

# POINT CLOUD BASED 3D MODELS FOR AGENT BASED SIMULATIONS IN SOCIAL DISTANCING AND EVACUATION

S. Nikoohemat<sup>a,\*</sup>, P. Godoy<sup>b</sup>, N. Valkhoff<sup>b</sup>, M. Wouters-van Leeuwen<sup>b</sup>, R. Voûte<sup>c</sup>, V. V. Lehtola<sup>a</sup>

<sup>a</sup> Dept. of Earth Observation Science, Faculty ITC, University of Twente, Enschede, The Netherlands - (v.v.lehtola, s.nikoohemat)@utwente.nl

<sup>b</sup> INCONTROL Simulation Software BV, The Netherlands - (paula.godoy, nienke.valkhoff, marlies.wouters)@incontrolsim.com

<sup>c</sup> CGI Nederland BV, The Netherlands - robert.voute@cgi.com

## Commission IV, WG IV/5

**KEY WORDS:** Point Clouds, Indoor 3D Modeling, Agent Based Simulation, Disaster Management, Social Distancing, Evacuation

### ABSTRACT:

Point clouds serve as the raw material for various models, such as Building Information Models (BIM). In this work, we investigate the reconstruction steps needed to create models that can be utilized directly for agent-based simulations. The input data for the reconstruction is captured with an indoor mobile mapping system. To show the prominence of this idea, we run social distancing and evacuation simulations on the reconstructed models. The simulations are run with multiple agents using a vision-based pedestrian model and A\*-based path finding algorithm. The limitations of this approach are discussed. The video of the simulation is shared with the audience.

Link to the video: <https://youtu.be/r2D3IxXt7Ls>

## 1. INTRODUCTION

Social distancing is now more important than ever. Often, the question in exceptional conditions boils down to how can living continue while the fear of illness is present. These questions may be answered with realistic planning using measured data. Such planning allows for optimizing between various factors, such as space, time, and the number of people.

The mobile laser scanning techniques of today can be used to accurately measure the physical indoor environments we live and work in [Lehtola et al., 2017]. The outcome of the scanning is a three-dimensional (3D) point cloud. A point cloud, when reconstructed into a model, can be used as a platform in which to run simulations that predict different situations. We refer to the simulation environment here as a simulation 'sand-box'. These sand-box simulations can be used to find answers to questions related to social distancing and evacuation.

A sand-box model representing a building can be used to predict the outcome of ordinary, extraordinary, or emergency scenarios. In order to demonstrate the multi-functionality of the model we reconstruct and run two different scenarios. First, crowd simulation taking physical distancing<sup>1</sup> into account can, among others, give insights into the maximum capacity of infrastructure for complying with physical distancing rules and test protocols and plans for social distancing measures. The topic is timely, as Covid-19 pandemic has brought a new normal to people's life regarding the distance they should keep to others. Second, we perform a simulation of emergency evacuation, which is a common application of pedestrian simulation models. The interest lies in identifying not only the total time necessary to evacuate a building, but also the locations where bottlenecks can appear.

Results can be obtained quickly. By connecting the modern techniques, the scanning of a building for a point cloud, reconstruct-

\*Corresponding author.

<sup>1</sup>Physical distancing is sometimes referred to as social distancing in the health context.

ing a sandbox model, and running the agent-based simulation in the sandbox is doable within the same day, see Figure 1.

To the best of our knowledge, this is the first work that investigates connecting scanned point clouds to agent-based simulations. We build strongly on the previous work in 3D indoor reconstruction and show that with some additional steps, the automated creation of simulation-suitable models is possible. The works closest to ours consist of (1) indoor reconstruction methods that form a boundary representation or standardized BIM models (e.g. IFC) [Nikoohemat et al., 2020a] and of (2) (commercial) agent-based simulation methods that use the standardized BIM models as simulation sandboxes, e.g. [Sun and Turkan, 2020]. The BIM standards are, however, not made for agent-based simulations but for other, e.g. construction, purposes. Therefore, our contribution lies in investigating the shortcut between a 3D point cloud and a functional agent-based simulation.

The paper is organized as follows. We begin with a review on the related work. Then we elaborate the techniques used to reconstruct sand-box models. In the results section, we present a sand-box model of a fire brigade building in Haaksbergen and the outcome of social distancing and evacuation simulations.

## 2. RELATED WORK

In general, indoor 3D models can be used for multiple simulation purposes. Different simulation engines have different requirements for the sand-boxes they can run simulations in. Computational fluid dynamics simulations use voxelized or finite element models to model, for example, ventilation [Li and Nielsen, 2011], humidity [Ciuman and Lipska, 2018], and thermal user comfort [Buratti et al., 2017]. Lighting simulations use different spatial and temporal resolutions and can cover basically anything between part of a room to a whole building [Baloch et al., 2018]. Acoustics simulations use detailed geometry but also surface material properties of e.g. a concert hall [Vorländer, 2013].

Building information models (BIM) are a set of different type of models used in architecture, engineering and construction (AEC).

