AGILE: GIScience Series, 6, 28, 2025. https://doi.org/10.5194/agile-giss-6-28-2025
Proceedings of the 28th AGILE Conference on Geographic Information Science, 10–13 June 2025.
Eds.: Auriol Degbelo, Serena Coetzee, Carsten Keßler, Monika Sester, Sabine Timpf, Lars Bernard
This contribution underwent peer review based on a full paper submission.

Identification and Visualization of Unscanned Areas Within a Building Based on the Building's Outer Hull.

Georgios Iliopoulos [1], Robert Voûte [1,2], and Peter van Oosterom [1]

[1]Delft, University of Technology, Department of Architectural Engineering & Technology
[2]CGI Nederland BV, Department of Geo-ICT

Correspondence: Robert Voûte (r.voute@tudelft.nl)

Abstract. 3D evaluation of indoor environments is important for first responders and their commanders. Both are unfamiliar with the situation that they will find. While doing their work they can scan the environment they are working in and process that data in real time. To be able to do this efficiently both the scanned areas and the unscanned parts need to be assessed giving indications where to go to next. In this research an approach to identify, classify and visualize unscanned areas within the building is proposed. Using RGB and Depth (RGB-D) cameras and Collaborative Visual SLAM a pointcloud of the interior of the building is generated. By combining the pointcloud with the building's outer boundary it is possible to evaluate which areas have been visited by the responders, which have not and identify potential hidden spaces. The proposed solution implements that by employing voxels. The result can be seen on the online 3D web-viewer. The research shows promising results but also aspects that will have to be improved like the alignment of indoor and outdoor parts. Also the narrow field of view of the used LiDAR device (iPhone) brought limitations.

Algorithm, case study.

[CV6] Usability of maps, [DM5] Modelling 3D, temporal and uncertain phenomena, [TA11] User community of EO services and applications.

Keywords. Indoor pointcloud, first responders, voxelization, voxel transversal, unscanned spaces

1 Introduction

Localization in indoor environments is significant for emergency response and can assist first responders (i.e., fire brigade, police, paramedics) in their operations and communication (Dilo and Zlatanova, 2011). Since GNSS accuracy will be limited indoors, other wireless-based localization solutions have been proposed (e.g., WiFi, Bluetooth Low Energy, RFID tags, Ultra-wideband, etc.), which come with their own limitations in terms of cost and accuracy (Tseng et al., 2022; Smit et al., 2021). In addition to that, floor plans are not always available or they are outdated. Furthermore, using static 2D plans in a dynamic 3D environment confines the possibilities of operation management and raises the importance of having 3D representations of the building in question to increase the situation awareness(Smit et al., 2021). According to Tseng et al. (2022), another approach to tackle the localization issue is visual Simultaneous Localization and Mapping (visual-SLAM). Visual-SLAM is capable of reconstructing a 3D scene in the form of a point cloud from RGB and Depth images, and keeping track of the sensor's position at the same time (Abaspur Kazerouni et al., 2022). Therefore, this method is suitable for emergency situations and can bridge the information gap described before, regarding the 3D representation of the building and the position of the people who operate in it, fulfilling the needs for spatial information and increasing the situational awareness of the agencies involved (Smit et al., 2021).

Having a 3D representation of a building's interior in point cloud form is undoubtedly a valuable resource in emergency situations. However, visualizing progress will further enhance the informational value and situation awareness. Specifically, it is crucial to have a means of tracking progress, showing scanned areas and identifying areas that yet have to be observed simultaneously. This capability can greatly assist in planning the scanning process by allowing for the efficient allocation of personnel to different locations. Furthermore, the ability to pinpoint unobserved areas after scanning is completed adds significant value. These areas are of particular interest to emergency responders, such as the police, who may be concerned about concealed spaces where people or objects of interest could be located. Addressing this specific information gap is the core focus of this proposal. To solve this problem, a volumetric approach using 3D voxels is followed.

2 Methodology

The problem this research tried to solve is the detection of which spaces inside a building are not observed by the sensors which are carried by first responders during an emergency operation. The sensors are RGB and Depth cameras and collaborative visual SLAM is used to generate a map of the interior of the building. To solve the problem, the usage of voxels is chosen. This section, describes the methodology followed to solve the problem and elaborates on the steps that were carried out towards the goal.

One of the many ways to represent the 3D model of a building, for instance IFC, CityGML and IndoorGML, is the usage of voxels (Boguslawski et al., 2022). According to Xu et al. (2021), discretization of 3D point clouds using voxels is "...a common and effective way to organize and structure 3D point clouds." and they "...can be used to structurally represent discrete points in a topologically explicit and information-rich manner.". These two properties mean that the neighbors of a voxel are known and semantics can be assigned to each voxel. Furthermore, voxels enable the representation of occupied and free volumes, in the same fashion as occupancy grids in 2D (Gehrung et al., 2019).

