

Exploring Isovist Applications in Third-Person View Visualisations of Outdoor Space Boundaries Using Point Clouds

A thesis submitted to the Delft University of Technology in partial fulfilment of
the requirements for the degree of

Master of Science in Geomatics for the Built Environment

Xenia Una Mainelli

August 30, 2022

Supervisor TU Delft: Ir. Edward Verbree
Supervisor TU Delft: Dr. ir. Martijn Meijers
Supervisor CGI: Ir. Robert L. Voûte
Co-reader: Dr. Jesús Balado Frias
External Supervisor: Ir. Herman W. de Wolff



Xenia Una Mainelli: *Exploring Isovist Applications in Third-Person View Visualisations of Outdoor Space Boundaries Using Point Clouds* (2022)

© ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

The work in this thesis was carried out in the:



GIS Technology group
Architectural Engineering + Technology
Faculty of Architecture & the Built Environment
Delft University of Technology

In cooperation with:



Supervisors: Ir. Edward Verbree
Dr. ir. Martijn Meijers
Ir. Robert L. Voûte

Co-reader: Dr. Jesús Balado Frias

Abstract

The kinds of physical spaces present in the real world are becoming ever more complex, and the locations defining the boundaries between these spaces are often arbitrary. Distinguishing between which spaces count as ‘outdoors,’ and which count as ‘indoors,’ becomes more difficult when ‘semi-outdoor’ and ‘semi-indoor’ spaces are considered. Integrating these different spaces within geovisualisations is difficult because data on the spaces are often collected and stored separately. Many existing navigational applications avoid the explicit differentiation between different types of spaces, or choose to only visualise one type of space. Additionally, these applications rarely identify which areas are visible to users at their present positions, and which areas are occluded.

This thesis explores the potential of utilising point clouds directly in geovisualisations to communicate information about the types of spaces surrounding a hypothetical user in a real-world environment. Raw point cloud data is collected on three different transitional spaces, all of which contain an outdoor element. These point clouds are classified into four different ‘space-types’ (outdoor, indoor, semi-indoor, and semi-outdoor), and visibility analysis is performed on them directly. The resulting information on space-type and visibility is combined within multiple different data visualisations, the concepts of which have been designed using a list of requirements based on existing literature. The visualisations show that there is potential for direct use of point clouds in communicating information about spaces to a user, and that discerning between visible and occluded spaces, has potential value to a user orienting themselves within their environment with aid of a geovisualisation.

Keywords: Outdoor-Indoor Boundaries, Geovisualisation, Point Cloud, Visibility Analysis

Acknowledgements

My three supervisors have all been incredibly generous with their time and knowledge throughout the completion of my thesis. Thank you Edward for the care you took of your geomatics master thesis students, supporting us during the entire process while often lending a critical eye, and for being willing to place so much trust in me together with Martijn. Thank you Martijn for being such an involved third mentor, even when it was not technically required! I greatly appreciated the time you took out for supervising this thesis, and how you were present at all of the bi-weekly meetings with Edward, ready to give your insights on interesting concepts. And thank you Robert for regularly yielding to chairing our initially thrice-weekly, online meetings with the students at CGI, as well as informally spending time with us all within and outside of the CGI office. You were always sure to get me into contact with people who could be of help, gracefully allowed my research to change its course on multiple occasions, and did it all with a great sense of humour.

On a more personal note, without my parents and sister, I certainly would not have had the experiences and opportunities along the way that ultimately led me to studying for a master's degree in The Netherlands. Finally, to the city of Delft for looking so alluringly beautiful back on that 2015 summer's day when I visited as an unsuspecting high school student – in the back of my mind I had already become convinced that I needed to study here at some point. (And of course to Delft's Johannes Vermeer for his through-views!)

Contents

Abstract	ii
Acknowledgements	iii
1 Introduction	1
1.1 Outdoor-Indoor Visualisation Integration	2
1.2 Isovists	4
1.3 Learning from Dutch Golden Age Painters	7
1.4 Point Clouds	9
1.5 Research Questions	10
1.5.1 Sub-Research Question 1: Classifying Point Clouds	10
1.5.2 Sub-Research Question 2: Exploring Visualisation Techniques	11
1.5.3 Sub-Research Question 3: Evaluating Visualisation Designs	12
1.5.4 Sub-Research Question 4: Reflecting on Visualisations’ Usefulness	12
1.6 Scientific and Societal Relevance	12
2 Related Research	15
2.1 Outdoor-Indoor Boundary Visualisation	15
2.2 The Wayfinding Experience	16
2.3 Isovists as Visualisation Tools	16
2.4 Visibility Analysis on Point Clouds	17
2.5 Data Visualisation Practices	18
3 Methodology	21
3.1 Data Collection	21
3.2 Data Classification	22
3.3 Visibility Analysis	23
3.4 Visualisation Techniques	25
3.4.1 List of Requirements	25
3.4.2 Understanding the User’s Needs	26
3.5 Space Classification and Visibility Analysis	27
4 Results	29
4.1 Space-Type Classification	29
4.2 Visibility Analysis: Hidden Point Removal	32
4.3 Visualisation Designs	35
4.3.1 The “Floor Plan” Designs	35
4.3.2 The “Entranceway” Designs	38
4.3.3 The “Walled” Designs	40
4.3.4 The “Isovist” Designs	40
4.4 Data Visualisations	44

5 Discussion	51
5.1 “Floor Plan Enclosures” Evaluation	55
5.2 “Floor Plan Markers” Evaluation	56
5.3 “Doors” Evaluation	57
5.4 “Walls” Evaluation	58
5.5 “Isovist Paths” Evaluation	59
5.6 “Isovist Paths Outline” Evaluation	60
5.7 Limitations	61
6 Conclusions	63
6.1 Addressing the Research Questions	63
6.1.1 RQ 1: Point Cloud Classification	64
6.1.2 RQ 2: Visualisation Techniques	64
6.1.3 RQ 3: Evaluation of Techniques	66
6.1.4 RQ 4: Reflection on the Use of Isovist	67
6.2 General Conclusion	68
7 Future Work	71
A List of Requirements	73
A.1 User Experience Justification	74
A.1.1 Visualisation must be designed with a layman in mind as the end user	74
A.1.2 Updating the visualisation view must involve minimal latency	74
A.2 Software Justification	75
A.2.1 Must not demand too much processing power from device	75
A.2.2 Must be spatially dynamic	75
A.2.3 (OPTIONAL EXTRA) Must be temporally dynamic	75
A.3 Intuitiveness Justification	76
A.3.1 User must be able to link physical environment features with digital twins	76
A.3.2 Boundary locations must be easily identifiable	76
A.3.3 Different space boundary types must be clearly distinguishable	76
A.3.4 ‘Non-isovist’ space must be clearly distinguishable from isovist space	77
A.4 Design Justification	77
A.4.1 Must be visually minimalistic to avoid visual clutter	77
A All Data Visualisation Designs	79
Bibliography	81

List of Figures

1.1	Visual description of Yan, Diakit�, and Zlatanova (2019)’s definitions of four space types	3
1.2	Example of a 2D isovist	4
1.3	Map comparison for the 2D isovist example	4
1.4	Panorama photograph of Bank Junction in the City of London, United Kingdom	5
1.5	2D isovist example with an angle of 120�, and view limit	5
1.6	2D Isovist Example with Three Isovists Intersecting One Another	5
1.7	Wellington statue at Bank Junction demonstrating a 2D isovist weakness	6
1.8	2D isovist example with reflective and transparent boundaries	6
1.9	Three examples of Dutch Golden Age painters’ depictions of complex indoor environments	8
1.10	Samuel Dirksz van Hoogstraten painting of a house interior	8
1.11	Pieter de Hooch Painting split into different space types	9
1.12	Stanford Bunny as a point cloud	9
2.1	Nikander et al. (2013)’s indoor floor plan with satellite imagery concept	16
2.2	Dalton and Dalton (2015)’s <i>Contour Isovists for Ten Simple Volumetric Spaces</i>	17
2.3	Point clouds of David statue from Katz, Tal, and Basri (2007)	18
2.4	Visual description of Katz, Tal, and Basri (2007)’s point cloud inversion	18
2.5	Two depictions of Bertin (1967)’s <i>Visual Principles</i>	19
2.6	(Munzner 2015)’s effectiveness ranking of visualisation channels, dependent on data type	20
2.7	Munzner (2009)’s “Nested Model for Visualization Design and Validation”	20
2.8	Threats and validations in Munzner (2009)’s model	20
2.9	Brehmer and Munzner (2013)’s <i>Typology of Abstract Visualisation Tasks</i>	20
3.1	Mobile phone point cloud scans of four different space boundary types	22
3.2	Performing <i>CloudCompare</i> ’s HPR operator on the Stanford Bunny	24
3.3	Steven’s Psychophysical Power Law	25
4.1	Point cloud space-type classification process in <i>CloudCompare</i>	30
4.2	Cleaned point clouds of four space boundaries, showing RGB values and space-types	31
4.3	Before and after HPR is performed on the “Main Entrance” point cloud	32
4.4	Before and after HPR is performed on the “Car Park Entrance” point cloud	33
4.5	Before and after HPR is performed on the “Pedestrian Entrance” point cloud	33
4.6	Before and after HPR is performed on the “Car Park Stairs” point cloud	34
4.7	CAD building used to demonstrate design concepts of proposed visualisations	35
4.8	“Floor Plan Thresholds” design: Lines indicate space boundary thresholds	36
4.9	“Floor Plan Enclosures” design: Non-traversable boundaries are bordered	37
4.10	“Floor Plan Enclosures (greyscale)” design	38
4.11	“Floor Plan Markers” design: shapes indicate space boundary thresholds	38
4.12	“Doors” design: Shapes of doorways indicate space boundary thresholds	39
4.13	“Doors (greyscale)” design	40
4.14	“Walls” design: Walls provide information on occluded spaces	41

4.15 “Isovist Paths” design: Pathways to entrances are visualised	42
4.16 “Isovist Outdoor Area” design: Entire isovist is visualised	43
4.17 “Isovist Outdoor Area with Green Paths” design	43
4.18 “Isovist Green Outdoor Area” design: Non-outdoor spaces are in greyscale	43
4.19 “Isovist Paths (greyscale)” design	43
4.20 “Isovist Paths Outline” design: Paths are outlined	43
4.21 All data visualisations for the car park entrance data set from viewpoint 1	45
4.22 All data visualisations for the car park data set from viewpoint 2	46
4.23 All data visualisations for the pedestrian entrance data set from viewpoint 1	47
4.24 All data visualisations for the pedestrian entrance data set from viewpoint 2	48
4.25 All data visualisations for the car park stairs data set from viewpoint 1	49
4.26 All data visualisations for the car park stairs data set from viewpoint 2	50
5.1 The six most differing data visualisations	52
5.2 “Floor Plan Enclosures” visualisations for each space boundary	55
5.3 “Floor Plan Markers” visualisations for each space boundary	56
5.4 “Doors” visualisations for each space boundary	57
5.5 “Walls” visualisations for each space boundary	58
5.6 “Isovist Paths” visualisations for each space boundary	59
5.7 “Isovist Paths Outline” visualisations for each space boundary	60
6.1 Screen capture of <i>Google Maps</i> ’ 3D modelling of buildings	66
A.1 Compilation of all data visualisation design concepts described in section 4.3	80

