

Direct Use of Indoor Point Clouds for Path Planning and Navigation Exploration in Emergency Situations

Algan Mert Yasar^{1,2}, Robert Voûte^{1,2}, Edward Verbree²

¹ CGI Nederland B.V. George Hintzenweg 89, 3068 AX Rotterdam, The Netherlands
algan.yasar@cgi.com, robert.voute@cgi.com

² Delft University of Technology, Faculty of Architecture and the Built Environment, Architectural Engineering and Technology,
Digital Technologies 2628 BL Delft, The Netherlands
a.m.yasar-1@student.tudelft.nl, r.voute@tudelft.nl, e.verbree@tudelft.nl

Keywords: Indoor navigation, Path planning, Game engine, RRT, Real-time, Point cloud.

Abstract

This study investigates the feasibility of directly utilizing 3D indoor point clouds for real-time indoor navigation, particularly to enhance emergency response processes. Traditional indoor navigation research primarily focuses on creating navigation systems from pre-existing indoor models, resulting in a graph representation that simplifies spatial relationships, requires post-processing, and delivers results only afterwards, often overlooking real-time obstacles and complex layouts such as those in modern office floors. This research proposes an original approach by leveraging real-time generated 3D models using HoloLens 2 sensors, which combine RGB images and depth sensor output to create a comprehensive point cloud. The study explores path planning directly within these point clouds without the need for extensive preprocessing or segmentation, aiming to provide immediate navigation support with minimal delay. Utilizing the Rapidly Exploring Random Trees (RRT) algorithm, the research seeks to minimize preprocessing and swiftly visualize navigable paths, evaluating the system's performance in terms of processing time and path viability. This approach addresses the limitations of traditional graph-based methods and the challenges posed by outdated or unavailable indoor models, offering a promising solution for real-time emergency navigation assistance.

1. Introduction

Emergency response scenarios demand urgency and precision, as the nature of such events often involves saving lives and minimizing damage (Kapucu and Garayev, 2011). Navigating an unfamiliar location such as a high-rise building without prior knowledge, can be a significant challenge for emergency responders who need to quickly locate resources like fire extinguishers, safe spots, and alternative exits. Even a few seconds saved through an improved navigation system could have a substantial impact on the outcome of an emergency. Current indoor navigation systems primarily rely on satellite signals (such as GNSS) and local Wi-Fi or Bluetooth signals for positioning, which might offer great accuracy under normal circumstances. However, these systems often fail in emergency scenarios where power outages or structural damage can disrupt local networks and block satellite connections. Furthermore, these systems typically depend on pre-existing building data like floor plans or Building Information Models (BIM), which may be outdated or poorly maintained (Nikoohemat et al., 2020), leading to inaccurate and potentially dangerous guidance during emergencies. Most indoor navigation solutions represent the environment in 2D and simplify the spatial complexity into nodes and connecting lines, which is efficient for everyday use but inadequate for the intricate layouts and unexpected obstacles found in emergency scenarios (Boguslawski et al., 2022).

This study aims to leverage the capabilities of an existing system (Morlighem et al., 2020; Smit et al., 2021; van Schendel, 2022) where 3D indoor point clouds are generated and expanded in real-time as field agents navigate through the environment. We utilize visual HoloLens 2's internal SLAM system for positioning, which does not depend on external connectivity other than local network connections provided with the system for data transfer. This research emerged from an effort to find a methodology that could utilize available 3D point cloud data with minimal

processing to keep up with the real-time data influx. An unconventional approach was adopted, using a methodology popular in the robotics industry that focuses on speed and iterability. Additionally, the study uses data obtained from three different scanners: one is the Microsoft HoloLens 2, used to evaluate what could be achieved with the current system, and the other two are survey-grade mobile laser scanners (MLSs), namely GeoSLAM ZEB Horizon RT and Leica BLK2GO. Even though their workflows do not fit with the real-time mapping system, they are employed in this research to see what is achievable if sensors with higher point density and wider fields of view become available.

