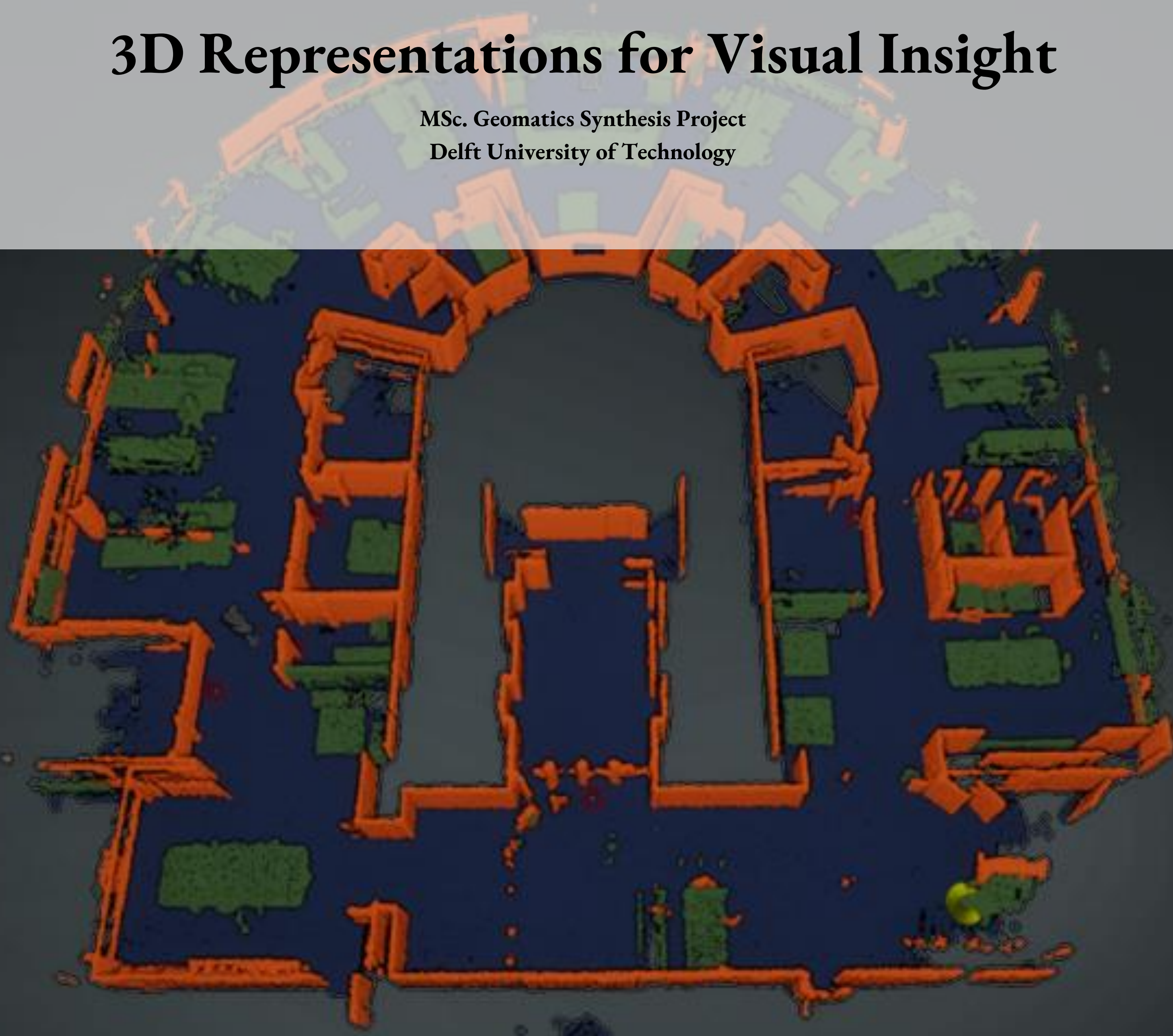
3D Representations for Visual Insight

MSc. Geomatics Synthesis Project
Delft University of Technology



Runnan Fu
Yuzhen Jin
Zhenyu Liu
Xenia U. Mainelli
Theodoros Papakostas
Linjun Wang

3D Representations for Visual Insight

MSc. Geomatics Synthesis Project
Delft University of Technology
June 18, 2021

Runnan Fu
Yuzhen Jin
Zhenyu Liu
Xenia U. Mainelli
Theodoros Papakostas
Linjun Wang

Mentors:
Edward Verbree
Robert Voûte

Abstract

As a method that can accurately represent 3D spatial information, point cloud visualisation for indoor environments is still a relatively unexplored field of research. Our client for this project, the Dutch National Police, requested a variety of potential solutions for visualising (unfamiliar) indoor environments that can be viewed by both external command centres, and internal operations units. Currently, unknown interior layouts (or layouts that are different in practise to what is stated on paper) can have serious, sometimes even life-threatening, consequences in time-sensitive situations. This project uses a game engine to directly visualise point cloud data input of indoor environments. The primary aim is to find ways of clearly communicating a point cloud of an environment to a layman viewer through intuitive visualisations, to aid decision-making in high-stress moments. The final product is a variety of visualisation concepts, hosted within a game engine in order to allow users to navigate throughout (part of) a building, and customise certain interaction features. To aid the layman viewer, various interpretation methods (e.g. cartography) are considered. The Unreal Engine 4 (UE4) project was designed and developed based on the requirements given by Dutch Police, and consisted of 4 modules: data preprocessing, render style, functional module, and User Interface (UI). An indoor point cloud dataset is used for the implementation, while corresponding mesh and voxel models are also respectively generated and evaluated as reference objects. The implemented software product is evaluated based on a Structured Expert Evaluation Method and finally our project result demonstrates that point cloud has unique advantages for visualisation of indoor environments especially in pre-processing efficiency, detail level, and volume perception.

Table of Contents

1. Introduction

Demand for visualisations of 3D environments has been on a continuous rise over the last half century, as technological developments offer opportunities for more realistic, more detailed, and more complex techniques. The potential that 3D environment visualisations have, is immense; from game design to modelling scientific experiments, a plethora of fields have discovered the value that visualising 3D environments has. Representing 3D objects using methods like boundary representations, meshing, and voxelization, are common. But the increasing demand for more precise, detailed, and up-to-date models, puts these methods at a disadvantage: these visualisation techniques lose detail by making assumptions about environments as they interpolate (and sometimes even extrapolate) the available data, and require a processing step to convert the input data into the final model, which takes time. Point clouds on the other hand, retain more detail, can be more easily updated, and are closer representations of reality due to their being (usually raw) sensor output. They can provide color, geometry, and attribute information, which can then be used to extract semantic information. A point cloud's disadvantages are that a layman viewer can have difficulty in recognising what a point cloud is representing (a low number of points exacerbates this), and that too high a number of points will clutter a visualisation. Finding the right visualisation techniques to perform upon point cloud data to overcome the disadvantages mentioned, and exploit the strengths of point clouds, is a relatively unexplored field of research (please refer to the Literature Review section for more on this). Though it must be mentioned, point clouds have been directly used in several non-academic applications, such as video games, notably *Scanner Sombre*, a cave exploring experience, and educational applications like the London Science Museum's Shipping Galleries project, that scans old museum exhibitions for archiving. But given that academic attention has been limited, this project's research focus lies in the cartography of point clouds; can we harness the potential that point clouds have, to as effectively (or more effectively) convey the contents of an environment to a layman, in the way that other 3D environment visualisation techniques already can?

The client for this project was the Dutch National Police, who requested a variety of visualisations of indoor environment point cloud data, for use by both command centres and first responders to a scene. The police force often encounter situations in which unfamiliar buildings need to be entered, where floor plans can be out of date, or real-time situations are different to what was briefed. A lack of a quick-to-comprehend and up-to-date overview, can have serious — and sometimes even life-threatening — consequences in time-sensitive situations, when command centres direct first responders on the basis of out-of-date information. This project focused on the visualisation and perception of indoor environments, with an emphasis on the direct use of point clouds. Alongside the research aim of exploring point cloud cartography and potential, the intended final product for this project's design element was a variety of visualisation concepts, that would have a focus on being intuitive, so that they are easy for a viewer to understand in high-stress moments. Using a game engine to host a demonstration of the indoor environment visualisation options we developed, allows users to navigate throughout a (part of) building, and customise a variety of features. To aid the layman viewer, we considered various interpretation methods (e.g. cartography), and options for interaction with the data (e.g. adjusting the transparency of particular features). The process of software design and development was a key aspect of this project, and integral to the success of our two final products – this paper and our visualisation demonstration. We compared our direct point cloud data visualisations with other visualisation techniques like meshing and voxelization, reporting the advantages and disadvantages that the different methods have, with regard to visualisation effect, ease of perception, and processing efficiency.

Section 2 explains this project's objectives, and Section 3 expands upon the objectives, laying out this project's research questions. Section 4 details the methodology for this project, and Section 5 dives into a literature review of existing work related to this project. Section 6 describes the requirements defined for this project's visualisation output, and Section 7 tackles the core of this project, namely Project Design & Development, explaining how we built our product prototype. Section 8 demonstrates the results, i.e. the visualisation(s). Section 9 describes and evaluates our results, and Section 10 provides some concluding remarks on this project. Finally, Section 11 dives into suggestions for how this work can be developed in further research beyond the scope of this project.

2. Objectives

Our objective in this synthesis project was to work towards finding innovative and creative methods of visualising indoor point cloud data. Visualisation of indoor point cloud data may have not yet received much attention in the field of research, but has great potential for situations where incoming geographic information needs to be visualised in (near) real-time, and where interpolation of data is not desired. The intended output is, expansion upon existing research regarding point cloud visualisation potential, specifically with regard to real-time indoor data, and creating a visualisation concept that demonstrates the ideas explored, showing the strength and value of direct point cloud use in data and environment visualisation. In the following section, we state our research question, and break it down into the sub-goals that we needed to achieve in order to be able to answer our initial question.

3. Research Questions

Our main research question is:

"Using a game engine, can we create unique or insightful visualisations from point clouds, that provide clear perceptions of indoor environments, to assist police in first response activities?"

With limited project time, we focused on the visualisation of the indoor point cloud data of a single storey in one building. Additional functions, such as indoor positioning and path planning, were not our top priority, considering we wanted to predominantly explore point clouds, despite appreciating their potential for enhancing the visualisation's innovativeness. Our first main goal for this project was having a semantically enriched point cloud, where points within the cloud have at least one category, namely the type of architectural feature or other object that they represent. Having a point cloud with this information, meant that we could move onto our next goal, which was importing the (now sufficiently classified) point cloud into a suitable game engine, and being able to work with the point cloud in such an application. The third goal of this project was the most important one for the design element of the project; finding ways of representing and visualising the data successfully. For this, we had a list of requirements and time set aside for creative exploration, to guide our software design process. Our final main goal was to reflect on our method, and compare our output with other indoor visualisation techniques, identifying its strengths and weaknesses, and where our project could be taken in the future, beyond the scope of this MSc. Geomatics synthesis project. Below is a more detailed explanation of each of these four main goals that guided our project.

3.1. Indoor Object Recognition and Semantic Marking

We needed a way of segmenting and recognising building features of indoor environments (such as walls, roofs, and floors) and other indoor objects, from a raw point cloud dataset. These features would be used to form a closed or semi-closed indoor space, for which we would soon need to find a way of allowing a user to navigate within. Though initially having planned for needing to classify the data ourselves, we ended up receiving semantically enriched data from our client. The labelling of points was fairly basic, and there were some incorrect classifications as well as stray points without classification, but receiving this classified data greatly helped in speeding up this stage of the project. More information on point cloud segmentation is given in Section 7.2.1.

3.2. Integration of Point Cloud Data with 3D Game Engine Software

Most current, major 3D game engine software, such as *Unity* and *UE4*, supports point cloud data processing. For this sub-problem, the primary goal was to select a suitable 3D game engine, which required a comprehensive assessment on the difficulty of using the software, the level of support for point cloud data, and how effectively the software can produce the desired visualisations. Two teams assessed the two most promising game engines, *Unity* and *UE4*, and recorded their findings using the same list of evaluation points, which guided our decision making. Details on this evaluation process can be found in Section 7.1.

3.3. 3D Visualisation and Perception Suitable for Crisis Environment

Having decided upon the game engine to be used, we moved on to the core goal of this project, which was exploring ways of visualising point clouds, such that they can be of use in the client's crisis situations. This required a preliminary literature review of best visualisation practices, and the main bulk of this project can be found in the UE4 Project Development & Design phase. Among other considerations, this third goal included removing disruptive elements, exploring the usefulness of different perspectives (first or third person view, and bird's eye view), and determining how to best visualise different architectural components at an appropriate level of detail.

3.4. Reflection on our Method, and Comparison with Other Work

Our final goal was to complete some form of user testing, in order to be able to critically reflect on our project and its fulfillment of the initial requirements list, and identify weaknesses that future work could strengthen. We also wanted to compare our project with other more common 3D environment visualisation methods, like meshing and voxelization, in terms of visualisation effect, enhancement of perceptual ability, and processing efficiency. The intention of this was to highlight in which situations point clouds could potentially be a more successful form of visualisation.

4. Methodology

Unlike natural science research or behavioral science research, which is concerned with using theories to explain phenomena and to explore new truths, Design Science Research (DSR) focuses on design and investigation of artifacts, including algorithms and methods, in a problem context for fixing issues in that context. So, the Design Science Research Methodology (DSRM) (Hevner & Chatterjee, 2010; Wieringa, 2014) has obvious solution-oriented characteristics.

As described in Section 2, the outputs of this project are designing a 3D indoor point cloud visualisation, which is a typical artifact, and a design science research report. Therefore, this report's methodology structure follows DSRM requirements (Hevner et al., 2004; Johannesson & Perjons, 2014; Peffers et al., 2007) of being organised into five phases: (1) the determination and description of a problem, (2) the definition of goals for the product, (3) the design and development of the *UE4* project, (4) the implementation of the product, and (5) the evaluation of the product.

4.1. The Determination and Description of a Problem

Design Science Research (DSR) needs to specify its research problem in the first phase, which can be refined into design problems and knowledge problems (Wieringa, 2014). The determination of research problems should be based on user demands and relevant literature to show the importance, value, and places of these problems in visualisation of the 3D indoor environment. Since the project is closely related to practical application scenarios, information from the industry is also very important. The work for this phase was essentially completed through means of the Project Initiation Document of this synthesis project. However, not all details of the research problem could be known at the start of the project, and so its details should be iterated continuously throughout the entirety of the project, based on feedback from result implementation and evaluation. The final version of relevant work done for this phase, is discussed between Section 1 and Section 3 of this report.

4.2. The Definition of Goals for the Product

The second phase is closely related to the first. Due to the solution-oriented characteristics of DSRM, research questions need to be translated into the goals of the product. This translation extracts and summarises the requirements for the product from the users and relevant literature. Due to limitations of time and manpower, these product requirements have been grouped into four priority categories: *MUST HAVE*, *SHOULD HAVE*, *COULD HAVE*, and *WILL NOT HAVE*, based on MoSCoW analysis and focus on high-priority items. The relevant works of this phase are shown in Section 6.

4.3. The Design and Development of the UE4 Project

In this phase, the *UE4* project should be designed and developed based on the requirements from Section 4.2. First, we compared the two mainstream 3D game engine platforms (*Unity* and *UE4*) and chose the one that is most suitable for this project. Then the workflow of overall module design for this *UE4* project is shown in this phase. For each module, preprocessing, including UI system, functional system, and render styles, the design purpose and design of its subcomponents are introduced. The details of this phase are given in Section 7.

4.4. The Implementation and Testing of the Product

In this phase, the point cloud dataset is imported into the software product to present the results and show the preprocessed data, main UI elements, and all functionalities that can be used to prove that this product can meet basic requirements of this research, which are the *MUST HAVE* items in PID's MoSCoW table. The result of this phase can be used to improve the definition of project goals, *UE4* project design and development in previous phases. The details of this phase is shown in Section 8.

4.5. The Evaluation of the Product

Finally, the implemented software product needs to be evaluated by user testing for how well it meets the requirements, such as whether the *SHOULD HAVE* and *COULD HAVE* items in PID's MoSCoW table are met based on self assessment (such as compare with mesh and voxel model) and feedback from CGI and the Dutch National Police. The Dutch National Police is the final user of this software so it has a deep understanding of the actual requirements of the crisis indoor environment. CGI is an information technology consulting and integration company, which has many exploratory experiences in the field of indoor 3D visualisation and can provide technical advice from an industry perspective. A comparison between our result, mesh, and voxel models is also necessary in this phase. The result of this phase can also be used to improve the previous phases and form the conclusion of this research report. The details of this phase are shown in Section 9.