To be able to determine unscanned parts of the interior it is necessary to be aware of the outer limits. The outer boundary of the building describes these. The proposed solution relies on the prerequisite that the building's outer boundary is available and aligned with the interior scan. The former part is guaranteed in the Netherlands since the outer boundaries of all buildings are available on 3DBAG (see Figure 1). The later condition requires manual processing see subsection 3.1 since it is not possible to automate the alignment. Having the exterior shell of a building and the data (i.e., keyframes with RGB and Depth images and associated camera parameters) to reconstruct the point cloud of its interior — simulate the scanning process, someone can analyze in volumetric fashion which areas are not yet visited. To organize the analysis of the outer boundary minus the scanned areas the use of voxels has been introduced. By creating a voxel grid that contains the bounding box of the outer boundary it is possible to classify its voxels in three categories (Figure 2):

- Voxels that contain the boundary.
- Voxels which are outside of the building.
- Voxels which are inside of the building.

Having this classification of the voxels, by reconstructing the point cloud incrementally one keyframe at a time it is possible to detect which interior voxels contain points and which voxels correspond to the void space between the camera's position and the voxels that contain points. In this case, a keyframe is considered the 3D pose with its corresponding a pair of RGB-D images, which are captured in a frequency of 2 times per second. Doing this for every keyframe and updating the voxel grid accordingly, at the end of the scanning process, it is known which voxels are occupied by the point cloud and which voxels were seen by the camera but are empty (the detailed steps are given in subsection 3.2). It is important to highlight that this solution considers the scene to be static, which means that a voxels that is labeled as 'Occupied' can not change status. The proposed solution can be discretized in the following steps:

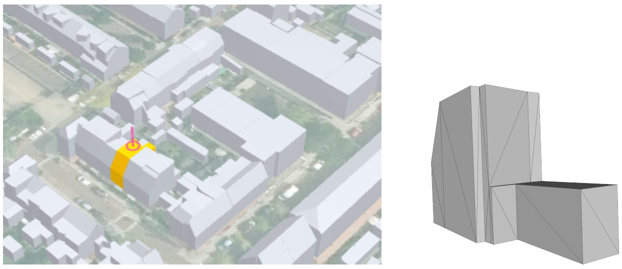

Figure 1. 3DBAG. On the left, an extract from 3DBAG from the city of Delft. The highlighted building is the one used as the case-study. On the right, the the triangular mesh of the building's outer boundary downloaded from 3DBAG.

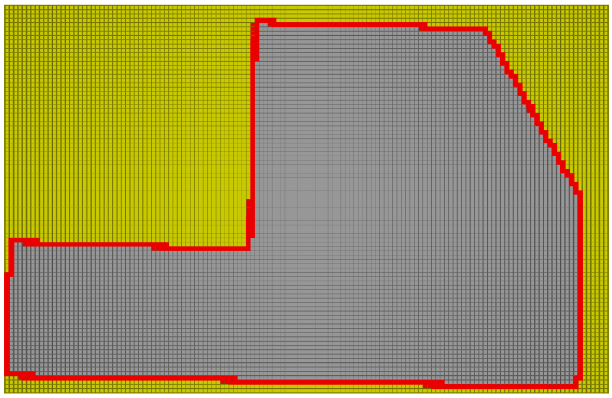

Figure 2. Section of the initial voxelization of the building's bounding box. Yellow voxels are the exterior of the building, the red voxels are the voxels occupied by the building's outer boundary and the gray voxels are those in the interior of the building.

1. Retrieve outer boundary from 3DBAG.
2. Scan the interior of the building.
3. Align the scan with the outer boundary.
4. Voxelize the working domain — bounding box of the building's outer boundary.
5. Reconstruct the point cloud per keyframe.
6. Voxelize the point cloud.
7. Mark voxels that contain points and void voxels — implement voxel traversal algorithm.
8. Visualize the result.

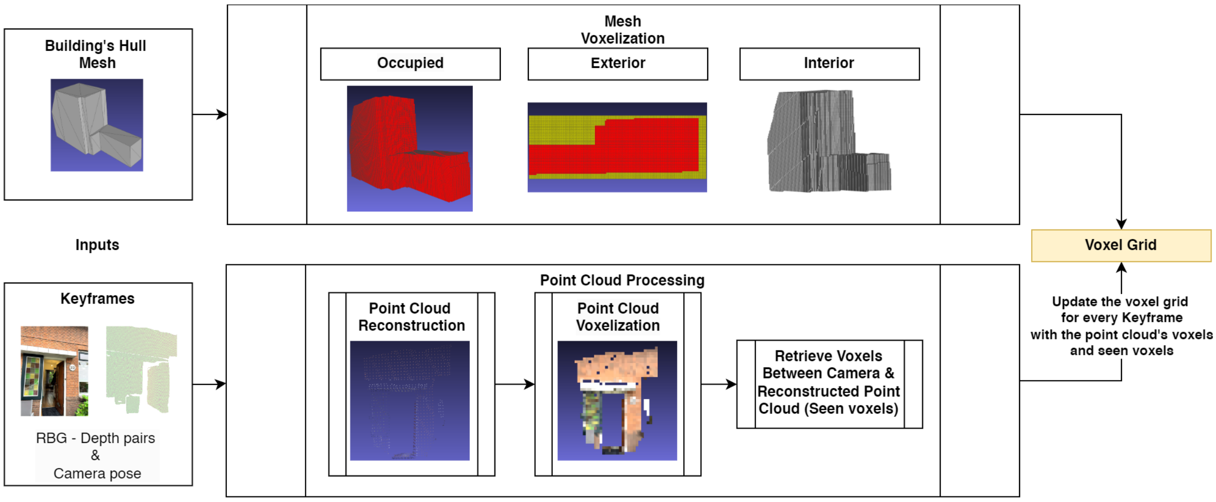

Figure 3. Flowchart

3 Implementation details

The main idea of the proposed solution is that the building's outer boundary is aligned with the point cloud. Initially, the bounding box of the outer boundary is voxelized and the voxels are organized into 3 categories (see Figure 2). Following that, the point cloud is reconstructed incrementally and the voxel grid is updated accordingly. The individual steps are explained in the following sections.

