therefore, the riser in the beginning or end of a stair cannot be detected correctly, which is also the case for sloped surfaces.

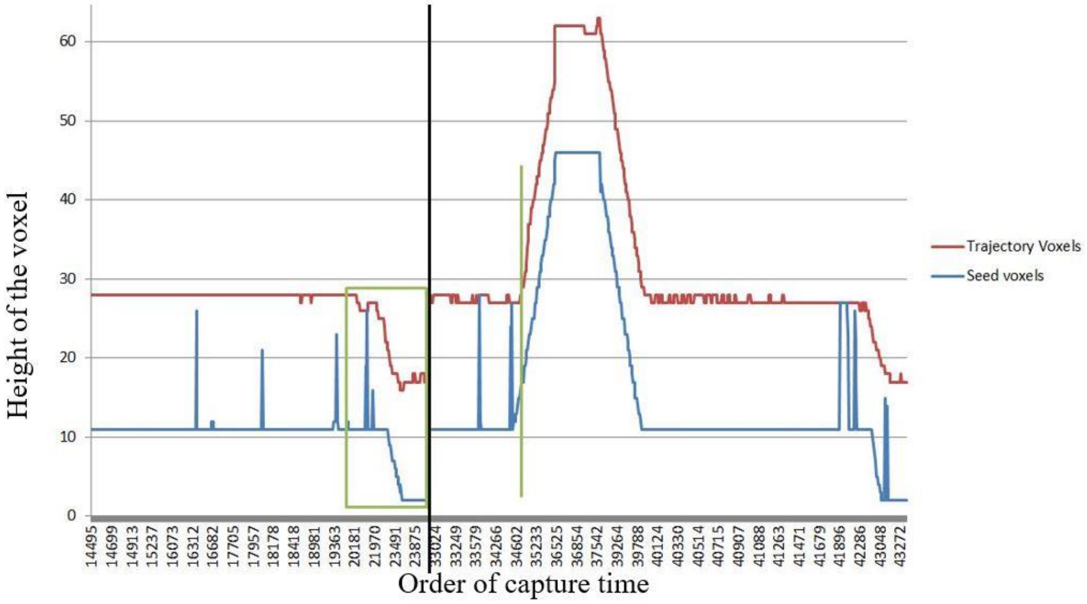

Figure 5: Difference in change in height between the trajectory and the geometry model when a stair is entered. Trajectory voxels (red), seed voxels (Blue), box (green)

This results in the wrong classifications of the regions at the beginning of stairs or slopes. In this step, the beginning and end of a stair or slope are detected based on the ordered seed voxels, the geometry, and the size of the classified regions. Compared to the work of Staats, Diakité, Voûte, & Zlatanova (2017), the new algorithm is more specific at detecting these features. Besides this, it is also possible that the scanner gets held above furnishing elements. If the voxels of the trajectory are projected down, these furniture objects will be classified as a certain type or surface; a stair, slope or a horizontal surface. These errors will be corrected during this step.

The next step consists of filling small gaps in the regions. The MLS scans even below furniture. As previously discussed, a WS is supposed to be free of obstacle for a pedestrian. The remaining voxels above the regions after the cleaning and the analysis are assumed to be furniture objects. For each surface region, the voxels up until the actors' height are retrieved and removed from the identified surfaces. This results in the final WS.

Implementation

The developed algorithms have been tested on a point cloud of a university building. This indoor environment contains stairs, slopes, furniture, and a large theatre-like structure with rooms inside of it, see figure 6

Figure .

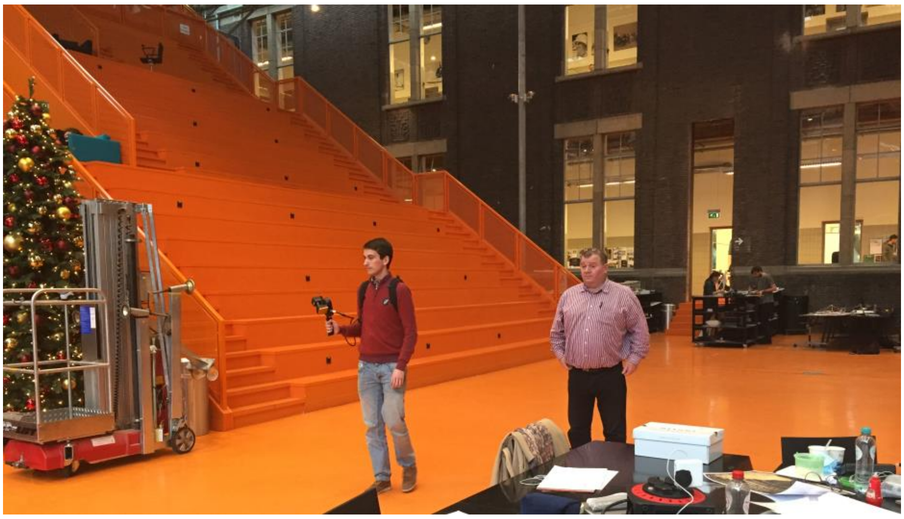

Figure 6: Image of the testing location

This space is scanned during opening hours, which resulted in a point cloud which includes dynamic objects, see figure 7.

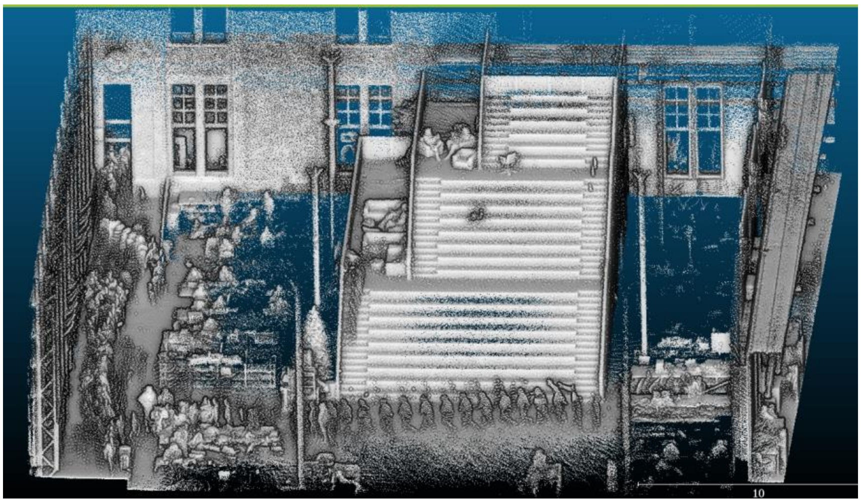

Figure 7: Image of the point cloud of the MLS device with dynamic objects.