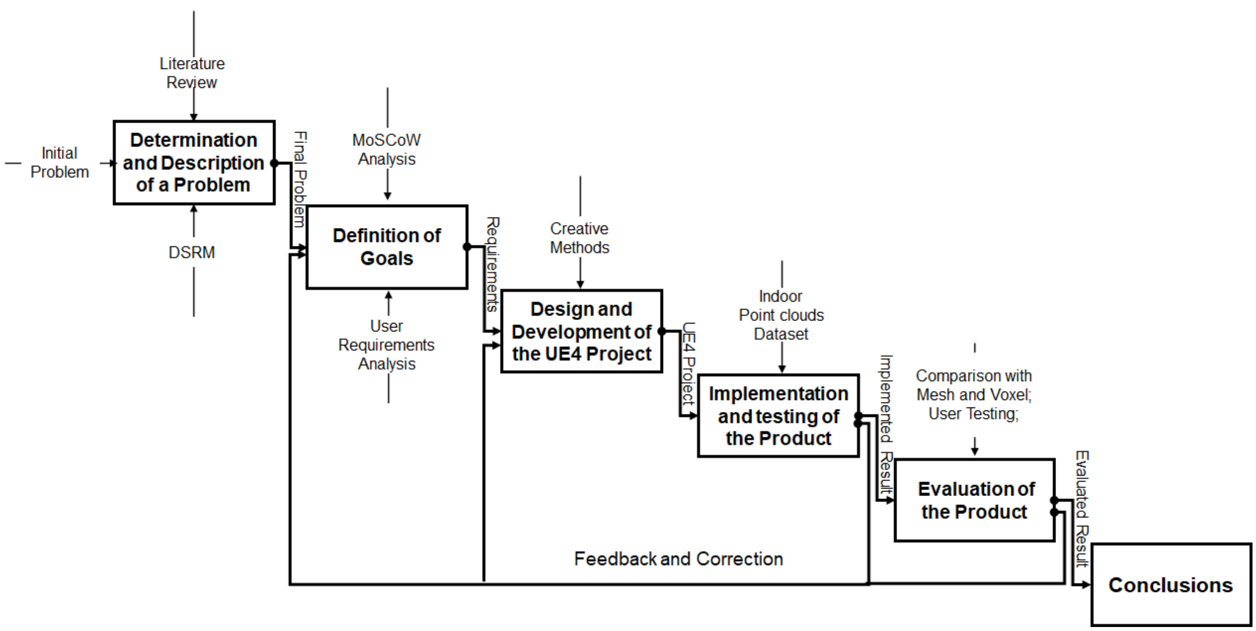


Figure 1: DSRM workflow

5. Literature Review

This section dives into related work to form a basis for investigating the added value of point cloud based visualisations and visualisation techniques for indoor point cloud data. The first subsection discusses the existing work related to 3D data representations including point cloud models, mesh models and voxel models. The second subsection describes existing research on visualisation of point cloud models and visualisation techniques for indoor point cloud models. Finally, existing research on the application of game engines for point cloud visualisation will be discussed.

5.1. 3D Data Representations

Point cloud data cannot be integrated into many 3D applications, and is therefore typically used as input to generate 3D mesh models and voxel models, which are then used for further analysis and applications (Xiong et al., 2013). Some researchers argue that mesh models provide more comprehensive representation compared to raw point clouds, as meshes can exploit the most important geometric features of a 3D object (Lv et al., 2021). Additionally, with continuous, smooth surfaces, meshes can deal with the noise and occlusion problems that point clouds have, which has an impact on applications such as scene interpretation (Bassier et al., 2020). Voxel models are also a common way to represent point cloud data in the applications of air pollution, noise, and wind analysis, and (indoor) navigation (Gorte & Zlatanova, 2016). In comparison to point clouds, voxel models are easy to use and understand, and algorithms to process them are typically much simpler, which also makes them more reliable, robust, and easy to parallelise (Nourian et al., 2016).

Figure 2: Different representations of 3D point cloud data. (Source: Poux, 2021)

However, extracting mesh and voxel models from point cloud data is time consuming, and can result in some degree of information loss or error (Bassier et al., 2020). While tools exist to automatically generate mesh and voxel models from point cloud data (Bassier et al., 2020; Julin et al., 2019), an additional computation and generation step is still needed, unlike with the direct use of point clouds. In using original data, point cloud models are closer to the real-life situations they depict (Cao et al., 2019), which is important for applications such as training exercises. In addition, recent development of algorithms and new rendering platforms can sufficiently address noise, occlusion, uneven data distribution, and sparsity problems of point clouds, and better support the direct use of point clouds. For the reasons above, direct point cloud visualisation and analysis is beginning to attract attention from academia and industry (Poux & Billen, 2019).

5.2. Visualisation of Point Cloud Models

To be able to use and analyse point clouds to better understand the built environment, it is important to perform point cloud segmentation and classification, and then visualise point clouds in a proper way. For point cloud segmentation, numerous approaches have been applied and developed, including RANSAC, PCA and region growing (Che et al., 2019). After clustering points into different segments, semantic features (e.g. ground, roof, trees or wall) can be extracted from segments by classification methods. Many authors (Bremer et al., 2013; Hackel et al., 2016; Munoz et al., 2008) have successfully extracted various features from point clouds using rule-based, random forest and other methods. For instance, (Bremer et al., 2013) can classified point clouds into ground, ground inventory, wall, wall inventory, roof, artificial poles and trees. However, these studies are often devoted to extracting as many classes as possible and pursuit of higher precision processing results, and therefore the algorithms are complex and time-consuming. In our project, users are more concerned with the general characteristics of the space, and a fast data pre-processing time is also important. Therefore, we think it is necessary to create a simpler classification for processing indoor point clouds in this project.

In terms of point cloud visualisation, existing research mainly focuses on data integration for a more detailed and accurate 3D model (Abdullah et al., 2017; Nebiker et al., 2015), web-based visualisation system for interactive exploration and inspection of point clouds (Discher et al., 2019), and the use of immersive VR display systems for a better user experience (Tredinnick et al., 2018). However, there is a lack of research on how to visualise point clouds to provide clear perceptions of indoor environments.

Research on visualisation principles and techniques for 2D geometries such as points, curves, and surfaces is relatively mature, and many of the findings can be simply extended to point clouds and 3D space. For instance, (Neuville et al., 2018) summarised over 20 static retinal variables such as hue and lightness, and used them to produce more effective 3D city model visualisation results. In addition to these variables derived from 2D cartography, there are many variables unique to 3D objects, such as perspective, shadows, and shading.

Several authors have studied 3D visualisation and came up with very interesting ideas. For instance, (Török et al., 2014) found that when viewed from the ground with a self-centered frame of reference, the user is able to find a relatively better route to the target location than when viewed from a bird's eye view. (Li et al., 2013) suggested that only the most relevant 3D content should be selected and visualised in a more attractive way in order for users to effectively focus on and understand the information of interest. There are also studies focused on 3D indoor visualisation techniques. For instance, (Florio et al., 2019) suggested a combination of a scene elements removal function, transparency function and an exploded map to reveal occluded parts.

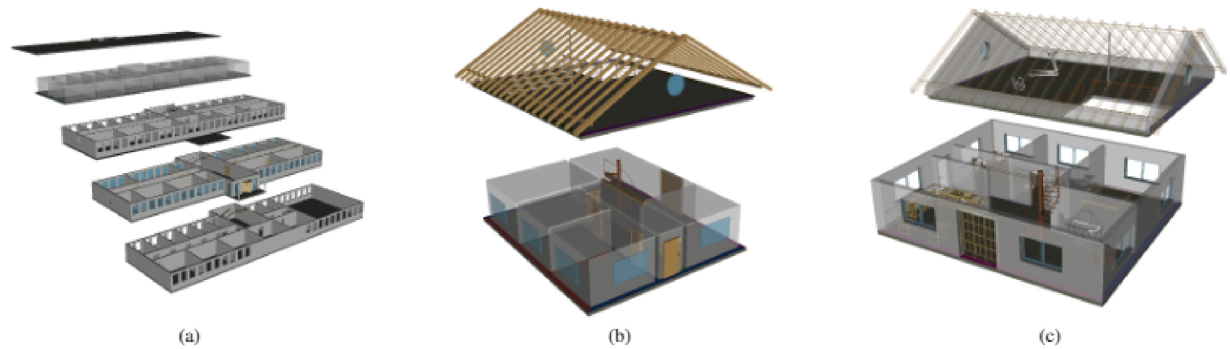


Figure 3: A combination of exploded (a), cut-away (scene elements removal) (b), and ghosting views (transparency) (c) to reveal inner building structures. (Source: Florio et al., 2019)

In general, while there has been much research on 3D and indoor visualisation techniques, the discussion of integration with point clouds and special usage scenarios is still lacking, making this study innovative and challenging.

5.3. Use of Game Engines for Point Cloud Models

Game engines such as *Unity* and *UE4* are 3D real-time creation platforms widely used to develop 3D games with realistic visual effects and immersive experiences, and have seen adoption by other industries such as film and television industry ("Unreal Engine," 2021). These software not only facilitate game application creators, but also allow users to experience, control, and analyze objects of interest in a virtual environment, which is useful for remote teaching, learning, and training (Kharroubi et al., 2021). (Barczak & Woźniak, 2020) also emphasized the advantages of game engines in terms of reuse. With few modifications, components and functions of game engines can be extended and integrated to a variety of logic and rules, which is useful for creating an application that works with multiple datasets.

In the fields of architecture, archaeology and cultural heritage, the use of game engines to create fine-grained, fast-rendering, interactive 3D models has attracted a great deal of attention (Kharroubi et al., 2021; Merlo et al., 2013; Virtanen, Daniel, et al., 2020). Many authors have explored the possibilities and added value of visualising 3D (indoor) mesh models in game engines (Carbonell et al., 2020; Merlo et al., 2013; Virtanen, Julin, et al., 2020). The purposes of those research mainly are making virtual models closer to reality or supporting applications such as cultural heritage propagation and training. Some scholars have also studied the integration of point cloud data with game engines, and they mainly focus on developing new algorithms or tools to make game engines more supportive of point cloud visualisation (Kharroubi et al., 2021; Virtanen, Daniel, et al., 2020). On top, to our knowledge, there is no work on exploring indoor point cloud visualisation techniques to enhance spatial perception using game engines.

6. Product Goals Analysis

Heading to the Product Goals Analysis, the implemented software product needs to be evaluated by user testing for how well it meets the requirements, such as whether the *SHOULD HAVE* and *COULD HAVE* items in PID's MoSCoW table are met based on feedback from the clients. The MoSCoW table was used firstly for the project's structured workflow plan, and is also used in this phase to evaluate the level of integration of the final outcome. The predefined goals and final results will be evaluated.

Table 1: MoSCoW table

MoSCoW	Feature	Client	Result	Notes
MUST (HAVE)	*Viewable on a desktop viewer*	Eases the potential of the visualisation, and does not demand client to invest in additional hardware/software to use our product	DONE	Executable software desktop viewer.
	Navigable	A fundamental demand from the client, that the user is able to "look around" the represented environment	DONE	This includes features like panning, zooming, rotating, etc.
	Quick to render	A fundamental demand from the client	DONE	The visualisation will be designed for use in time-sensitive situations. This is the main reason for opting for a game engine
	Visible to both command centre, and those within environment	A fundamental demand from the client	PARTIALLY DONE	In order for the visualisation to be shown to those in-situ, the software must be integrated depending on what equipment the police uses (e.g. VR goggles, tablet device, smartphone device, Portable PC)
	Either a first person or third person view	The most important thing for the client, is that the visualisation is understandable	DONE	First-person view Third-person view Bird's eye view (navigable)
	Filter out unnecessary data	Client wants to have as clear and uncluttered a visualisation as possible – the target user is someone not necessarily acquainted with data viz, so it must be intuitive	DONE	Settled in the preprocessing stage -cleaning the point cloud,deciding on which data can be removed while retaining an understandable visualisation

SHOULD (HAVE)	*Customisability for viewer, can show/hide different features*	A request from the client	DONE	The software offers, switching styles, transparency and feature customisation (point size) functionalities
	Show the (close to) live positions of first responders within the building	A request from the client	PARTIALLY DONE	The software attempts to simulate the first responder's environment, providing visual insight of the scene
COULD (HAVE)	*Integrated first and third person view*	Laymen might use this visualisation, so being able to switch between views quickly, can help with orientation	DONE	Supported functionality of the software
	Include surrounding environment (outside of the building)	Client said that this would be nice to see	NOT DONE	Not supported- Will be a potential future research topic
	Multiple buildings visualised	Client mentioned this as a possibility, but not an expectation	NOT DONE	Not supported- Will be a potential future research topic
	Virtual Reality or Augmented Reality options		NOT DONE	Not supported, not explored
	Make use of voxels and planes (instead of purely the point clouds)	The client wants the clearest and most accurate visualisation, but doesn't mind how this is achieved	PARTIALLY DONE	Mesh models and Voxel models of the dataset were reconstructed and evaluated. Direct point cloud visualisation was finally used.
WILL NOT (HAVE)	*Identify objects of interest for the client*		NOT DONE	This aspect was evaluated , however not further explored

7. UE4 Project Design & Development

This section first gives a comparison between two mainstream 3D game engine platforms (*Unity Game Engine* and *UE4*), and how we choose the platform most suitable for this project. The second subsection then introduces module design and development ideas for our software product to be developed within that engine, based on the goals from [Section 6](#).

7.1. 3D Game Engine Platform

For the purpose of the project, in order to test our visualisations' rendering abilities and provide an integrated application for our client (the Dutch National Police) to use in first-response activities, there was the need for a platform to visualise, process, integrate and provide the product. We had to explore the visual (re)presentation and the cartographic insight potential of point clouds, without generating a concrete 3D model, thus avoiding changing the nature of the point clouds (e.g. point cloud to vector). The product software was designed in a game-oriented scope, to provide, if possible, insightful interactive functionalities, aiding police units in decision-making, SA (Situation Awareness). A game-oriented approach also offers the possibility of dynamic optimisation and further improvement in future research.

Aiming to find the most appropriate platform for the project's implementation, two well-known Game Engines were identified for evaluation. "A game engine is conceptually the core software necessary for a game program to properly run" ("Game engine," 2021). The core functionalities that a game engine typically offers are a rendering engine ("renderer") for 2D or 3D graphics, a physics engine or collision detection (and collision response), and localisation, among others. The two Game Engines that were set to test are *Unity Game Engine* and *UE4*. These engines were chosen for their popularity, reputation, and the fact that they are provided free of charge. They both also offer point-cloud plugins/modules, an essential feature for this project, which were tested as part of the evaluation to choose the most suitable game engine for the project. A brief summary of the evaluation comparing the engines' capabilities, is shown in the following table.

Table 2: Comparison between *Unity* and *UE4* game engines

Item of Evaluation	unity **Unity**	UNREAL ENGINE **UE4**
Supported programming language(s)	C#	C++
Point cloud plug-in/module (links, price, etc.)	2 plug-ins: - PCX (free): github.com/keijiro/Pcx - Point Cloud Viewer and Tools (not free) https://assetstore.unity.co	- LiDAR Point Cloud (free official plug-in) pointcloudplugin.com/ - Cesium for Unreal (free official plug-in, is suitable for UE version 4.26)

	m/packages/tools/utilities/point-cloud-viewer-and-tools-16019?locale=EN	
Number of points import supported	PCX: Easy import of dataset	LiDAR Point Cloud: Easy import of dataset
Loading time	PCX: Satisfactory	LiDAR Point Cloud: Satisfactory
Rendering time	PCX: Satisfactory	Satisfactory
Segmentation functionality	Not explored	Different parts (roof, floor, etc.) of the raw file can be merged.
Classification attributes exploitation	Not explored	Eligible

From the evaluation, *UE4* was selected as the hosting platform for the project's design and implementation. A major advantage of *UE4* is this team's former experience with C++, which is supported in *UE4* but not in *Unity*. Secondly, there's an official point cloud plugin in *UE4*, providing better support for the visualisation of point clouds, in comparison with *Unity*'s plugin, which is not official and also not free of charge.

7.2. Modules Design & Development

This subsection introduces the design basis and development ideas of four software product modules, including the data preprocessing, render styles, functional system, and UI system. The chart below describes the workflow of module design and development.

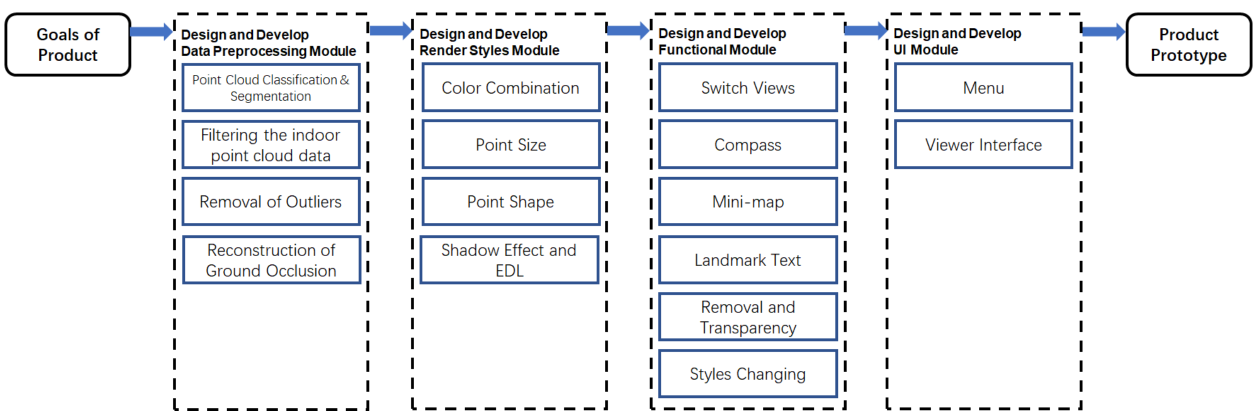


Figure 4: Workflow of modules design and development

7.2.1. Data Preprocessing Module

7.2.1.1. Point Cloud Classification & Segmentation

[Section 5.5](#) indicated the importance of classification and segmentation operations in point cloud applications. Segmentation groups the discrete points into different object classes, and this process can split the point cloud data into many small subsets, significantly reducing the computation complexity in subsequent preprocessing phases. Classification operations can give point cloud objects semantic information to better understand the environment. Segmented objects with semantics in the point cloud data may be demonstrated by means of different rendering styles, set with different parameters for outlier removal and occlusion repair algorithms, and switched between the displayed and hidden states independently.