3.1 Outer boundary & Point cloud alignment

The proposed solution was tested on a three-story house in the city of Delft (see Figure 4) which was downloaded from 3DBAG. It was chosen as a proper option due to the fact that it contains a single residence. Consequently, someone can access all the areas inside the buidling, compared for instance to other houses which share the floors interchangeably. The data were captured using iPhone 15 Pro Max using an open source application called RTAB-Map - 3D LiDAR Scanner (Labbé) and the data are stored in an SQLite database. The point cloud and the outer boundary were aligned manually using CloudCompare and the result is depicted in Figure 5. For this step it was chosen to transform the outer boundary so that it contains the point cloud and not the other way around.

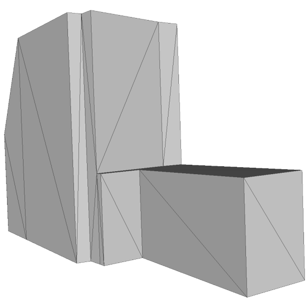

Figure 4. Outer boundary of the building used to test the solution.

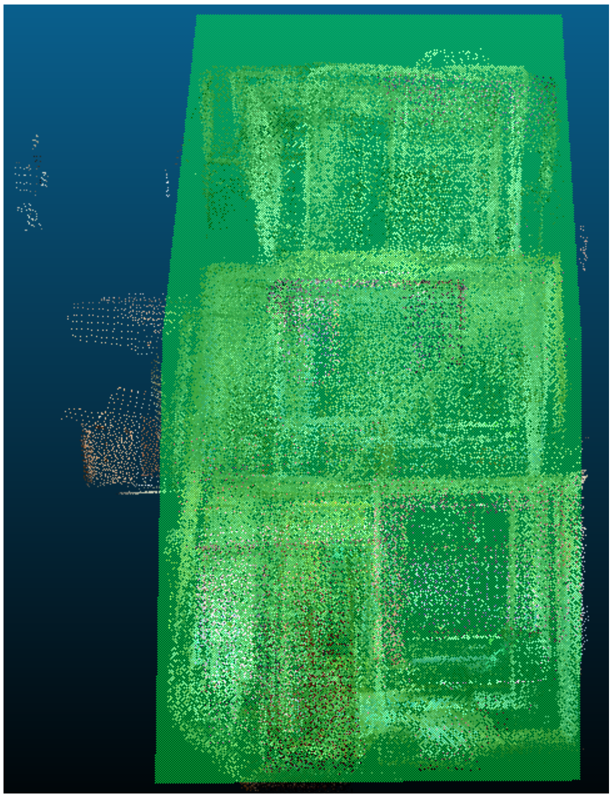

Figure 5. Result of outer boundary and point cloud alignment in CloudCompare. The green surface represents the outer boundary and the points are the reconstructed point cloud of the interior. The points that lay outside of the building are points captured from the exterior of the neighboring house, while the device was on the balcony which can be seen in Figure 4.

3.2 Point cloud reconstruction - Scan simulation

The captured data are stored in a SQLite Database. To reconstruct the point cloud someone would need the pairs of RGB and Depth images of each keyframe (e.g., see Figure 6) along with the pose (i.e., the intrinsic and extrinsic parameters). The reconstruction of the point cloud was implemented using Open3D library. The result of the reconstruction of the point cloud of a single keyframe is depicted in Figure 7. Figure 8 depicts a staircase reconstructed from a series of 22 keyframes while the complete point cloud is depicted in Figure 9.

Figure 6. Example of RGB and Depth images captured with iPhone. These images show the first key-frame captured during the scanning process of the house used as the case study for the current proposal.

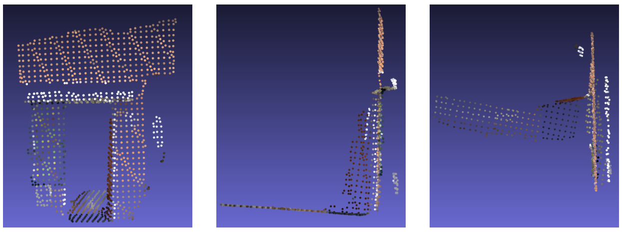

Figure 7. Point cloud reconstructed from the first keyframe. The pair of RGB and Depth image that reconstructs this point cloud is the one in Figure 6.

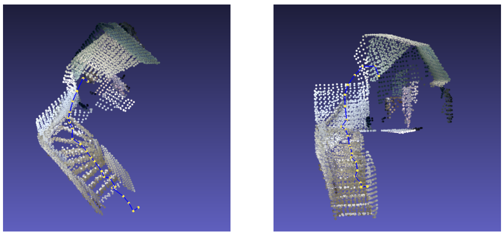

Figure 8. Reconstructed point cloud that represents a staircase and part of the upper floor and ceiling. The blue line represent the trajectory of the camera and the yellow points the location where the images were captured. From the left to the right, the point cloud is illustrated from the front, the side and the top respectively.