The data capture (including the point cloud and trajectory) was done using the Zeb Revo laser scanner. This scanner has a maximum range of 30 meter, 43.200 points/sec, scanning resolution of 0.625° horizontal and 1.8° vertical, angular field of view 270° x 360°, absolute position Accuracy 3 – 30cm and a relative accuracy of 1 –

3cm. The algorithms of the approach were implemented using Python 2.7 and PostgreSQL 9.5. The overall was tested on an Intel(R) Core i7-7820HQ CPU at 2.90 GHz, 32.00 GB RAM laptop, running Windows 10 on a 500 GB SSD. The point cloud exists of around 14 million points and is based on a local coordinate grid.

Implementation results

The first step of the process exists of the voxelization of the point cloud. If a voxel contains a point, the voxel is saved. Otherwise the voxel will be removed. The voxel size is based on the smallest element that needs to be identified. In this approach, the smallest element is a riser of a stair. The risers of a stair’s step should be less or equal to 15 cm, according to the ISO (2011) standard. Thus, if the voxel size is 15 cm, a riser is represented by one voxel. However, the correct form of the riser can hardly be detected if data is noisy. Therefore, a smaller voxel size of around 5-7 cm is used (Staats, Diakité, Voûte, & Zlatanova, 2017).

The dynamic objects detection relies on the number of unique scanning timeframes contained inside a voxel. Different tests showed that the best results are achieved when voxels with less than two scanning seconds are filtered out. Increasing the threshold resulted in the loss of parts of the model that were not or could not be scanned thoroughly. To minimize such loss, the threshold needs to be as low as possible. Most of the dynamic objects cloud be removed from the voxel model with the cleaning process, see figure 8. The remaining dynamic objects are objects that were at the same place for a longer period of time or parts that overlap in different scanning rounds. These objects are not filtered in the proposed method.

The objects that were scanned for a longer period of time are scanned more thoroughly and have a better voxel representation, see figure 3. One possible way to detect these dynamic objects is by detecting their shape.

The SLAM algorithm gives better results if the data is captured while ensuring a loop closure (GeoSLAM, 2016). This results in multiple scans of the same part of a building which increases the change of overlap in different scanning rounds. These parts are not represented by a human shape but form smaller residual noise objects. To detect these residual noise objects, the scanning times could be divided into separated data frames, as suggested by Litomisky & Bhanu (2013). By analysing these data frames, the residual noise objects can probably be detected, which should be addressed in further research.

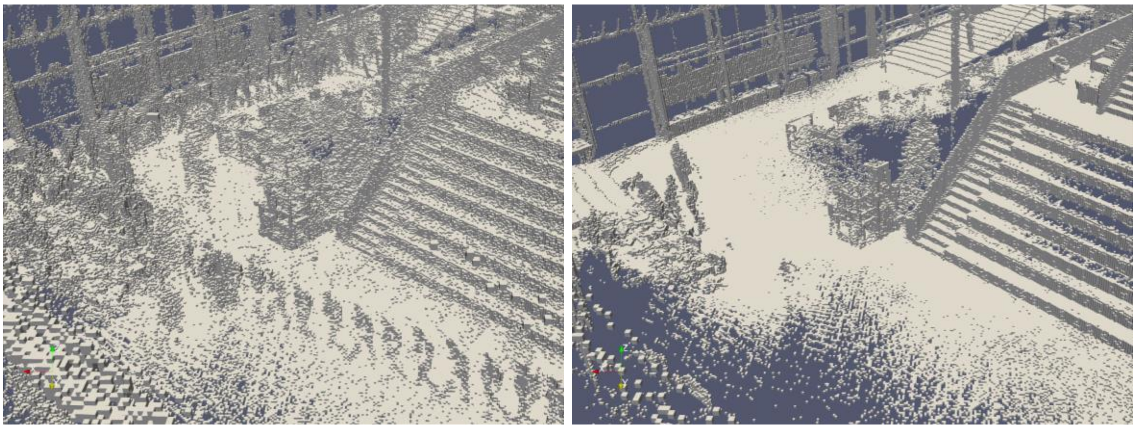

Figure 8: Voxel model before cleaning (left), voxel model after cleaning (right)

After the cleaning of the voxel model, the trajectory is analysed into three types: stair, slope or horizontal. The type classification is based on angles between successive trajectory points. There are three parameters necessary for this process; a minimal angle, a maximal angle, and several connected elements as illustrated in table 1. If a point is within these parameters, it is classified as a stair or slope element. The resulting unclassified points will be marked as horizontal elements, see figure 9.

Type	Minimal angle In degree	Maximal angle in degree	Connecting elements

Stair	7,1	60	4
Slope	2,3	18,4	2

Table 1: Trajectory classification parameters

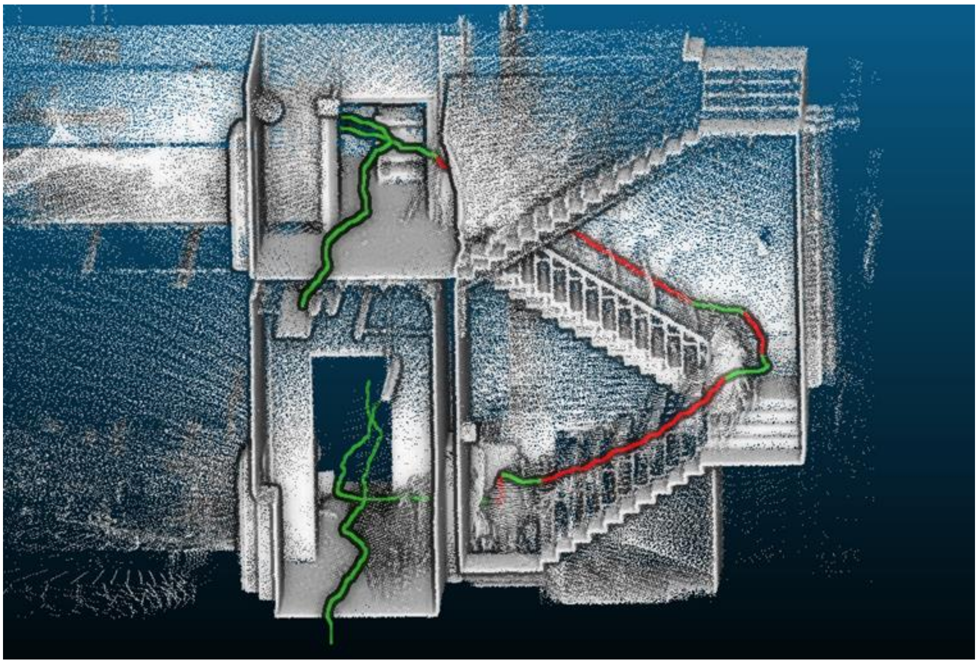

Figure 9: Analysed trajectory. Horizontal (green), stair (red)

After the classification of the individual trajectory points, the same spatial structure of the voxel model is used to voxelize the trajectory. After this, the trajectory is projected down on the model. These seed voxels are the input for the region growing step and therefore referred to as seed voxels, see figure 10.