In this project the point cloud data needed to be classified into 4 categories: (1) roof, (2) architecture components (walls, columns, etc), (3) non-architecture components (tables, cabinets, etc), and (4) ground. This division method can be applied to all indoor environments, and can meet the needs of visualisation. Further segmentation, such as splitting non-architecture components into separate objects, was considered unnecessary for the project's objectives. Over-segmentation not only increases the processing time, but also increases the amount of information available. Not all information is equally important in a crisis environment, and forces users to spend extra cognitive time sifting through the valid information. This is why we avoided overly complex classification methods in this phase.

Upon splitting the point clouds, the roof plane and ground plane should be extracted first. Plane detection methods, such as RANSAC and region growing, can find all the planes in the original point cloud data, in which the two planes with the largest number of points are usually roof and ground. Then the average Z-value and the variance of the Z-value for these two planes is calculated. The plane with the higher average Z-value is the roof, and the other plane represents the ground. The Z-value variance of the points in both planes is very small, a fact that can be used to check the extraction results. Secondly, we extract the architectural components and the non-architectural components. All the points except the roof and ground are placed on a XY plane to create a two-dimensional grid with the smallest block size possible. For each grid block, if the maximum Z-value of all points in it is less than the average Z-value of the roof (due to the accuracy and noise of the data, the tolerance is needed), all points in this grid block will be marked as non-architectural components, and otherwise be marked as architectural components.

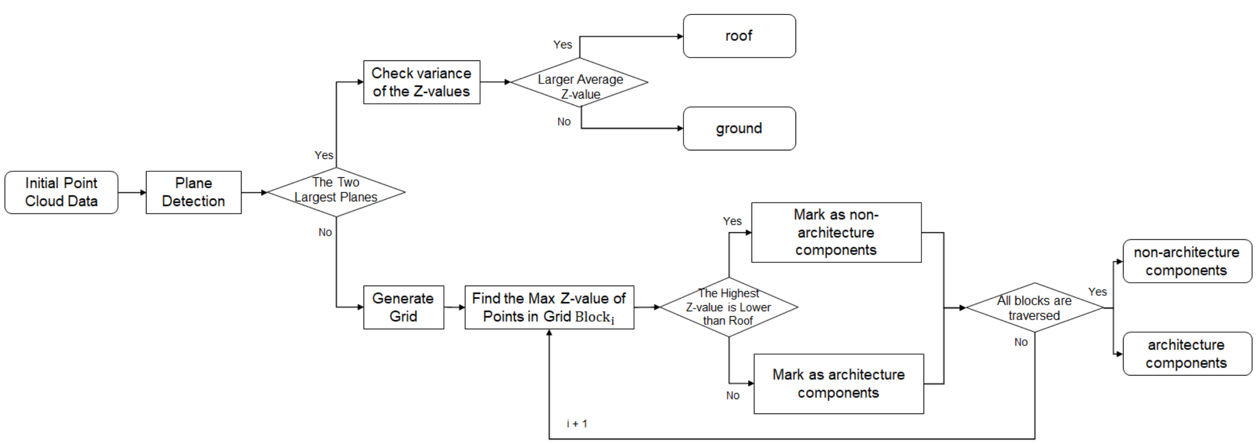


Figure 5: Workflow of point cloud segmentation

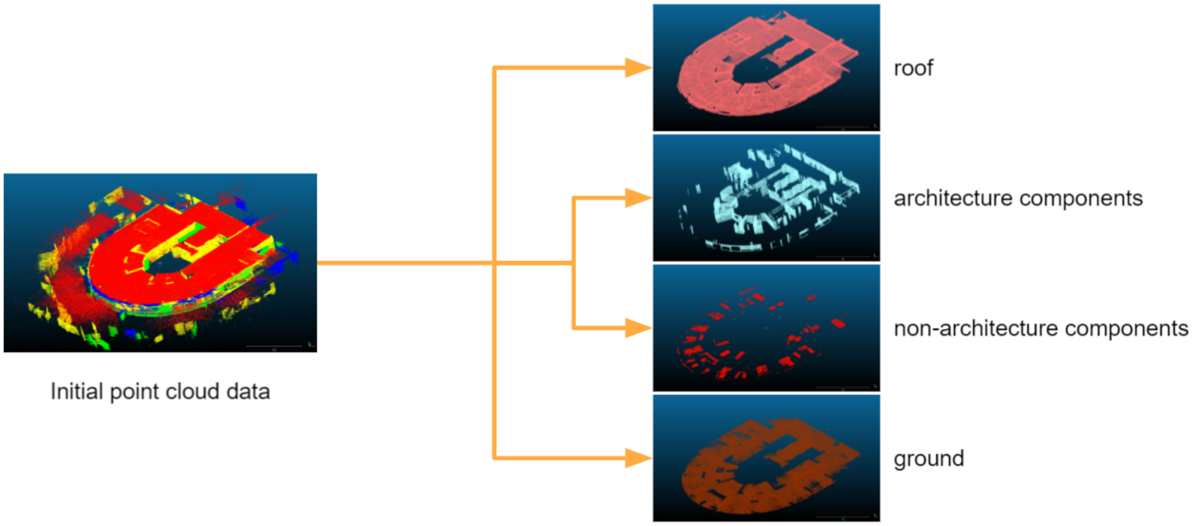


Figure 6: Four segmented classes with their corresponding semantic information

7.2.1.2. Filtering the Indoor Point Cloud Data

The indoor environment is usually not completely enclosed, which means that a LiDAR scanner inside a room can obtain part of the external environment objects from openings in the external wall, such as windows. We must filter out these external points from the indoor space, and remove them from the data. A 2D grid can be created and all the points can be projected onto it. Each grid block will be individually identified as an indoor area, only if there are a certain number of roof and ground points inside it, and the point density exceeds a set threshold. Only the points in the indoor area should be preserved.

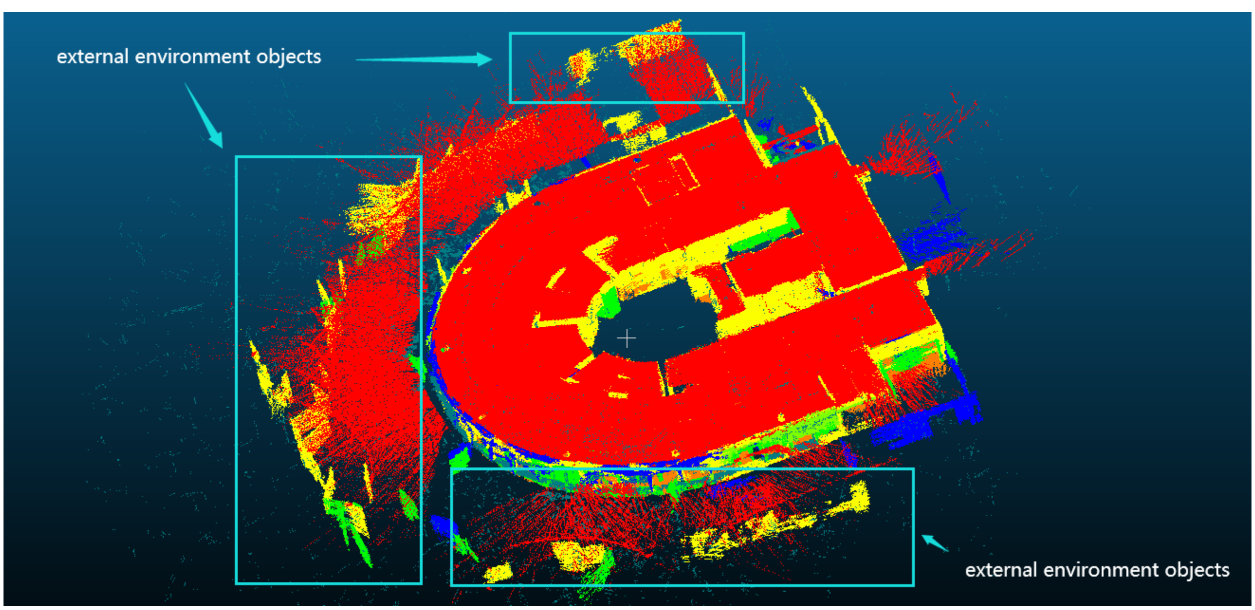


Figure 7: External Environment Objects

7.2.1.3. Removal of Outliers

Outliers are the noisy, sparse, and temporarily incoherent parts in point cloud data, which is a common problem in raw data captured directly by the scanner (Ning et al., 2018). This issue can result in unfaithful rendering results of point clouds. After the collision property is added to the point cloud model, the outlier can lead to limited exploration of the indoor environment. For example, noise may block the navigable area. Therefore the denoising and point cloud cleaning process is necessary in the preprocessing phase.

The outliers can be filtered from a point cloud based on the number of neighboring points they have in the search area. Different parameters, including radius of search area and minimum neighbor points number, can be applied to different objects. A good outlier removal method should find a balance between denoising, feature-preservation, and avoiding degradation of the input (Rakotosaona et al., 2019). For example, some objects, such as non-architecture components, need to retain certain characteristics and therefore cannot be over-smoothened.

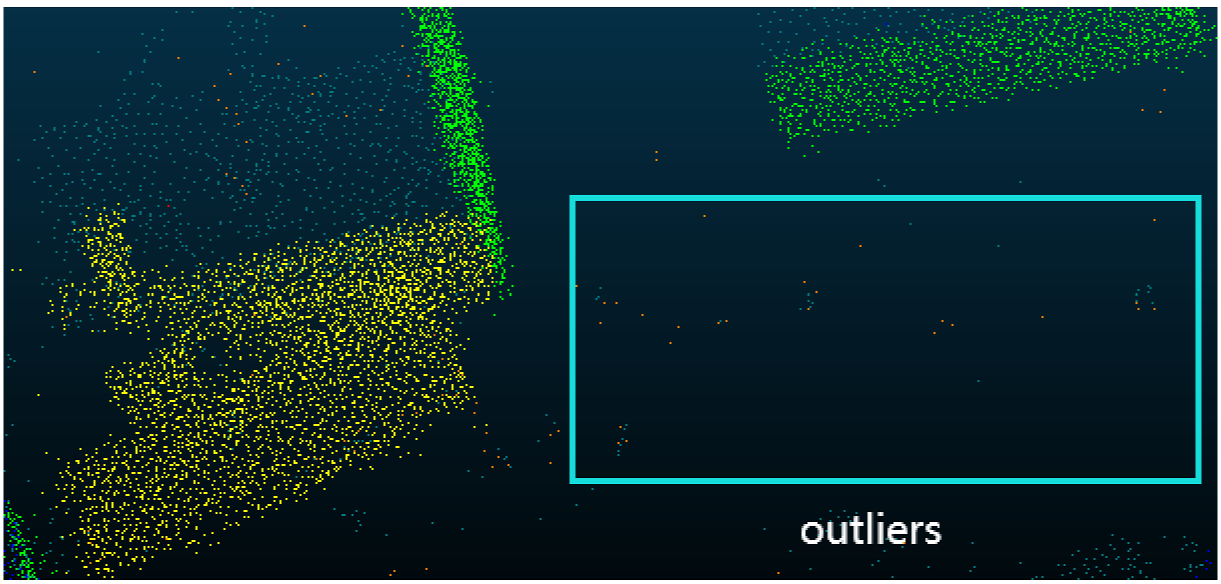


Figure 8: Point Cloud Outliers

7.2.1.4. Reconstruction of Ground Occlusion

Occlusion means that the collection of spatial data is incomplete, which is also a common issue in point cloud data (Balado et al., 2020). Occlusion according to the plane, can be divided into ground occlusion and facade occlusion (Friedman & Stamos, 2012). The former causes the user to fall through holes in the ground (no points in the occlusion areas) while exploring the indoor environment. The latter creates non-existent entrances and exists in enclosed or semi-enclosed spaces.

However, the reconstruction occlusion methods without introducing new supplementary datasets, are based on reasonable assumptions rather than real world data. So using this kind of method to solve point cloud occlusion problems runs the risk of adding incorrect spatial information to the data, which may be misleading to users. We need to avoid unnecessary occlusion reconstruction, such as most facade occlusions, because the shape of the occlusion area is usually irregular, which is easy to recognise with the user's eye. The repair of the ground occlusion is necessary, otherwise it will seriously affect the free exploration of the indoor environment. However, there is no need to repair the point cloud ground occlusion directly. What is needed, is the insertion of a transparent plane object with collision property, which can be considered as the real ground surface in *UE4*, and make it overlap with ground after importing the point cloud data into the game engine. This guarantees that the camera in the *UE4* project will not end up out of the bounds of the point cloud data region.

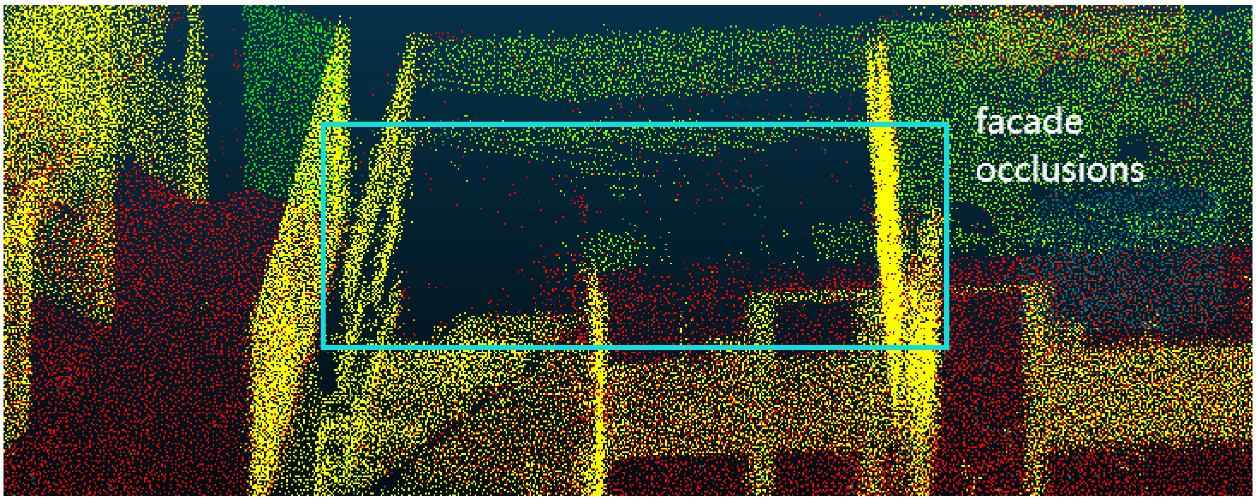


Figure 9: Facade Occlusions

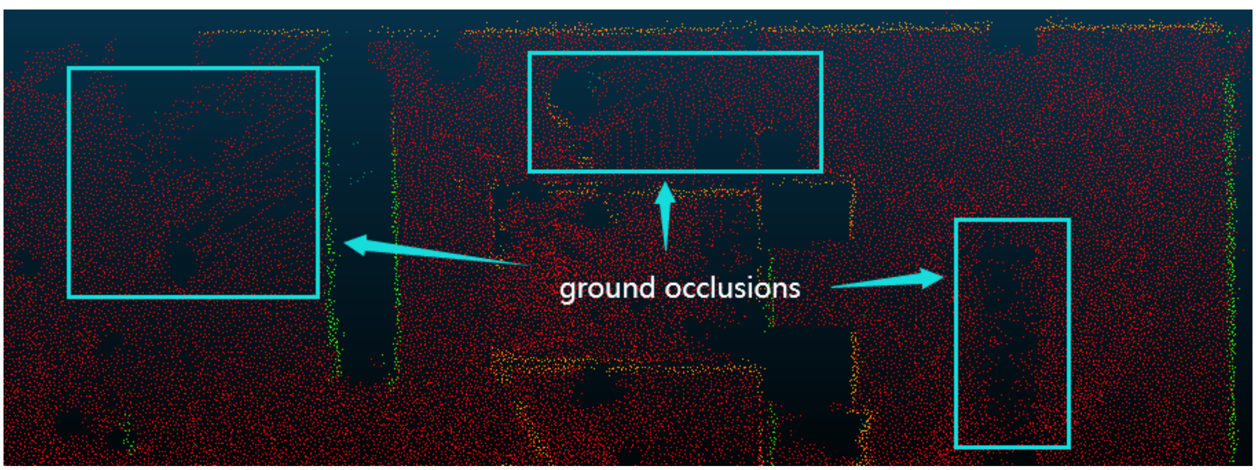


Figure 10: Ground Occlusions

7.2.2. Render Styles Module

In order to make point clouds more realistic and truly help users better understand interior spaces, we need to be aware of the broad variety of possible visualisations, and use visualisation techniques wisely. For principles important for a good visualisation, (Andrienko et al., 2020) has made summaries (Table 3), many of which can be simply applied to our project.