3.3 Voxelization

To perform the volumetric analysis of the explored and not explored areas, 10 centimeters voxels are utilized. The voxelization can be discretized into two main steps: 1. voxelization of the building's mesh and 2. voxelization of the point cloud. The latter can be further divided into: (i) voxelization of the point cloud generated by a single keyframe

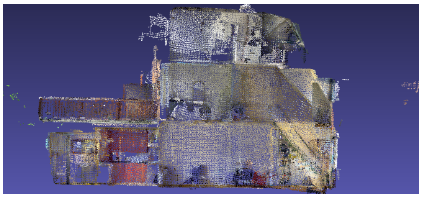

Figure 9. Side-view of the complete point cloud.

and (ii) marking of the void voxels between the camera and voxels that contain points.

To support this goal a custom voxel grid class was implemented. Apart from the origin and the voxel size, the class stores for each individual voxel its Status, which can take 1 of 5 possible values. Each one of these statuses has a different color associated which is used later on for the visualization. More in detail:

(i) the 'Building hull' Status describes the voxels that are occupied the the building's outer boundary.

(ii) as 'Scanned' are marked the voxels that contain points from the generated point cloud.

(iii) the 'Air' Status corresponds to the void which are between the camera and the 'Scanned' voxels and.

(iv) the 'Unknown' voxels are essentially the interior voxels after the initial voxelization of the bounding box.

(v) As 'Temporary' are marked the voxels which are generated on the fly when the camera is outside of the bounding box, for instance when entering the building.

Initially, the bounding box of the mesh is voxelized as shown in Figure 2 using Open3D. Having that, for each point cloud generated from a pair of RGB and Depth images a temporary sparse voxel grid is created (i.e., many empty cells – see Figure 10) which has the same origin and voxel size as the voxel grid that corresponds to the bounding box. Since the two grids have the same origin and voxel size, two voxels with the same index correspond to the same position in 3D space. Therefore it is safe to mark voxels as 'Scanned' in the custom voxel grid based on the grid generated from Open3D.

3.4 Labeling of void voxels

The next step is to determine which voxels are visible from the camera but contain no points, in other words which voxels represent void or empty space. For each Keyframe, the position of the camera is known as well as the voxels which contain points from the reconstructed point cloud. Therefore, marking void voxels can be done by using voxel traversal (Amanatides et al., 1987). This process is visualized in Figure 11 in 2D, but easily extends to 3D. In Figure 11, the purple square is the position of the camera

Figure 10. Sparse voxel grids generated from two different point clouds from different keyframes. In the top row the voxels grid corresponds to the point cloud depicted in Figure 7. In the bottom row from the left to the right the point cloud from a single keyframe is illustrated and the corresponding voxel grid.

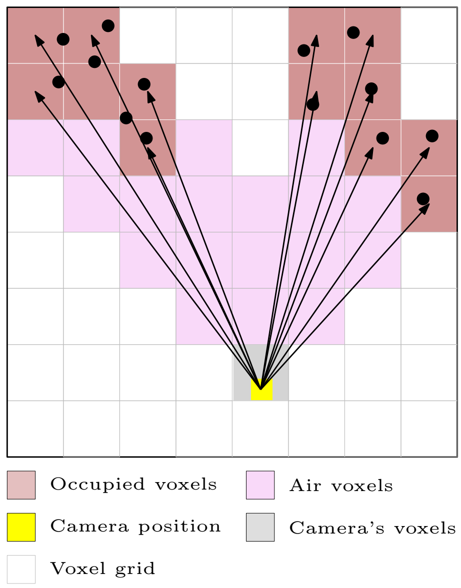

Figure 11. Voxel traversal in 2D. The black arrows represent the rays from the camera to the centers of voxels that contain points.

and the grey voxel is the voxel in which the camera is located; the black circles represent the reconstructed point cloud and the dark cyan voxels are the voxels that contain points. The black lines which run between the center of the camera's voxel and a target voxel. The voxels that are crossed by the lines are the ones that are identified as void space. The implementation of the algorithm is adopted from (Gyurgyik, 2023). The result of this process is visualized in Figure 12 and Figure 13. Since the algorithm marks as 'Seen' only the voxels that are between the camera and the voxels that contain points, it is noticeable, for instance, in Figure 12 that the 'Seen' voxels form a line. In addition, in Figure 13, a cone-shaped structure is created by the 'Air' voxels. This is because the algorithm does not cast arrays in all directions – within a specific radius – but only from the camera to the 'Scanned' voxels.

In summary, this section outlined the key steps and approaches employed to solve the problem this proposal focuses on. The data inputs are two: 1. the building's outer boundary from 3DBAG and 2. the RGB and Depth images and the camera poses from the SQLite database. Then the boundary is used to voxelize its bounding box and categorize the voxels into:1. 'Exterior' 2. 'Interior' and 3. 'Boundary' voxels. Following that, by retrieving the RGB and Depth image pairs and the camera's parameters for each keyframe, a point cloud is generated and voxelized ('Scanned' voxels). Having that, the 'Air' voxels are marked by using fast voxel traversal. Only the 'Unknown' voxels can be marked as 'Air'. This process is iteratively done until all the keyframes are processed and the voxel grid is updated. This pipeline is visualized in the flowchart in Figure 3.