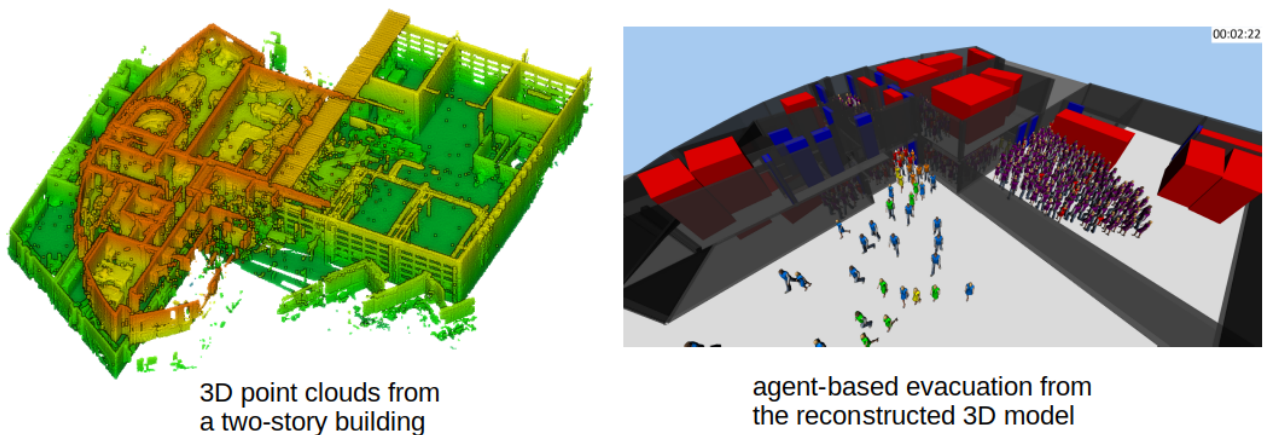


Figure 1: (a) Perspective view of the 3D point cloud of a building with two floors from which we compute (b) a reconstructed sandbox model that we use in agent-based simulation.

BIM models that comply with industry foundation class (IFC) standards are referred to as IFC models. Agent-based simulations have human sized objects that are moved on floor surfaces. These surfaces are 2D manifolds that have a topology that connects all the different floors of a building into one model. Agent-based simulations can be run on BIM-based sand-boxes for evacuation planning [Marzouk and Al Daour, 2018, Sun and Turkan, 2020, Xu et al., 2020] and epidemic risk assessment [Harweg et al., 2020].

There are two ways to use BIM models for simulation purposes: 1. Using an existing BIM model which may not reflect all the recent changes of the building and it is as-design. 2. Create a new BIM model from new measurements which is referred to as scan-to-bim and as-is in the most literature [Bassier and Vergauwen, 2019, Macher et al., 2017, Nikoohemat et al., 2020b]. Scan-to-bim models can be created manually by bim experts or (semi-) automatically by computer algorithms. In the recent years, many researchers proposed smart solutions such as cell-decomposition, constructive solid geometry and shape grammar to create 3D models and BIM from point clouds [Mura et al., 2016, Ochmann et al., 2019, Nikoohemat et al., 2018, Tran H. et al., 2019]. Some of the approaches are limited to specific building structures such as Manhattan World or 2.5D buildings [Ikehata et al., 2015, Turner et al., 2015], others offer more flexibility in terms of adoption of multi-story buildings and complex structures [Ochmann et al., 2019, Bassier and Vergauwen, 2020, Tran and Khoshelham, 2020]. For our simulation purposes, we are focusing on reconstructing the volumetric walls by exploiting the topology between the permanent structures similar to [Nikoohemat et al., 2020b, Ochmann et al., 2016], detecting doors [Nikoohemat et al., 2018, Flikweert et al., 2019] and modeling multi-story buildings [Macher et al., 2017, Ochmann et al., 2019].

Hence, for rapid planning purposes, it makes sense to study how point clouds can be automatically converted into simulation sand-boxes for agent-based simulations.

### 3. INDOOR 3D RECONSTRUCTION METHODOLOGY

The point clouds for our agent based simulation are collected by indoor mobile laser scanners (IMLS). With mobile laser scanners, we are capable of scanning large buildings in a short time and then use the data firstly to create a 3D model and then import the model into the simulation software. To use the model for the

simulation, we need to understand what objects from the indoor environments are required. Permanent structures (walls, floor, ceilings), doors and furniture are detected in the point clouds and reconstructed in the final model. Note that in this work, the exact semantic of each furniture is not necessary and the furniture plays the role of obstacles in the simulation.

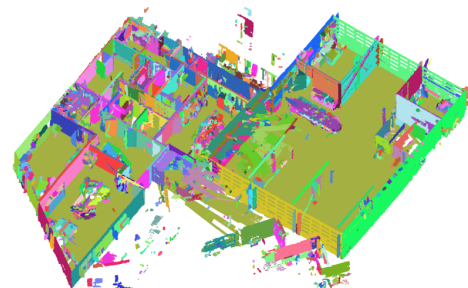


Figure 2: Surface growing planar segmentation. Each segment is represented by a vertex in the graph. Colors are random

#### 3.1 Reconstruction of Walls

The most important items in our agent based modeling approach are the layout of the rooms and how they are connected. This narrows down our reconstruction method to correctly modeling walls, navigable space and doors. Walls provide the layout of the space, furniture provide the non-navigable space and doorways provide the information of room connections.

The input point cloud is noisy and there is clutter (furniture and people) in it which makes the detection of walls cumbersome. We use a piece-wise planar reconstruction to tackle this problem but first the levels of the buildings should be separated from the input point clouds. Each level is a horizontal space which encapsulates rooms and furniture in the same level. Since for this data the trajectory from the MLS scanner is available, by using the method described in [Nikoohemat et al., 2018], the points associated with the trajectory from the same level is collected which results in level separation. Then each level is reconstructed separately. A z-histogram can be used to separate the levels for build-

ings scanned with a terrestrial laser scanner [Oesau et al., 2014], since a trajectory is not available.

**3.1.1 Plane fitting and adjacency graph:** Our approach exploits planar primitives since they are the dominant geometry in man-made structures. A surface growing method [Vosselman et al., 2004] is applied to detect the planar surfaces (see Figure 2). Similar to [Nikoohehmat et al., 2020b], an *adjacency graph*  $G_{adj} = (V, E)$  is created from the planes where each vertex  $v \in V$  represents a plane and each edge  $e \in E$  encodes the relation between two adjacent planes. This formulation allows us to avoid Manhattan-World assumptions. Note, at the stage of creating the graph there is no perception of semantics and hence planes can belong to any object. Since, planes are infinite primitives, we fit an enclosing minimum rectangle in the 3D space to the supporting points of the plane (segment) and accordingly some properties are calculated to encode the geometry of planar segments in the graph. From now on each plane or segment or minimum rectangle share the same properties. Therefore each plane, i.e., vertex  $v \in V$  in the graph inherits following properties which are calculated from the supporting points