As sensor technology continues to advance rapidly, the potential for integrating higher-quality sensors into real-time navigation system becomes increasingly feasible. Notable developments, such as the introduction of LiDAR sensors in consumer devices like Apple's mobile phones and tablets starting in 2020, highlight the trend towards more accessible and powerful sensing technologies (Díaz-Vilariño et al., 2022). The recent release of Apple's Vision Pro during the course of this research further underscores the progress in wearable sensor technology. These advancements suggest that features typically reserved for high-end mobile laser scanners could soon be available in smaller, more affordable, and wearable formats.

Employing the modified bi-directional Rapidly Exploring Random Trees (RRT) algorithm for pathfinding, this method treats the points in the point cloud as obstacles, using any available empty space to establish a path from the start to the endpoint. The RRT algorithm is chosen for its efficiency in rapidly processing and adapting to new environmental data, crucial for navigating dynamically changing indoor environments during emergencies.

Unlike traditional navigation systems that simplify environmental data into 2D maps, this solution maintains the three-dimensional complexity of the environment. This approach

allows for effective navigation through and around physical obstacles within complex indoor settings, which is essential in emergency scenarios where environments are prone to sudden alterations such as obstructions from debris or changes in accessible paths.

For example, in emergency scenarios requiring navigation through complex and altered indoor layouts, traditional graph-based navigation systems might not swiftly adapt to such changes. These systems often require time-consuming reprocessing of environmental data to reflect new conditions. In contrast, the use of a real-time 3D model in this research enables the immediate adaptation of navigation paths to the current state of the environment, providing responders with up-to-date and reliable routing.

The proposed navigation system can swiftly adapt to the dynamically changing environment of an emergency. By enhancing decision-making processes and providing efficient paths to safety or to other agents in distress, the system optimizes the use of real-time data with minimal processing.

This navigation solution thus offers an alternative to conventional methods by leveraging advanced scanning technology and pathfinding algorithms, facilitating more effective and reliable navigation in complex indoor environments during emergency responses.

2. Related Works

Indoor navigation is often considered under few main topics: indoor positioning, and pathfinding, mapping (modelling), and communication of the found path. The indoor positioning aspect of this project has been addressed by preceding studies (Morlighem et al., 2020; Smit et al., 2021; van Schendel, 2022). Therefore, our study will focus on the latter. This section follows the natural progression of indoor navigation: first, modelling; second, pathfinding; and lastly, navigation support/communication. Each of these subsections will be considered from the perspective of emergency response.

2.1 3D Indoor Mapping Techniques

The importance of accuracy in indoor modelling has been previously emphasized; there is a need for an up-to-date model, whether it be a floor plan, BIM, or 3D model. An up-to-date 2D floor plan would still represent a complex office floor as a single room. Therefore, the requirement is an up-to-date, 3D indoor model. Building Information Models (BIMs) could be an option. However, a case study by the European Commission (Carbonari et al., 2020) indicated that out of 21 building logbook initiatives, only 8 require updates after renovations, and only 3 are digitalized and accessible. Consequently, even if an up-to-date 3D model exists, it might not be available. To summarize, 2D models are inadequate for representing complexity, and existing 3D models could be inaccessible or outdated. Therefore, real-time 3D reconstruction appears to be the most viable option for emergency response scenarios.

2.2 3D Path Planning

Using a 3D model for navigation is often achieved with a navigation graph. This method requires segmenting and classifying the 3D model to extract the navigable surface of the floor plane. An example is the work of Balado et al., (2019), where they use a 3D point cloud to calculate the route. While this is a robust methodology, it takes computational time. Flikweert et al., (2019) proposed a methodology that automates the process from point clouds to navigation graphs, which can perform even better with high-capacity hardware. However, simplifying the

obstacles and navigable surface into a 2D representation is not feasible in a dynamic environment such as emergency response. Another approach by Broersen et al., (2016) proposes extracting the 3D empty space and structuring it into a graph to achieve pathfinding. This also requires time and processing, but their implementation of an octree approach allows them to quickly index their graph, even though it is a 3D structure. Lastly, the work of Fichtner et al., (2018) combines the use of octree and navigation graphs to streamline the process even more. However, these methodologies would require reprocessing as the dataset expands in real-time and depend on a secondary product. Therefore, we sought a method that could work with an unstructured, unprocessed point cloud.