Table 3: Principles for a Good Visualisation (Adapted from: Andrienko et al., 2020):

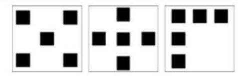
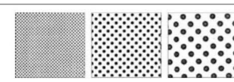
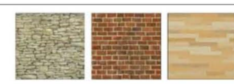
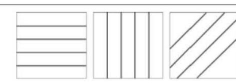
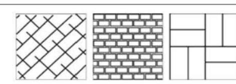
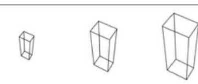
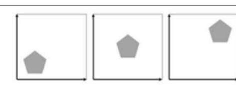
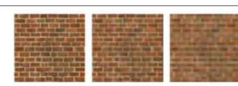
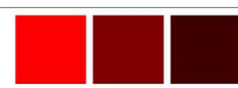
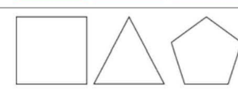
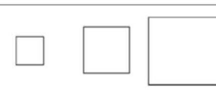
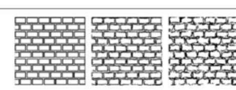
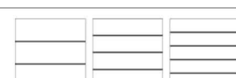
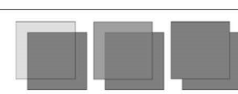
Principles	**Definition**
Utilise space at first	Taking into account the analysis tasks.
Respect properties	Make sure that the properties of the visual variables are consistent with the properties of the value domains of the data components.
Respect semantics	Knowledge of the domain and common sense should be used in the creation of visual displays.
Enable seeing the whole	The visualisations need to be designed in a way that it facilitates gain-ing an overview of all the data items that are shown.
Include labels, titles, and legends	The interpretation of patterns strongly depends on explanation of how the data components are encoded by visual means and appropriate labelling of them.
Avoid excessive display ink	Avoid visual components that do not communicate useful information.
Consider employee redundancy	Use redundant ways to communicate important information.
Enable looking at data from multiple perspectives	When the number of data components is higher than the number of effective visual variables then we should create multiple representations for exploring the data from different perspectives. Data items on multiple visualisations can be linked by interactive operations.
Rely on interactivity	This can facilitate comparison and complement visual representation in many other ways.

In terms of visualisation techniques, (Neuville et al., 2018) have summarised over 20 static visual variables (Table 4), as well as several 3D rendering parameters:

1. Projection: parallel or perspective;
2. Camera: position, orientation, and focal length;
3. Lighting: direct, ambient, or artificial light;
4. Shading;
5. Shadow;
6. Atmospheric effect;
7. Viewport variations: change the projection (parallel or perspective) progressively and degressively in order to efficiently reduce occlusion issues in 3D geospatial environments.

Based on these theoretical foundations, we selected several techniques that are suitable for our project for experiments in order to get a better rendering result.

Table 4: Static Visual Variables (Source: Neuville et al., 2018)

Visual Variable	Author (Date)	Example
Arrangement	Morisson (1974)	
Crispness	MacEachren (1995)	
Grain	Bertin (1967)	
Hue	Bertin (1967)	
Lightness/Value	Bertin (1967)	
Material	Carpendale (2003)	
Orientation	Bertin (1967)	
Pattern	Carpendale (2003)	
Perspective height	Slocum et al. (2010)	
Position	Bertin (1967)	
Resolution	MacEachren (1995)	
Saturation	Morisson (1974)	
Shape	Bertin (1967)	
Size	Bertin (1967)	
Sketchiness	Boukhelifa et al. (2012)	
Spacing	Slocum et al. (2010)	
Transparency	MacEachren (1995)	

7.2.2.1. Comparison of Different Color Combinations

We set three different color styles: one of the styles (the default one) is in brighter color, enabling users to achieve good visual effects in a strong light environment; the other two styles are in darker color, enabling users to achieve good visual effects in a low light environment.

For the default style, we use a color scheme that is closer to the real materials of objects to enable the user to quickly understand what is there in the indoor environment as soon as they start the program. Those two darker color styles can highlight the architecture and non-architecture respectively (with the bright orange color to attract user's focus) according to users' needs.

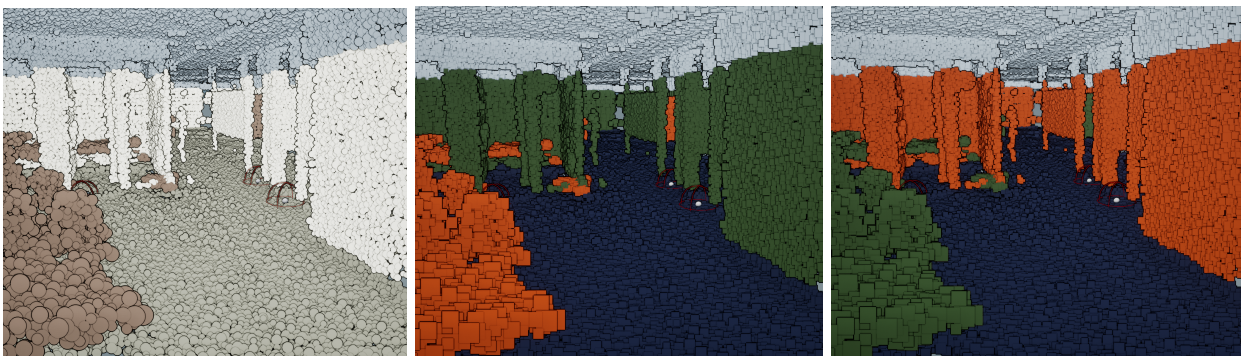


Figure 11: Default style (left), style 1 (middle) and style 2 (right).

7.2.2.2. Comparison of Color and Material

We experimented with the "material" used for the surfaces of the indoor environment, exploring how changing the texture affects the perception of the visualisation. As can be seen below in Figures 12 and 13, applying the Cobblestone_Rough(*UE4*) material to the voxel model, shows the greatest success in creating smooth surfaces that do not demand attention from the viewer. The mesh model is less successful in not distracting the eye, but does have more of an effect than when applying this material feature to the point cloud model. Unless the points within the point cloud model were to be increased in size such that the individual points' surfaces were to be visible in enough detail, adding this material variable serves little purpose. But upon points being increased in size too much, we run into the issue of large points degrading the visualisation in another way (see experiment 3. below), and thus this becomes a question of finding the correct balance between the different variable values, and determining whether or not introducing each new variable has a net positive effect, or simply adds to the complexity of the visualisation.

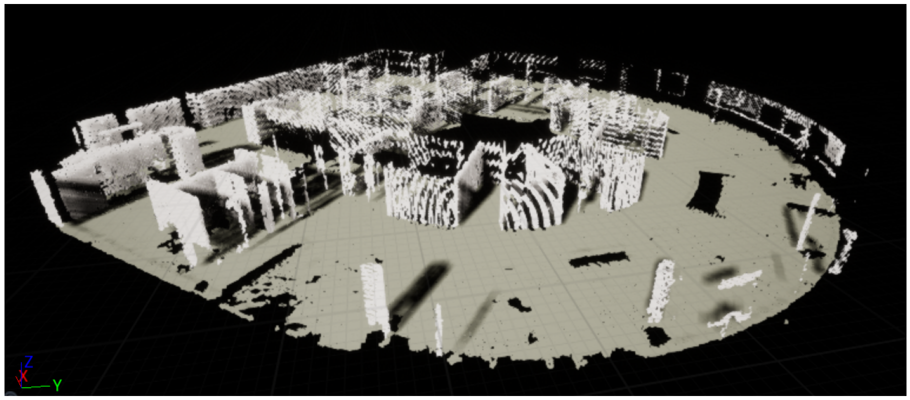


Figure 12: Voxel Model of Floor and Architectural Features, rendered with materials

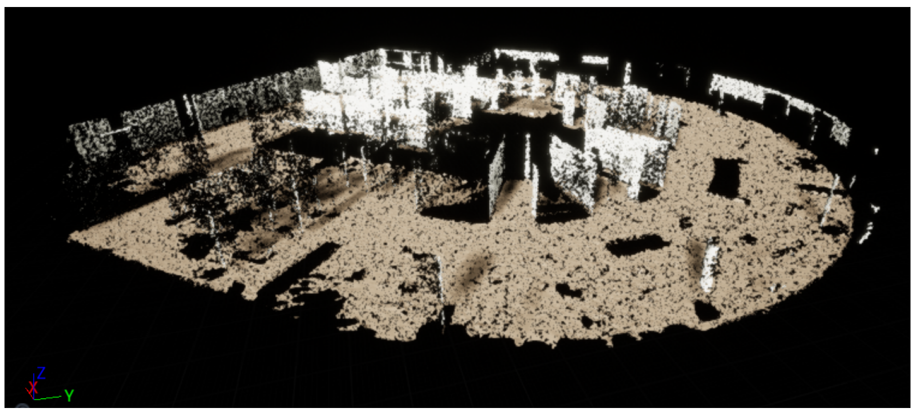


Figure 13: Mesh Model of Floor and Architectural Features, rendered with materials

7.2.2.3. Comparison of Different Point Size and Point Shape

The size of the points of the point cloud has an impact on the visualisation effect of the object. When the points are small (especially when the density of the point cloud is low), the formed object will not show specific shapes and contours, and the sense of volume will be low, which will reduce the user's perception of the object. When the points have larger sizes, a more complete surface of the object can be formed, but at the same time, details will be lost. With the *UE4*, we enable the user to adjust the shape and size of the points of the point cloud according to personal preferences to get a better visualisation effect. The adjustment of the point size makes it possible to circumvent the effects of the point cloud density such as forming voids between points when the density is too low (Kharroubi et al., 2019).

For data where the point density is already sufficient, if point size is setted too large, the structure of the object will be unclear. Since we want this visualisation to work with more point cloud data, we choose a medium point size as the default value and provide an adjustment slider for the user to customise the point size based on data characteristics and personal preferences, which is described in Section 7.2.3.6.

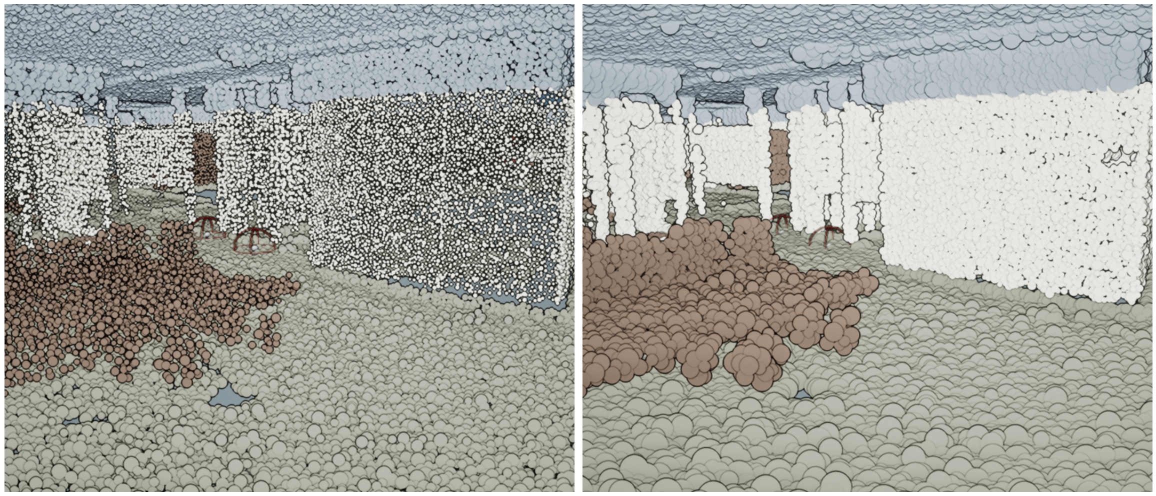


Figure 14: Reduce point size of wall and non-architecture (left) and enlarge point size of floor and roof (right)

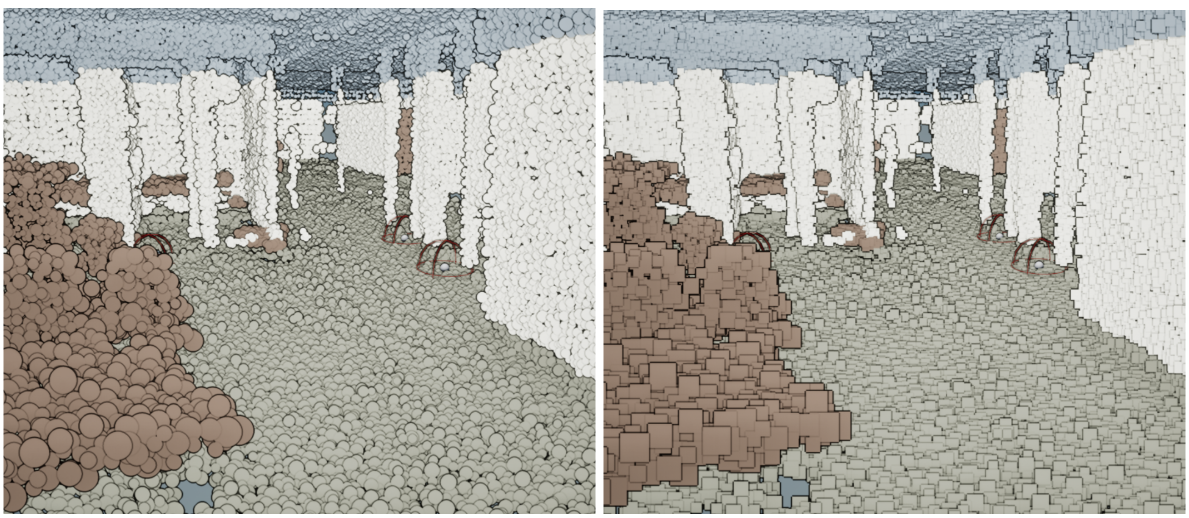


Figure 15: points in sphere shape (left) and points in cube shape (right).

7.2.2.4. Comparison of Shadow Effect and Eye-Dome Lighting (EDL)

Point cloud data often does not have realistic 3D and depth information, so additional techniques to enhance the depth perception of the scene are necessary. In *UE4*, there are two simple ways to support this task: Shadow Casting and Eye-Dome Lighting (EDL). Shadow casting in our case is applying static shadow to point clouds. As a non-photorealistic lighting model, the Eye-Dome Lighting (EDL) can group points close to each other and shade their outlines, which accentuates the shapes of objects within a point cloud (Ribes & Boucheny, 2011).

Our experiments with EDL and shadow casting are shown in Figures 16 and 17, and we found that EDL can greatly improve the visual quality for point clouds, but the application of shadow casting failed to produce a more desirable effect. Although shadow casting can give the viewer a sense of depth and space to some extent, for point cloud data that uses color rather than real materials, it might distract the eye and cannot do what it was intended to do, i.e., make objects feel grounded in the world. Moreover, since realistic light effects in indoor environments are often complex, we might have to take into account the position, direction and brightness of the lights if we want better results from shadow casting. This can make it difficult for visualisation applications to quickly adapt to other point cloud data without more manual modification. Therefore, we chose to apply only EDL to point cloud data.

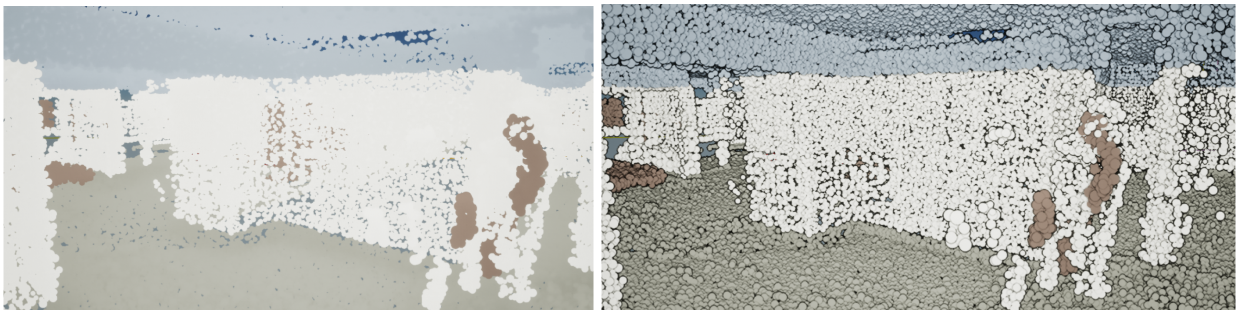


Figure 16: Visualisation Results without EDL (left) and with EDL (right).

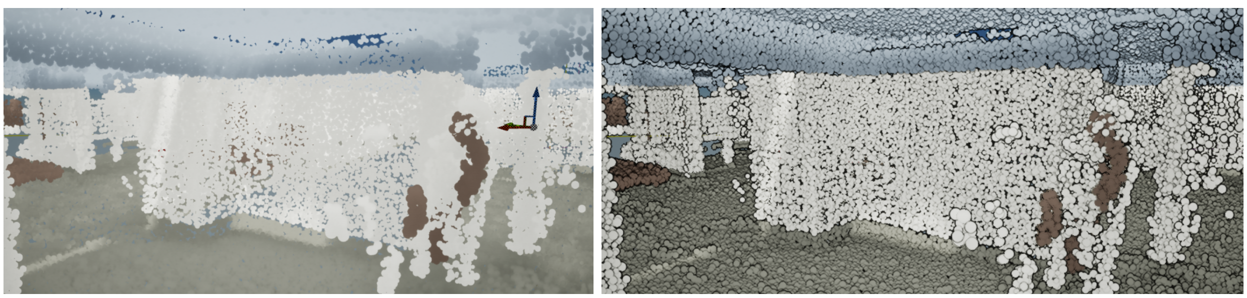


Figure 17: Visualisation Results with Shadow Casting (left) and with Shadow Casting and EDL(right).