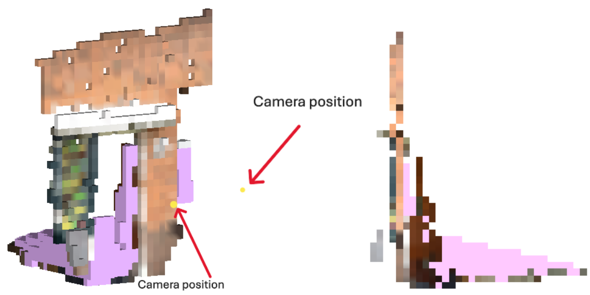

Figure 12. 'Scanned' and 'Air' (in purple) voxels of the point cloud generated from keyframe 1 from different views. The yellow point is the location of the camera.

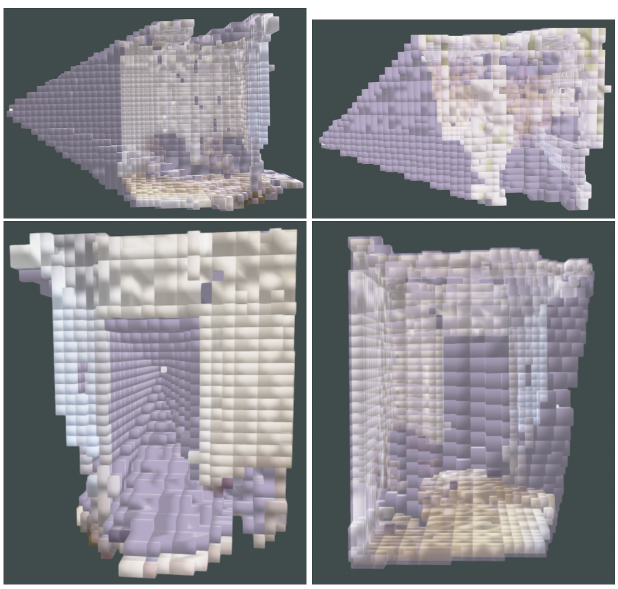

Figure 13. 'Scanned' and 'Seen' (in transparent purple) voxels of the point cloud generated from keyframe 20 from different views. The white point is the location of the camera.

4 Results

As formulated in section 1, the proposal addresses how to enrich spatial information gathered during emergency response operations using mobile scanning devices. The methodology described in section 2 is applied to answer this research question. Since the results of the algorithm appear complicated on static images, it is recommended that the readers refer to the online 3D viewer to visualize and inspect the data.

The prerequisite of the proposed solution is that the outer boundary of the building is available and aligned with the point cloud captured with the iPhone. Acquiring the outer boundary of the building was achieved using 3DBAG data and storing the triangular mesh of the building in .obj format. Following that, the mesh was manually aligned with the point cloud. This process does not give optimal results which raises some issues which affect the results and will be explained later. In Figure 15 is depicted the resulting visualization of the voxel grid. As seen in the figure, the red voxels are the ones that are occupied by the mesh (top-left), the green-blue the ones that are 'Unknown' (top-right) and the purple ones the 'Seen' voxels (bottom-left).

To analyze the results, the existence or absence of 'Unknown' voxels will be discussed since they are complementary with the 'Seen' voxels. Starting from the exterior side of the building as it is visible in Figure 15 it is noticeable that there are too many voxels marked as 'Unknown'. This indicates that there is a big volume of the building that has not been visited. Although, this is not the case since there are many 'Unknown' voxels between the the 'Scanned' and the 'Mesh' voxels. The reason for this is the thickness of the wall as shown in Figure 14 and the alignment of the mesh with the point cloud which is not 100% accurate. Therefore, to tackle that issue it would be interesting to implement a method that would automatically align the mesh with the point cloud. Upon examining the house's interior, it is evident that marking the 'Scanned' voxels was successful. As seen in Figure 16 where there are points available it correctly marks the corresponding voxels as 'Scanned'. The 'Unknown' voxels exist where there are no points captured, for instance on the right side of the couch. The cause of missing points during the scanning process can be either occlusion or the limited angle of the device sensor. Another case is visualized in Figure 17. In the ceiling of the room there is a gap where no points are captured. As a result, there are no 'Scanned' voxels there and consequently there are 'Unknown' around this gap because the voxel traversal algorithm was not applied towards this area. The cause of the missing points is that the iPhone has a quite narrow field of view and has to be pointed directly towards the surface being scanned. Figure 18 and Figure 19 depict two examples of hidden spaces that can identified. Figure 18, depicts the 'Unknown' voxels that are below the surface of the scanned couch. In a similar manner, Figure 19 depicts the 'Unknown' voxels that are no the underside of staircase towards the exterior wall. These two cases are good examples of hidden spaces that can be identified.

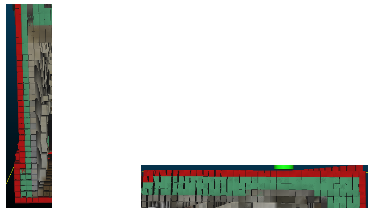

Figure 14. Section details of the 'Unknown' voxels (green) between the 'Mesh' voxels (red) and the 'Scanned' voxels (colored).

5 Suggestions for future work

This research serves as a proof of concept for identifying hidden spaces using the outer boundary of a building and the scan of its interior. It verifies the feasibility and the effectiveness of the suggested solution. For future work, the following suggestions are significant.