List of Tables

1.1	A summary of difficulties in combining indoor and outdoor data in visualisations . . .	3
5.1	“Floor Plan Enclosures” Visualisation Evaluation Summary	55
5.2	“Floor Plan Markers” Visualisation Evaluation Summary	56
5.3	“Doors” Visualisation Evaluation Summary	57
5.4	“Walls” Visualisation Evaluation Summary	58
5.5	“Isovist Paths” Visualisation Evaluation Summary	59
5.6	“Isovist Paths Outline” Visualisation Evaluation Summary	60

Acronyms

2D Two-Dimensional. [4](#)

3D Three-Dimensional. [4](#)

AOI Area of Interest. [16](#)

CAD Computer Aided Design. [35](#)

GIS Geographic Information System. [15](#)

GNSS Global Navigation Satellite System. [2](#), [3](#)

HPR Hidden Point Removal. [17](#), [23](#), [32-34](#), [53](#), [68](#), [71](#)

LAS LASer. [21](#)

LiDAR Light Detection And Ranging. [9](#)

PLY Polygon. [21](#)

RGB Red-Green-Blue. [21](#)

Glossary

convex hull The smallest convex set containing a set of n points. [17](#)

digital twin The digital representation of a real-world environment or system. In this work it refers to 2D or 3D digital models of physical environments. [2](#), [26](#)

Grasshopper A graphical algorithm editor or visual programming language environment within [Rhinceros](#) (formerly a plugin). [12](#), [27](#)

indoor space An area bounded by a roof, and walls on all sides. [1](#)

isovist The total space that is visible from a given viewpoint. It is a representation of space that is often used in architectural analysis, in both 2D and 3D forms. [4](#), [10](#)

outdoor space An area not bounded by a roof or walls. [1](#)

point cloud A (typically large) set of dimensionless points in 3D space, often produced by scanning a physical environment or by means of photogrammetry. [9](#), [10](#)

Rhinceros A 3D computer graphics application, commonly referred to as *Rhino3D*, developed by Robert McNeel & Associates ([2022](#)). [xiii](#), [12](#), [27](#)

semi-indoor space An area bounded by a roof, and with a significant amount of area not bounded by a wall. [2](#)

semi-outdoor space An area bounded on all sides by walls, but not bounded by a roof. [2](#)

space boundary The zone between two different types of spaces. Also referred to as a “transitional space”. [6](#), [10](#)

visualisation channel A way in which a [visualisation mark](#)'s appearance can be altered to encode information visually e.g. position, length, colour saturation, size, shape. Also known as a “visual variable”. [18](#), [25](#)

visualisation mark A geometric primitive that is used to visualise data, e.g. a point, a line, an area. [xiii](#), [18](#)

Chapter 1

Introduction

Navigation tools for the everyday user, from the ubiquitous *Google Maps* to more niche positioning applications, are perpetually developing features to help users better understand physical environments, and to make it easier to compare virtual representations of environments with their real-life counterparts. **Outdoor space** environment visualisation has seen impressive headway in the past decade alone, but successful visualisation techniques for illustrating the configurations of **indoor space** environments, are still lacking (Bauer, Ullmann, and Ludwig [2015](#), 6). The integration of outdoor and indoor spaces within digital, visual representations of space is also still very much in its infancy, and poses difficulties for multiple reasons.

Striving for seamless integration between outdoor and indoor environments is in the interest of end users of many a navigational tool, be it *Citymapper*¹ or a campus map contained within a university’s mobile application. Consider an individual who has made an appointment at a hospital, but has not visited this hospital before, and thus possesses limited knowledge about the hospital’s interior layout. They can enter the hospital’s address into their navigational application of preference on their smartphone, which will advise them on the most suitable driving routes, show train or bus connections, and provide walking instructions on both “ends” of their journey. The only snag is that this navigational application understands the desired destination to be just outside one of the hospital’s entrances. At this point in their journey the individual becomes reliant upon signage provided by the hospital (or alternatively, has to find a service desk where an employee can aid them), to find the hospital wing or room where they are expected. Though powerful in many other respects, widely available mapping services tend to become quite crude when indoor navigation is required of them. Given that the majority of journeys involve at least one endpoint within a building, this lack of continuity between outdoor and indoor navigation is a weakness that has very legitimate repercussions for plenty of individuals.

One might argue that at institutions frequently visited by newcomers, like hospitals, museums, public libraries, and universities, etc., it can be expected that there will be adequate signage to help visitors find their destination from the point at which their mapping application drops them off. It might follow that it is thus non-essential to develop mapping services to fulfil this function of navigation transcending the boundaries between the outdoors and indoors. However, a similar argument could have been made just over a decade ago for why mobile mapping applications of outdoor environments were unnecessary: a physical map already provided all of the necessary information. Why develop a mobile application that would mime the task hitherto being performed adequately by a paper map? A simple answer is innovation. There is the hope that a newer development (that will in most instances be an initially weaker substitute for its predecessor) will have enough potential to be developed until it one day supersedes its precursor’s capabilities.

1. **Citymapper** is a mapping application for urban journeys and uses public transit data

This thesis is an exploratory look at methods of digitally visualising physical environments to further aid users' understanding of the spaces around them when using navigational applications. Although any results will not immediately solve a lack of seamless navigation between outdoors and indoors, they can provide interesting concepts that might lead on to form the basis of better developed future implementations.

1.1 The Challenges of Integrating Outdoor and Indoor Visualisations

One cause for why combined outdoor-indoor space visualisation is not substantially developed, is that information on the two types of spaces is often collected and stored separately by different organisations. Organisations tend to have particular objectives for their collected data, and to avoid unnecessary expenditures they will aim to only collect the most relevant data. In a majority of cases, comprehensive data collection of the entirety of an environment's indoor and outdoor areas would be superfluous for completing the desired task. Data on indoor environments is also often not publicly available if it has been recorded by the occupants or owners of a building for private purposes.

Spatial data is often collected in different representations, at varying resolutions, and some spaces are recorded for use in grid-based models, while others are recorded in a form to suit network-based models (Zlatanova et al. 2014). When datasets on indoor and outdoor spaces *are* suitable for use in the same model, problems still prevail, as identical elements within models can be used to represent different objects. For example, in a network model for an indoor space, nodes might represent the centres of rooms and other destinations of interest (OGC 2019, Section 2), while for an outdoor space, nodes can denote the beginnings and endings of roads (with the roads themselves being represented by edges running between nodes). Another critical factor to consider is that in order to implement user positioning features within mapping tools, GNSS is the predominant player in outdoor environments, while locating a user in indoor environments is usually Wi-Fi- or Bluetooth-dependent.

The existence of semi-indoor space and semi-outdoor spaces only serves to further complicate the matter. Ambiguity often arises when using these terms, so this work will follow the example of (Yan, Diakit , and Zlatanova 2019, 9) where semi-indoor spaces are top-bounded and semi-outdoor spaces are side-bounded (see Figure 1.1 for a visual explanation). Examples of these types of spaces that are neither fully indoor nor outdoor, are covered footpaths (semi-indoor), and enclosed gardens (semi-outdoor). It can at times be useful to exploit these types of spaces' features when path-planning; a semi-indoor space can shield the user from the elements, such as rain or intense sunshine, while a semi-outdoor space might offer an outside oasis, away from bustling thoroughfares. Adversely, a semi-outdoor space can obstruct a user's journey if it is visualised in the same way as other publicly accessible spaces on a map.

A final problem that hampers a smooth union of indoor and outdoor environments into singular visualisations, is that different physical environments and their digital representations have both usually been designed for different users. Outdoor environment applications are often designed for vehicular drivers, and an indoor environment is nearly always designed for walking individuals (note how many indoor environments and their subsequent digital twins do not fully consider the needs of, or information required by, those with accessibility needs, such as wheeled users). It follows that resulting spatial visualisations have usually been customised and equipped with the relevant tools for particular user groups, where subsidiary information is left out. Cyclists are a fitting example of a user group that must often navigate and/or transcend spaces designed for multiple different user groups, but are regularly forgotten about by designers of mapping tools.

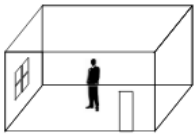
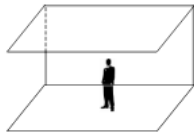
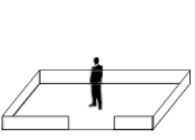
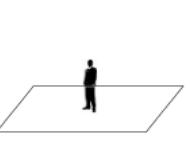




Environment	I-space	sl-space	sO-space	O-space
Definition	$C^I \geq \beta \ \& \ C^S \geq \delta$	$C^I \geq \eta \ \& \ C^S \in [0,1]$ except $C^I \geq \beta \ \& \ C^S \geq \delta$	$C^I < \eta \ \& \ C^S \in [0,1]$ except $C^I < \alpha \ \& \ C^S < \gamma$	$C^I < \alpha \ \& \ C^S < \gamma$
Example				
Scene				
Thresholds	$0 \leq \alpha \leq \eta \leq \beta \leq 1 \ \& \ 0 \leq \gamma \leq \delta \leq 1$			

Figure 1.1: Yan, Diakit , and Zlatanova (2019)'s definitions of the spaces based on the proposed definition framework explained visually

Difficulties of Representing Indoor and Outdoor Spaces in the Same Visualisations

- I Data is collected and stored separately for indoor and outdoor spaces
 - (a) Collection is usually performed by different organisations
 - (b) Indoor data is often not publicly available
 - (c) Indoor and outdoor data is collected in different representations, and at varying resolutions
- II Navigation models can differ, i.e. grid-based versus network-based models
 - (a) When the same navigation models *are* used, identical elements can represent different objects in different types of space, e.g. nodes might represent indoor rooms as well as outdoor road junctions
- III For user positioning, outdoor models use GNSS, while indoor models use Wi-Fi or Bluetooth
- IV Semi-indoor and semi-outdoor spaces complicate space definitions
- V Indoor and outdoor environments are designed with different users in mind, e.g. drivers versus pedestrians

Table 1.1: A summary of some of the difficulties faced when attempting to represent indoor and outdoor spaces in the same visualisation. This work focuses on problem IV

In spite of these challenges, there has been a surge in interest for finding ways of unifying indoor and outdoor spaces within wayfinding applications, due to the far-reaching practical applications that successful solutions could have. Valuable research has been done on combating some of the aforementioned hurdles for unifying indoor and outdoor data (Yan, Zlatanova, and Diakit  2021; Cantarero Navarro et al. 2021), but as of yet there seems to be limited focus on the role or needs of the end user, the person who must ultimately interact with the wayfinding tool that is displaying this spatial data to them. Even if seamless integration of indoor and outdoor data is attained, it means little if it cannot help the end user with their wayfinding task, or in comparing digital twins with physical counterparts. Additionally, information about the nature and accessibility of a transitional space between different environments is often required (Alattas 2022). For example, a user who is told

that a particular door leads to an indoor space will need additional information if they are restricted by the width of a wheelchair or a pushchair, or if that door is for personnel only.