7.2.3. Functional Module

We have mentioned static visualisation styles, but in order to effectively analyze and understand a complex dataset, users usually need to look at a part of the data or look at the data from different perspectives. This means that we need some interaction and enhancement techniques. (Andrienko et al., 2020) have distinguished the following types of interaction:

1. Changing data representation.
2. Focusing and getting details.
3. Data transformation.
4. Data selection and filtering.
5. Finding corresponding information pieces in multiple views.

Based on these, we implemented our functional models including switching views, navigation compass, Mini-map, landmark text, transparency adjustment and styles changing. Some of these functions are inspired by shooters and adventure 3D games, which are well designed for the exploration functions needed by players (users) in a crisis environment.

7.2.3.1. Switch Views

The sense of presence and embodiment are the key elements of user experience in an immersive virtual environment (Gorisse et al., 2017). To improve the sense of presence and the sense of embodiment, many video game developers regarded the viewpoint notion as an essential component. In our project, we mainly focus on the first-person view, the third-person view, as well as the bird's eye view. The third-person view is an important way to improve the awareness of the virtual space while perceiving the avatar acting within the environment (Gorisse et al., 2017). The first-person view is the most suitable way to induce the high sense of embodiment according to some studies (Slater et al., 2010). From the bird's eye view, it becomes easy to explore the whole virtual environment because of the ease of grasping the relative positional relationship between the avatar and the surrounding environment (Awashima et al., 2017). Switching the above three views is not only an essential method to improve the sense of presence and embodiment of the virtual environment, but also helps users to discover the indoor environment more comprehensively.

With the different camera locations in the *UE4*, we can implement the function of switching views in our project. For the first-person and third-person view, the camera locations can be considered as static with reference to the avatar (Salamin et al., 2010). We attached the camera at the center of the avatar's head for the first-person view, and put the camera a short distance behind the avatar. For the bird's eye view, the camera should be put at a location above the point cloud model and have a fixed elevation relative to the roof. To explore the indoor environment in the bird's eye view, the roof of the building should be transparent when the user changes into the bird's eye view. The switching between different views can be done by pressing specific keys on the keyboard.

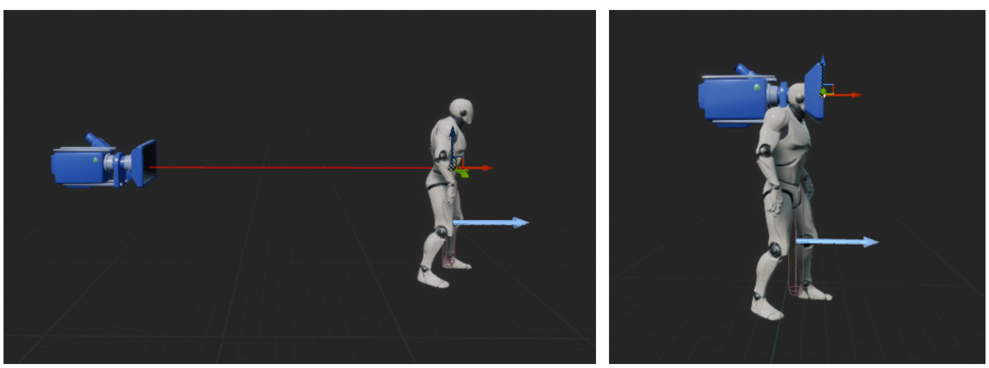


Figure 18: Camera location in third-person view (left), and camera location in first-person view (right)

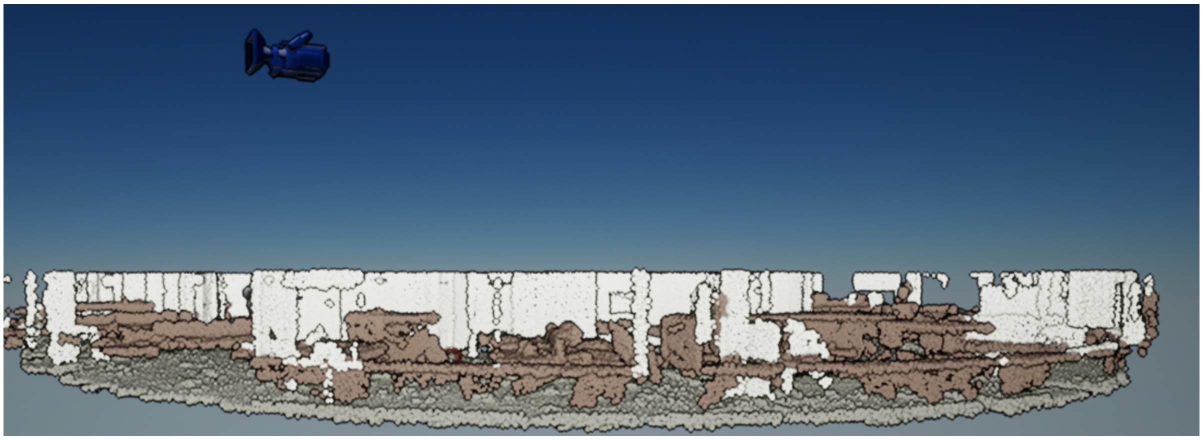


Figure 19: Camera Location in Bird's Eye View

7.2.3.2. Compass

As a common navigation tool in the real world, compass is often used in many video games to help users know the orientation, especially in role-player games (Xiao, 2019). Compass can serve as an essential way to improve the experience of interaction with the virtual environment. In our project, the compass and the mini-map are combined to provide navigation aids for the users.

The compass representation can be very flexible for different purposes in different video games. For our point cloud viewer, we applied a compass like a dynamic ruler placed at the top of the screen and the fixed arrow was removed, which has some advantages over the conventional compass. In this way, users can easily catch the information from the compass as their sight overlaps the avatar's sightline, and there can also be more information added on the dynamic ruler. We put some marks for crucial parts in the building (room, stairs, doors), and then we can also add the marks on the compass. When users move the camera, the positions of the marks on the compass will alter correspondingly.

Figure 20: The Compass used in this project

7.2.3.3. Mini-Map

To improve the usefulness of maps, people often use location maps, which show the geographical location of the area covered by a topographic map at a smaller scale with reference to a larger administrative or physical geographical unit(Kraak & Ormeling, 2020). Location maps help in geolocation at a more general level than more detailed topographic maps (Robinson, 1995). Geolocation is related to the use of topographic maps directly in the field and the contents of topographic maps are compared with objects in the field to assess their actual location and the spatial relationships between them (i.e. direction and distance) (Zagata et al., 2021). Global websites often use maps with mini-maps for car and pedestrian navigation. Their interfaces are moving towards more intuitive and fast navigation (Horbiński et al., 2020).

A mini-map (or minimap) is a miniature map that is often placed at a screen corner in video games to aid players in orienting themselves within the game world ("Mini-Map," 2021). In game fields such as 3D computer games and VR games (VR-Virtual Reality), the mini-map has become standard and it plays an important role in facilitating the character's movement and use of virtual interface in the game (Zagata et al., 2021). With the benefits of mini-maps, the knowledge of the indoor space and one's location may significantly change the user's perception of the indoor environment in order to help them navigate at a higher speed. This drastically contributes to the police in their first response activities.

In our design, we extracted the color scheme of the mini-map from a first-person shooter game (Counter-Strike: Global Offensive) to better display the environment and the character. We also add a light blue triangle area on the character icon to show the character's facing direction. This can help users better know their location and facing direction combined with the information of surrounding objects.

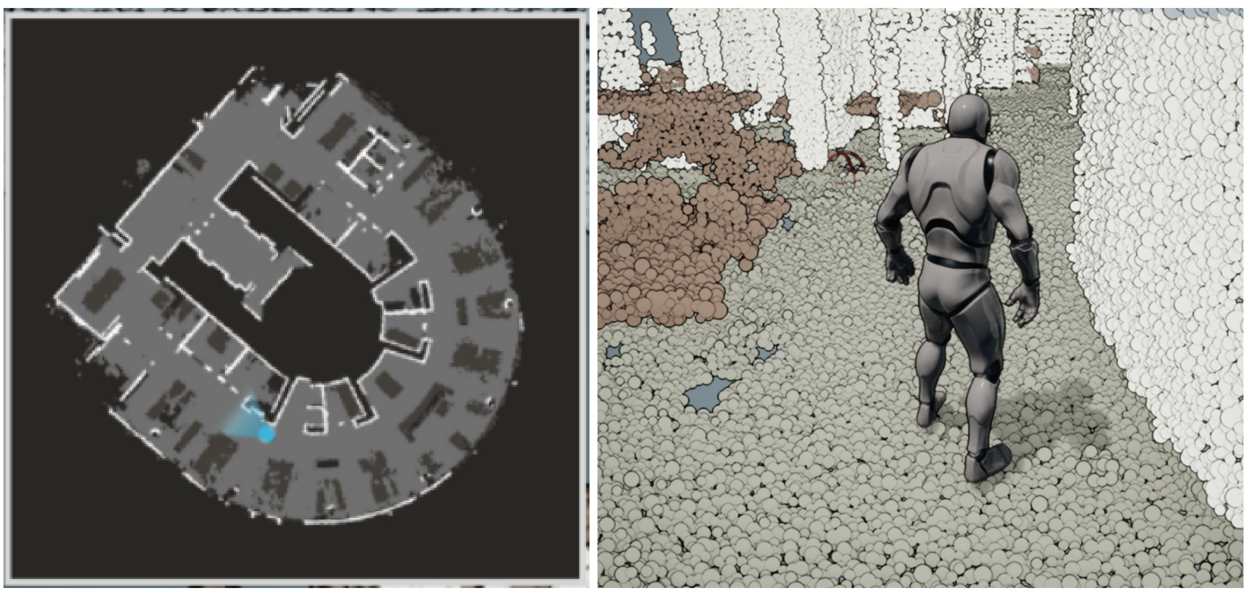


Figure 21: The mini-map shows the character's location and facing direction.

7.2.3.4. Landmark Text

The quick response situation in a potential police raid, being dynamic, raises the great importance of the operator's Situation Awareness, standing as a crucial detail for the performance and the decision making procedure. Being tactical and strategic systems, the police first response units and command personnel, rely on Situation Awareness for the optimal course of action determination (Endsley, 1995). In order to enhance the user's perception and visual insight in the model environment frame and thus enhance the SA, the semantic depiction of certain landmarks in the model is essential. Landmarks are static objects with predefined distinctive features in the model environment (Gim et al., 2021) and are often used as cues for action in situations of route navigation (Foo et al., 2005). As mentioned in (Hilton et al., 2021), being a key component in the development of spatial knowledge, landmarks are also linked to other neighboring locations to resolve ambiguous situations. Using visual cues has been described as a serial learning task, part of the basic navigation task and location awareness (Caplan et al., 2001).

The software's functionality attempts to exploit the landmarks' essential semantic importance for the visual perception and the situation awareness (SA). Landmarks are placed in key positions (i.e. room entrances in the model environment), as spherical object meshes. They were placed in those positions, specifically, in the threshold areas between rooms and the main corridor navigable space, given the major role that landmarks play at intersections, contributing to the fruitful navigation of routes (Hilton et al., 2021). In practice, the functionality takes advantage of the egocentric model representation in the 3^{rd} person perspective. In an egocentric reference frame, the locations are depicted with respect to the perspective of a perceiver (Klatzky, 1998). The user, navigating through space in the 3^{rd} person perspective is referred to as an ego with a definable axis of orientation (Klatzky, 1998). Apart from the visual insight of each room location, when the user interacts with the collision range of the landmarks (spherical meshes), there is also an event firing, providing the user a verbal indicator for a supplementary enhancement of his/hers location awareness. The verbal indicator is a visual text animation (Pop up Text), that fades in and progressively fades off notifying the user that he/she crossed a landmark and is entering a room.

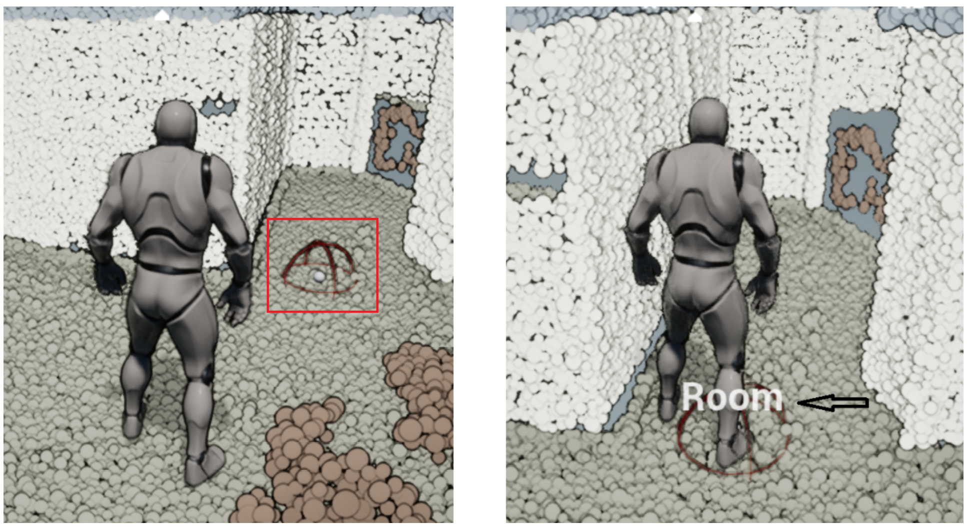


Figure 22: The collision sphere-landmark in a room entrance (left) and part of the Pop Up text animation when the character enters the room area (right).

7.2.3.5. Removal and Transparency

As 3D indoor point cloud data typically detail the interior of the environment as well as its exterior surface, it is necessary to apply some techniques to facilitate gaining an overview of all the data. According to (Florio et al., 2019), there are three major ways to support this task: cut-away, ghosted, and exploded views. Cut-away is basically removing outside-parts of a 3D scene to reveal occluded parts. Ghosted view is rendering the occluding surfaces semi transparent. (Uchida et al., 2020) believes transparency is also useful for analysing and understanding point cloud data. Exploded view is a diagram obtained by displacing components of an 3D object horizontally or vertically to show the internal structure of the object. Exploded view is mainly used to show the distribution of internal space or traffic flow in buildings, or to show the complex hierarchical structure of urban systems, which is thus not useful for this project.

In our project, we allow users to remove the roof as in most cases it is not the most interesting or important part for the user and can also be distracting or cause occlusion. Since users often need to see ground, architectural and non-architectural components to understand the interior environment, it can be considered unnecessary to allow these three classes to be removed as well.

Users are also allowed to adjust transparency to see the objects behind walls. In this way, users can have a thorough perception of the whole indoor environment and it helps them better perform in navigating and path finding which plays an important role in police first response activities. To render transparency, we cannot simply apply color with a transparency value to points, otherwise the volume of points (represented as cubes or spheres) and the overlap between points will cause visual blur (Figure 23 left). Instead, we created a realistic glass material and made the volume of the point not rendered (Figure 23 right). In this way, we have not only preserved clearly the position of the translucent object, but also allowed the parts behind the object to be partially rendered.

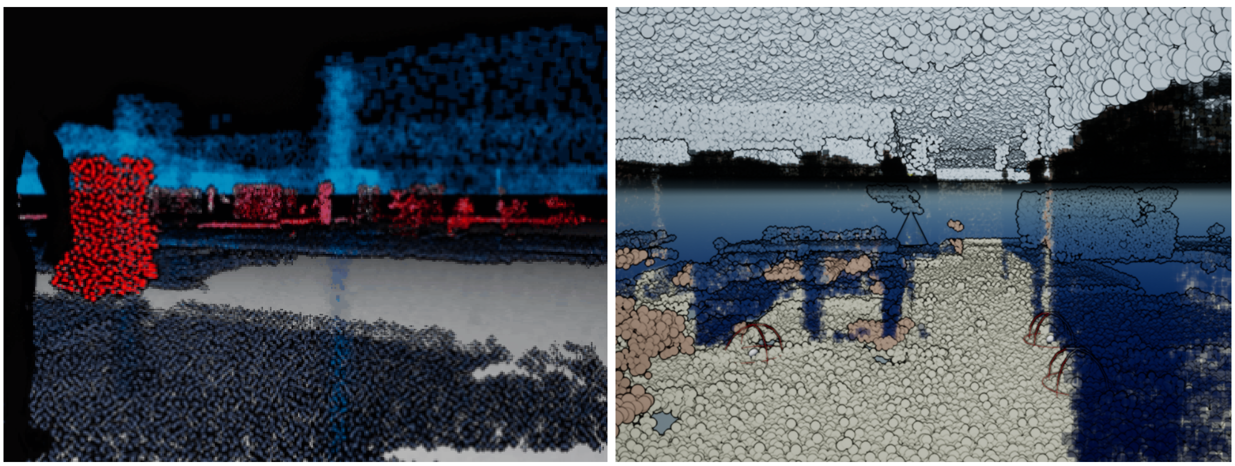


Figure 23: using color with a transparency value for the wall (left) has a worse visualisation result compared to using a realistic glass material (right).