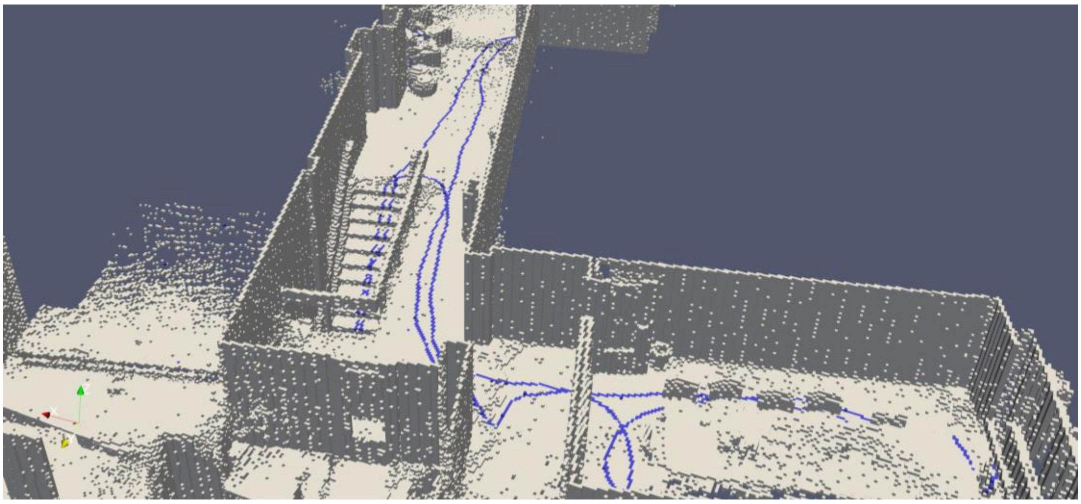

Figure 10: Seed voxels (blue) in the voxel model (white)

As described above, the detection of doors is based on a vertical-check and a horizontal-check. With the vertical-check, the distance between a seed voxel and the ceiling is calculated. If the ceiling is high enough, the doors can be detected when a specific pattern appears, see figure 11. This way, doors can be detected by the height alone.

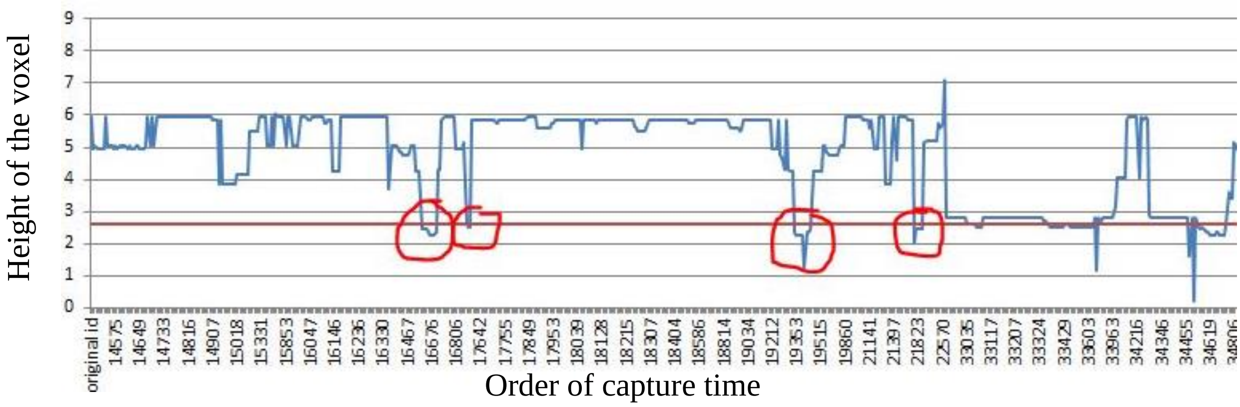

Figure 11: Identified doors along the voxel trajectory (horizontal), distance between ceiling and floor (vertical).

Doors cannot be detected based on the height check if the difference in height between the ceiling and the doorframe in not high enough or if the ceiling consists of pipes and air vents. This can be seen in the irregularity of the second part of the blue line in figure 12. At this position, the ceiling contains a lot of pipes which resulted in the identification of many false positive doors. Therefore, the side horizontal-check is implemented as extra verification.

The horizontal-check is implemented at a height of 1.60 meters above the seed voxels. At this height, there are less furniture objects. For each seed voxel, the number of neighbouring voxels within a radius of 50 cm, which is the size of a doorway, are added which results in the orange line. A door is only identified as a door if there is peak in the vertical-check and a peak in the horizontal-check, see figure 12. The green boxes represent doors, as can be seen in figure 13. The data in the red box has a spike in the horizontal-check but has no spike in the vertical-check and will therefore not be classified as a door.