$$v = v(N_{\text{points}}, \text{min}_{\text{rectangle}}, \mathbf{n}, \theta, \text{centroid}, \text{label}) \quad (1)$$

where the  $N_{\text{points}}$  is the number of supporting points of the plane, the  $\text{min}_{\text{rectangle}}$  is the enclosing rectangle of the segment in the 3D space where the plane is fitted, the  $\mathbf{n}$  is the normal vector of the plane, the  $\theta$  is the angle between the normal vector and positive direction of the  $Z$ -axis and the  $\text{centroid}$  is the centroid of the supporting points. The  $\text{label}$  can be either of *wall*, *floor*, *ceiling* or *clutter* which are decided later. Likewise, each  $e_i \in E$  is created from a pair of  $(v_i, v_j)$ . Similar to vertices, each edge also has properties which encode the relation between two planes and their supporting segments:  $e(\text{dist}_{\text{adj}}, \text{angle}_{\text{adj}}, \text{type}_{\text{adj}})$  where  $\text{dist}_{\text{adj}}$  is the distance between closest points of two segments,  $\text{angle}_{\text{adj}}$  is the angle between normal vectors of two planes, and  $\text{type}_{\text{adj}}$  is the type of relation between two planes which we explain later. An edge  $e \in E$  is added to the graph  $G$  if  $\text{dist}_{\text{adj}}$  is less than a *threshold* (e.g., 0.1 meter). Otherwise, two planes are not considered adjacent. Similarly, if two planes are not intersecting, meaning they are coplanar or parallel (= parallel normal vectors) then an edge between them is not established. The  $\text{type}_{\text{adj}}$  has an important role in defining the semantic of the segment (e.g., wall or clutter). To define the type of an edge, previously all *planes* ( $v \in G$ ) are classified to *almostHorizontal* and *almostVertical* given a threshold angle of  $45^\circ$ . An edge can have three types:

for each  $e \in E$ :

$\text{type } e_{(i,j)} == \text{"wall - ceiling"}$  iff  $v_i$  is *almostHorizontal* and  $v_j$  is *almostVertical* and  $\text{centroid}(z)_{v_i} > \text{centroid}(z)_{v_j}$

$\text{type } e_{(i,j)} == \text{"wall - floor"}$  iff  $v_i$  is *almostHorizontal* and  $v_j$  is *almostVertical* and  $\text{centroid}(z)_{v_i} < \text{centroid}(z)_{v_j}$

$\text{type } e_{(i,j)} == \text{"wall - wall"}$  iff  $v_i$  is *almostVertical* and  $v_j$  is *almostVertical*

From the given formulation you can see the plane of a ceiling candidate ( $v_{\text{almostHorizontal}}$ ) should be above the wall candidate and in reverse for a floor candidate. Furthermore, note a "ceiling - ceiling" or a "floor - floor" relation does not exist because if two ceilings (or two floors) are adjacent and coplanar, their planes are merged in an early stage. If they are not coplanar, like two sloped ceilings next to each other, then the edge type is not necessary and is redundant information because the relation

to adjacent wall candidates defines whether the object is clutter or the ceiling. Next step is to label each vertex based on the number of edges and the type of edges.

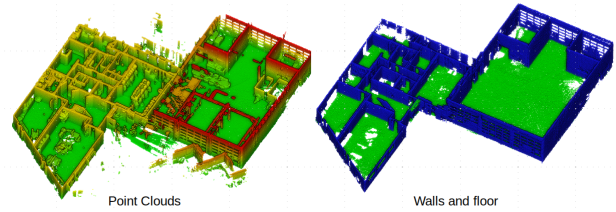


Figure 3: Left: Point clouds of the first floor, Right: detected walls and floor

**3.1.2 Wall detection:** Each vertex  $v \in G$  is representing a plane and its associated segment which obtains a label  $\text{label}(\text{wall}, \text{floor}, \text{ceiling}, \text{clutter})$ . The degree of each vertex is the number of connected edges,  $\text{deg}(1, n-1)$  where  $n$  is the number of vertices in the graph. Given an adjacency graph, the degree and the type of edges for each vertex are analyzed to label the vertex. Segments (vertices) belonging to ceilings (or floors) are easy to recognize as they are connected to several *almostVertical* planes, for example a ceiling vertex (*almostHorizontal*) should have two or more "wall - ceiling" edges and no "wall - wall" edge. Therefore, objects like tables or planar objects near the ceilings (e.g. beams) are classified as clutter since they have "wall - wall" edge. Normally objects like cabinets and cupboards form a horizontal-vertical relation similar to the ceiling-wall structure. To avoid including these segments as ceiling candidates, the *almostHorizontal* segment is labeled as a ceiling candidate if it is not covered with another horizontal segment above it.

A vertex obtains a *wall* label if it is *almostVertical* and it has at least one "wall - ceiling" edge. Note that a "wall - floor" edge for a wall is not necessary because often walls are occluded with furniture near the floor. Rest of the vertices obtain the clutter label (furniture, people, noise). Figure 3 shows the detected walls and floor from the point clouds. In the figure you can see the clutter near the ceiling and most of the furniture in the middle of the rooms are removed automatically.

Note, because of the presence of clutter (furniture and people) part of the walls are occluded, hence there is a gap in some part of the data which causes some walls not to be connected to each other or to the floor. Our method is able to tackle this problem, otherwise the simulation can perform incorrectly where walls are not connected. As long as there is a slight connection between the wall and the ceiling, partially scanned walls are detected. In the section 3.2, we explain how modeling connects the wall, floor and ceiling to create a watertight model.

## 3.2 Modeling Walls

Modeling contains a piece-wise planar detection and creation of enclosing oriented rectangle of permanent structures (walls, floor and ceilings) from point clouds and connect them to reconstruct a 3D model. From the previous step, segments belonging to permanent structures are selected for modeling. An operator quickly checks the automated labeled structures to make sure there is not a mislabeled object. If piece of a clutter is mislabeled as a wall the label is changed. This is a sanity check and should not take more than several minutes.