In the ongoing desire to make the digital twins of physical environments as accurate as possible (be it for business incentives, and/or wanting to reduce navigational errors made by users), one faces further dilemmas. Where unnecessary information is visualised in the name of gaining a comprehensive view, there is the risk of information overload for the user (Kraak and Ormeling 2021), and lagging loading of real-time data has been shown to impair user experiences (Khalid et al. 2014). Puikkonen et al. (2009) explicitly recommended the simplification of digital indoor maps for limiting the user’s cognitive load. Finding a way of representing 3D space in a more simplistic fashion might solve these issues, and when specifically applied to the boundaries of outdoor spaces, could provide new research in this field.

1.2 Enter: The Isovist

One such simplified representation of a space, is the **isovist**. An isovist can be best defined in the following way: “Given a point x in a space P , the isovist at x , V_x , is the subset of P visible from x .” (Davis and Benedikt 1979).

An application of isovists that explains their nature well, is the Art Gallery problem, a visibility problem devised by Victor Klee in 1973 that poses the question, how many stationary guards (or security cameras) does one require to surveil a particular amount of an area (the amount usually being ‘all,’ or ‘nearly all’)? (Couto, Souza, and De Rezende 2007, 1). Isovists can come into play when trying to conceptualise the problem, as their main function is calculating how much of an area is visible from a particular vantage point. With computer software it becomes relatively simple to calculate how multiple vantage points’ isovists intersect one another, and what space is being redundantly surveilled twice or more.

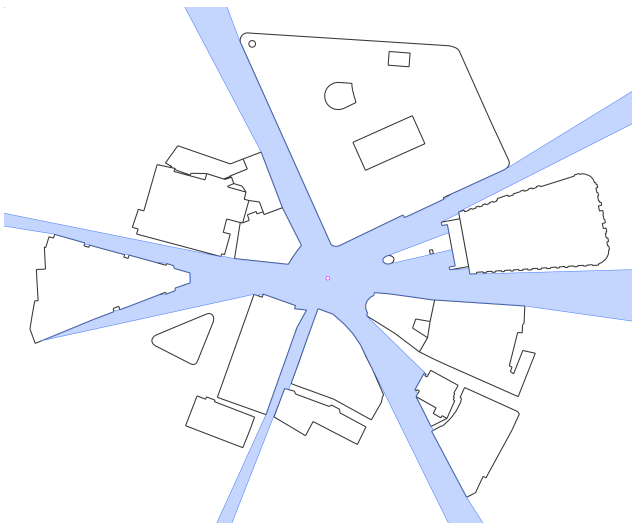


Figure 1.2: A demonstration of a 2D isovist where the blue area represents the ground surface area in direct line of sight from the central dot (the user’s position). Map data taken from *OpenStreetMap*

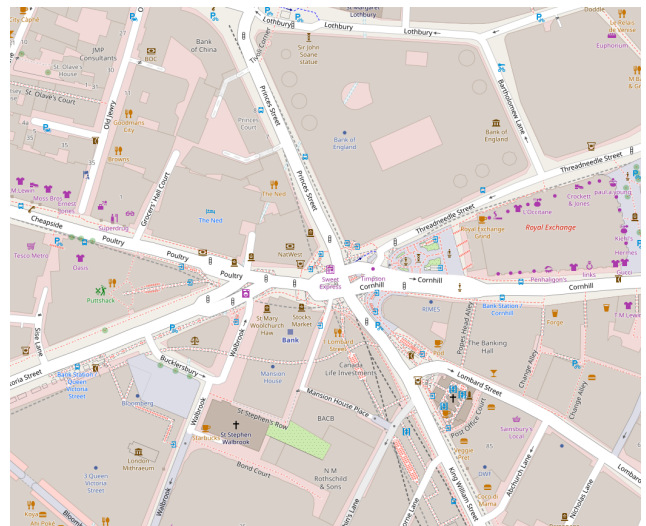


Figure 1.3: A screenshot taken from *OpenStreetMap* (OpenStreetMap contributors 2022), showing the centre of Bank Junction in the City of London, United Kingdom, as the user’s position that is depicted in isovist form in

Figure 1.2

Figure 1.2 shows how a 2D isovist is often visualised, with a point indicating the location of the viewer, and a shaded area indicating the extent of the ground surface area visible to them. This basic visualisation can be developed in many ways to become more accurate or informative. **Figure 1.5**

gives the isovist of a person with a 120 degree view, rather than the less realistic 360 degree view, which concurrently provides an understanding of the direction in which the user is facing. [Figure 1.6](#) meanwhile shows how multiple isovists from different perspectives interact with one another, where the darker shaded area indicates an overlap between the three separate viewpoints' isovists. It is easy to imagine how useful such an overview of multiple isovists could be when directing people to a certain location, describing landmarks that are visible from each individual's perspective.



Figure 1.4: A panorama photograph of Bank Junction in the City of London, United Kingdom, showing six of the nine roads that lead to the junction

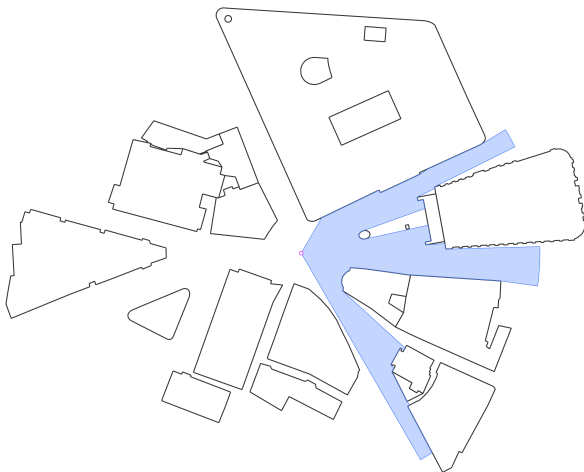


Figure 1.5: An example of how a 2D isovist can be customised: A viewer facing east has a peripheral vision of 120°, with a view limit of only about 100 metres

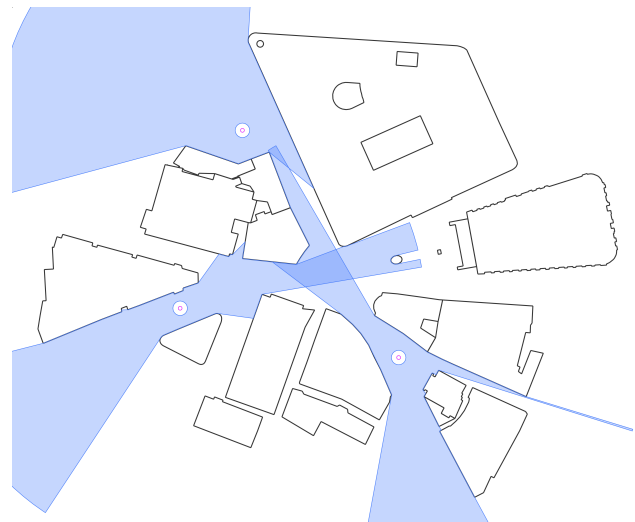


Figure 1.6: An example of how multiple 2D isovists can be contained in the same visualisation. The darker shaded areas represent where the isovists overlap one another

A weakness of these 2D isovists is that they do not take into account the heights of obstacles, or materials from which they are made. [Figure 1.2](#) shows a small, circular obstacle towards the right end of Bank junction, and treats this in the same way as all of the black lines – a solid wall with a height too high for the user to be able to see beyond it. In reality, this small circular object is a statue of the Duke of Wellington, and a user standing not too close in front of it can certainly see the Royal Exchange behind and above the statue's head. Additionally, the statue's plinth is far wider than the statue itself, meaning that the 2D isovist incorrectly interprets the statue as occupying a column of space as wide as the plinth, and infinitely high. That a user might be able to see what is behind a window or other threshold is also not accounted for. Some isovist-generating applications do allow for boundaries to be assigned different attributes, such as being a window or a reflective surface. [Figure 1.8](#) shows an isovist created using the (McElhinney [2022](#)) template, where cyan lines represent transparent boundary surfaces (floor-to-ceiling windows for example), and dark blue lines represent

reflective surfaces (mirrors). The resulting isovist visualisation depicts four different kinds of visible space: the visible space that is accessible in blue, the directly visible but inaccessible space in cyan, the space seen indirectly via a reflective surface in red, and imaginary space seen behind a mirror in yellow. This example shows how isovists can be visualised in different ways in order to be specialised for different tasks. It also highlights the importance of defining what counts as part of an isovist; is it only directly accessible, visible space, or do reflected images also count?