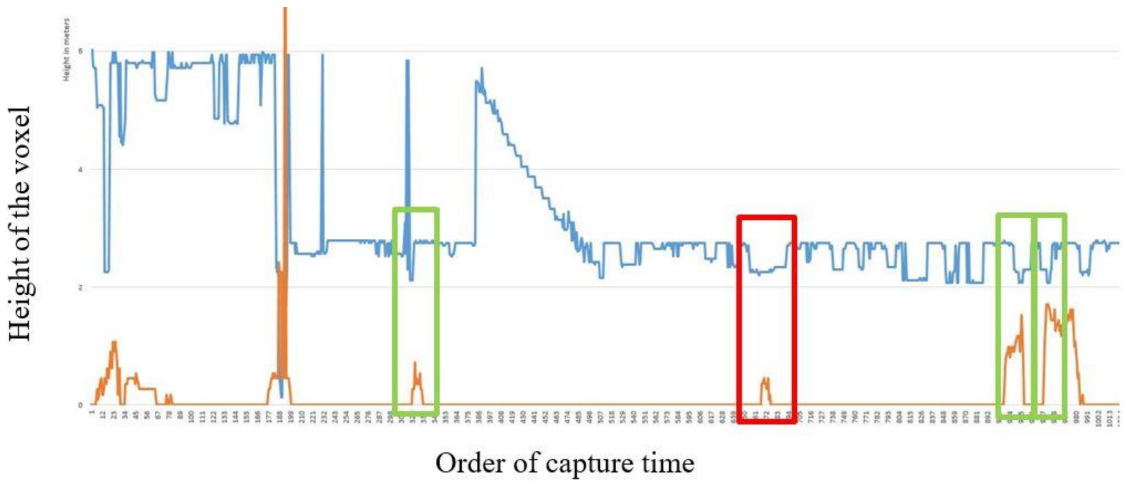

Figure 12: Height change between the seed voxels and the ceiling (blue, in meters). Number of voxels around the seed voxels at a height of 1.60 meters (orange, in number of voxels * voxel size). Green and red boxes from left to right: box1, box2, box3, box4.

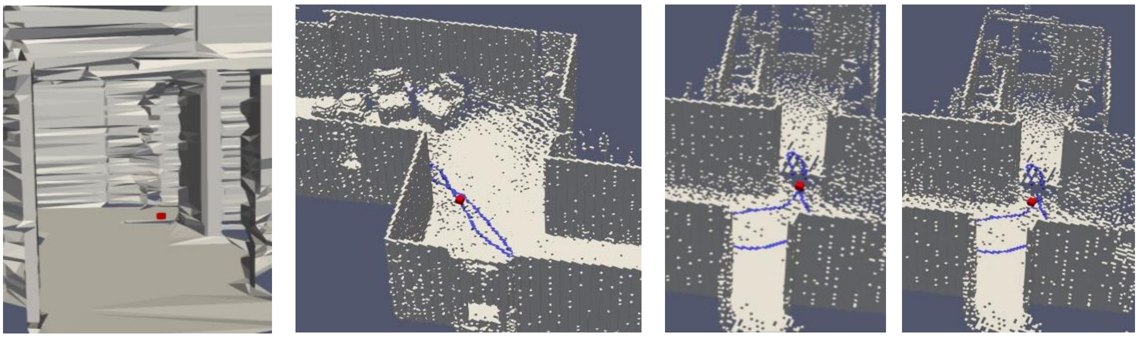

Figure 13: Image of the boxes in figure 13. From left to right: box1 entryway, box2 no entryway, box3 entryway, box4 entryway.

The used point for this research contains nine doors. Each time the trajectory crosses a door it can be detected therefore, nineteen detectable passages were present, see table 2. The implemented method detected twenty-two possible passages. From these passages, six false positives were present. These false positives were places where a height check peak appeared due to the piping in the ceiling and the seed voxel were close to a wall. In this way, a peak appeared in the horizontal-check. As visually inspection showed, these walls were always at one side of the seed voxel. The total amount of correctly detected passages is 84,2 percent where each of the nine doors was detected once.

Total amount of doors	Detectable passages	Detected passages	False positives	Percentage detected passages in %	Percentage correctly detected passages in %
9	19	22	6	115,79	84,21

Table 2: Total amount of identified doors

Further research is needed to improve the door detection method. First, the algorithm should be automated to detect the peaks. The detection is now based on visual inspection of the vertical and horizontal-check, see figure 12. Second, it is also interesting to further improve the horizontal-check. The current horizontal-check counts all the voxels around the seed voxel at a specific height. If an entryway is passed there is, in most cases, a door frame which results in objects on the left side and right side of the seed voxel. If the algorithm is improved to only check the left and right side at a specific part, a lot of falls positives can be removed from the result. Third, the voxels that form the doorway need to be identified and removed. In that way, the region growing process can split each different room.

As discussed earlier, the ST_ClusterDBSCAN algorithm only processes 2D clusters (PostgresSQL, 2017). Therefore, the whole voxel model is region grown into horizontal regions. Only regions containing seed voxels are kept. The region growing process is designed to account for the horizontal eight neighbour adjacency of each processed voxel. Because the voxel space subdivides the original point cloud to its smallest extend, a neighbour voxel has a distance of the value 1. This value represents the actual voxel size of around 5-7 cm. According to Pythagoras, the four corner voxels have a distance of √2, which is the distance parameter for the region growing process.

Two stopping criteria are used during the region growing process: the first one occurs when a door voxel is reached, and the second one occurs when a voxel is found to have at least 2 voxels above itself. The space above these voxels is occupied and is not representing WS. The door voxels are currently not identified and are currently not used as a stopping criterion but will be in future research. The result of this process can be see figure 14.

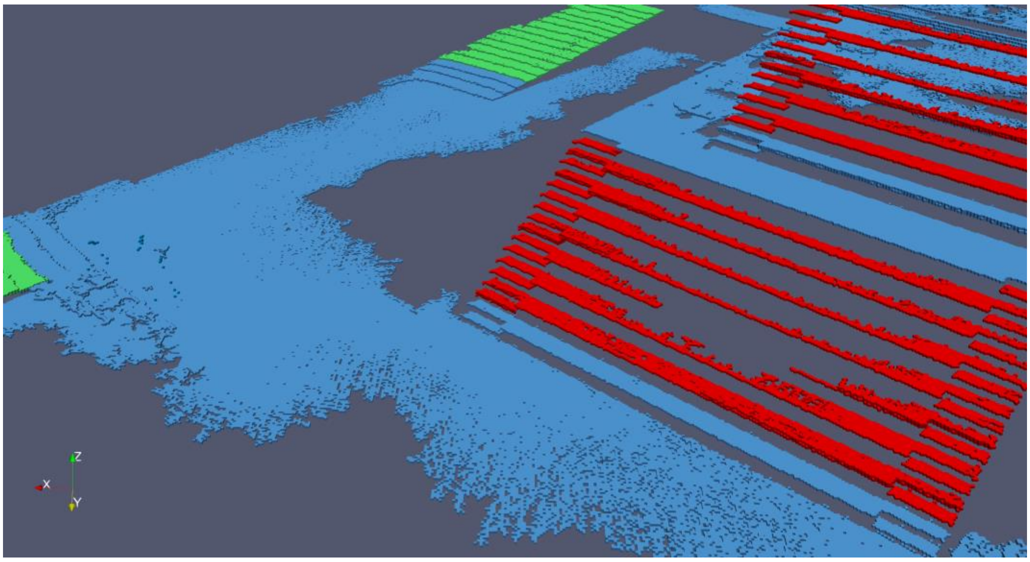

Figure 11: Region growing of the seed voxels. Slope (green), horizontal (blue) and stairs (red).