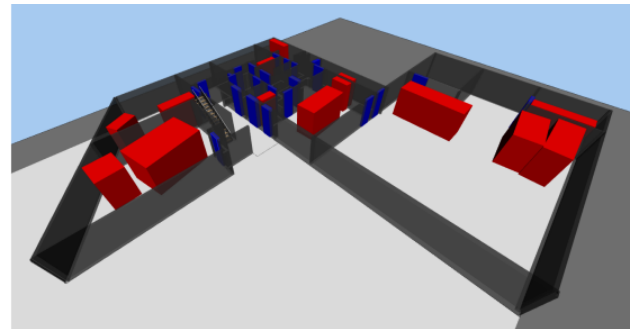
For simulation purposes, walls are the most important structures and they should be connected to create a topologically correct model. Since walls are occluded with furniture, some of the walls are not connected to each other or to slabs (floors and ceilings). Not connected walls can be fixed by intersecting planes of a pair of segments and obtaining the intersection line. Then the enclosing rectangle of each segment is extended to the intersection line given a  $d_{\text{extension}}$  threshold. The threshold guarantees that (almost) perpendicular walls on both sides of a narrow hallway are not extended to each other. As a result of this operation, the minimum rectangle and the centroid of the segments are updated but the normal vectors remain the same.

Surfaces of interior walls are normally scanned from both sides. Therefore, there are two sides of a wall which form a wall structure. First, we should identify which faces belong to the same wall and merge them as one wall object (one segment). Each wall is a parametric object with properties inherited from the segment and its associated plane. For each pair of wall segments, if two adjacent segments are almost coplanar or parallel meaning the angle between their normal vectors is less than  $\theta (< 5^\circ)$  then they form a volumetric wall and merge into one wall object. The thickness of the created volumetric wall is inherited from the distance of two planes. Accordingly, the parameters of the enclosing cuboid and the plane are updated. Note that instead of the enclosing rectangle an enclosing oriented cuboid is created. The normal vector of each face of the b-rep is oriented inside the room. For walls which are scanned from one side such as facade walls a fixed thickness is introduced and the wall is offset in the opposite direction of the normal vector. The curved walls are modeled as small planar walls (see Figure4, the second floor).

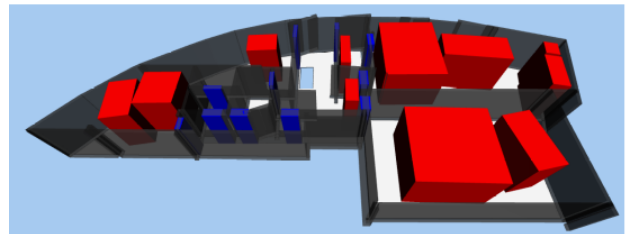
### 3.3 Door and Furniture Detection

Doors are necessary for our simulation application. Since we use a mobile laser scanner for scanning, a trajectory of the scanning route is created alongside the main point clouds. A trajectory simply is a collection of discrete points representing the mobile laser scanner positions. Any door that the scanner operator went through it during the scanning can be detected. Similar to [Nikoohemat et al., 2020b], we use the intersection between the scanner trajectory and the wall to identify the position of doors. Then a door template with fixed height and width is fitted to the intersection location. The normal vector of the door plane and the thickness of the door are inherited from the wall. The door lower edge is intersecting the floor plane to make sure the door is connected to the ground. This approach detects at least one door per scanned room, for example if the operator enters from one door and exits the other door both doors can be detected. Furthermore, if the door is closed after the operator goes through it which appears as a closed door in the data, still it can be identified because it is intersected by the trajectory. If a door is closed or the room is not accessible during the scanning, then that space can not be accessible for simulation as well.

In the labeling step, furniture is labeled as clutter including noise and people. For simulation purposes, the navigable space is required and furniture is added to the model as obstacles (non-navigable). From the segments which are labeled as clutter, large adjacent segments which have overlap with the floor are selected. Since our agents for simulation are people and they move on the floor, hence objects like lamps or clutter near the ceiling or on the walls are not included in the model. This way, we keep the model uncluttered for agents. Finally, an oriented bounding box (cuboid) is created from each cluster of furniture and is added to the model.



First Floor



Second Floor

Figure 4: 3D model with doors (blue) and furniture as obstacles (red boxes)

Element type	Format	Details
Walls	OBJ	B-Rep, one file for each floor
Doors	OBJ	B-Rep (cuboid), one file per floor
Obstacles	OBJ	B-Rep (cuboid), one file per floor
Staircases	OBJ	manually added, future work

Table 1: The format used to represent the reconstruction result as a sandbox. B-rep means boundary representation (see text), floor is referring to building levels.

### 3.4 Reconstructed sandbox model

The reconstructed sandbox model consists of four types of OBJ-format files, as detailed in Table 1. The format uses boundary representation (B-rep) for the object modeling (e.g. a wall instance), which is commonly used in CAD (computer aided design) and solid modeling. Walls span the size of the sandbox, but also limit the movement within it. After the placement of walls, door objects are used to create openings on these walls, as they would otherwise create isolated rooms. Finally, obstacles are placed to further limit the walkable space and staircases are placed to connect multi-floor environments. In this work, the staircase between the two floors is placed manually.

The simulation (e.g. evacuation) relies on these layers that we created from point clouds. A more robust 3D model reconstruction method results in less work in the simulation step and a correct simulation results. Four mentioned layers from the 3D model of point clouds are used for the simulation and ceilings are not required.

## 4. SIMULATION METHODOLOGY AND SETUP

For the agent-based simulations, a commercial software<sup>2</sup> is used. It is designed for pedestrian simulations in complex multi-layered

<sup>2</sup>INCONTROL simulation software Pedestrian Dynamics, <http://incontrolsim.com>

infrastructures [Van Toll et al., 2011], and can be used to evaluate, for example, the comfort and safety of up to 100000 agents under various environmental conditions. The functional principle of the simulation is briefly explained next, with more detail in [Bijsterbosch et al., 2012].