- It is crucial to change the way the void ('Air') voxels are marked. Towards this direction it is important to use an algorithm that uses ray-casting in a radius around the sensor and not only towards 'Scanned' voxels. This will reduce significantly the number of single voxels that remain marked as 'Unknown' at the end of the process.
- Automating the alignment of the point cloud with the building's outer boundary. This can be achieved with the following two approaches:
  - By using a GNSS-enabled drone. In this way, the drone could scan the entrance of the building from the level of the agents and apply feature matching between the data captured by the drone and the agents. In this way not only interior would be aligned with the outer boundary, but also it would be geo-referenced.
  - By using reference points outside of the building which will be measured with GNSS coordinates that are visible in the point cloud.
- Rectify the 'Unknown' voxels. To improve the results, it is suggested to implement post-processing steps that will filter 'Unknown' single voxels or groups of voxels based on some volumetric criteria.
- Depending on the requirements of the user (i.e. police or fire brigade) the size of the voxels might be set according to specific need e.g., it depends whether rooms or insides of walls need to be found.
- To improve the performance of the solution it is important to use an octree instead of voxels.
- To further improve this methodology it will be useful to incorporate detection of dynamic object, because currently the scenes are considered static.

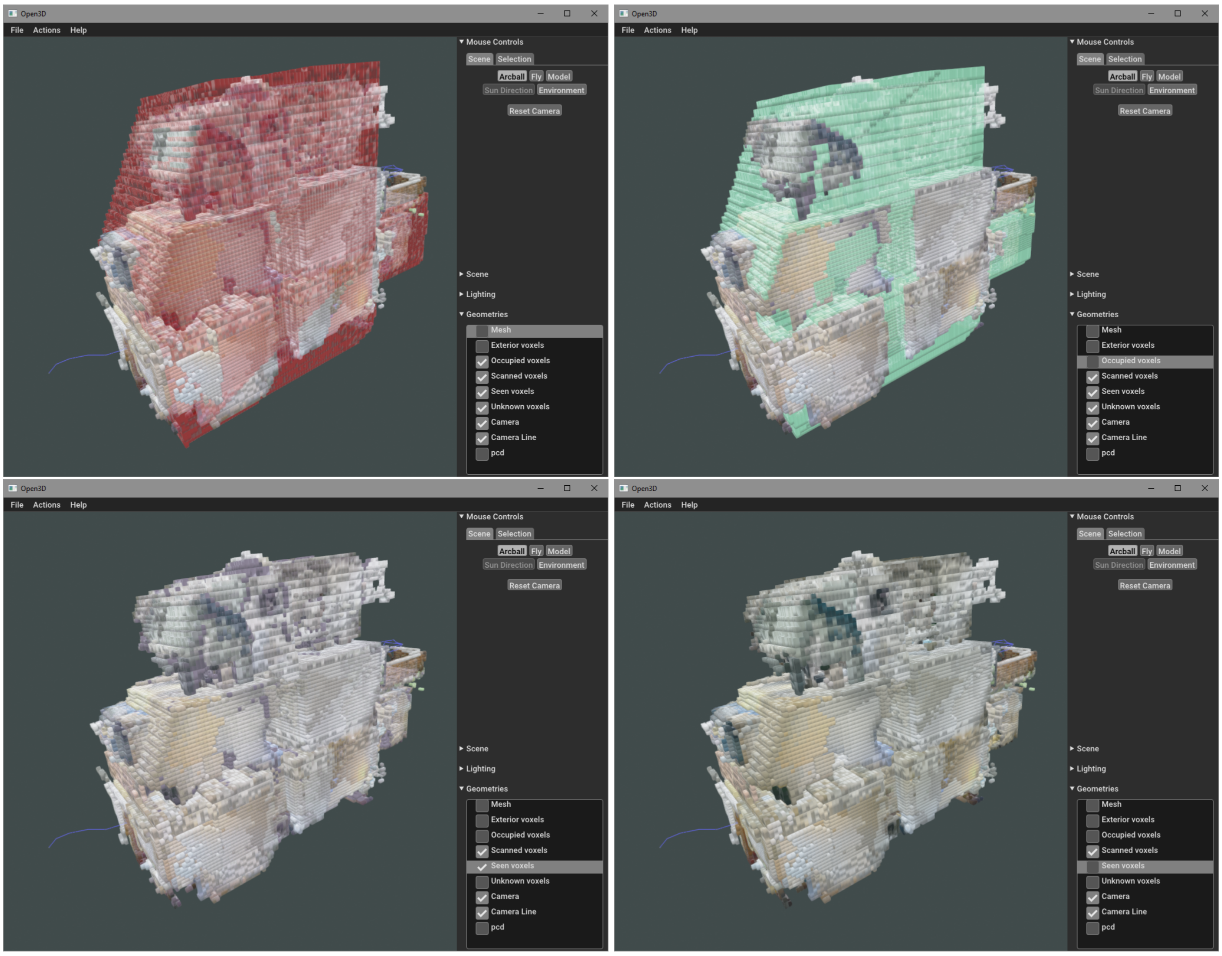

Figure 15. Visualization of the voxel grid. The colored voxels represent the 'Occupied' voxels - voxels that contain points.The red voxels are the ones that are occupied by the mesh, the green-blue the ones that are 'Unknown' and the purple ones the 'Seen' voxels. The blue line is trajectory followed by the camera.