The next step consists of the classification-check. This is an improved method compared to Staats, Diakité, Voûte, & Zlatanova (2017). The classification-check is needed because not all risers and not all slope parts can be detected based on the height change of the trajectory. Therefore, this check is based on the geometry of the model.

The classification check proposed in Staats, Diakité, Voûte, & Zlatanova (2017) was based on the closeness of stair or slope seed voxels. If a seed voxel is within a certain threshold and not classified as a stair or slope, it was classified as a stair or slope. This approach is not very specific and results in overshoot or undershoot of the stair parts, which looks like the left image of figure 15.

The proposed improved classification-check is more specific and is also based on the size of the risers or slope parts and their change in height. The process starts by analysing the different types of seed voxels in a region. For each region, the number of seed voxels per type are calculated. All the voxels of each region are updated to the type with the most seed voxels for that region, see the right part of figure 15.

For now, the type of a floor region is based on the type with the most seed voxels. It is interesting to further investigate what happens when more cases are introduced. A different decision could be made when there are two types with almost the same amount of seed voxels in a region. Future testing and research is needed to verify if this approach can be of use for the classification of stair or sloped surfaces.

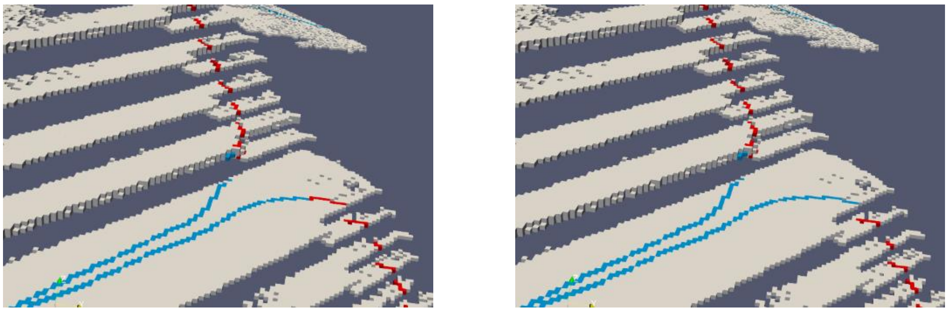

Figure 15: Seed voxels in regions of different types (left) and seed voxels in regions of one type (right)

The next step consists of the retrieval of the largest and smallest riser region of a stair or a slope. This is done by visiting all the seed voxel in the order they were captured with the laser scanner. If a stair seed voxel is reached, the size of that regions is added to a list. This goes on until the last seed voxel of that stair is reached. The resulting stair floor regions are ordered by size and the smallest and the fourth largest riser regions are saved. The fourth largest region is chosen instead of the largest region because a stair can contain horizontal parts. These have larger areas than regular risers and should therefore not be added to the list largest or smallest list.

After the smallest and largest regions of a stair or slope are known, the seed voxels are once again visited in order they were captured. A seed voxel belongs to a stair or a slope, if the following three rules apply:

(1) There is a specific height change in the ordered seed voxels

(2) There is a classified voxel of the stair or slope type within a specific distance to the current voxel

(3) The size of the regions is between the largest and the smallest regions closest to the stair

These rules can be translated to different parameters for stairs and slopes, see table 2. The detection of wrongly classified furniture objects will be discussed later on.

	Height change (in voxels)	Voxel of same class within	Size of the regions
Slope	Change = 1	40 voxels	0.5 x small size < size < 1.5 x large size
Stair	1 < Change < 5	10 voxels	0.5 x small size < size < 1.5 x large size
Furniture	Change > 10	-	-

Table 3: Parameters of the classification check.

The parameters of table 3 are the result of different tests on different datasets. What can be noticed, is that the number of voxel of same class within is much higher for a slope than for a stair. Tests showed that in some cases the first two parts of the slope are not classified as such. Because the height change over a certain distance is much smaller for a slope than for a stair, the voxel distance of a slope riser is much larger. If these rules are applied the beginning of stairs and slopes can be identified correctly, see figure 16.

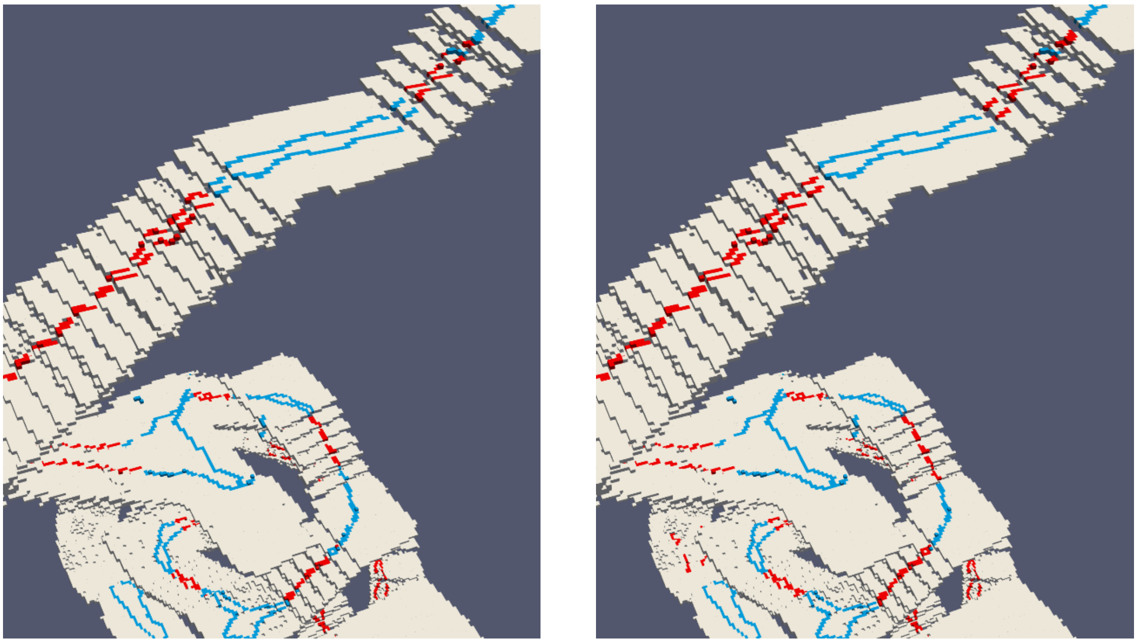

Figure 16: Seed voxels before the classification check (left) and seed voxels after the classification check (right)

Because the size of regions can be 1.5 times larger than the largest region or 0.5 times smaller than the smaller region, it is possible that some horizontal floors have voxel regions at different heights. These regions are within the range of the largest stair riser and are therefore classified as stairs, see figure 17. These kinds of exceptions need to be taken into account in further research.