Figure 1.7: A red line indicates the outline of the Duke of Wellington statue at Bank junction, showing the space it actually obscures, versus what [Figure 1.5](#)'s 2D isovist suggests

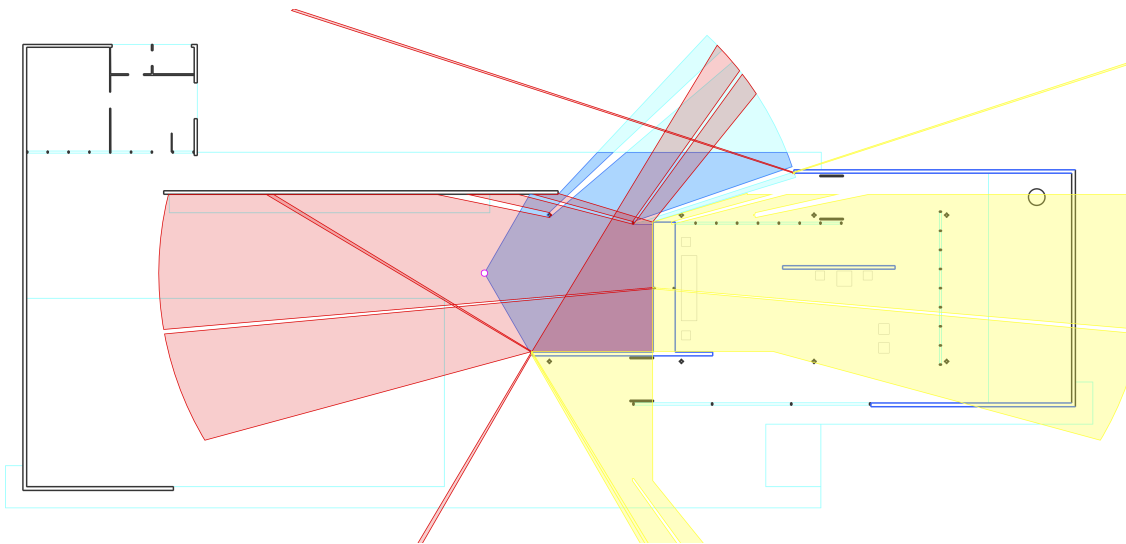


Figure 1.8: A 2D Isovist showing accessible (blue), inaccessible but visible (cyan), reflected (red), and spectral (yellow) spaces from a single standpoint. Though more accurate, it also leads to complexity. Made with (McElhinney [2022](#))

Even in their infancy, it was clear that isovists had potential for wide-reaching practical applications (Davis and Benedikt [1979](#), Section 3.4), from analysing architectural designs' perceived openness, to determining whether someone could have witnessed a crime from a particular position. But to the knowledge of this author, there has not yet been an exploration into their suitability for representing the area surrounding a [space boundary](#) to an end user in the context of enhancing their understanding of the space. Additionally, the optimal geovisualisation of 'non-isovists' (i.e., the space in a locale

that does not make up the isovist) seems to be a rather unexplored territory. In current visualisations of spaces, users receive a lot of information on what they *can* see, but very little about what they cannot see. Showing a user a wall that they can already see in their environment, does not provide the user with much new information, while showing them what is behind a wall, does provide new information. Clear visualisation of the digital twins of a user’s environment is useful for orientation purposes, but there is a limit to how much use detailed visualisations have. Architectural elements like walls and building façades take up a lot of room in a visualisation, but convey comparatively very little information. Meanwhile, doorways and other smaller features of importance are often not as obvious, though they do contain valuable information for a user.

Providing a means of visualising a scene that does not simply remove all areas not contained within the isovist, but finding ways of visualising these spaces too (while still retaining a clear distinction between the visible and hidden areas), could contribute new findings to the field of geovisualisation. This thesis will be exploring how the environments and transitions between outdoor space and three other types of space (indoor, semi-indoor, and semi-outdoor) can be clearly communicated to the user by means of intuitively visualising isovists of the space, as well as the occluded ‘non-isovist’ surroundings.

A very similar (and at times interchangeable) concept to the isovist is the viewshed, so inspiration could potentially be taken from how viewsheds are designed and developed (Petrasova et al. [2015](#), 77). While isovists are commonly used in indoor and built-up environments, viewsheds are often found in terrain analysis, and show the effects that a varying elevation can have on visibility.

1.3 Dutch Golden Age Painters as Forerunners of Geospatial Visualists

A number of Dutch Golden Age artists were aware of the complexities of visualising multiple rooms, and of incorporating both indoor and outdoor environments into a single painting. But not only were they aware of the complexities: they would actively seek out opportunities to depict these labyrinthine perspectives, as a way of demonstrating their artistic skill. Johannes Vermeer was notable for featuring windows in his domestic scenes, and often painted foregrounds of draping curtains and open doorways to convey the intimacy of the scene taking place beyond. Two of his most famous paintings, “The Little Street” (see [Figure 1.9a](#)) and “The Love Letter,” are examples of his penchant for open doorways and walkways, and the private scenes occurring beyond them.

Two of the “Master of Light”’s contemporaries went far further than Vermeer did in terms of depicting complex background layouts, namely Pieter de Hooch and Samuel van Hoogstraten. Both hailed from nearby Vermeer’s Delft, but are less popularly known in the present day. de Hooch was adroit at scenes where open doors offer teasing glimpses of other rooms or courtyards; [Figure 1.9c](#) shows one of his works where an indoor room in the foreground leads on to what looks like a courtyard, then to a passageway out into the street, and if one looks closely, there is a woman inside a doorway on the other side of the canal, overseeing what is happening in the foreground setting. Essentially, de Hooch has illustrated a path from indoor, to semi-outdoor, to semi-indoor, to outside space, and then back to indoor space again, all in one view! van Hoogstraten created one of the few still in-tact “perspective boxes,” where two peepholes provide opposing views of the same interior with multiple rooms and hallways, in near Escheresque fashion (Nakamura [2020](#)). In art, the points where the outside can be seen from within, and vice versa, are known as “through-views.” It is the objective of this thesis to find a modern, more visually simplistic equivalent of representing these through-views, in a way that is temporally dynamic (it changes with the user’s demands in real-time), such that a user can understand the kinds of space boundaries they are encountering.

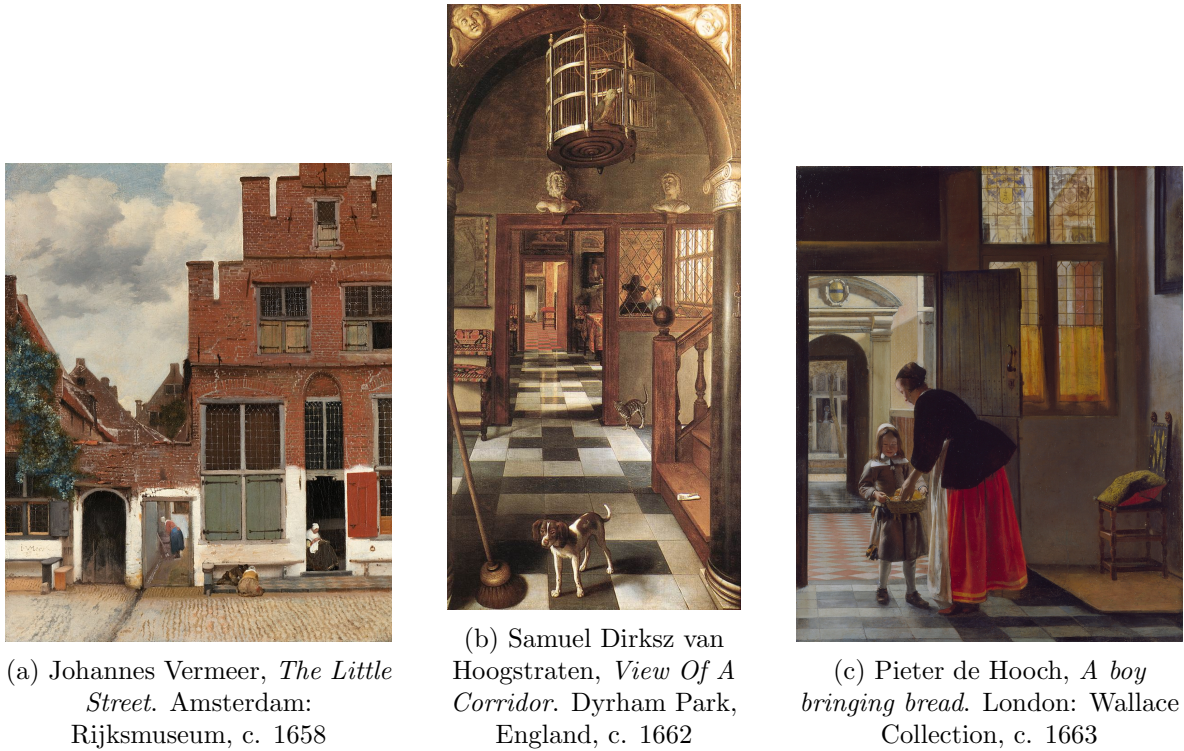


Figure 1.9: Three examples of Dutch Golden Age painters' depictions of complex indoor environments



Figure 1.10: Samuel Dirksz van Hoogstraten, *A Peepshow with Views of the Interior of a Dutch House*. London: The National Gallery, c. 1655.

To help demonstrate what is meant by visualising different spaces and boundaries in the form of isovists, [Figure 1.11](#) provides a simple example of how Pieter de Hooch's painting from [Figure 1.9c](#) could be divided into different types of spaces. The indoor space is red; semi-outdoor, yellow; semi-indoor, green; and outdoor, purple. This is a static view, and everything visible in the painting is of course a part of the painter's isovist. This stationary view already looks rather complex with all four space types present (including two instances of indoor space), so it is easy to see how a dynamic isovist visualisation that also incorporates methods of visualising hidden spaces around corners, would

require careful design consideration.



Figure 1.11: Splitting de Hooch’s *A boy bringing bread* (right, [Figure 1.9c](#)) into the different types of spaces depicted (left)

1.4 Point Clouds

[Point clouds](#) are sets of dimensionless points in space that have each been assigned a set of coordinates, and optionally additional values for any number of other attributes. Point clouds can be created by scanning an environment with a [LiDAR](#) scanner where a laser beam builds up a 3D image of a scene (other methods include photogrammetry, where point locations are triangulated from multiple 2D images). The laser beam is reflected back to the scanner upon encountering the first obstacle in its path, and this is the position at which a point will be placed. Enough points together form a point cloud that generally forms a recognisable enough image for the human eye.

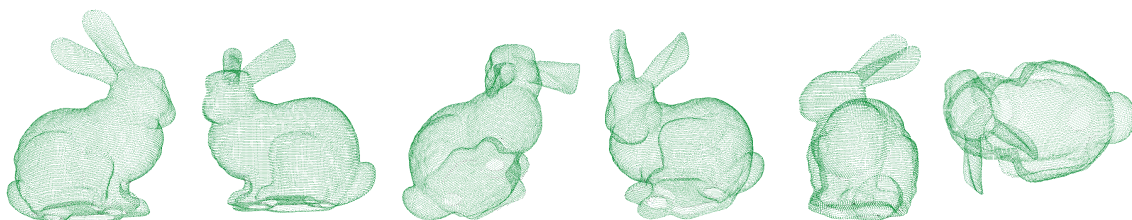


Figure 1.12: The “Stanford Bunny” (Stanford University Computer Graphics Laboratory [1994](#)) as a raw point cloud, viewed from six different angles

Points clouds are rarely directly used as the final form of visualisation of a space, most often undergoing transformations into triangulated meshes, voxel models, or other models like Building Information Models (BIM) (Liu et al. [2022](#); Xiong et al. [2013](#); Nebiker, Bleisch, and Christen [2010](#)). This work will incorporate the direct use of point clouds within the visualisations produced. The reason that

this thesis has opted for point cloud data as the input data, partially due to the emerging availability of LiDAR scanning on smartphones. If enough users were to opt to share their scans of environments (which could conceivably include scans taken passively while a user walks around with a phone in-hand), a substantial database of points could result in more accurate and extensive point clouds of transitional spaces. This would help with making more visualisations of more spaces, more accurately. Point clouds are also direct representations of space, and capture a scene, ‘as is.’ Even where specks of dust and other impurities might cause unwanted, rogue points to be created, points clouds remain precise representations of reality. Another advantage of point clouds for this specific research, is that LiDAR scans make no differentiation between the outdoors and indoors. There is no hierarchy attributed to the objects that have caused the scanner to record points at specific locations.