7.2.3.6. Styles Changing

To allow users to distinguish different objects, we use different colors for each class. We pre-set three different color styles (including the default one) and users can switch the style simply by clicking the style buttons in the UI interface.

To meet users' personal visualisation preference, we also add functions for users to change the point shape and point size. Users can adjust the parameters of point shape and point size through the UI interface for different objects parts respectively (wall, non-architecture, floor and roof part).

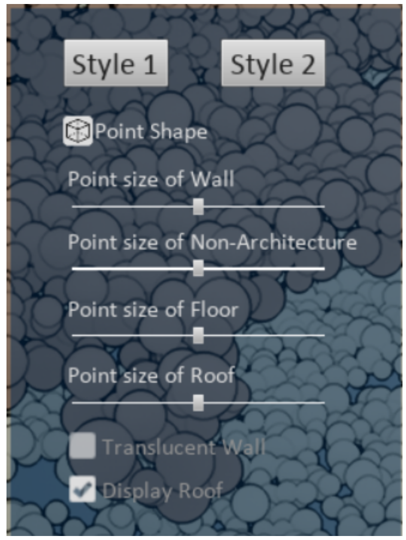


Figure 24: The Style changing UI.

7.2.4. UI Module

The UI of the software is the visualisation of computer's functions that interact with the users. In order to improve the interaction experience of users and make the virtual environment more immersive, we created a UI and applied it in our software. In general, the UI includes two parts: view interface and menus.

7.2.4.1. Viewer Interface

In our project, the viewer interface is a combination of all the viewer's functions, and all the components are set up in the HUD (Heads-up display). HUD is commonly used in video games, which is simply a collection of persistent on screen elements whose purpose is to indicate player status as well as to give instructions for the player (Wilson, 2006). According to Llanos(2013), HUD is one of the most informative and succinct ways to communicate what a game is all about. For the point cloud viewer, it's important to include the navigation aids and also some components to adjust the visualisation type of the point cloud. HUD can be used as a useful tool to gather all the elements, and will be displayed on the screen all the time while the viewer is running.

Many functions will be realized in the viewer interface. Firstly, there are some instructions for the keyboard operations at the right bottom of the screen(element 1 in Figure 25). Secondly, the navigation aids are added. The compass is put at the top of the screen and the mini-map is at the upper left corner. As suggested by game designers, mini-maps generally should not exceed 10% of the available display area (Adams, 2014). Element 2 in Figure 24 is the compass and element 3 shows the position and the size of the mini-map. Thirdly, we also added a canvas to encompass the elements of switching the point cloud style. Element 4 in Figure 25 shows the canvas, it includes different types of elements: button, slider, check box. This interface makes it possible to change the classification color of the point cloud, to switch the point shape, to adjust the point size, to change styles of the wall and the roof.

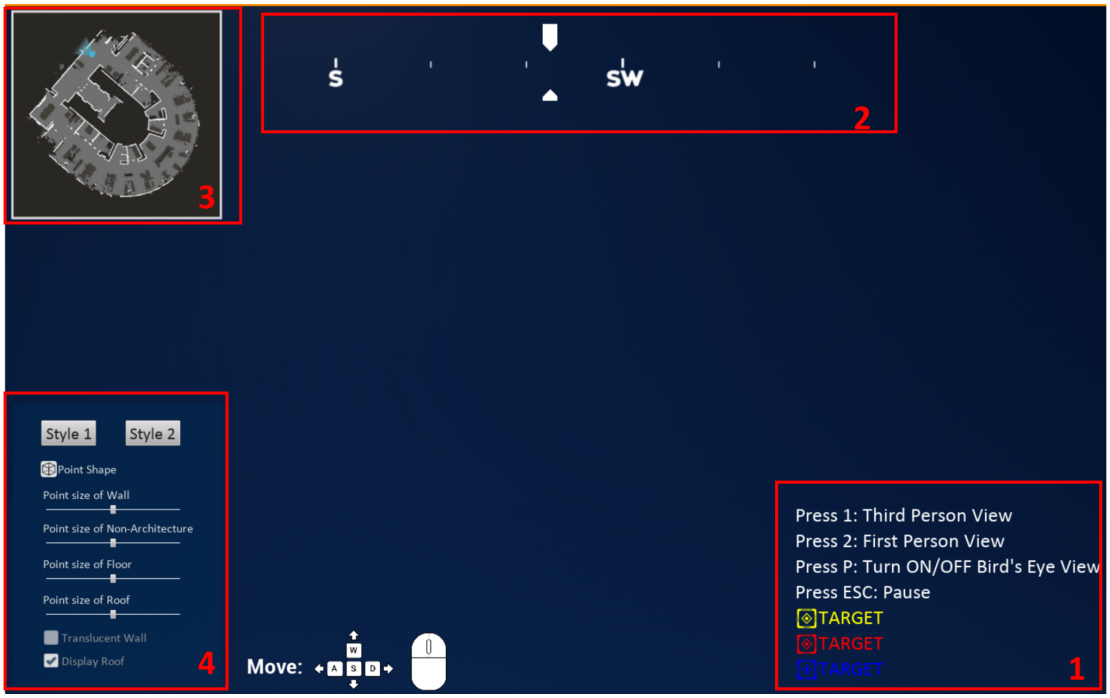


Figure 25: The viewer interface

7.2.4.2. Menu

A simple start menu and a pause menu are created in the software. The start menu is the initial interface of the software which will be displayed when users start the software, and it will provide the buttons to start the viewer and to quit the software. The pause menu will be displayed when the users press escape key on the keyboard, it provides the functions to resume the viewer and to go back to the start menu. Figure 26 shows the start menu and the pause menu. In both the start menu and pause menu, menu items are laid out in a vertical column.

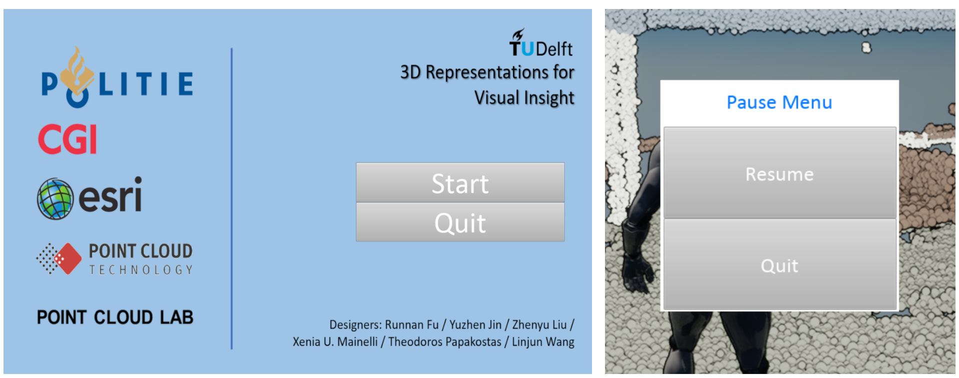


Figure 26: Start menu (left) and pause menu (right)

8. Visualisation Implementation Results

The implemented software product can be found in our Github repository (https://github.com/peterliu502/IndoorPointCloudViewer). The data we used to implement the visualisation is provided by CGI, which is the point cloud data from the first floor of the Safety Region building of Rotterdam-Rijnmond. To remove outliers and segment the data into different parts, we used the method mentioned in Section 7.2.1 to preprocess the data. Then the fours part of the data (roof, floor, wall, non-architecture) will be imported into *UE4* with the Lidar Point Cloud plug-in.

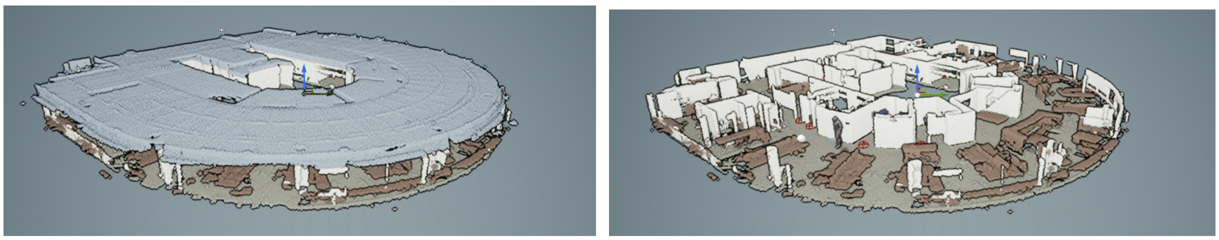


Figure 27: The point cloud data with roof (left) and without roof (right)

Then we can put the avatar into the model, and implement all the functions. The viewer interface will be displayed with the avatar. Uses can act as the avatar to move around the indoor environment. The mini-map and compass in the interface will aid the navigation, and there are three landmarks in different colors put in the building which can be seen on the compass. In the mini-map, users can identify the facing direction of the avatar. These landmarks represent key locations in the indoor environment. To change the style of the point cloud, users can interact with the components in the canvas.

Figure 28: Some visualisation results: (A) View the point cloud data in first-person perspective, and the shape of the points is set as circles. (B) View the point cloud data in first-person perspective, and the shape of the points is set as squares . (C) View the point cloud in third-person perspective. (D) View the point cloud data in bird's eye view, the positions of the avatar and the red target can be identified . (E) Set the point size as the smallest size. (F) Set the point size as the biggest size . (G) View the point cloud in style 1. (H) View the point cloud in style 2.

9. Result Evaluation

9.1. Comparison with Mesh and Voxel Models

In this section, we use the same point cloud file to generate corresponding mesh and voxel models respectively as reference objects. A greedy surface triangulation algorithm is applied on the point cloud with normals, to obtain a triangle mesh based on projections of the local neighborhoods. For the voxel model, its size is set as 0.05m , with intentions to keep as much detail as possible. The comparison considers three aspects, including preprocessing time, retained details, and volume perception.

9.1.1. Preprocessing Time

Using the point cloud format directly has a significant efficiency advantage during the preprocessing phase because both the voxel and mesh models require additional transformation processing.

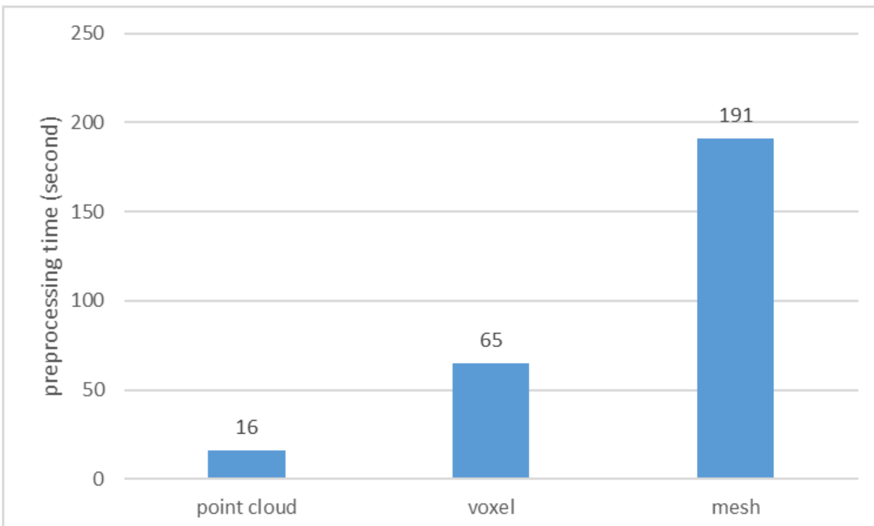


Figure 29: Preprocessing time(seconds) of point cloud, mesh and voxel models

9.1.2. Retained Details

Considering the retained details of each model(Point Cloud, generated Mesh, generated Voxel), it is apparent that the Point Cloud model keeps most of the details. The Voxel model is relatively qualitative, however, the result is not optimal compared with the Point cloud, as it still does not preserve some important details. In particular, indoor outliers are usually very difficult to be perfectly removed, so the Voxel approach complicates their handling in terms of collision inside the *UE4* environment. The Voxel Model's degree of detail loss is highly relative to the designated Voxel size. Concerning the Mesh model, after the reconstruction, a large number of surfaces are getting smooth and in parallel there is a significant detail loss. Additionally, the conversion of non-architecture components to meshes without further splitting them is challenging, while converting them directly to meshes yields notable detail loss and even complicates the components' recognition.

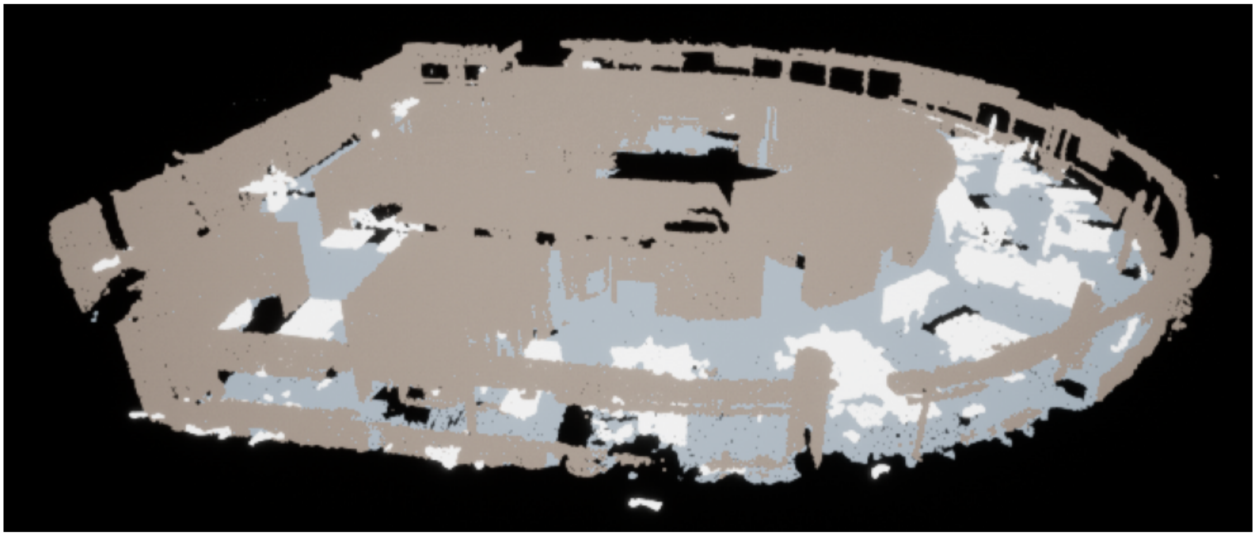


Figure 30: Point Cloud Model, rendered in *UE4*

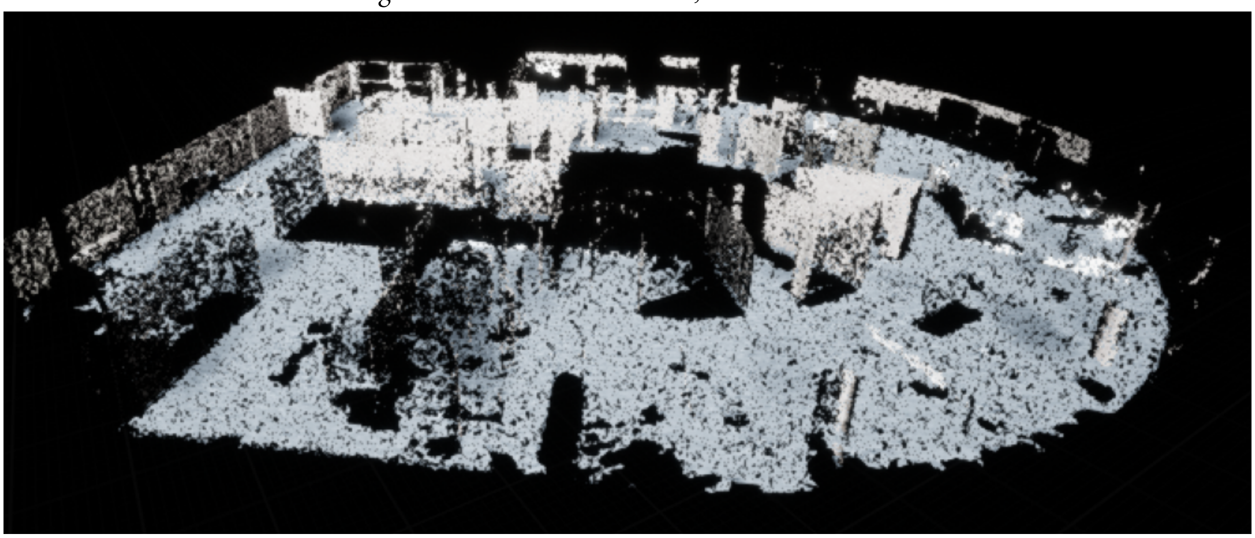


Figure 31: Mesh Model, rendered in *UE4*

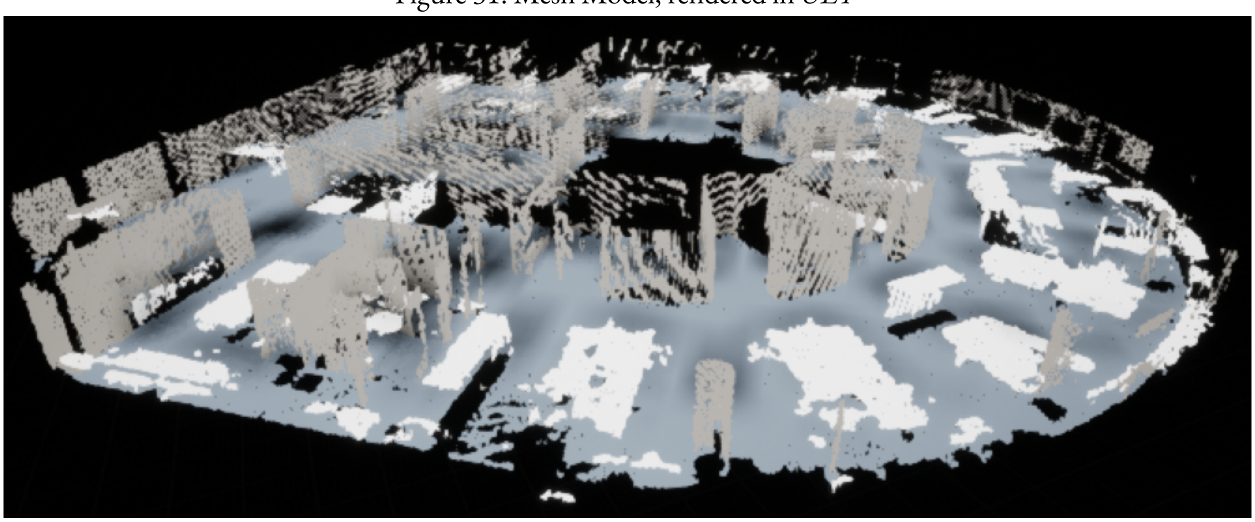


Figure 32: Voxel Model, rendered in *UE4*

9.1.3. Volume Perception

Another aspect which is considered particularly important for the visualisation's effectiveness is the volume perception a potential model can provide. The Point Cloud yields the most optimal result, since the data pretty much represent the real environment. The remaining models(mesh,voxel), being products of reconstruction, develop certain effects that degrade the volume perception, due to accumulated errors in the reconstruction phase. The mesh approach of building accurate polyhedra is rather challenging and does not provide optimal volume perception. The Voxel model's volume perception is significantly better, but still is inaccurate in certain parts and it highly depends on predefined approximations(voxel size, voxelization approach among others).