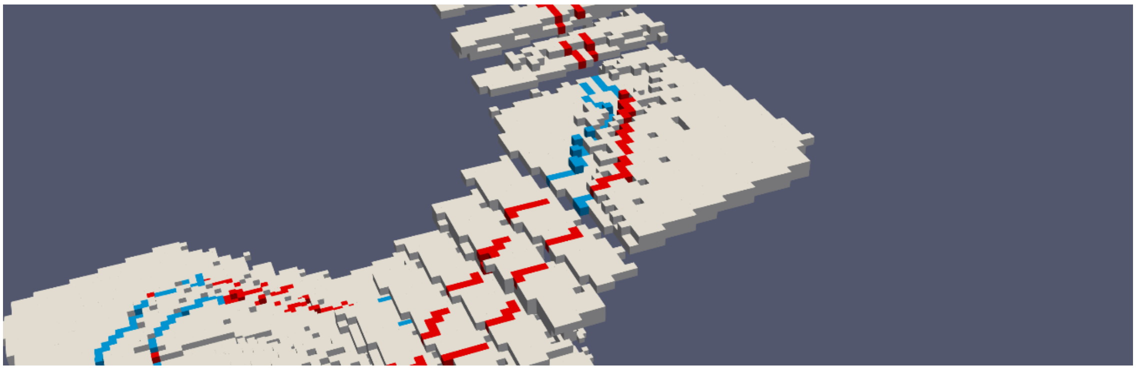

Figure 17: Horizontal floor classified as stair

The discussed improvements are tested on two different point clouds. The results of the previous method and improved method can be seen in table 4 and 5. The percentage of detected risers is higher, and the percentage correctly detected risers has also increased. The percentage of point cloud 2 is a little bit lower because there was a part of a stair where the laser scanner was not held above the stair. The false positives only increased with one. The falls positives that are present are mostly created by the trajectory classification.

Point cloud	Total amount of risers	Detected risers	Falls positives	Percentage detected risers	Percentage correctly detected risers
1: figure 14	68	64	1	94,11	92,65
2: figure 17	206	175	7	84,95	81,55

Table 4: Results of the previous method

Point cloud	Total amount of risers	Detected risers	Falls positives	Percentage detected risers	Percentage correctly detected risers
1: figure 14	68	70	2	102,94	100,00
2: figure 17	206	207	7	100,49	97,09

Table 5: Results of the improved method

These tests are also implemented for the checking of the slope. Since there was only one slope present in the first point cloud, this is the only reference data. As can be seen in table 6 and 7, there were no falls positives and the percentage of correctly detected slope parts is increased with around 12%.

Project	Total amount of slope parts	Detected slope parts	Falls positives	Percentage detected slope parts	Percentage correctly detected slope parts
1: figure 14	34	30	0	88,24	88,24

Table 6: Results of the previous slope detection method

Project	Total amount of slope parts	Detected slope parts	Falls positives	Percentage detected slope parts	Percentage correctly detected slope parts
1: figure 14	34	34	0	100,00	100,00

Table 7: Results of the improved slope detection method

If the laser scanner is held above furnishing elements during the data capture, these objects contain seed voxels. Because there was no change in height in the trajectory, these elements are mostly classified as horizontal floors. These wrongly classified surfaces are detected by a large jump in the height value (the z-axis). If the change in height is larger than the threshold classified in table 3, the points are detected and removed Furniture objects below this threshold are not detected as such, as can be seen in the red boxes, see figure 18.

Figure 18: Classified furniture objects (white), stair seed voxels (red) and horizontal seed voxels (blue). The furniture elements in the red box cannot be detected as furniture objects.

The specified parameters are currently based on two datasets. Further testing is required to find the ideal values that work in all kind of circumstances. In figure 19, the result of the classification check can be compared to the situation after the region growing process without the classification check.

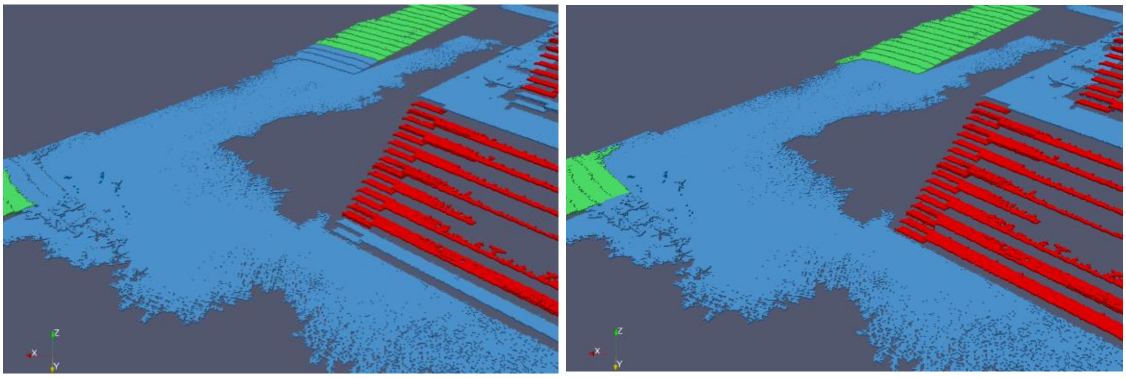

Figure 19: Result after the region growing (left) and the result after the classification check. Stair (red), horizontal (blue) and slope (green)

Small gaps can appear in the floor regions. In this step, these small gaps are filled. If the distance separating two voxels is below two, this is identified as a gap and will be filled with new voxels.