The simulation starts with multiple agents placed in different locations. During the simulation, the agents find a path from their current position to another a-priori given position in the environment. The model uses the Explicit Corridor Map (ECM) [Geraerts, 2010] as a navigation mesh, which is a subdivision of the entire walkable space into connected polygonal areas, see Fig 5b. It is essentially a graph consisting of vertices and edges. The edges of the ECM form the medial axis: a set of curves describing the middle of the walkable space, see Fig. 5.

When planning a path to a goal location, an agent tries to find a route along the network's edges, using a modified A\* algorithm. Because each network node is associated to the closest points on obstacles, the resulting route is actually a corridor: a set of polygons and circle segments (drawn around the closest points), describing the free space that the agent can use around the route. The algorithm called Indicative Route Method (IRM) [Karamouzas et al., 2009] smoothly steers an agent through a corridor while following an indicated path. To avoid agents running into one another, the method uses a vision-based collision avoidance algorithm based on simple behavioral heuristics [Moussaïd et al., 2011].

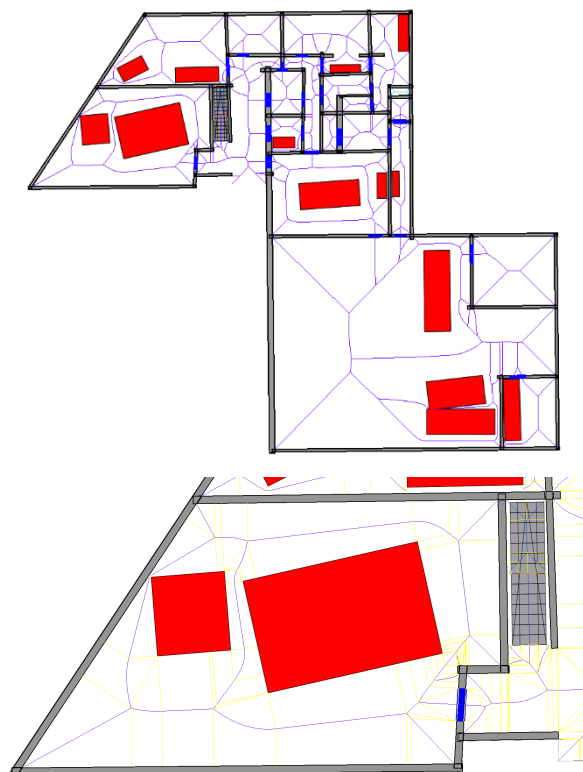


Figure 5: (a) The simulation sandbox. (b) A close-up that shows the medial axes (lines in blue) and the explicit corridor mesh that utilizes the equidistant lines to obstacles (in yellow).

#### 4.1 Physical distancing scenario

This scenario is simulated with 30 agents, which move between the different rooms trying not to trespass each other's private space. The implemented rules for physical distancing in this work

follow the regulation of The Netherlands, meaning that a minimum distance of 1.5 m is kept between agents at all times. In the 2D representation, the agents are characterized by their body radius (small disk) and their private radius (large disk), where the body radius provides the strict interaction with walls and other agents, and the private radius defines the physical distancing that should be avoided by others. By simulating physical distancing, one can, for example, identify the bottleneck areas where physical distancing rules are most likely to be violated, as well as the duration in time how long these violations last, see Figure 6 (right).

#### 4.2 Evacuation scenario

The scenario is built with 300 agents distributed over the rooms proportionally to the surface area of each room. All agents simultaneously start to evacuate the building, for instance due to a fire or a natural disaster. The time taken by agents to evacuate the building is assessed and the different rooms of the building can be compared to identify where the highest delays and densities occur during evacuation conditions. Such information can assist decision making of where to add emergency doors or determining the maximum number of people allowed, not in the whole facility, but in each room of the facility (see Figure6-left).

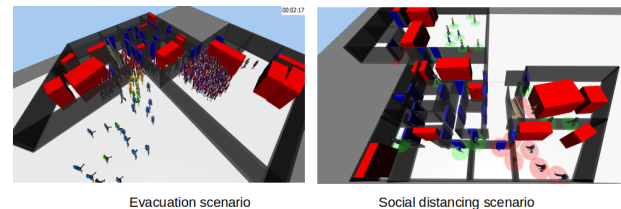


Figure 6: 3D view of evacuation simulation and social distancing. The evacuation figure shows the bottleneck in the first floor for the evacuation is the door in the big hall where a lot of agents are gathered and consequently the evacuation time increases. In the right image, the red circles show where the 1.5 distancing are violated.

## 5. RESULTS AND EVALUATION

### 5.1 Data

Our 3D model reconstruction method is tested on the point cloud of a two-storey building with 27 rooms. The point cloud properties present different challenges to our method such as a non-Manhattan World structure, heavy clutter, curved walls, different room sizes with arbitrary orientations, and curtain walls with glass surfaces. The point cloud is collected with a pushing-cart mobile laser scanner and the point noise is between 3-5 cm. A trajectory is also provided with the point cloud and it is used for the door detection.

### 5.2 Parameter selection and model accuracy

The point cloud is sub-sampled to less than 10 million points to accelerate the surface growing process which results in a point spacing of 2cm and nearly 1000 segments for both floors. The parameters are selected according to [Nikoohemat et al., 2020b]. For the surface growing, a minimum distance to surface of 8 cm is used which for 5 million points per floor takes less than 2 minutes on an Intel Core i7 (2.2 GHz) with 16 GB RAM. The adjacency graph is created using a 10 cm proximity distance between segments and small segments with less than 500 supporting