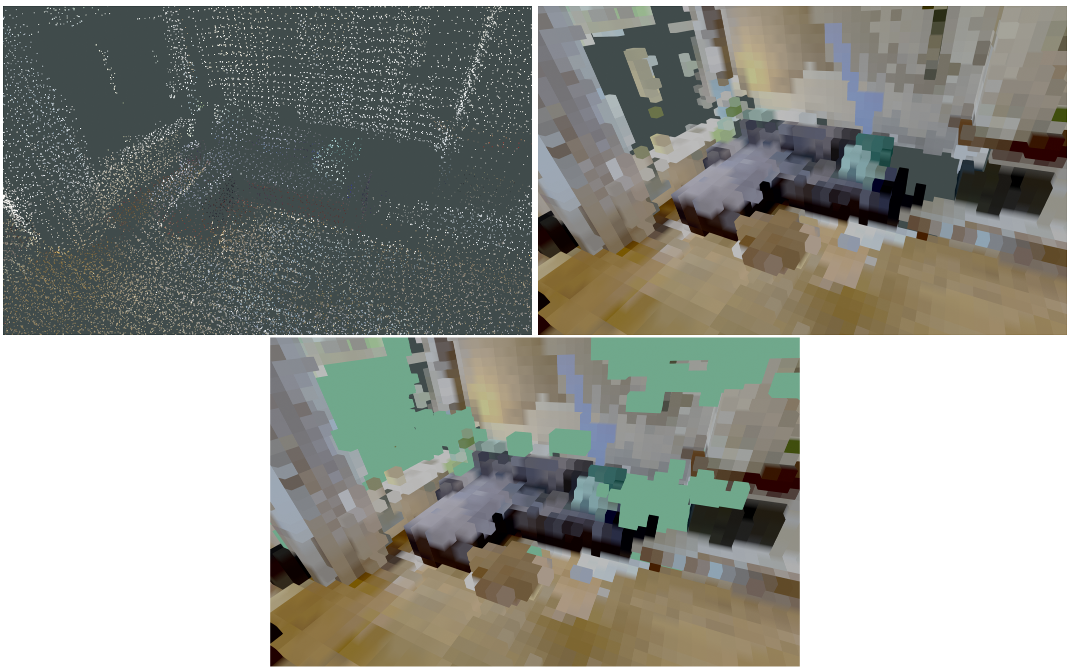

Figure 16. The living room of the scanned house and the resulting voxel grid.

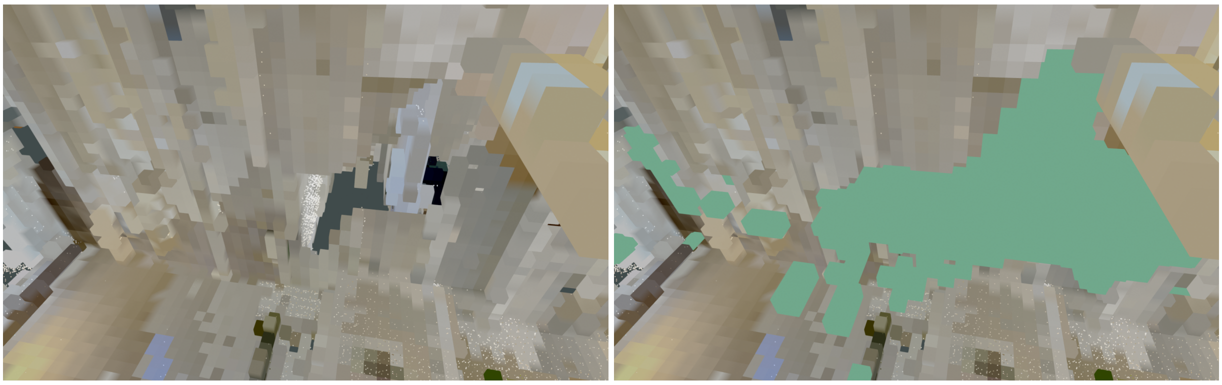

Figure 17. No data capture on the ceiling of the living room.

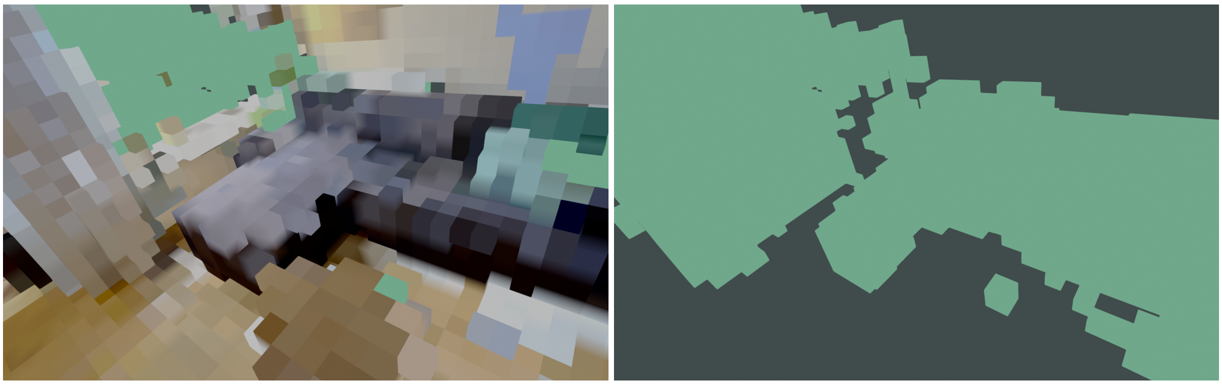

Figure 18. Hidden space in the sofa. On the left the 'Occupied' voxels are visualized along with the 'Air' voxels (in green). On the right only the 'Air' voxels are visualized which are hidden in the sofa.