The last step is dedicated to the removal of furniture objects. At this stage, we consider all the remaining voxels above the identified regions as furnishing objects. The voxels above each region is retrieved depending on the size of the agent. This allows to obtain the final walkable voxel space, see figure 20.

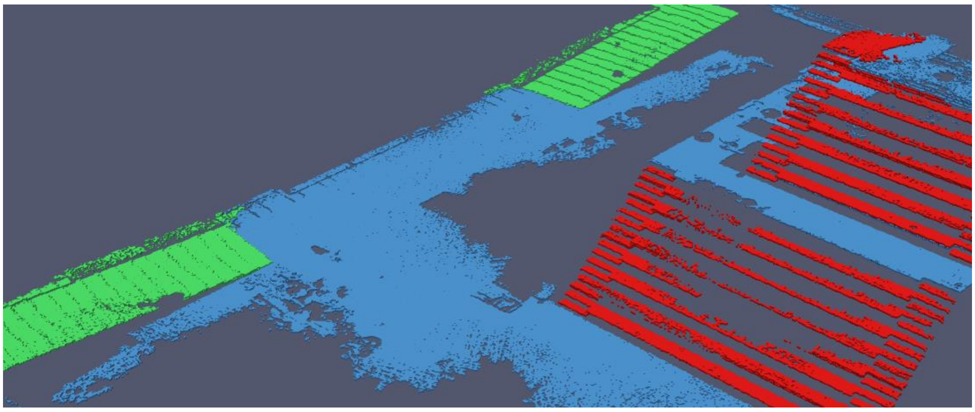

Figure 20: Final walkable voxel space. Stair (red), slope (green), horizontal (blue).

Comparing of the area (m2) of the floor regions

After the last processing step, the final navigable space is detected, and the reliability can be tested. This is done by comparing the surface area resulting from the proposed method to the area found on a CAD model of the same location. Two different spaces were selected to perform the comparison: a corridor and a part of the first floor. The results are illustrated in table 8. The area difference found between the two types of data is around 10%. Visual inspection showed that most voxels are missing in the corner of the rooms or at the sides of corridors. Furthermore, there were points missing where dynamic objects were still present. As discussed earlier, these objects were on the same location during the data capture.

Checking type	Hallway in m2	First floor orange rock in m2
CAD model	74,0	68,0
7,3 cm voxel model	67,7	61,3
Difference between CAD and voxel model in %	-8,5 %	-9,9 %

Table 8: Results between the m^2 of a specific region measured in the proposed method and a CAD model of the same space.

Conclusion

With the current approach it is possible to automatically identify structural elements of building such as floors, stairs, slopes, doors, and furniture objects, based on the point cloud of the building and the trajectory of the MLS device. This method voxelizes the data from the MLS and efficiently identifies different kinds of walkable voxel spaces using spatial properties based on the trajectory. The method makes it possible to create a continuous WS for pedestrians in buildings, including several floors, stairs, and elevations. In comparison to the previous method Staats, Diakité, Voûte, & Zlatanova (2017) the door detection has been implemented and end points of non-flat surfaces are detected better. Because the method can detect and remove dynamic objects at the time of scanning, data capture of the environment can be done even during business hours. It is suitable for any type of room and is free of any constraints related to the configuration (e.g. Manhattan-like oriented) because the internal structure of the building is already encapsulated in the trajectory of the MLS.

This paper proposed an improved method for detection of doors and classifications of stairs and slopes. The detection of doors is based on a vertical and horizontal-check. Implementation results show high percentage of detected doors (115,8 percent). 84.2 percent of these are actual doors, while the rest are falls positives. Because the trajectory crosses a doorway multiple times, it is also possible to have multiple detections. In the examples shown, each of the nine doors were detected at least once. Both vertical-check and horizontal-checks make it possible to detect doors when there is a small height difference between the ceiling and the doorframe. Besides this, the approach requires only two constraints, which is improvement compared to the three proposed by Nikoohemat, Peter, Elberink, & Vosselman (2017).

This paper also presents an improved way of correctly identifying stairs and slopes. The detection of risers, including falls positives, is improved from 89.5 to 101.8 percent. Neglecting the falls positives, the detection of risers is improved from 87.1 to 98.5 percent. The detection of slopes is improved from 88.2 to 100 percent.

These improvements ensures features can be identified more accurately. The detection of doors is a step forward towards identification of rooms.

Future work

Several aspects can be improved and stand as future research. We list some possibilities in the following.

Improving door detection and automating the process

The door detection algorithm can be further improved by adapting the horizontal-check. Currently all the voxels around the current voxel are investigated to specify which give falls positives (besides a wall or other objects). Door frames are usually existing at both sides. Therefore, the approach can be improved by checking if voxels are present of the left and right side of the current voxel. Furthermore, the detection of doors from the graph, now done manually, needs to be automated. To be able to identify the space of separate rooms, the door voxels between the doorframe should be identified.

Identification of dynamic objects, which do not move during the data collection

As discussed in the implementation, dynamic objects that were at the same place during the data capture are not detected with the current method. More research is required to identify these objects. A possible way could be to use reference shapes to detect the pedestrians, which are still present in the voxel model. Another option would be to separate the different time frames.

Generation of a node network

Path planning based on the voxel model requires a lot of time, because there are many voxels to be visited and a lot of possible paths. In many cases it is better to generate a more generic navigable node network. A natural extension of this method could include the automatic generation of navigation networks in an open standard like IndoorGML or other types of node networks.

Generate an indoor map

To view the route through the building from the start point to the endpoint, a map is needed. This map should be produced from the captured data because this will be the data the node network is generated on. In this way, both datasets are describing the same aspects of the indoor space.

References

Anagnostopoulos, I., Pătrăucean, V., Brilakis, I., & Vela, P. (2016). Detection of Walls, Floors, and Ceilings in Point Cloud Data. Construction Research Congress.

Boguslawski, P., Mahdjoubi, L., Zverovich, V., & Fadli, F. (2016). Automated construction of variable density navigable networks in a 3D indoor environment for emergency response. *Automation in Construction, 72*, 115-128. doi:10.1016/j.autcon.2016.08.041

Broersen, T., Fichtner, F. W., Heeres, E. J., Liefde, I. d., Rodenberg, O. B., Verbree, E., & Voûte, R. (2016). Using a linear octree to identify empty space in indoor point clouds for 3D pathfinding. *19th AGILE Conference on Geographic Information Science.*

Brown, G., Nagel, C., Zlatanova, S., & Kolbe, T. H. (2013). Modelling 3D Topographic Space Against Indoor Navigation Requirements. In J. Pouliot, S. Daniel, F. Hubert, & A. Zamyadi (Eds.), *Progress and New Trends in 3D Geoinformation Sciences* (pp. 1-22). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-29793-9_1

Budroni, A., & Boehm, J. (2010). Automated 3D reconstruction of interiors from point clouds. *International Journal of Architectural Computing, 8*, 55-73.