1.5 Research Questions

Combining the concepts of space boundary, isovist, and point cloud, the research question of this thesis became, “To what extent could point clouds be successfully used to visualise third-person view isovists of, and occluded spaces surrounding, outdoor space boundaries, in a way that helps users of the visualisation better understand these environments?”

In order to tackle this faceted research question, sub-research questions were established to help guide the research process. Being able to answer the sub-research questions should in turn result in an answer for the main research question.

The sub-research questions are as follows, and explained in more detail below:

1. How can point cloud data be best classified into types of spaces and space boundaries, as well as classified as ‘visible’ or ‘occluded,’ in the context of a third-person view?
2. What visualisation techniques can be used upon a point cloud to visualise different types of spaces within and outside of an isovist, and how can the space boundaries be made clearly distinguishable?
3. Which visualisation techniques are the most successful when evaluated against a list of requirements?
4. What does the visualisation of isovists from a third-person view offer, that other visualisation techniques of outdoor space boundaries do not, and does visualising the non-isovist area have added value?

1.5.1 Sub-Research Question 1: Classifying Point Clouds

The first sub-research question addresses the first stage of the research, namely data collection and processing. This thesis has chosen for input data to be collected in the form of point clouds. The reasoning behind this decision is that point clouds are direct representations of space, can contain much detail about a space, and are able to record all kinds of spaces. The individual points in a cloud can contain multiple additional attributes such as colour, and in the case of this thesis, the type of space to which they belong, can also be assigned. They are a raw form of data, taken directly from the output of a scanner, and have not been manipulated, which would otherwise risk a loss in accuracy or information, not to mention the additional processing time required for transforming raw data into certain model types. With continual advances in handheld technologies, it is not credulous to suggest that the near future could see the vast majority of smartphone users (unwittingly) having a LiDAR scanner at their disposal. This means that users who choose to share data on their surroundings, could easily collect point cloud data, potentially even passively as they walk through a space with phone in-hand. If a large enough repository of point cloud data on building interiors and surroundings is built up in this way, it could become feasible to apply (derivatives of) the visualisations that will

be developed in this project, upon that data, ready for direct consumption by individuals requiring navigational assistance.

The choice was made to use the raw point cloud data in the visualisation process, rather than first manipulating it into another form. For example, generating a 3D mesh is a common end use for point clouds, and it has been claimed that a mesh “provides more comprehensive representation” than a point cloud (Lv, Lin, and Zhao [2022](#)). Though meshes have a lot of potential for clear and efficient visualisation of space, there is the risk that a point cloud can be inaccurately transformed, such that the mesh incorrectly represents the space (a rogue point within a point cloud can create a plane in the mesh that does not correctly represent reality).

Though point clouds do have many advantages, their shortcomings must not be forgotten, so that these can be mitigated as far as possible, and so that inherent limitations are acknowledged. The main difficulty with point clouds is the sense of clutter that they can create, and that impurities or temporary obstacles in an environment can be picked up by a sensor, resulting in rogue points occupying what would otherwise be deemed to be ‘empty space’. Especially if mass collection of point clouds occurs in dynamic environments, temporary users of the space will invariably be scanned along with the stationary environment. On the other end of the spectrum, a point cloud with too few readings is unlikely to clearly represent the 3D space – it is naturally more intuitive for a user to understand that a continuous plane within a mesh represents a wall, than to use their mind’s eye to connect sparse points contained in the same plane, and form the image of a wall.

The primary research component of this question entails an exploration of ways in which the point cloud can be best classified into visible and occluded space, and into the different types of spaces (indoor, outdoor, etc.). But before that can be done, data collection will need to take place in order to obtain workable point clouds of the types of spaces with which we are dealing. It would be ideal to do so with a mobile application, to best mimic how an intended end user experiences their surroundings.

Classification options within *CloudCompare* (CloudCompare [2022](#)) software will be explored, and although automatic classification of point clouds is not the focus of this thesis, looking into algorithms that could automatically classify the point clouds into the different types of spaces (such as determining the space type by calculating how many walls are bounding an area) could add value to the project’s end product.

It is anticipated that determining whether points in a point cloud are visible or occluded from a particular point of view, will provide a significant challenge, as the nature of point clouds means that common visibility analysis methods cannot be (straightforwardly) used, ray-casting being a notable example. (Existing methods are discussed in [section 2.4](#)).

Answering sub-research question 1 will involve an evaluation of the most appropriate software and existing methods for data collection, cleaning, and classification. It will entail categorising the acquired point cloud data into different space types, and performing visibility analysis upon this classified data. Definitively choosing the software to be used in this project is needed for answering this sub-question.

1.5.2 Sub-Research Question 2: Exploring Visualisation Techniques

Sub-research question 2 tackles the core and defining focus of this thesis, namely geo-data visualisation. Using various software and experimenting with different visualisation techniques, this portion of the thesis is meant to be creative and exploratory, hopefully resulting in novel methods for visualising 3D space. In order to provide some structure to the design process, a list of requirements detailing what a successful visualisation should (not) contain and be able to do, will be imperative for ensuring that the visualisation(s) fulfil the intended end user’s needs. This list of requirements will require supporting literature, as well as guidelines for how the designs can be evaluated with respect to the extent to which the requirements have been fulfilled. Without a guide such as this, there is a danger that resulting designs might be creative, but not of any practical use, or even that there is existing

literature already proving that such methods of visualisation are not a good means of communicating space.

Initial design concepts can be sketched out on paper so as not to allow the limits of a digital design package, restrict the creativity of ideas generated. Then the designs with the most perceived potential can be translated into digital sketches for a more polished communication of ideas. These visualisation designs should then be brought to life using various software deemed suitable for the task.

The 3D computer graphics application, *Rhinoceros* (Robert McNeel & Associates 2022), has a powerful graphical algorithm editor, *Grasshopper* (Rutten 2022), which looks promising for this project. Used by many architects, *Grasshopper* can generate isovists, but lacks direct support for point clouds. A point cloud processing plugin such as *Cockroach* for *Rhinoceros* may need to be used to overcome this hurdle. Another potential software candidate was *CityEngine*, due to its widespread use in the field, but unfortunately it “does not support LiDAR data nor are tools provided for the automatic extraction of surfaces out of point clouds” (ESRI R&D Center Zurich 2018).

Due to the limitations of lag times and difficulty in updating models with real-time data in applications such as *Rhinoceros*, a game engine such as *Unity* (Unity Technologies 2021) could facilitate more a immersive user experience. A game engine can provide a better example of how a visualisation interface would look and be operated in real-time with incoming live data streams.

1.5.3 Sub-Research Question 3: Evaluating Visualisation Designs

The third sub-research question follows on naturally from sub-research question 2, where an evaluation of the various designs developed, will be performed. This will require use of the list of requirements (also created in the previous step), and a written evaluation of the different designs against the list of requirements, will be given. Not all requirements can be evaluated in a binary manner, and some evaluation will involve subjective reasoning, which is why user testing would add value to this project. However, a large and diverse enough group of participants would need to be gathered, in order for user testing to be worthwhile. With only a few participants, it would be difficult to draw definitive conclusions, and if participants have significant experience with geovisualisations, this can also affect their perceptions of the visualisations.

Within the scope of this project, the intention is that the various designs are compared and contrasted with one another, and weaknesses are identified such that future design iterations can effectively build upon the work completed here.

1.5.4 Sub-Research Question 4: Reflecting on Visualisations’ Usefulness

In contrast to sub-research question 3, this sub-research question has a broader focus on the geovisualisation field as a whole. In answering this sub-research question, it is the intention that this research can be put in the broader context of geovisualisation and isovist usage. A considered conclusion of the span of this project must address this sub-research question.

1.6 Scientific and Societal Relevance

As likely already inferred from this introduction, the societal relevance of this project is very practical. Were this research to find that the isovists are a successful method for visualising outdoor space boundaries to end users (that is, they positively affect user experience by decreasing the time taken for a user to recognise a boundary type, and give the user confidence in knowing what lies beyond the boundary), this approach could be integrated into a variety of navigational applications. Were it to be discovered that they are not a successful tool for enhanced situation awareness, this can be a contribution to the existing scientific literature in the domain of visualising outdoor-indoor boundaries.

This work opens a field of research about transitional spaces, and how these are experienced by individuals. Most visualisation applications still keep the outdoor separate from the indoor, and only visualise one or the other, but the users of these applications need to transcend the boundaries between these spaces in nearly all of the journeys that they take. Cities are building more non-standard structures with every passing year; some private building planning applications require publicly accessible rooftops in order to gain building permission; many structures contain public footpaths underneath them; and metro networks are continually growing. Spaces are becoming more difficult to make sense of, as some spaces can no longer be classified into the binary ‘outdoor’ or ‘indoor.’ While the urban environment is becoming more quirky and inventive, new technologies are needed to describe these environments, so that users of these environments can make sense of them.

(At long last) it is also becoming less acceptable not to cater for different accessibility needs. This thesis indirectly works towards providing more useful information for users with certain mobility needs, so that they can understand which spaces are safe for them. This is because the concept of classifying areas into different ‘space types’ in this thesis, can be applied to other information about a space, such as classifying step-free areas, or spaces with wider doorways.

An additional practical application of this work could be for emergency situations where responders to a scene are directed by an external command centre. This visualisation technique of employing isovists might help command centres in understanding what their responders can and cannot see from their perspectives. A simple representation like an isovist, of the responders’ views, could reduce the cognitive load of the commanders, and help focus on relevant information. Overlapping isovists would quickly highlight to command centres where their responders have shared view, allowing them to more effectively communicate, and provide directions and instructions.

Chapter 2

Related Research

This brief review of existing literature will cover the following areas of research:

1. Endeavours to visualise outdoor-indoor boundaries for end users
2. Tools that help users navigate indoor and outdoor spaces
3. The use of isovists as a visualisation tool for end users
4. Performing visibility analysis on point clouds
5. Data visualisation techniques for effective communication

2.1 Outdoor-Indoor Boundary Visualisation

As touched upon in the introduction, there have been significant contributions in the past decades to the body of work on integrating indoor data with outdoor data. But focus on the visualisation of these boundaries *for the end user* is still in relative infancy. Nikander et al. (2013) explored integrating indoor and outdoor data into a single mobile navigation application by converting CAD floor plans into GIS data (see Figure 2.1). The floor plans were overlaid onto a satellite view, resulting in a very clear distinction between indoor and outdoor, but although comprehensive, resulted in a busy-feeling interface. There is also only the possibility of a bird’s-eye view, and floor plans are notorious for often being out-of-date – buildings sometimes never acquire an updated floor plan after construction is complete, despite building processes regularly deviating from initial blueprints.