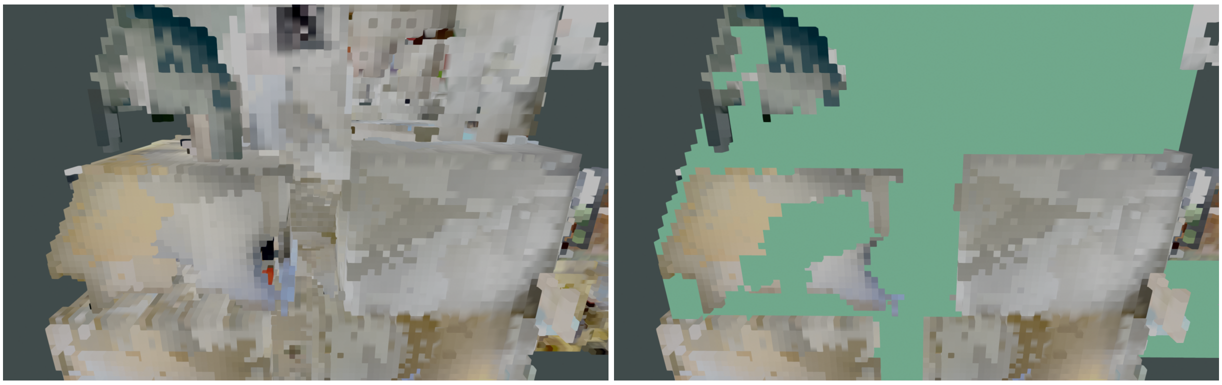

Figure 19. Hidden space under the staircase. On the left image there is an empty space visible which is under the staircase. On the right image it can be noticed that this space is filled with 'Air' voxels (in green).

6 Data and Software Availability

The research data and the code are available via the following DOI link. The final result of the process can be viewed in this 3D web viewer.

7 Declaration of Generative AI in writing

The authors declare that they have not used Generative AI tools in the preparation of this manuscript. All intellectual and creative work, including the analysis and interpretation of data, is original and has been conducted by the authors without AI assistance.

8 Acknowledgments

The authors would like to thank assistant professor Dr.ir. BM Meijers for the guidance regarding the development of the 3D web viewer. Last but not least, the constructive remarks of the reviewers were much appreciated and incorporated to improve the quality of the paper.

References

Abaspur Kazerouni, I., Fitzgerald, L., Dooly, G., and Toal, D.: A survey of state-of-the-art on visual SLAM, Expert Systems with Applications, 205, 117 734, https://doi.org/10.1016/j.eswa.2022.117734, 2022.

Amanatides, J., Woo, A., et al.: A fast voxel traversal algorithm for ray tracing., in: Eurographics, vol. 87, pp. 3–10, Citeseer, 1987.

Boguslawski, P., Zlatanova, S., Gotlib, D., Wyszomirski, M., Gnat, M., and Grzempowski, P.: 3D building interior modelling for navigation in emergency response applications, International Journal of Applied Earth Observation and Geoinformation, 114, 103 066, https://doi.org/10.1016/j.jag.2022.103066, 2022.

Dilo, A. and Zlatanova, S.: A data model for operational and situational information in emergency response, Applied Geomatics, 3, 207–218, https://doi.org/10.1007/s12518-011-0060-2, 2011.

Gehrung, J., Hebel, M., Arens, M., and Stilla, U.: A fast voxel-based indicator for change detection using low resolution octrees, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W5, 357–364, https://doi.org/10.5194/isprs-annals-IV-2-W5-357-2019, 2019.

Gyurgyik, C.: fast-voxel-traversal-algorithm, https://github.com/cgyurgyik/fast-voxel-traversal-algorithm/tree/master, accessed: 2024-09-05, 2023.

Labbé, M.: RTAB-Map, http://introlab.github.io/rtabmap/.

Smit, B.-P., Voûte, R., and Verbree, E.: creating 3D indoor first responder situation awareness in real-time through a head-mounted AR device, ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, V-4-2021, 209–216, https://doi.org/10.5194/isprs-annals-V-4-2021-209-2021, conference Name: XXIV ISPRS Congress <q>Imaging today, foreseeing tomorrow</q>, Commission IV - 2021 edition, 5;9 July 2021 Publisher: Copernicus GmbH, 2021.

Tseng, P.-Y., Lin, J. J., Chan, Y.-C., and Chen, A. Y.: Real-time indoor localization with visual SLAM for in-building emergency response, Automation in Construction, 140, 104 319, https://doi.org/https://doi.org/10.1016/j.autcon.2022.104319, 2022.

Xu, Y., Tong, X., and Stilla, U.: Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry, Automation in Construction, 126, 103 675, https://doi.org/10.1016/j.autcon.2021.103675, 2021.