Diakité, A. A., & Zlatanova, S. (2016). Valid Space Description in BIM for 3D Indoor Navigation. *International Journal of 3-D Information Modeling (IJ3DIM), 5*, 1-17. Retrieved from http://EconPapers.repec.org/RePEc:igg:j3dim0:v:5:y:2016:i:3:p:1-17

Díaz Vilariño, L., Boguslawski, P., Khoshelham, K., Lorenzo, H., & 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., Verbree, E., Zlatanova, S., & Diakité, A. (2017). Indoor modelling from SLAM-based laser scanner: door detection. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 355.

Ekholm, A. (1996). A conceptual framework for classification of construction works. *Electronic Journal of Information Tech-nology in Construction ITcon, 1*, 25-50.

Ekholm, A., & Fridqvist, S. (2000). A concept of space for building classification, product modelling, and design. *Automation in Construction, 9*, 315–328.

Fichtner, F. W., Diakité, A. A., Zlatanova, S., & Voûte, R. (2018, 6). Semantic enrichment of octree structured point clouds for multi-story 3D pathfinding. *Transactions in GIS*, 233-248.

GeoSLAM. (2016). *ZEB-REVO User's Manual v1.0.2.* unpublished.

Holenstein, C., Zlot, R., & Bosse, M. (2011, 9). Watertight surface reconstruction of caves from 3D laser data. *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 3830-3837). doi:10.1109/IROS.2011.6095145

ISO. (2011). ISO/FDIS 21542: Building construction -- Accessibility and usability of the built environment.

Józsa, O. (2012). *Analysis of 3d dynamic urban scenes based on lidar point cloud sequences.* Master's thesis, Budapest University of Technology and Economics.

Khoshelham, K., & Díaz-Vilariño, L. (2014). 3D Modelling of Interior Spaces: Learning the Language of Indoor Architecture. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-5*.

Li, G. (2014). *Automatic detection of temporary objects in mobile lidar point clouds.* Master's thesis, University of Twente.

Litomisky, K., & Bhanu, B. (2013). Removing Moving Objects from Point Cloud Scenes. In X. Jiang, O. R. Bellon, D. Goldgof, & T. Oishi (Eds.), *Advances in Depth Image Analysis and Applications: International Workshop, WDIA 2012, Tsukuba, Japan, November 11, 2012, Revised Selected and Invited Papers* (pp. 50-58). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-40303-3_6

Macher, H., Landes, T., & Grussenmeyer, P. (2016). Validation of point clouds segmentation algorithms through their application to several case studies for indoor building modelling. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B5*.

Nikoohemat, S., Peter, M., Elberink, S. O., & Vosselman, G. (2017). Exploiting indoor mobile laser scanner trajectories for semantic interpretation of point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*(4), 355.

PostgresSQL. (2017). 8.9. Spatial Relationships and Measurements. Retrieved from http://postgis.net/docs/manual-dev/ST_ClusterDBSCAN.html

Quintana, B., Prieto, S. A., Adán, A., & Bosché, F. (2018). Door detection in 3D coloured point clouds of indoor environments. *Automation in Construction, 85*, 146-166. doi:https://doi.org/10.1016/j.autcon.2017.10.016

Rabbani, T., Van Den Heuvel, F., & Vosselmann, G. (2006). Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36*, 248-253.

Sirmacek, B., Shen, Y., Lindenbergh, R., Zlatanova, S., & Diakite, A. (2016). Comparison of ZEB1 and Leica C10 indoor laser scanning point clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, III-1*, 143-149. doi:10.5194/isprs-annals-III-1-143-2016

SMART. (n.d.). 5.4.3.45 IfcSpace. Retrieved from http://www.buildingsmart-tech.org/ifc/IFC4/final/html/schema/ifcproductextension/lexical/ifcspace.htm

Staats, B. R., Diakité, A., Voûte, R. L., & Zlatanova, S. (2017, 9). 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.

Suzuki, T., Kitamura, M., Amano, Y., & Hashizume, T. (2010, 10). 6-DOF localization for a mobile robot using outdoor 3D voxel maps. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems,* (pp. 5737-5743). doi:10.1109/IROS.2010.5652983

Thill, J.-C., Dao, T. H., & Zhou, Y. (2011). Traveling in the three-dimensional city: applications in route planning, accessibility assessment, location analysis and beyond. *Journal of Transport Geography, 19*, 405-421. doi:10.1016/j.jtrangeo.2010.11.007

Turner, E., Cheng, P., & Zakhor, A. (2015, 4). Fast, Automated, Scalable Generation of Textured 3D Models of Indoor Environments. *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, 9*, 409-421. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6985553

Vo, A.-V., Truong-Hong, L., Laefer, D. F., & Bertolotto, M. (2015). Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing, 104,* 88-100.

Yan, L., Liu, H., Tan, J., Li, Z., Xie, H., & Chen, C. (2016). Scan Line Based Road Marking Extraction from Mobile LiDAR Point Clouds. *Sensors, 16*, 903. doi:10.3390/s16060903

Zheng, Y., Peter, M., Zhong, R., Oude Elberink, S., & Zhou, Q. (2018, 6). Space Subdivision in Indoor Mobile Laser Scanning Point Clouds Based on Scanline Analysis. *18(6)*.

Zlatanova, S., Liu, L., & Sithole, G. (2013). A Conceptual Framework of Space Subdivision for Indoor Navigation. *Publication of 3D GIS*. Retrieved from https://3d.bk.tudelft.nl/szlatanova/thesis/html/refer/ps/ISA_2013_SZ_LL_GS_final.pdf

Zlatanova, S., Liu, L., Sithole, G., Zhao, J., & Mortari, F. (2014). Space subdivision for indoor applications. *Delft University of Technology, OTB Research Institute for the Built Environment*. Retrieved from http://repository.tudelft.nl/islandora/object/uuid:c3ef4c87-9c35-4d05-8877-a074c3f7fdbf?collection=research