Path planning methodologies for humans, in contrast to robotics, require much more structure and classification to be understood by the end-user, which does not fit with our objective. Therefore, we adopted a methodology that emerged in the field of robotics: Rapidly Exploring Random Trees (LaValle, 1998). This simple algorithm is not computationally demanding yet effective. In addition to the base algorithm there are various studies that proposed improvements such as bi-directional search, target oriented exploration and adaptive extension (Wang et al., 2022; Zheng et al., 2022).

2.3 Visualization and Path Communication

The final part of this section involves the visualization and communication of the path. We use the term "communication" because the output of our pathfinding methodology is a list of coordinates, which might be adequate for an autonomous drone but not for a field agent. Our end user, a field agent, has limited time and resources to comprehend complex data. Patel and Grewal, (2022) conducted a user study comparing augmented reality (AR)-based indoor navigation with traditional 2D maps. Most users found the AR-based method to be less mentally and physically tiring and were able to reach their objectives faster. The study noted that a limitation was the need for users to hold a mobile phone in front of them for navigation support. The authors proposed the use of an AR-headset as a further improvement, which could enhance the user experience by allowing hands-free navigation. Additionally, preceding studies related to this project (Morlighem et al., 2020; Smit et al., 2021) utilized the Unity game engine for visualizing 3D graphics, and the current workflow is optimized around this system. Therefore, we decided to visualize the path in Unity in a way that can be overlaid onto the real world using the HoloLens's mixed reality view. This approach ensures that the path is easily understandable and can be followed by the field agent in real-time, enhancing their navigation efficiency during emergency responses.

3. Methodology

3.1 System Overview

Diagram and overall explanation of the system is seen in Figure 1 below. Point cloud processing and coordinate picking actions are only applicable for the proof of concept and are due to the limitations of mobile laser scanners and the development environment. The 3D modelling system that this project is designed for can provide point clouds as well as the location of field agents (initial coordinates) and allows for the selection of goal coordinates. Therefore, the pre-processing section will not be required.

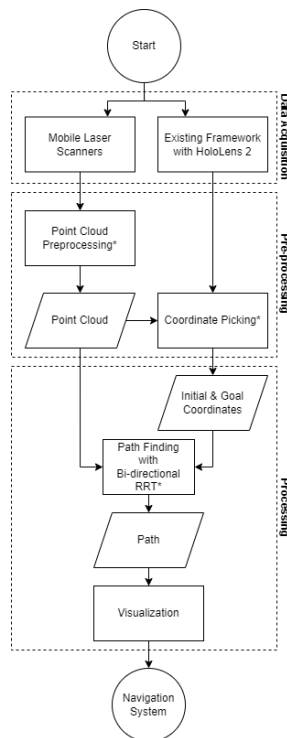


Figure 1. Methodology Flowchart

3.2 Data Collection Equipment and Environments

Our study employed the HoloLens 2 and two mobile laser scanners, the GeoSLAM Zeb Horizon RT and Leica BLK2GO, across three environments: an office building, an apartment complex, and a basement. The office environment was selected for its complex layout and reflective surfaces like glass and monitors, which pose challenges for scanning accuracy. The apartment complex and basement, chosen for their simpler structures and minimal pedestrian traffic, facilitated a focused evaluation of the system's routing and visualization capabilities under controlled conditions. All environments were scanned with the appropriate permissions to comply with data privacy regulations, ensuring GDPR (General Data Protection Regulation) compliance.