points are not included. After automatic labeling of permanent structures, the operator corrects the label of some of the mislabeled walls and slabs which was equal to 6percent of the total number of planar segments in the graph. Most of the corrections occurred near the ceiling because of the clutter which covered the ceiling. Since our application is for evacuation purposes the enclosure of spaces are important. So the operator checks that the walls belonging to the facade of the building are correctly present in the data. Interior walls could be wrong and we do not check every individual room as this is in the contrary to our purpose of automatic modeling and simulation. We do not spend more than 10 minutes for the manual inspection and correction. Experimentally, if the data is noisy, the planar segmentation can be erroneous and consequently the time of correction of labels increases. If some of the interior walls are missing it results in bigger rooms (two smaller rooms will be connected). Although such cases result in a less correct model, for creating the simulation is not a problem. To make sure the spaces are enclosed by walls and slabs in section 3.2 an automatic extension operation was performed. A threshold of 1.0 meter is applied for  $d_{\text{extension}}$ . Finally, we check the accuracy of the permanent structure detection using a point-wise comparison with the manually labeled points. The precision, recall, and F1-score for wall labeling are 0.90, 0.96, and 0.93, respectively. In total, a correct model is achieved for 27 out of 30 doors and 24 out of 27 rooms. The missing doors are added by the operator just in case a room does not have a door. Otherwise, a door per room is enough to run the simulation.

### 5.3 Sandbox model requirements and reconstruction

The agent-based simulation sets some requirements to the model that we obtain from the reconstruction. These are our observations from trying to run simulations on the reconstructed sandbox model.

- Doors need to be detected correctly. If a room does not have any doors, this means the agents cannot leave the room, and this is detected and flagged as an error in the simulation process and in the next iteration the door is added.
- The walls need to be watertight at the cross-sections of wall elements. Otherwise, artificial holes emerge and the agents will use these to find new routes.
- Only those obstacles that block movement must be taken into account. For this method there is no limitation of vertical walls and horizontal floor and ceiling. Any arbitrary orientation is supported. Curved walls are represented with many small planar surfaces, see the second floor in Figure4.
- Stair cases must be correctly placed to connect multi-floor environments. As with doors, missing stair cases are detected during the start of the simulation if some floors are disconnected from the exits. Also, the correct slope for the stair case is important as it is used to slow down the moving agents.

The obstacles were well placed automatically. However, as mentioned, we encountered problems with door detection. Therefore, three missing doors were inserted manually in a 3D modeling software.

We had to add the stair case manually to the model using a 3D modeling software, as adding the stairs automatically is part of the future work.

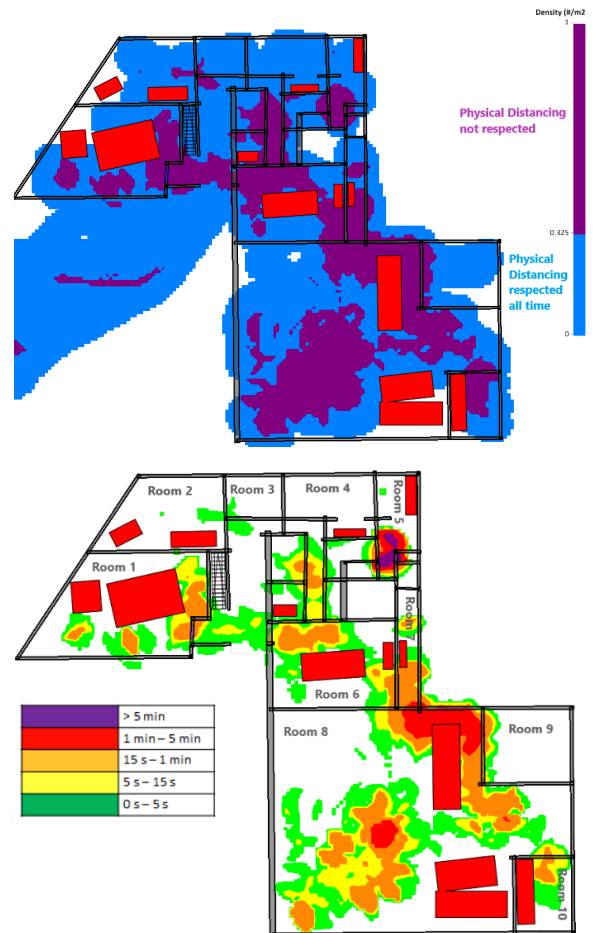


Figure 7: (a) The map of Respected Social Distancing., (b) Social Distancing Timing.

### 5.4 Simulation Scenarios

For the physical distancing scenario, the interest lies in identifying the areas where the minimum distancing is most likely to be violated. Therefore, we investigate the density of people per square meter, using two threshold values. Given the physical distance of 1.5 m and an agent radius of 0.239 m, the threshold for the density level is first set to 0.3254 persons/m<sup>2</sup>. In Figure 7, we see an abundance of violations of the physical distance in the multiple narrow corridors formed by the obstacles and the tight hallways, but also that the duration of most of these violations is very short (from 5 to 15 seconds). The bottlenecks can be pinpointed by the door between Rooms 7 and 8, as well as the door of Room 5.

In the simulation of the evacuation scenario, the highest densities (agents/m<sup>2</sup>) and delays (s) are experienced when the agents evacuate from Room 8, see Figure 8 and Figure 9. The number of agents placed inside the building is purposefully very large to highlight the properties of the sand box model, for example, we can see that the flow of agents from Room 8 saturates into a steady state so that a maximum flow velocity for this bottleneck can be determined. The maximum delay experienced exiting from this room is 3 minutes, which is 4.5 times higher than the maximum delay observed in Room 2, see Figure 8 (right).