9.2. User Testing Results

Ideally, our product prototype would be tested upon multiple target market users (employees of the Dutch police force), using a combination of eye tracking software, keylogging, real-time qualitative feedback from the user, and a concluding survey. Given time limitations and social distancing regulations at the time of working, testing with target market representatives was deemed unfeasible. But in order to at least gain some preliminary insights into the success of our product prototype, we opted for a survey based on the Structured Expert Evaluation Method, and invited the mentors and clients of this project to participate in the user testing moment, requesting their input on a variety of questions. When analysing the results, it is important to bear in mind that all of the respondents to the survey are very familiar with point cloud research and application, meaning that testing the success that our visualisation would have with the layman viewer, was not possible.

The survey was split up into four sections: The first asked for initial impressions, the second covered the initial user requirements, the third asked about the functionalities of the visualisation, and the final section contained questions about the visualisation styles. None of the questions were compulsory to fill in, there were a mix of answer formats (open text, linear scale checkbox, multiple-choice grid), and an explanatory text was included at the start of each section. Any questions that asked for a rating on a feature or functionality, used a scale between 1 and 5, where '1' was always the most 'negative' response option, and '5' the most positive. In total, six responses were received, all of which contained completed answers for the vast majority of the questions.

From the form's second section on the prototype's fulfillment of the initial list of user requirements, the rendering speed requirements received the most positive score (every user graded it a 5), while the Ease of Comprehension received the lowest score (3.0 with a standard deviation of 1.2). This is certainly not optimal, as it is of utmost importance that the client can understand the new and unexpected situations that the visualisation shows them. This is a key takeaway that must not be glossed over, and forgotten about when planning future developments for the visualisation game.

The third section regarding functionalities of the game, showed the mini-map to be the most popular feature (all users rated it a 4 or 5, with 5 meaning "You find the mini-map very useful"), followed closely by the Bird's Eye View feature. The least appreciated feature was clearly Landmark Text (a score of 2.2 with a standard deviation of 1.3), and the Compass and Switching Between Views also received a mere 3 (both with fairly high standard deviations like the Landmark Text feature).

Though this user testing step shed light on the advantages and disadvantages of the visualisation and its various styling options, the results ought to be taken with a grain of salt due to there only being a total of six responses.

Looking at the standard deviation scores helps determine how secure the means are in representing the views of all respondents. Features with a high mean and low standard deviation, can be cautiously deemed successful, such as Rendering Speed, Mini-Map, First Person View, and Bird's Eye View. Features with a high mean and high standard deviation (E.g. Navigability, Third Person View, Display Roof), cannot be labelled successful until the qualitative feedback on these features are sifted through, to come to the root of the disparity between the responses. There is a fair likelihood that the concept and design of the feature itself is not the problem, but that there is a failure in conveying the feature's intended purpose clearly to the user. Low scores with high standard deviations (such as Landmark Text and Translucent Wall), could be features that have potential, or divide opinion, and should not be immediately discarded simply because of the low score, but further inspected to find their flaws. Low scores with low standard deviations can probably be discarded in future editions of the visualisation, but there were no instances of this kind of feature within the responses.

Table 5: User Testing Survey Responses

Form Section	Question	Mean Score	Standard Deviation
2. Fulfillment of User Requirements	Rendering Speed	4.8	0.4
	Navigability (e.g. zoom, rotate, pan features)	3.8	1.3
	Ease of Comprehension	3.0	1.1
	Level of Detail / Clutter	4.0	0.6
	Customisability (Helpfulness)	3.3	0.8
	Customisability (Whether it is liked)	3.8	0.6
	Innovation	3.3	0.5
3. Functionalities	Mini-Map	4.5	0.5
	Compass	3.0	1.1
	First Person View	4.0	0.6
	Third Person View	3.7	1.0
	Bird's Eye View	4.2	0.8
	Switching Between Views (1st person, 3rd person, bird's eye)	3.3	1.5
	Landmark Text	2.5	1.4
4. Visualisation Styles	Translucent Wall	2.5	1.0
	Display Roof	3.8	1.0

Preferences for the two different style settings were perfectly split (two users preferred Style 1, two preferred Style 2, and the remaining two had no preference). And visualising the point clouds as either spheres or cubes also did not show a unanimous preference (three users liked the spheres, two preferred the cubes, and one did not have a preference).

The survey enquired about the perceived usefulness of customising the sizes of the points for the four different classes, as well as which sizes for the different classes would be best to use as the default size setting for any game initiation. The responses were mixed, further highlighting the importance of performing more user testing (or receiving more survey responses) before continuing with developing the visualisation. Answers to the questions on the perceived usefulness of the "changing point size" options spanned the full range between (1) not useful, and (5) very useful, for every class, bar non-architectural components. Answers to the questions on the best default size for the points, also ranged greatly within the confines of scores 1 and 5, aside from the architectural components, where everyone agreed that the current default size is the best.

Table 6: User Testing Survey Responses (continued)

Question	Mean Score	Standard Deviation
Perceived usefulness of “changing point size” option for walls	3.0	0.8
Perceived usefulness of “changing point size” option for roof	2.75	0.5
Perceived usefulness of “changing point size” option for floor	2.75	0.5
Perceived usefulness of “changing point size” option for non-architectural components	2.6	0.5

9.2.1. Highlights

In general, all the respondents were very positive about the prototype game, praising different features that appealed to them. According to one user’s summary, the controls were “intuitive and straightforward to use, lowering the entry barrier for users as much as possible.” Features that were popular across the board, are shortly described here:

9.2.1.1. Rendering Speed

All users bar one rated the rendering speed a 5, and users commented that “the environment runs smoothly”, and “frame rates stay consistently interactive”. This is promising for when the visualisation may need to be used upon larger datasets or even real-time data being streamed into the application.

9.2.1.2. Views

The Bird’s Eye view was very popular among the six users who completed the feedback form, with two mentioning it in the “initial thoughts” section, before they were even queried on it. The first-person view was considered the most valuable by the client, while the third person view and bird’s eye view were more popular with mentors on this project.

9.2.1.3. Mini-Map

All users were positive about the mini-map, praising the value it would have when exploring larger datasets with the same game setup.

9.2.2. Limitations

Qualitative feedback shed a light on the reasoning for certain scores, some practical suggestions for bettering useability, and some more complex thoughts on fundamental changes that would still need to be made to have a finished product ready-to-use by the client. The most elementary weakness when reflecting on the initial list of requirements, is that the product lacks the "innovative" streak we were hoping to achieve. As one user commented: "It's a straightforward, but well-implemented solution". The limited time for this project meant that getting a functional game was the priority, and devising creative functionalities came later. With more time in general, and more time dedicated to creative brainstorming, we believe that our product could become far more unique and unusual than its current status. The recommendations from the written feedback can be distributed into four categories:

9.2.2.1. Practical Weaknesses

- Changing the keys one uses for changing views e.g. use the '1' and '3' keys for changing to the first and third person views, respectively.
- It is not possible to pause the game in Bird's Eye View, nor see the compass feature in that view.
- The player figure is still partially visible at times, in first person view.
- A smoother transition between changing viewpoints, e.g. a gradual zoom or fade, rather than an abrupt change in scene, could make it easier on the eye.
- It is difficult to control the viewpoint in the Bird's Eye View.
- The transparency option needs to be more carefully considered and redesigned. Although the use of materials has solved the blurring problem caused by the volume of the point clouds and their overlapping with each other, the transparency effect now takes on the visualisation of the skybox and thereby rendering all objects behind the wall less visible, which creates a less-than-pleasant visual effect. This problem might be alleviated by adjusting material shading modes, colors, transparency values, etc.

9.2.2.2. More Customised Interaction

- Offering an alternative to the WASD keys for navigation, such as a combination of the mouse and arrow keys.
- The speed at which the user can navigate the point cloud, should be customisable.
- One user required better sensitivity for the mouse and keys in order to be able to comfortably use the visualisation.

9.2.2.3. Missing Functions

- Adding path planning, such as finding the shortest path to a particular target.
- Option to query objects, such as their size (thus, number of points included in that cluster), and the object or feature class to which they belong.

- Importing another point cloud dataset and making a new level automatically.

9.2.2.4. UI Design Issues

- Improving the layout and design of the menus.
- Moving the Landmark Text to another spot in the scene, to be less of a distraction. Or have it as a continuous information feature on one dedicated spot on the screen.
- Making the figure smaller and taking up less of the screen space.

10. Conclusions

This project aimed to answer the main research question:

"Using a game engine, can we create unique or insightful visualisations from point clouds, that provide clear perceptions of indoor environments, to assist police in first response activities?"

The determination of the research problems were based on user/client demands and relevant literature to show the importance, value, and places of these problems in visualisation of the 3D indoor environment, while in parallel aim to assist police first response activities. The *UE4* project was also designed and developed based on the aforementioned requirements. To conclude, point clouds are very suitable data models for indoor environment visualisation, with significant advantages in pre-processing efficiency, detail level, and volume perception, in comparison to meshes and voxels. Preprocessing directly affects the quality of a model, and is thus a very important step. It consists of four main parts: denoising, determining the boundary, segmentation and occlusion repair. Regarding the denoising algorithm a balance should be found, between denoising quality and feature-preservation, in order to avoid unnecessarily degrading the input data. Besides that, more adaptive denoising and boundary-determined algorithms are needed to accommodate different input data in future research. Dividing point cloud data into four classes: roof, architecture components, non-architecture components, and ground, is a solution that takes into account both efficiency and visualisation requirements. Only minimal and necessary occlusion fixes (such as ground occlusion) should be performed, as it is not helpful to include guesswork rather than real details into models. In general, mainstream 3D engines already have good support for point clouds, which can achieve rich visualisation and interaction effects.

Concerning the visual rendering style of point clouds, research is still in its infancy. Starting from aspects of color combination, point size, shadow effect, and EDL, we designed some styles suitable for indoor environments based on theoretical visualisation techniques and experiments. We provide visualisation styles that allow users to both quickly understand the interior environment and focus on objects of interest. Our experimental conclusions related to point cloud visualisation techniques, including the use of color has better effect than material, EDL is needed to enhance the visual effect, it is recommended not to use shadow casting in point cloud data, transparency effect is best achieved by using glass material, etc., which can provide reference for subsequent point cloud visualisation studies. In addition, we allow users to immerse themselves in the virtual environment to experience the point cloud indoor model, adjust the visualisation styles as well as the size and shape of the points, which provide an innovative way to use point cloud data. However, clearly communicating the design intent of these rendering style solutions to users is not easy, as can be seen from the User Testing Results section scores. How buttons, text descriptions, menus, etc. are designed and organised to allow users to quickly understand the role of different styles, and how to guide users to choose the appropriate style, etc., are issues that deserve further research. In addition, there is a gap between theory and users' real feelings, as well as between different users' needs and preferences, so it is necessary to do sufficient communication with users, which is still not enough in our project.

Furthermore, 3D shooters and adventure games are good learning materials for exploring indoor environments. Most of the functions in the game can be well developed based on the 3D game engine. Some functions borrowed from the game, such as the mini-map, also work well in real indoor environments. Nevertheless, it is important to notice that not all interaction logic from games is applicable to real-world applications. The interaction logic of many users (especially those with industry backgrounds) with 3D scenes is based on 3D maps, city modeling software or BIM software, which is rather different from video games. (For example, some users do not like to move their character using the WASD keys). Consequently, alternatives must be offered to them. All in all, the integration of other indoor point cloud applications, increasement of the user's interaction with the product, integration of outdoor environment visualisation, and expansion into a multi-flood model can further increase the practical value of indoor point cloud visualisation. Following, recommendations on further research concerning this project will be provided.

11. Further Research

This project's findings offer various possibilities for further improvement and optimisation in future research attempts. In parallel, since the project's implementation will probably be provided publicly – with respect to the principles of Open Science – the intention for further research is highlighted. Potential new research could focus on the enrichment of the software's functionalities, the exploration of new visualisations and interactions, as well as general optimisation guided by specific ad hoc user feedback.

More specifically, there are certain aspects that can be designated subjects for further research in the future. That would be, the increasement of the user's interaction with the product, providing custom landmarks(user defined), and also making use of the game engine for a further enhancement of the user's interaction with the indoor environment utilising VR devices. In parallel, research should also focus on the integration of other indoor point cloud applications, such as routing, blind area detection, change detection, and even VR and AR(Augmented Reality) applications. Additionally, future research could take over to expand the method, to apply for the indoor environment of multiple floors and even integrate the method to also combine the outdoor environment visualisation.

A potential future implementation could concern the optimisation of the data processing, providing more adaptive methods (parameters in the denoising algorithm are automatically adjusted based on input data) and critical position extraction(such as automatic door detection). Besides that, research could also focus on the identification of objects of interest of the potential user, making use of specific data models. The project's basic principles can also be used as a part of an integrated pipeline(software), enmeshing data acquisition - real time processing - visualisation - SA(Situation Awareness) for first response activities of the Dutch police, as well as, other related first response units that are interested in the 3D representations to gain visual insight of certain environments.

12. Acknowledgements

We are proud of what we have managed to achieve as a team of six over the course of these past couple of months, but as with any project, it was not possible without the help of certain key people. We would like to thank Edward Verbree and Robert Voûte for their active role as mentors throughout this process, guiding us in making our project stronger in a variety of aspects — from providing feedback on our reports, to directing us towards helpful resources — and ensuring we had the tools at hand to complete the project. In the kick-starting phase of the project, we had insightful talks with Bart-Peter Smit, Rico Richter, Soeren Discher, and Florent Poux, and we would like to thank them all for giving us practical advice on approaching this project, suggesting a rich selection of articles, and helping us in clearing up the many uncertainties we possessed at first. Bart-Peter provided expertise on the details of the project itself and desires of the client, while Rico and Soeren advised on how to choose a game engine, and along with Florent, opened our eyes to the sheer potential of raw point cloud data. Nanco Oudejans took the time out in later phases to critically assess our demo, providing personal thoughts on its strengths and weaknesses, as well as directing us towards literature regarding navigational aids and how different users perceive and interact with them. We are grateful for his honest thoughts on what elements were (not) beneficial to the final product, and his suggestions for additional features to be added to the project. Finally, we'd like to thank all of the mentors and clients who completed our feedback form, which should be an invaluable reference point for undertaking any future work related to this project.

13. Bibliography

Abdullah, C., Baharuddin, N., Mohd Ariff, M. F., Majid, Z., Lau, C., Yusoff, A., M Idris, K., & Aspuri, A. (2017). INTEGRATION OF POINT CLOUDS DATASET FROM DIFFERENT SENSORS. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-2/W3*, 9–15. https://doi.org/10.5194/isprs-archives-XLII-2-W3-9-2017

Adams, E. (2014). *Fundamentals of game design*. New Riders. http://cds.cern.ch/record/1664828

Andrienko, N., Andrienko, G., Fuchs, G., Slingsby, A., Turkay, C., & Wrobel, S. (2020). Principles of Interactive Visualisation. In N. Andrienko, G. Andrienko, G. Fuchs, A. Slingsby, C. Turkay, & S. Wrobel (Eds.), *Visual Analytics for Data Scientists* (pp. 51–88). Springer International Publishing. https://doi.org/10.1007/978-3-030-56146-8_3

Awashima, Y., Komatsu, R., Fujii, H., Tamura, Y., Yamashita, A., & Asama, H. (n.d.). *Visualization of Obstacles on Bird's-eye View Using Depth Sensor for Remote Controlled Robot*. 4.