3.3 Scanning Workflows and Data Preprocessing

The HoloLens 2 was used to represent real-time 3D modelling system, capturing RGB and Depth images along with the scanner's pose. This data was stored in an SQL database and processed at the edge server into point clouds, though specific details remain outside the scope of this document due to proprietary restrictions. The mobile laser scanners required initial preprocessing using their respective software suites to convert proprietary data formats into standard files (.ply, .las), necessary for subsequent visualization and analysis. This preprocessing included noise filtering and SLAM processing, streamlined to mirror potential future scenarios where more advanced sensors might be used without extensive manual intervention.

Furthermore, preprocessing included coordinate picking prior to path finding. This is an additional step required in the development process in contrary to the 3D modelling framework where initial location will be the field agent's location and goal location will be determined from the system interface (Figure 2).

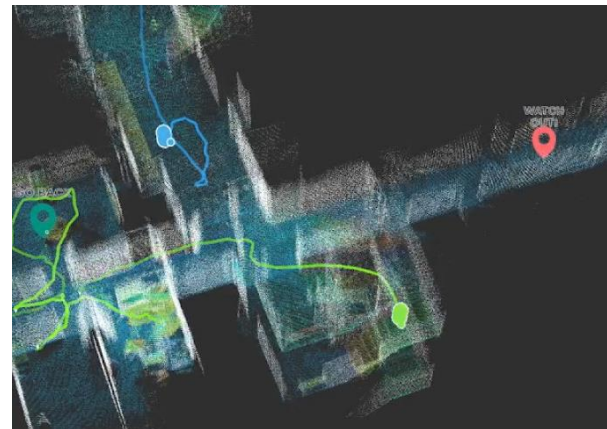


Figure 2. POI and Field agent view from 3D modeling system

3.4 Data Handling and Processing

While this research did not focus on sensor fusion, it emphasized the individual capabilities and data processing workflows of each scanner type. The HoloLens 2 and mobile scanners operated independently, capturing distinct datasets that were used to evaluate the system's performance across different technological capabilities and environmental complexities.

3.4.1 Data Structuring: As mentioned in previous chapters, high computational demand of 3D point clouds is often a bottleneck. In this study we addressed this in a similar method to Fichtner et al., (2018) and Broersen et al., (2016). An octree was used to structure the search space, enhancing the steering and collision checks, while the search trees of the RRT algorithm were structured with a KD-tree to achieve feasible iteration speeds.

3.4.2 Pathfinding and Route Planning: Pathfinding was conducted using a modified bi-directional Rapidly Exploring Random Trees (RRT) algorithm. Inputs included initial and goal coordinates, accompanied by the point cloud file either directly exported from the 3D modelling system or acquired with mobile laser scanners and pre-processed. The following sections explain the functions involved in path planning. The generalized process is illustrated in the flowchart below (Figure 6).

3.4.3 Collision Check: One of the most crucial functions of the algorithm is collision checking. In most RRT implementations, obstacles are depicted as 'regions,' so the collision check involves determining if the sample coordinates fall within these obstacle limits. However, in this implementation, obstacles are the points of the point cloud, representing an indoor environment. Therefore, the aim is to stay 'in' the obstacle.

To achieve this, an axis-aligned bounding box that roughly represents the size of an average adult human is used to check if there are 'points' in the path. Another challenge is defining the threshold for a collision. Since the data was not cleaned or filtered, noise in the scans could result in false collisions. Thus, we calculated the density of the obstacles to decide on this threshold. This density is sensor-specific and should be adjusted according to the data source.

3.4.4 Target-Bias: In standard RRT, sampling is uniformly random across the search space, which can result in inefficiencies in large spaces or narrow openings. A performance-enhancing solution is target-biased sampling. The algorithm still randomly explores but, with a probability of p , the tree is expanded from the nearest node to the goal, in the direction of the goal.

3.4.5 Step Size: Defined as the distance between current node (n) and sample node ($nsample$), step size is set as the width of the bounding box used for collision checking. As the algorithm checks for collisions around $nsample$, which is at the center of the bounding box, the previous check for n ensures the first half of the step is obstacle-free. Therefore, checking for $nsample$, results in a clear path. Bounding boxes and step size is seen in **Figure 3** Error! Reference source not found..