Kim and Wilson (2014, 1) also used floor plans, generating 3D indoor models “from CAD files and building footprints using *CityEngine* and its built-in procedural modelling approach,” showing how the application can generate ‘3D web scenes.’ This work demonstrates *CityEngine* (ESRI R&D Center Zurich 2018) in use for visualising 3D indoor models and user routes, but there was little focus on how to optimise the connection with outdoor space, aside from visualising building façades.

Applying the assets of the *IndoorGML* standard (OGC 2019) to outdoor areas, and combining this with *Open Location Code* and *OpenStreetMap* information for outdoor areas, Cantarero Navarro et al. (2021) designed an approach for modelling outdoor-indoor spaces that can provide information from multiple sources at different levels of abstraction. The shortfalls of the method were that many buildings are not already modelled in *IndoorGML*, and some outdoor spaces were not modelled (accurately) in *OpenStreetMap*, requiring manual generation or corrections of the models.



Figure 2.1: Nikander et al. (2013)’s prototype application with an indoor floor plan overlaying outdoor satellite imagery

2.2 The Wayfinding Experience

Transition entropy refers to the amount of exploratory visual scanning being performed, measured by how often a user’s gaze transitions between different Areas of Interest (AOI). Stationary entropy refers to how even the distribution of visual attention is across various AOIs; a lower stationary gaze entropy is indicative of difficulty with a navigation task. Kapaj, Lanini-Maggi, and Fabrikant (2021) demonstrated that users navigating with a map depicting “3D photorealistic buildings” had a higher stationary entropy than users given “2D building footprints,” suggesting that 3D visuals might help make navigation easier.

Makri, Zlatanova, and Verbree (2015) applied the concepts of the Dutch outdoor Junction Network System for cyclists, to indoor, pedestrian navigation, to positive effect. This showed that adapting an outdoor system for indoor use can be a viable option for making a user’s transition between different spaces, more seamless.

Bauer, Ullmann, and Ludwig (2016) found that users using more abstract navigation interfaces performed better in navigation tasks; they arrived at their destinations faster, and spent “significantly less visual attention on the screen.” For indoor navigation, marking indoor landmarks like doors and stairs, was also deemed helpful. Slightly at odds with Bauer, Ullmann, and Ludwig (2016)’s findings, Lanini-Maggi, Ruginski, and Fabrikant (2021) discovered that users tended to look at a first-person video more often than an abstract map during navigation tasks, and that receiving emotionally laden navigation instructions led to even more use of this first-person video perspective.

2.3 Isovists as Visualisation Tools

Following an evaluation of eleven methods for representing 3D isovists, Dalton and Dalton (2015) designed three further representations for 3D isovists in 2D form. The most successful diagram style was what they coined the “Contour Isovist,” a diagram that layered isovists from different eye heights on top of one another (see Figure 2.2).

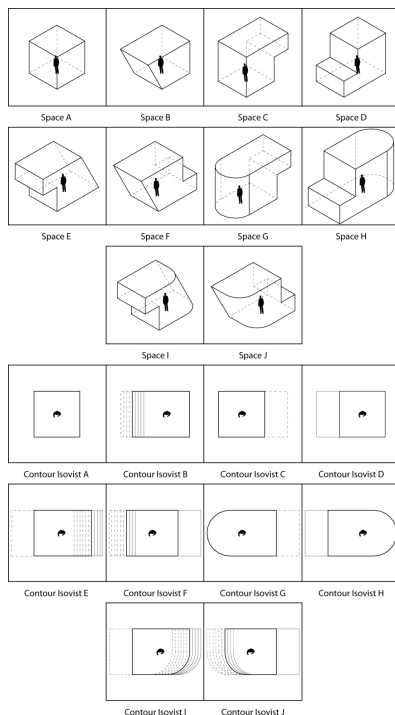


Figure 2.2: Image taken from Dalton and Dalton (2015). Dalton and Dalton (2015)’s *Contour Isovists for Ten Simple Volumetric Spaces*

It is also worth noting that an “embodied” 3D isovist can be a “more reliable predictor of the human perception of space compared to generic volumetric 3D isovists” (Krukar et al. 2020, 2319), because of the different ways in which we perceive vertical and horizontal space. Those authors argue that because of this, simply implementing the features of a 2D isovist in 3D space, will not make for a successful 3D isovist.

2.4 Visibility Analysis on Point Clouds

Díaz-Vilariño et al. (2018) showed how point cloud data can be generated into 3D isovists for indoor environments, by first discretising the cloud into voxel-based structures, and then running a ray-tracing algorithm on the voxels, classifying them as visible or occluded.

Peters, Ledoux, and Biljecki (2015) showed how the medial axis transform can be utilised for performing visibility analysis upon point clouds directly.

Katz, Tal, and Basri (2007) developed a “Hidden Point Removal (HPR) operator” as a form of visibility analysis that could be performed directly on point clouds. The operator determines which points in a point cloud are visible from a given viewpoint, without the need for surface reconstruction, or the estimation of normals. Their method involved inverting point clouds and extracting the points residing on the convex hulls of the “transformed” point clouds (these being the initially visible points). To give a more complete explanation, a sphere is defined with a centre at the camera position, and a radius R that reaches the furthest point away from the centre, meaning that all of the point cloud’s points are contained within the sphere. The points are now ‘reflected’ along the boundary of this sphere so that the new, transformed points are all on the outside of the sphere. Now a **convex hull** is made from all of the points, and given that the initially closest points to the camera, are now the furthest away, they should be components of the convex hull. These points that lie along the convex hull, should all be visible, and all points inside of the hull should be hidden. The method also works from a viewpoint within the point cloud, and “a change in camera rotation or field of view does not

require a re-calculation of visibility.”



Figure 2.3: Images taken from Katz, Tal, and Basri (2007). A point cloud of Michelangelo’s *David*; on the left the input to the Hidden Point Removal operator, on the right the output

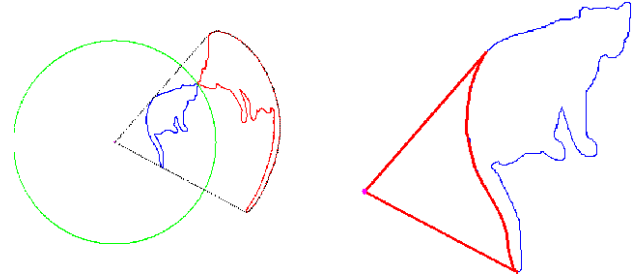


Figure 2.4: Taken from Katz, Tal, and Basri (2007): “Left: spherical flipping (in red) of a 2D curve (in blue) using a sphere (in green) centered at the view point (in magenta). Right: back projection of the convex hull. Note that this image is used only for illustration; in practice, R is much larger.”

One weakness to their method, is that if the distance between the viewpoint and certain points is small, large portions of the space can be occluded by the points in question. A smaller R leads to more false negatives (visible points are missing in the point cloud), while a larger R leads to more false positives (too many (occluded) points are contained in the point cloud).

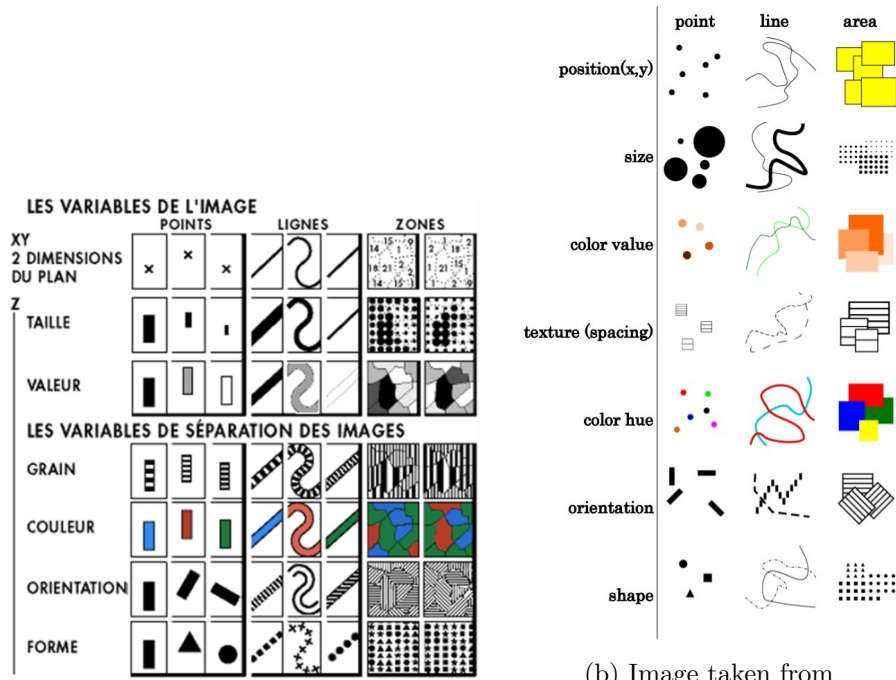
2.5 Data Visualisation Practices

Though existing for hundreds of years, data visualisation is a flourishing field that has been gaining growing awareness since the late twentieth century. Over the past few decades, extensive work and research has been carried out in the discipline, especially with regard to quantifying how individuals perceive and interpret data visualisations.

The cartographer Bertin (1967) used marks and channels to explain how to visually encode information effectively. **Visualisation marks** are the building blocks for representing data, and can be modified with **visualisation channels** to customise their appearance. A mark can be any geometric primitive, such as points, lines, areas, surfaces, and volumes. A channel can be a mark’s position, size, shape, colour (distinctions are drawn between hue, value, saturation, and luminance), orientation, grain, etc. A modified version of Bertin’s visual variables can be seen in **Figure 2.5b**.

It is crucial to understand the Expressiveness and Effectiveness Principles before applying visualisation channels to data. The Expressiveness Principle states that a visualisation must only encode all of the information it intends to encode, and not imply any other relationships within the data (Card 2008, 523). In practice, this means that the chosen visualisation channel should be suited to the type of data it is representing. The Effectiveness Principle states that a user must be able to perceive differences and patterns in the data. In practice, this means that the most important attributes should be encoded with the channels that are best at highlighting the characteristics of that data type.