Furthermore, by looking at the time series of the density at the exit door of Room 8 in Figure 8 (left), we observe that the congested state endures for a prolonged time. This indicates unsafe conditions by that door, since people pushing against each

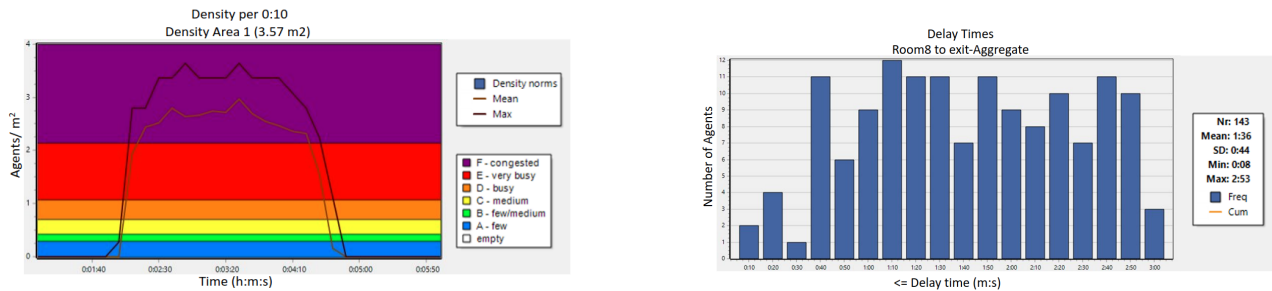


Figure 8: Left: Evacuation Density Area Crowded, Right: Evacuation Delay Distribution

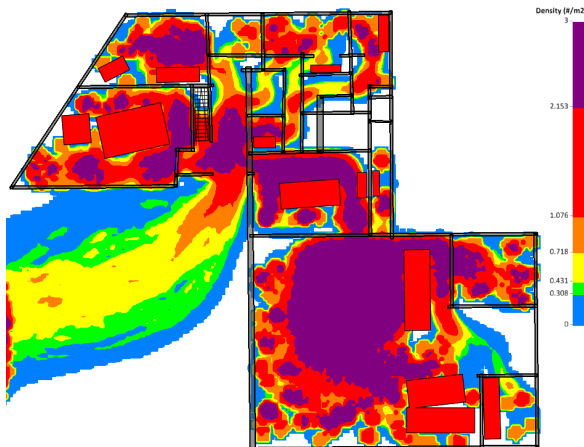


Figure 9: The result of evacuation density map. The colors are based on the number of people per square meter. The purple areas are dense area during the evacuation.

other in a dense crowd may cause panic and lead to unstable flow regimes, which can ultimately result in injuries and medical emergencies due to crushing or trampling.

## 6. DISCUSSION AND FUTURE WORK

The 3D modeling method from point clouds shows the steps of how to reconstruct required layers for agent-based simulation tasks. For our simulation scenarios, wall layout and doors are important layers during the modeling process. Our method successfully creates a 3D model with 90 percent overall accuracy for the simulation task. The reconstruction algorithm fails to model the curved wall in the second floor properly and some ill-posed segments are created. Although, this does not jeopardize the simulation process as long as there is no gap (hole) between walls. The challenge in modeling from point clouds lies in detecting and reconstruction of the permanent structures (walls, floors, doors). While the floor is easy, walls and doors can pose a problem in some cases, for example, walls which are partially scanned and heavily cluttered. The pair-wall intersection and extension handle the missing data perfectly and make sure the walls are not disjoint. However, some of the walls are mislabeled ( 6 percent of the total segments) which can be relabeled with visual inspection. We also noticed the thickness of the cuboid of some of the doors need to be increased to ensure that an opening is connecting two rooms so the navigation graph is not interrupted in doorways. Another interesting outcome of our experiment is that rooms without an opening or the missing doors can be flagged during the evacuation scenario and the model can be returned to the reconstruction step for further corrections. This is similar to solution suggested by [Nikoohemat et al., 2020a] for consistency

check of 3D models from point clouds. However, this can not be checked for missing walls or mislabeled walls.

Some of the created cuboids of furniture are intersecting walls. This unintended collision is because we generalize the shape of the furniture (chairs and table) to a 3D bounding box which near the walls causes an undesired intersection. However, this issue does not hurt the simulation results.

The path finding has multiple a-priori assumptions, which are justifiable but should be used consciously in the future work to avoid artefacts. For instance, in our work, the agents do know how to get to their destination. In other words, the agents do not get lost. This may be because they have had evacuation training or because the environment has enough directional signs to guide the agents. However, it is easy to think of cases where people do temporarily get lost, for example in airports. The motion speed of agents also builds on priors. In principle, A\* would allow for defining distinct areas where moving is slower or more complicated or more favorable, see e.g. [Lehtola et al., 2019]. Here, however we have treated all navigable space as similarly walkable (or runnable) and, except for the stairs that we added manually, there are no areas where moving is slower or more complicated, such as crawling spaces, or more favorable such as individually preferred routes. These could be automatically created from fusing point clouds to other data.

## 7. CONCLUSION

The scanning, reconstruction, and simulating can each be done quickly with the techniques we used. It is already possible to obtain results during the same day, which may spark societal and commercial interest. We present physical distancing and evacuation scenarios as examples of the multi-functionality of the used techniques for different simulation purposes. Simulation results can be used for enlightened decision making. The limitations of the presented technique are directly connected to scanning. Should the scan be complete (incomplete), the reconstruction will also be so, and the simulation will run on a sandbox that does (does not) accurately represent the reality. If the scan from a building can be trusted to be complete, there is no need to require standardized BIM models for simulation purposes, as scanned point clouds suffice. Finally, we note that should there be deficiencies in the point clouds, these can oftentimes be overcome by the information obtained from the scanning trajectory, especially if the trajectory goes through the doorways.

## ACKNOWLEDGEMENTS

The authors would like to thank the Fire Brigade Haaksbergen for making their buildings available for the test and data collection.