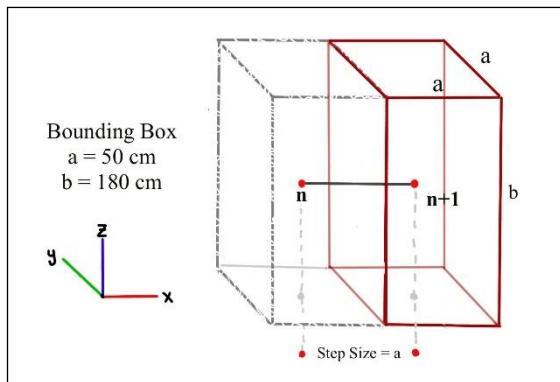


Figure 3. Bounding box and step size.

3.4.6 Bi-Directional Search: Search process involves two RRT trees: Tree A (TA) and Tree B (TB). TA starts the search from initial coordinates and expands towards the goal coordinates (due to bias). Meanwhile, TB starts at the goal coordinates and expands to the latest node of TA in a sequential search process.

3.4.7 Path Optimization, Normalization & Export: When a connection is established between initial and goal points, rewiring is executed which then checks if there is a path with lower cost. Furthermore, as the path represents a connection between crucial points of collision check bounding boxes, the coordinates are normalized to the floor level by reducing the Z component by 0.7 meters (**Figure 4**). This process is done to optimize the visualization results. Lastly, the coordinate list of nodes is exported in a plain text file.

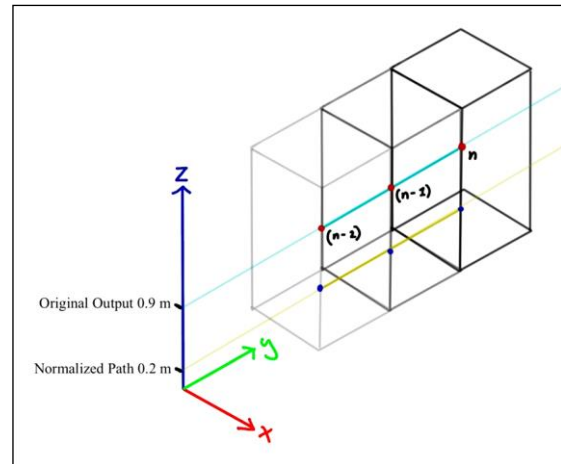


Figure 4. Normalized path illustration

3.4.8 Visualization: The user interface of the real-time 3D modelling system is developed with a game engine, which is also used for visualizing the 3D model. Consequently, the path visualization is done within the same environment. A C# script that accepts the resulting text file from the path planning is used to visualize the path. Since the path shares the same coordinate system with the point clouds, the final path, after normalization, appears approximately 0.2 meters above the surface and provides visual guidance that does not require additional directions (**Figure 5**).



Figure 5. Path visualized over the dataset in game engine.

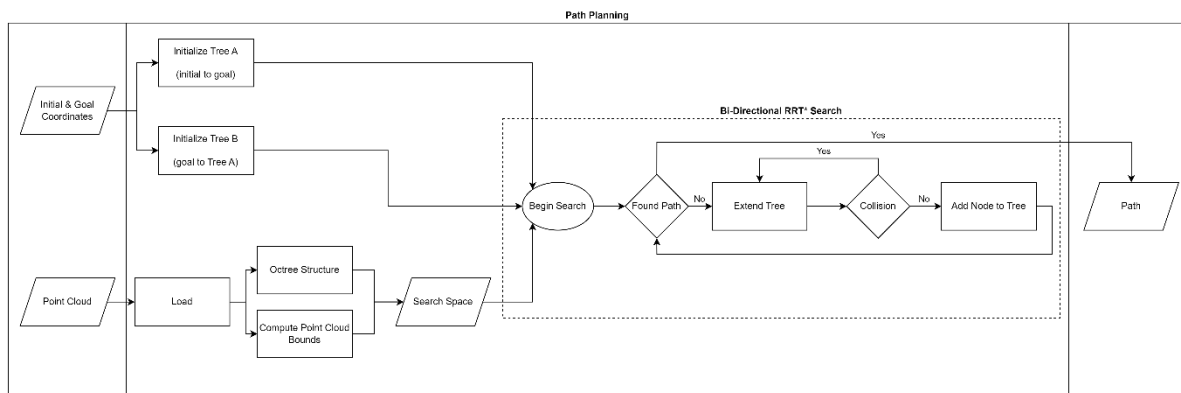


Figure 6. Path Planning Flowchart.