An order of effectiveness was developed by Munzner (2015, Ch 5) for channels, and the order is depen-



(a) Image taken from Bertin (1967). Bertin (1967)'s *Visual Principles*

(b) Image taken from Krassanakis, Mitropoulos, and Nakos (2013). Bertin (1967)'s *Visual Principles*, modified

Figure 2.5: Two depictions of Bertin (1967)'s *Visual Principles*: Figure 2.5a from Bertin's original work, and Figure 2.5b is a modernised version of the same concept, developed by (Krassanakis, Mitropoulos, and Nakos 2013)

dent on the type of data that the marks are representing (be it categorical, ordinal, or quantitative), see Figure 2.6.

An approach for tackling a data visualisation task and identifying weaknesses in the visualisation, is using the four layers of Munzner (2009)'s "Nested Model for Visualization Design and Validation." Figure 2.7 displays the four layers, which will be referred to here as: domain situation, data and task abstraction, visual encoding, and algorithm design. Each level feeds into the following level, meaning that any mistake made in a prior level, has consequences further along the visualisation pipeline. The intention of this model is to let the designer of a data visualisation see how to evaluate different aspects of the design, and to identify the threats in the way of creating a 'good' visualisation at each step of the way. In subsection 3.4.2, critical questions appropriate for each of the first three levels are posed and answered.

Brehmer and Munzner (2013) created a typology of visualisation tasks, including reasons for *why* a task is being performed by the user of a data visualisation, and *how* it can be executed. A graphic of this typology can be seen in Figure 2.9.

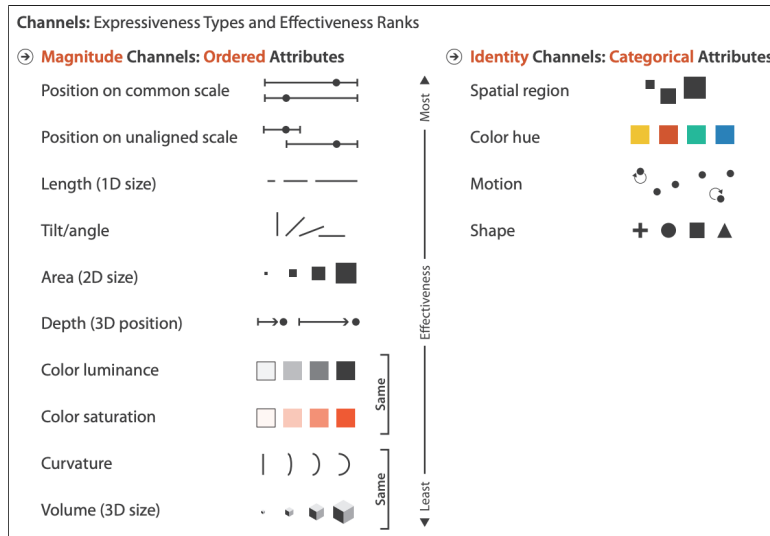


Figure 2.6: Image taken from Munzner (2015), Chapter 5. A ranking of the effectiveness of different channels, depending on the data type being visualisation

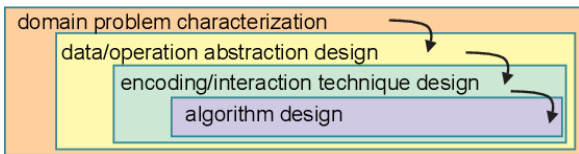


Figure 2.7: Image taken from Munzner (2009). Munzner (2009)'s "Nested Model for Visualization Design and Validation"

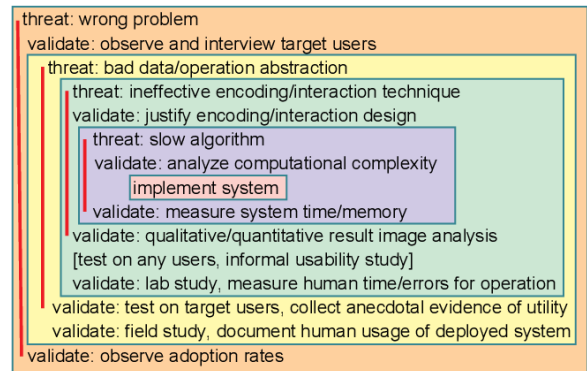


Figure 2.8: Image taken from Munzner (2009). Identifying the threats at each level of Munzner (2009)'s nested model in Figure 2.7, and how to validate the visualisation in each layer

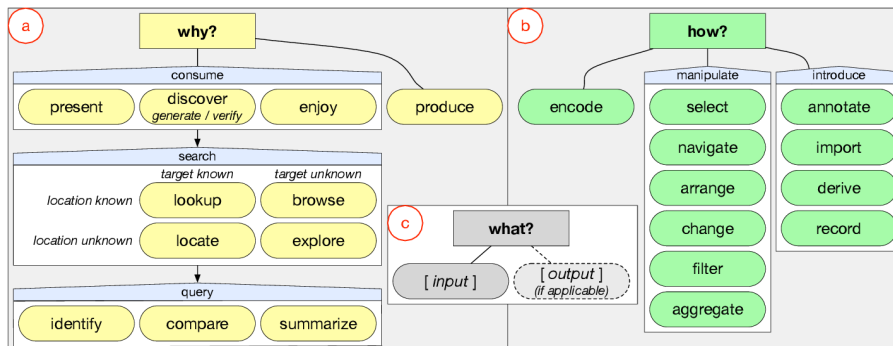


Figure 2.9: Image taken from Brehmer and Munzner (2013). Brehmer and Munzner (2013)'s *Typology of Abstract Visualization Tasks*

Chapter 3

Methodology

3.1 Data Collection

Data collection was partially performed first-hand by means of a smartphone (namely the iPhone 12 Pro model, which has a LiDAR sensor), using the *Scaniverse* application (Toolbox AI [2021](#)) to take LiDAR scans of environments. Initially, four different kinds of space boundaries in the local area were selected to test out the quality of the LiDAR scans, which can be described as follows:

1. **Main Entrance:** Outdoor to Indoor space boundary
2. **Car Park Entrance:** Outdoor to Indoor space boundary (this boundary could arguably also be classified as an Outdoor to Semi-Indoor one, as some walls contain gaps, and it does not ‘feel’ as though it is a truly indoor space)
3. **Pedestrian Entrance:** Outdoor to Semi-Outdoor space boundary (the Semi-Outdoor space subsequently leads on to the car park area, which is here considered to be an Indoor space)
4. **Car Park Stairs:** Outdoor to Semi-Indoor space boundary (the location was next to a set of stairs, but the stairs are not visible)

These locations were not exemplary examples of the types of space boundaries with which this thesis is dealing, but served the purpose of testing out the LiDAR software. *Scaniverse* produced some impressive results considering that scanning was being performed free-hand with a mobile phone (see [Figure 3.1](#) for the resulting raw point clouds), but there were some serious drawbacks. Firstly, it was immediately obvious that windows caused issues for the LiDAR scanner, which was not surprising, given that it is fairly well documented that LiDAR scans find windows troublesome (Smit [2022](#)). An example of this can be seen in [Figure 3.1a](#), a point cloud scan of the “main entrance” with glass walls and a revolving door. Secondly, the point cloud had a limited range for scanning. The application claims a maximum distance of five metres, but the quality of the scan already deteriorates before the five metre mark. This is not ideal for this project, as most users have a far greater field of vision.

Scaniverse is often used to generate mesh models from scans, but the point clouds can be exported in the [PLY](#) and [LAS](#) format, which was done in the case of this thesis, ready for cleaning. The points in the cloud also contain information on the colour attribute, recorded with the [RGB](#) colour model.

Though the self-made scans of the local environment are useful for testing visualisations, at this stage it is already clear that they will not have the capacity for serious future testing beyond the scope of this project, and that a much more complex test dataset can not be generated solely through use of the iPhone scans. One of the more well-known large and pre-existing datasets is the “Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS),” but as the name suggests, the data ‘only’ covers six indoor areas (or a total of 271 rooms), and not other space-types like (semi-) outdoor.

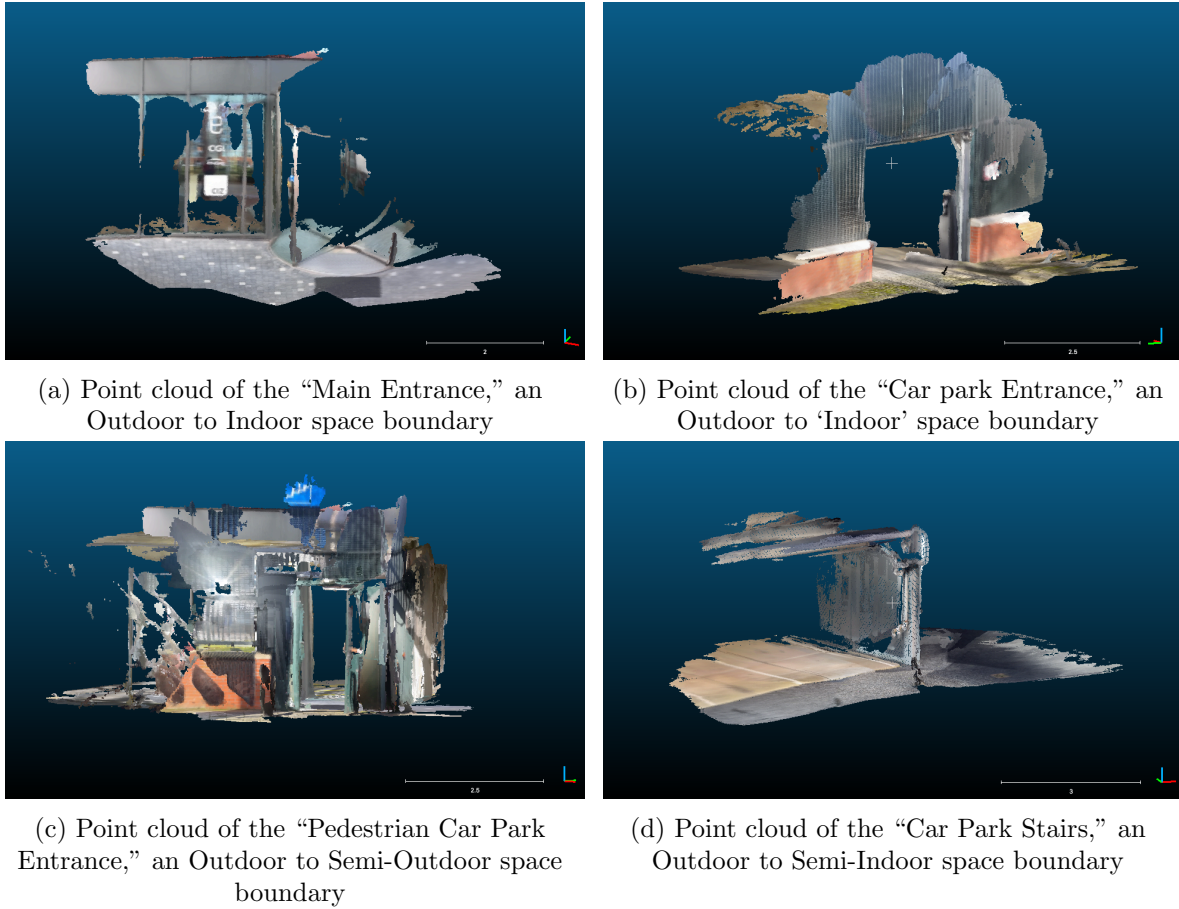


Figure 3.1: Point cloud scans taken using the *Scaniverse* application on an iPhone 12 Pro, and displayed in *CloudCompare*

3.2 Data Classification

With the scans made, these point clouds could be loaded as PLY files into *CloudCompare* (CloudCompare [2022](#)), open source software for processing point clouds, ready to be classified into different types of spaces. See [section 4.1](#) for a detailed description of how this was done, and the subsequent results.