Balado, J., González, E., Verbree, E., Díaz-Vilariño, L., & Lorenzo, H. (2020). AUTOMATIC DETECTION AND CHARACTERIZATION OF GROUND OCCLUSIONS IN URBAN POINT CLOUDS FROM MOBILE LASER SCANNING DATA. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *VI-4/W1-2020*, 13–20. https://doi.org/10.5194/isprs-annals-VI-4-W1-2020-13-2020

Barczak, A., & Woźniak, H. (2020). Comparative Study on Game Engines. *Studia Informatica*, 5–24. https://doi.org/10.34739/si.2019.23.01

Bassier, M., Vergauwen, M., & Poux, F. (2020). Point Cloud vs. Mesh Features for Building Interior Classification. *Remote Sensing*, *12*, 2224. https://doi.org/10.3390/rs12142224

Bremer, M., Wichmann, V., & Rutzinger, M. (2013). *Eigenvalue and graph-based object extraction from mobile laser scanning point clouds*. *II-5/W2*, 55–60. https://doi.org/10.5194/isprsannals-II-5-W2-55-2013

Cao, C., Preda, M., & Zaharia, T. (2019). *3D Point Cloud Compression: A Survey* (p. 9). https://doi.org/10.1145/3329714.3338130

Caplan, J. B., Madsen, J. R., Raghavachari, S., & Kahana, M. J. (2001). Distinct Patterns of Brain Oscillations Underlie Two Basic Parameters of Human Maze Learning. *Journal of Neurophysiology*, *86*(1), 368–380. https://doi.org/10.1152/jn.2001.86.1.368

Carbonell, C., Gunalp, P., Saorín, J., & Hess-Medler, S. (2020). Think Spatially With Game Engine. *ISPRS International Journal of Geo-Information*, *9*. https://doi.org/10.3390/ijgi9030159

Che, E., Jung, J., & Olsen, M. J. (2019). Object Recognition, Segmentation, and Classification of Mobile Laser Scanning Point Clouds: A State of the Art Review. *Sensors*, *19*(4), 810. https://doi.org/10.3390/s19040810

Discher, S., Richter, R., & Döllner, J. (2019). Concepts and techniques for web-based visualization and processing of massive 3D point clouds with semantics. *Graphical Models*, *104*, 101036. https://doi.org/10.1016/j.gmod.2019.101036

Endsley, M. R. (1995). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *37*(1), 32–64. https://doi.org/10.1518/001872095779049543

Florio, A., Trapp, M., & Döllner, J. (2019). *Semantic-driven Visualization Techniques for Interactive Exploration of 3D Indoor Models*. https://doi.org/10.1109/IV.2019.00014

Foo, P., Warren, W. H., Duchon, A., & Tarr, M. J. (2005). Do Humans Integrate Routes Into a Cognitive Map? Map- Versus Landmark-Based Navigation of Novel Shortcuts. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *31*(2), 195–215. https://doi.org/10.1037/0278-7393.31.2.195

Friedman, S., & Stamos, I. (2012). Online facade reconstruction from dominant frequencies in structured point clouds. *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 1–8. https://doi.org/10.1109/CVPRW.2012.6238908

Game engine. (2021). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Game_engine&oldid=1026978297

Gim, J., Ahn, C., & Peng, H. (2021). Landmark Attribute Analysis for a High-Precision Landmark-Based Local Positioning System. *IEEE Access*, *9*, 18061–18071. https://doi.org/10.1109/ACCESS.2021.3053214

Gorisse, G., Christmann, O., Amato, E. A., & Richir, S. (2017). First- and Third-Person Perspectives in Immersive Virtual Environments: Presence and Performance Analysis of Embodied Users. *Frontiers in Robotics and AI*, *4*, 33. https://doi.org/10.3389/frobt.2017.00033

Gorte, B., & Zlatanova, S. (2016). RASTERIZATION AND VOXELIZATION OF TWO- AND THREE-DIMENSIONAL SPACE PARTITIONINGS. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLI-B4*, 283–288. https://doi.org/10.5194/isprs-archives-XLI-B4-283-2016

Hackel, T., Wegner, J. D., & Schindler, K. (2016). Fast Semantic Segmentation of 3D Point Clouds with Strongly Varying Density. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *III–3*, 177–184. https://doi.org/10.3929/ethz-b-000126659

Hevner, A., & Chatterjee, S. (2010). Design Science Research in Information Systems. In A. Hevner & S. Chatterjee (Eds.), *Design Research in Information Systems: Theory and Practice* (pp. 9–22). Springer US. https://doi.org/10.1007/978-1-4419-5653-8_2

Hevner, A., R, A., March, S., T, S., Park, Park, J., Ram, & Sudha. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly*, *28*, 75.

Hilton, C., Wiener, J., & Johnson, A. (2021). Serial memory for landmarks encountered during route navigation. *Quarterly Journal of Experimental Psychology*, 174702182110207. https://doi.org/10.1177/17470218211020745

Horbiński, T., Cybulski, P., & Medyńska-Gulij, B. (2020). Graphic Design and Button Placement for Mobile Map Applications. *The Cartographic Journal*, *57*(3), 196–208. https://doi.org/10.1080/00087041.2019.1631008

Johannesson, P., & Perjons, E. (2014). Research Strategies and Methods. In P. Johannesson & E. Perjons (Eds.), *An Introduction to Design Science* (pp. 39–73). Springer International Publishing. https://doi.org/10.1007/978-3-319-10632-8_3

Julin, A., Jaalama, K., Virtanen, J.-P., Maksimainen, M., Kurkela, M., Hyyppä, J., & Hyyppä, H. (2019). Automated Multi-Sensor 3D Reconstruction for the Web. *ISPRS International Journal of Geo-Information*, *8*, 221. https://doi.org/10.3390/ijgi8050221

Kharroubi, A., Hajji, R., Billen, R., & Poux, F. (2019). CLASSIFICATION AND INTEGRATION OF MASSIVE 3D POINTS CLOUDS IN A VIRTUAL REALITY (VR) ENVIRONMENT. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLII-2/W17*, 165–171. https://doi.org/10.5194/isprs-archives-XLII-2-W17-165-2019

Kharroubi, A., Hajji, R., Billen, R., & Poux, F. (2021). CLASSIFICATION AND INTEGRATION OF MASSIVE 3D POINTS CLOUDS IN A VIRTUAL REALITY (VR) ENVIRONMENT. *ISPRS - International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*.

Klatzky, R. L. (1998). Allocentric and Egocentric Spatial Representations: Definitions, Distinctions, and Interconnections. In C. Freksa, C. Habel, & K. F. Wender (Eds.), *Spatial Cognition* (Vol. 1404, pp. 1–17). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-69342-4_1

Kraak, M.-J., & Ormeling, F. (2020). *Cartography ; Visualization of Geospatial Data*. CRC Press.

http://dx.doi.org/10.1201/9780429464195
Li, M., Bakillah, M., & Zipf, A. (2013). Embedding context-awareness to improve 3D geo-visualization for mobile users. *Gis.SCIENCE*, *26*, 81–86.
Llanos, S. C. (n.d.). *What Does the HUD Tell Us?: The Heads Up Display as a Meta-communication in Videogames*. 5.
Lv, C., Lin, W., & Zhao, B. (2021). Voxel Structure-based Mesh Reconstruction from a 3D Point Cloud. *IEEE Transactions on Multimedia*, *PP*, 1–1. https://doi.org/10.1109/TMM.2021.3073265
Merlo, A., Belenguer, C., Vendrell Vidal, E., Fantini, F., & Aliperta, A. (2013). 3D model visualization enhancements in real-time game engines. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *40*, 181–188. https://doi.org/10.5194/isprsarchives-XL-5-W1-181-2013
Mini-map. (2021). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Mini-map&oldid=1026826229
Munoz, D., Vandapel, N., & Hebert, M. (2008). *Directional Associative Markov Network for 3-D Point Cloud Classification*. https://doi.org/10.1184/R1/6552644.v1
Nebiker, S., Cavegn, S., & Loesch, B. (2015). Cloud-Based Geospatial 3D Image Spaces—A Powerful Urban Model for the Smart City. *ISPRS International Journal of Geo-Information*, *4*, 2267–2291. https://doi.org/10.3390/ijgi4042267
Neuville, R., Pouliot, J., Poux, F., De Rudder, L., & Billen, R. (2018). A Formalized 3D Geovisualization Illustrated to Selectivity Purpose of Virtual 3D City Model. *ISPRS International Journal of Geo-Information*, *7*(5), 194. https://doi.org/10.3390/ijgi7050194
Ning, X., Li, F., Tian, G., & Wang, Y. (2018). An efficient outlier removal method for scattered point cloud data. *PLOS ONE*, *13*(8), e0201280. https://doi.org/10.1371/journal.pone.0201280
Nourian, P., Goncalves, R., Zlatanova, S., Ohori, K., & Vo, A. (2016). Voxelization Algorithms for Geospatial Applications. *MethodsX*, *3*. https://doi.org/10.1016/j.mex.2016.01.001
Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302
Ph.D, F. P. (2021, April 11). *How to represent 3D Data?* Medium. https://towardsdatascience.com/how-to-represent-3d-data-66a0f6376afb
Poux, F., & Billen, R. (2019). Voxel-Based 3D Point Cloud Semantic Segmentation: Unsupervised Geometric and Relationship Featuring vs Deep Learning Methods. *ISPRS International Journal of Geo-Information*, *8*, 213. https://doi.org/10.3390/ijgi8050213
Rakotosaona M.-J., La Barbera V., Guerrero P., Mitra N. J., & Ovsjanikov M. (2019). *PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds*. https://arxiv.org/abs/1901.01060v3
Ribes, A., & Boucheny, C. (2011). *Eye-Dome Lighting: A non-photorealistic shading technique*.
Robinson, A. H. (1995). *Elements of cartography. - 6th ed.* Wiley.
Salamin, P., Tadi, T., Blanke, O., Vexo, F., & Thalmann, D. (2010). Quantifying Effects of Exposure to the Third and First-Person Perspectives in Virtual-Reality-Based Training. *IEEE Transactions on Learning Technologies*, *3*(3), 272–276. https://doi.org/10.1109/TLT.2010.13
Slater, M., Spanlang, B., Sanchez-Vives, M. V., & Blanke, O. (2010). First Person Experience of Body Transfer in Virtual Reality. *PLoS ONE*, *5*(5), e10564. https://doi.org/10.1371/journal.pone.0010564
Török, Á., Nguyen, T. P., Kolozsvári, O., Buchanan, R. J., & Nadasdy, Z. (2014). Reference frames in virtual spatial navigation are viewpoint dependent. *Frontiers in Human Neuroscience*, *8*. https://doi.org/10.3389/fnhum.2014.00646
Tredinnick, R., Casper, G., Arnott-Smith, C., Peer, A., & Ponto, K. (2018). *Using Virtual Reality to Study*

Health in the Home. https://doi.org/10.1109/VAR4GOOD.2018.8576886
Uchida, T., Hasegawa, K., Li, L., Adachi, M., Yamaguchi, H., Thufail, F. I., Riyanto, S., Okamoto, A., & Tanaka, S. (2020). *Noise-robust transparent visualization of large-scale point clouds acquired by laser scanning*. http://dx.doi.org/10.1016/j.isprsjprs.2020.01.004
Unreal Engine. (2021). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Unreal_Engine&oldid=1028463656
Virtanen, J.-P., Daniel, S., Turppa, T., Zhu, L., Julin, A., Hyyppä, H., & Hyyppä, J. (2020). Interactive dense point clouds in a game engine. *ISPRS Journal of Photogrammetry and Remote Sensing*, *163*, 375–389. https://doi.org/10.1016/j.isprsjprs.2020.03.007
Virtanen, J.-P., Julin, A., Handolin, H., Rantanen, T., Maksimainen, M., Hyyppä, J., & Hyyppä, H. (2020). INTERACTIVE GEO-INFORMATION IN VIRTUAL REALITY – OBSERVATIONS AND FUTURE CHALLENGES. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLIV-4/W1-2020*, 159–165. https://doi.org/10.5194/isprs-archives-XLIV-4-W1-2020-159-2020
Wieringa, R. J. (2014). Research Goals and Research Questions. In R. J. Wieringa (Ed.), *Design Science Methodology for Information Systems and Software Engineering* (pp. 13–23). Springer. https://doi.org/10.1007/978-3-662-43839-8_2
Wilson, G. (2006, February 3). *Off With Their HUDs!: Rethinking the Heads-Up Display in Console Game Design*. GAMASUTRA.
Xiao, J. (n.d.). *A Study of Navigation Aids in Video Games*. 52.
Xiong, X., Adan, A., Akinci, B., & Huber, D. (2013). Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction*, *31*, 325–337. https://doi.org/10.1016/j.autcon.2012.10.006
Zagata, K., Gulij, J., Halik, Ł., & Medyńska-Gulij, B. (2021). Mini-Map for Gamers Who Walk and Teleport in a Virtual Stronghold. *ISPRS International Journal of Geo-Information*, *10*(2), 96. https://doi.org/10.3390/ijgi10020096

Appendix A: Product Feedback Form

Survey was sent out using [Google Forms](), and had five sections in total, including the introductory section requesting the respondent's name and information on the operating system on which they were running the .exe file that contained the visualisation game. The italicised texts below are explanatory comments that the respondent could read when filling in the form.

If you haven't already, please run the .exe file in the "IndoorPointcloudsViewer" folder that was sent along in the same email as the link to this form.
Please first explore the game – try out the different features, and get a feel for how to use it – before answering the following questions. If you are having trouble running the file, please contact us!

1. First Impressions
    a. Any initial thoughts regarding the visualisation 'game'?
    b. What features do you like at first glance? What enhances the experience for you?
    c. Which features do you not understand, are difficult to use, or do you think are badly designed?

At the beginning of this project, we wrote out a list of requirements for our visualisation, and would like to now find out the extent to which we have fulfilled these requirements.
We'd like to ask you to give the different requirements a rating, from 1 (very weak) to 5 (very strong).
[Questions 2.a to 2.g are all multiple choice with five options (a scale from 1 through to 5). Question 2.h is an open-ended question with no character limit for the answer.]

2. Fulfillment of User Requirements
    a. Rendering Speed
    b. Navigability (e.g. zoom, rotate, pan features)
    c. Ease of Comprehension
    d. Level of Detail / Clutter
    e. Customisability (Helpfulness)
    f. Customisability (Whether it is liked)
    g. Innovation
    h. Summarising Thoughts (An opportunity to explain any of the scores you just attributed above)

We implemented multiple different functionalities and features into our visualisation game, and are curious to hear whether you like them, find them useful, or just a waste of space!
*You can first rate the feature on a scale of 1 (negative) to 5 (positive), and then optionally provide a comment if you'd like to elaborate. E.g. maybe you rated something a 5 for *liking* the feature concept, but want to mention that you'd have given it a 1 if the question were about *ease of use*. Or maybe you really like a feature, but have an idea for improving it further.*
[Questions 3. a, c, e, g, i, k, and m are all multiple choice with five options (a scale from 1 through to 5). Questions 3. b, d, f, h, j, l, and n are all open-ended questions with no character limit for the answers.]

3. Functionalities ("You can first rate the feature on a scale of 1 (negative) to 5 (positive), and then optionally provide a comment if you'd like to elaborate.")
    a. Mini-Map
    b. Mini-Map Thoughts
    c. Compass
    d. Compass Thoughts
    e. First Person View
    f. First Person View Thoughts
    g. Third Person View
    h. Third Person View Thoughts
    i. Bird's Eye View
    j. Bird's Eye View Thoughts
    k. Switching Between Views (1st person, 3rd person, bird's eye)
    l. Switching Between Views Thoughts
    m. Landmark Text (The pop-up text that describes the new area being entered)
    n. Landmark Text Thoughts

And finally, we tried to include some customisation options for the user, and want to hear which ones you like, as well as which settings you prefer the most, so that we can make this the default starting style.

[Questions 4. a and b are both multiple choice with three options, the first two being the visualisation style in question, and the third being "no preference". Questions 4. c (i, ii, iii, iv), d (i, ii, iii, iv), e, and g are all multiple choices with five options (a scale from 1 through to 5). Questions 4. f, h, i, and j are all open-ended questions with no character limit for the answers.]

4. Visualisation Styles
    a. Do you prefer Style 1 or Style 2?
    b. Do you prefer the points to be displayed as spheres or as cubes? (Click on Point Shape button)
    c. For which points do you find the "changing size" option the most & least useful?
        i. Wall
        ii. Non-architectural components
        iii. Floor
        iv. Roof
    d. What do you find to be the best default style regarding points size?
        i. Wall
        ii. Non-architectural components
        iii. Floor
        iv. Roof
    e. Translucent Wall
    f. Translucent Wall Thoughts
    g. Display Roof
    h. Display Roof Thoughts
    i. Thoughts on "Targets" feature – the option to include targets within the game, and see their locations on the compass. Would it be useful? What could make it better?
    j. FINAL REMARKS: Please leave any comments that you may still have, that weren't addressed by any of the previous questions in this form.