4. Results & Discussion

4.1 System Performance and Future Directions

The system's effectiveness was primarily measured by the speed and success of the pathfinding process, with the RRT algorithm tailored to prioritize rapid, safe route generation over finding the shortest path. This approach was chosen to reflect the urgent need for emergency navigation, where conditions can change unpredictably. The current implementation serves as a proof of concept, with plans for future tests to validate and refine the system under a broader range of real-world conditions. The sample results for the longest test path is given below.

Testing hardware:

- Processor: Intel i7-10850H CPU @ 2.70GHz
- RAM: 32 GB DDR4
- GPU: NVIDIA Quadro T1000 4 GB DDR6

Dataset:

- Environment: Basement (Figure 7)
- Sensor: GeoSLAM Zeb Horizon RT
- Point Cloud Size: 1,407,430 points
- Path Length: 83 meters
- Average Time per 1000 Iterations: 0.61 seconds
- Total Run Time (octree generation and ~15,000 iterations): 11.89 seconds

Note that the hardware capacity used in testing is significantly lower than that of the edge computer. Therefore, we consider these rates promising.

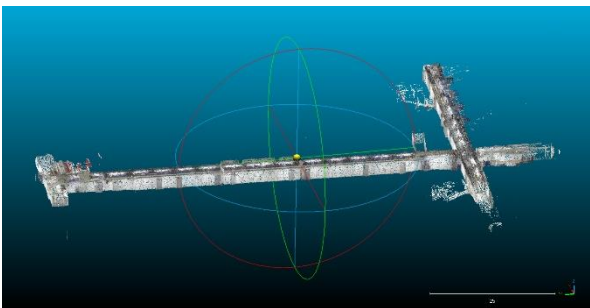


Figure 7. Basement dataset (Scale bar = 25 m).

4.2 Challenges

4.2.1 Sensor limitations: On one hand, it is possible to generate a point cloud in real-time. However, the resulting point cloud often has gaps where the algorithm ends up leaving the point cloud, or there are obstacles that are not completely scanned, hence considered as viable openings. On the other hand, some sensors can quickly scan the environment with much higher coverage, resulting in a dataset that is perfect for this methodology. However, it is not yet possible to access the point cloud without preprocessing. A comparison of the same environment with different scanners is shown below (Figure 8). Due to field of vision limitations, the ceiling is not covered by the HoloLens scan.

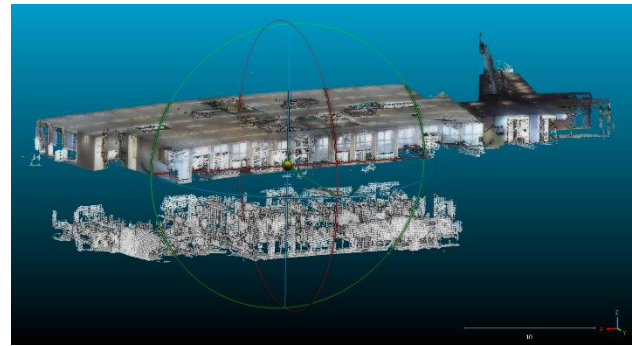


Figure 8 Office environment with Mobile Laser Scanner (top), HoloLens 2 (bottom).

4.2.2 Step size: As mentioned in the methodology (3.4.2), the step size is set to 0.5 meters to accommodate the bounding box for collision checks. While this is effective for collision checking, a small step size negatively affects performance in large areas (e.g., the basement dataset). Variable step size implementations have proven to be useful in such scenarios (Zhang et al., 2019). However, using a variable step size would cause the collision check method to fail.