## REFERENCES

- Baloch, A. A., Shaikh, P. H., Shaikh, F., Leghari, Z. H., Mirjat, N. H. and Uqaili, M. A., 2018. Simulation tools application for artificial lighting in buildings. *Renewable and Sustainable Energy Reviews* 82, pp. 3007–3026.
- Bassier, M. and Vergauwen, M., 2019. Clustering of Wall Geometry from Unstructured Point Clouds Using Conditional Random Fields. *Remote Sensing* 11(13), pp. 1586.
- Bassier, M. and Vergauwen, M., 2020. Unsupervised reconstruction of Building Information Modeling wall objects from point cloud data. *Automation in Construction* 120, pp. 103338.
- Bijsterbosch, J., van Toll, W. and Pitsch, H., 2012. About the pd crowd simulation framework. In: *Proceedings of the Winter Simulation Conference*, pp. 1–6.
- Buratti, C., Palladino, D. and Moretti, E., 2017. Prediction of indoor conditions and thermal comfort using cfd simulations: A case study based on experimental data. *Energy Procedia* 126, pp. 115–122.
- Ciuman, P. and Lipska, B., 2018. Experimental validation of the numerical model of air, heat and moisture flow in an indoor swimming pool. *Building and Environment* 145, pp. 1–13.
- Flikweert, P., Peters, R., Díaz-Vilarino, L., Voûte, R. and Staats, B., 2019. Automatic extraction of a navigation graph intended for IndoorGML from an indoor point cloud. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci* 4, pp. 271–278.
- Geraerts, R., 2010. Planning short paths with clearance using explicit corridors. In: *2010 IEEE International Conference on Robotics and Automation, IEEE*, pp. 1997–2004.
- Harweg, T., Bachmann, D. and Weichert, F., 2020. Agent-based simulation of pedestrian dynamics for exposure time estimation in epidemic risk assessment. *arXiv preprint arXiv:2007.04138*.
- Ikehata, S., Yang, H. and Furukawa, Y., 2015. Structured indoor modeling. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1323–1331.
- Karamouzas, I., Geraerts, R. and Overmars, M., 2009. Indicative routes for path planning and crowd simulation. In: *Proceedings of the 4th International Conference on Foundations of Digital Games*, pp. 113–120.
- Lehtola, V., Montewka, J., Goerlandt, F., Guinness, R. and Lensu, M., 2019. Finding safe and efficient shipping routes in ice-covered waters: a framework and a model. *Cold regions science and technology* 165, pp. 102795.
- Lehtola, V. V., Kaartinen, H., Nüchter, A., Kaijaluoto, R., Kukko, A., Litkey, P., Honkavaara, E., Rosnell, T., Vaaja, M. T., Virtanen, J.-P. et al., 2017. Comparison of the selected state-of-the-art 3d indoor scanning and point cloud generation methods. *Remote sensing* 9(8), pp. 796.
- Li, Y. and Nielsen, P. V., 2011. Cfd and ventilation research. *Indoor Air* 21(6), pp. 442–453.
- Macher, H., Landes, T. and Grussenmeyer, P., 2017. From Point Clouds to Building Information Models: 3D Semi-Automatic Reconstruction of Indoors of Existing Buildings. *Applied Sciences* 7(10), pp. 1030.
- Marzouk, M. and Al Daour, I., 2018. Planning labor evacuation for construction sites using bim and agent-based simulation. *Safety science* 109, pp. 174–185.
- Moussaïd, M., Helbing, D. and Theraulaz, G., 2011. How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences* 108(17), pp. 6884–6888.
- Mura, C., Mattausch, O. and Pajarola, R., 2016. Piecewise-planar Reconstruction of Multi-room Interiors with Arbitrary Wall Arrangements. *Computer Graphics Forum* 35(7), pp. 179–188.
- Nikoohemat, S., Diakit , A. A., Lehtola, V., Zlatanova, S. and Vosselman, G., 2020a. Consistency grammar for 3d indoor model checking. *Transactions in GIS*.
- Nikoohemat, S., Diakit , A. A., Zlatanova, S. and Vosselman, G., 2020b. Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management. *Automation in Construction* 113, pp. 103109.
- Nikoohemat, S., Peter, M., Oude Elberink, S. and Vosselman, G., 2018. Semantic Interpretation of Mobile Laser Scanner Point Clouds in Indoor Scenes Using Trajectories. *Remote Sensing* 10(11), pp. 1754.
- Ochmann, S., Vock, R. and Klein, R., 2019. Automatic reconstruction of fully volumetric 3D building models from oriented point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 151, pp. 251–262.
- Ochmann, S., Vock, R., Wessel, R. and Klein, R., 2016. Automatic reconstruction of parametric building models from indoor point clouds. *Computers & Graphics* 54, pp. 94–103.
- Oesau, S., Lafarge, F. and Alliez, P., 2014. Indoor scene reconstruction using feature sensitive primitive extraction and graphcut. *ISPRS Journal of Photogrammetry and Remote Sensing* 90, pp. 68–82.
- Sun, Q. and Turkan, Y., 2020. A bim-based simulation framework for fire safety management and investigation of the critical factors affecting human evacuation performance. *Advanced Engineering Informatics* 44, pp. 101093.
- Tran, H. and Khoshelham, K., 2020. Procedural Reconstruction of 3D Indoor Models from Lidar Data Using Reversible Jump Markov Chain Monte Carlo. *Remote Sensing* 12(5), pp. 838. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.
- Tran H., Khoshelham K., Kealy A. and D az-Vilari o L., 2019. Shape Grammar Approach to 3D Modeling of Indoor Environments Using Point Clouds. *Journal of Computing in Civil Engineering* 33(1), pp. 04018055.
- Turner, E., Cheng, P. and Zakhori, A., 2015. Fast, Automated, Scalable Generation of Textured 3D Models of Indoor Environments. *IEEE Journal of Selected Topics in Signal Processing* 9(3), pp. 409–421.
- Van Toll, W., Cook, A. F. and Geraerts, R., 2011. Navigation meshes for realistic multi-layered environments. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE*, pp. 3526–3532.
- Vorl nder, M., 2013. Computer simulations in room acoustics: Concepts and uncertainties. *The Journal of the Acoustical Society of America* 133(3), pp. 1203–1213.
- Vosselman, G., Gorte, B. G., Sithole, G. and Rabbani, T., 2004. Recognising structure in laser scanner point clouds. *International archives of photogrammetry, remote sensing and spatial information sciences* 46(8), pp. 33–38.
- Xu, Z., Wei, W., Jin, W. and Xue, Q.-r., 2020. Virtual drill for indoor fire evacuations considering occupant physical collisions. *Automation in Construction* 109, pp. 102999.