4.2.3 Target bias: Another method to overcome the challenges of large areas is introducing a target bias. This improves performance in simpler layouts; however, the bias probability also introduces a failure probability in complex layouts. As seen in Figure 9 below, both trees quickly advanced to their objective yet resulted in a failure due to iterating towards opposite sides of a wall and not being able to connect. The red and yellow dots represent the nodes that are closest to each other (0.2 meters apart). This challenge is especially evident in scenarios with multiple floors.

Diagnosis: Nearest Nodes and Attempted Connection

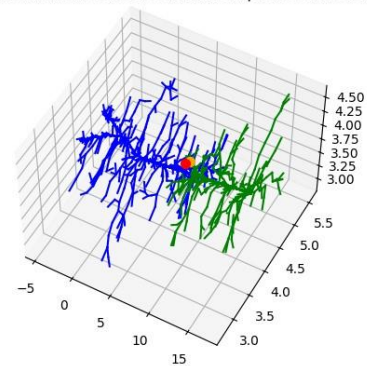


Figure 9. Failed due to biased sampling.

4.3 Discussion

Results show that it is possible to achieve efficient path planning by direct use of 3D indoor point clouds. However, it is not feasible with the current 3D modelling system due to limitations of the sensors. First of all, coverage of the HoloLens 2 FOV (field of view) and Range is inadequate for this methodology as HoloLens 2 data has gaps and unscanned obstacles which is then considered as empty space and used for path planning by the RRT search. Second, data acquired by mobile laser scanners were suitable for search. On the other hand, streaming the point clouds in real-time with these devices is not yet possible.

Another aspect to mention is the existence of drift in the scans. Especially with the HoloLens 2 and BLK2GO scans. Loop closure is practiced with Zeb Horizon RT scans while BLK2GO scans conducted without it to observe the effects. There is visible drift even in comparably short scans (>1 m drift in scans less than 10 minutes). For our methodology it does not effect the results as the path finding and visualization share the positioning and coordinate system of the point cloud. However, it could result in inaccuracies if georeferencing is required.

4.4 Future works

4.4.1 Parallel Processing / Multithreading: Processing the search from both trees simultaneously instead of the current sequential methodology could yield greater improvements in performance.

4.4.2 Continuous and Variable Collision Check: In our research bounding box hence, the collision check is limited to a large rectangle box. In a scenario where there is an opening that only allows the user to crawl through, our method will fail. However, using a smaller bounding box that represents the minimum space required to pass could result in an unfeasible path (e.g. a path that requires to crawl under tables when there is an opening large enough to walk next to it). Implementing a variable method that changes the size after certain amount of failed iterations could solve this problem. In addition, a method that can check around the proposed path constantly instead of bound based controls (without hindering the performance) can facilitate variable step size and provide performance increase.

4.4.3 Visualization Improvements: Implementing gaussian splatting to the point clouds could result in a visual that is much easier to comprehend for the observers (commanding agents) compared to the unstructured and semi-transparent look of the point clouds.

Acknowledgements

We would like to acknowledge that we have utilized ChatGPT to enhance the readability and clarity of this document.

References

- Balado, J., Díaz-Vilariño, L., Arias, P., Frías, E., 2019. POINT CLOUDS TO DIRECT INDOOR PEDESTRIAN PATHFINDING. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLII-2/W13*, 753–759. <https://doi.org/10.5194/isprs-archives-XLII-2-W13-753-2019>
- Boguslawski, P., Zlatanova, S., Gotlib, D., Wyszomirski, M., Gnat, M., Grzempowski, P., 2022. 3D building interior modelling for navigation in emergency response applications. *Int. J. Appl. Earth Obs. Geoinformation* 114, 103066. <https://doi.org/10.1016/j.jag.2022.103066>
- Broersen, T., Fichtner, F.W., Heeres, E.J., Ivo, de L., Rodenberg, O.B.P.M., Verbree, E., Voûte, R., 2016. Project Pointless: Identifying, visualising and pathfinding through empty space in interior point clouds using an octree approach., in: *Proceedings of the 19th AGILE Conference on Geographic Information Science*. Presented at the 19th AGILE Conference on Geographic Information Science., Helsinki, Finland.
- Carbonari, G., Ricci, M., Calderoni, M., Dourlens-Quaranta, S., 2020. *Building logbook state of play: report 2 of the study on the*

development of a European Union framework for buildings' digital logbook. Publications Office, LU.

- Díaz-Vilariño, L., Tran, H., Frías, E., Balado, J., Khoshelham, K., 2022. 3D MAPPING OF INDOOR AND OUTDOOR ENVIRONMENTS USING APPLE SMART DEVICES. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLIII-B4-2022*, 303–308. <https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-303-2022>
- Fichtner, F.W., Diakit , A.A., Zlatanova, S., Vo te, R., 2018. Semantic enrichment of octree structured point clouds for multi-story 3D pathfinding. *Trans. GIS* 22, 233–248. <https://doi.org/10.1111/tgis.12308>
- Flikweert, P., Peters, R., D az-Vilari o, L., Vo te, R., 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. IV-2/W5*, 271–278. <https://doi.org/10.5194/isprs-annals-IV-2-W5-271-2019>
- Kapucu, N., Garayev, V., 2011. Collaborative Decision-Making in *Emergency and Disaster Management*. *Int. J. Public Adm.* 34, 366–375. <https://doi.org/10.1080/01900692.2011.561477>
- LaValle, S.M., 1998. Rapidly-exploring random trees : a new tool for path planning. Iowa State Univ. Annu. Res. Rep.
- Morlighem, C., Chatzidiakos, C., Feenstra, J., van Schendel, M., Hurkmans, R., 2020. Deployment of Indoor Point Clouds for Firefighting Strategy. <https://repository.tudelft.nl/islandora/object/uuid%3A1dcb4e17-bc41-4529-8303-414deb07f330?collection=education>
- Nikoohemat, S., Diakit , A.A., Zlatanova, S., Vosselman, G., 2020. Indoor 3D reconstruction from point clouds for optimal routing in complex buildings to support disaster management. *Autom. Constr.* 113, 103109. <https://doi.org/10.1016/j.autcon.2020.103109>
- Patel, V., Grewal, Dr.R., 2022. AUGMENTED REALITY BASED INDOOR NAVIGATION USING POINT CLOUD LOCALIZATION. *Int. J. Eng. Appl. Sci. Technol.* 6, 67–77. <https://doi.org/10.33564/IJEAST.2022.v06i09.009>
- Smit, B.-P., Vo te, R., Verbree, E., 2021. CREATING 3D INDOOR FIRST RESPONDER SITUATION AWARENESS IN REAL-TIME THROUGH A HEAD-MOUNTED AR DEVICE. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. V-4-2021*, 209–216. <https://doi.org/10.5194/isprs-annals-V-4-2021-209-2021>
- van Schendel, M., 2022. Merging of Topometric Maps of Indoor Environments. <https://repository.tudelft.nl/islandora/object/uuid%3A92e1ea3a-3a34-449c-b0b4-731d6480266a>
- Wang, X., Wei, J., Zhou, X., Xia, Z., Gu, X., 2022. AEB-RRT*: an adaptive extension bidirectional RRT* algorithm. *Auton. Robots* 46, 685–704. <https://doi.org/10.1007/s10514-022-10044-x>
- Zhang, Z., Wu, D., Gu, J., Li, F., 2019. A Path-Planning Strategy for Unmanned Surface Vehicles Based on an Adaptive Hybrid Dynamic Step-size and Target Attractive Force-RRT Algorithm. *J. Mar. Sci. Eng.* 7, 132. <https://doi.org/10.3390/jmse7050132>

Zheng, Z., Bewley, T.R., Kuester, F., Ma, J., 2022. BTO-RRT: A rapid, optimal, smooth and point cloud-based path planning algorithm. <https://arxiv.org/abs/2211.06801>