Although (semi-)automatic classification of the point clouds would be of benefit to the project – primarily because it would reduce the manual input required, and decrease the time taken for classification – there are limited existing tools available that are capable of classifying point clouds into outdoor and indoor spaces. Most software focuses on the classification of different types of indoor spaces or objects within the space, or different types of terrain in the outdoors.

Plane segmentation can be performed fairly straightforwardly, and this information could then be used in order to determine to what kinds of space different points belong, for example:

- I **Outdoor:** Space is bounded by a ground plane, and a maximum of one wall plane from each separate building structure in the vicinity
- II **Semi-Outdoor:** Space is bounded by a ground plane, and at least two wall planes from the same building structure
- III **Indoor:** Space is bounded by a ground plane, a roof plane, and is entirely surrounded by wall planes

IV **Semi-Indoor**: Space is bounded by a ground plane, a roof plane, and is not entirely surrounded by wall planes

The difficulty with this strategy, is that unlike building models, a point cloud does not always have information about the two sides of a wall. This means that one might have a 2D-like wall structure if the scan has only been taken from one side, meaning that semantic information about the inside area can not be inferred by an automatic classification. A rough plane segmentation method native to *CloudCompare* is used to aid manual classification, and is described in [section 4.1](#).

3.3 Visibility Analysis

Visibility analysis is a popular topic in architectural and geomatics research fields, with there being a variety of existing techniques from which to choose. As described in [subsection 1.5.1](#), performing visibility analysis on a point cloud is an interesting problem. The inherent nature of points within point clouds being dimensionless, means that common ray-tracing techniques are often unsuccessful, given that a ray is cast and only returns to the sensor when it encounters an obstacle. Even with a dense point cloud, a ray can trace a path that does not encounter any points, despite there being a multitude of points that a person would immediately identify as forming a solid wall, for example. Common mitigation techniques include simply increasing the width of the ray, so that the ray has a greater chance of encountering points along its path, but this can lead to perimeteric regions being less accurately documented. Another technique involves converting the point cloud into a voxel grid, whereby every voxel is either filled (it contains a point from the point cloud) or empty (it does not contain a point). The larger the cell size chosen, the greater the chance that the ray will encounter a voxel labelled as filled.

Upon realising the limitations of performing visibility analysis directly on a point cloud, the voxel grid solution was initially considered a suitable solution. However, this does add another step of processing to the classification stage, which is undesirable, due to the potential this has for slowing down the generation of visualisations. The method works as follows: The point cloud is converted into a voxelised space, and then a voxel traversal algorithm is written in order to find out which voxels the ray passes through.

A previously discussed alternative solution from [section 2.4](#) is the research of Katz, Tal, and Basri (2007), which presented a novel technique for visibility analysis upon point clouds. *CloudCompare* implemented this research into a plugin known as qHPR, a [Hidden Point Removal \(HPR\)](#) operator. [Figure 3.2](#) shows the operator being performed on the Stanford Bunny point cloud (Stanford University Computer Graphics Laboratory 1994), previously introduced in [section 1.4](#). We can see that HPR helps with making it clear that the back of the bunny is what was initially in view. A point cloud allows us to see points that would be hidden if the point cloud were to be transformed into a mesh (or another closed, opaque surface). [Figure 3.2d](#) clearly shows the location of the cross-section of visible versus occluded points.

This operator is performed on the same four point clouds of different space boundaries shown in the previous section ([3.1](#)), and the results can be viewed in [section 4.2](#). It is important to note that these point clouds were scanned from relatively static positions, meaning that there are few points within the cloud that were truly obscured from the viewer, in some of these environments.

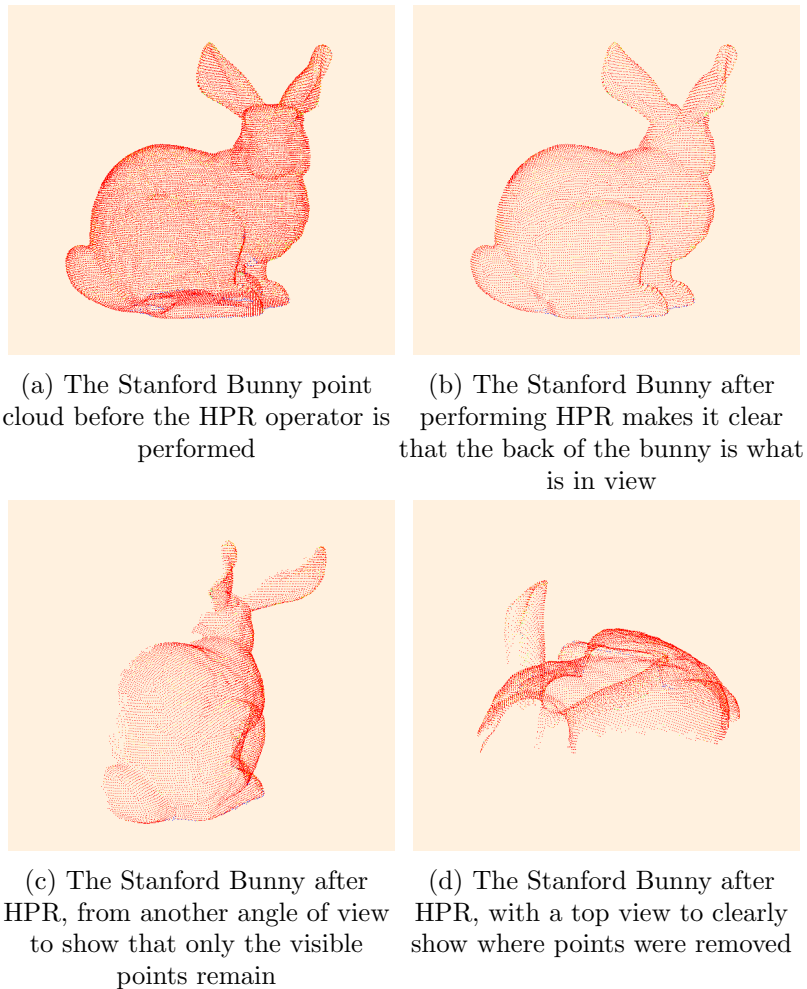


Figure 3.2: Performing *CloudCompare*'s HPR (Hidden Point Removal) operator on the Stanford Bunny (Stanford University Computer Graphics Laboratory [1994]), with an octree level of 16. Figure 3.2a shows the bunny before HPR is performed, and the other three images show the bunny after HPR, from varying angles

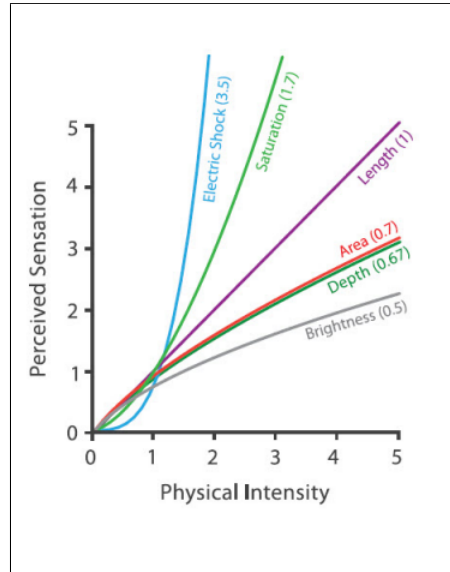


Figure 3.3: Image taken from Munzner (2015), chapter 5. Steven's Psychophysical Power Law shows how area, depth, and brightness are under-perceived, saturation is over-perceived, and length is accurately perceived

3.4 Visualisation Techniques

3.4.1 List of Requirements

Designing potential data visualisations is a stage of this project where creativity is needed if novel visualisations are to be realised. Though creativity and interdisciplinary thinking are necessary to develop new and/or unusual ideas, the design process also requires structure in order to guide decision-making. A list of requirements can detail what an ideal visualisation should contain, and specifies how this can be quantified and measured. The list acts as a guide for making design choices during the visualisation building process, and can later also be used to evaluate the extent to which final data visualisations are a success, from a range of perspectives. A list of requirements is a practical way of ensuring that concepts from an initial ideation phase can be concentrated into viable visualisations, without stifling creativity. For each requirement, a reason for why it is relevant to this project is given, as well as how its success can be quantified and measured in the context of the produced visualisations. Where applicable, supporting literature is provided to justify any reasoning. The List of Requirements can be found in [Appendix A](#).

In addition to the requirements list, data visualisation good practices will be consulted and employed. Particularly the effectiveness of different visualisation [visualisation channels](#) needs to be taken into account, as well as how users perceive these channels. Steven's Power Law (see [Equation 3.1](#)) is a useful method for determining how the physical magnitude of a stimulus is perceived in subjective terms by an individual, as follows:

$$\text{Perceived Sensation, } S = \text{Physical Intensity, } I^N \quad (3.1)$$

Where N is an exponent dependent on the modality type (e.g. brightness, area, or length). For length, the perceived magnitude is directly proportional to the actual magnitude (thus, $N = 1$), whereas brightness is under-perceived ($N \approx 0.3 - 0.5$), meaning that when using brightness to visualise a scale of values, more brightness is needed in order to bring across the change in value on the scale. Area is also under-perceived, while colour saturation is over-perceived.

First, concept visualisations will be sketched on paper to allow for a free-flowing and faster design process. Then the realisable designs will be transformed into digital images to ensure consistency be-

tween different designs, so that designs can be more easily compared and contrasted with one another. See [section 4.3](#) for images of the data visualisation concepts to be developed into visualisations.

3.4.2 Understanding the User’s Needs

Meditating on the analysis framework of (Munzner [2009](#)), described in [section 2.5](#), helped guide the initial visualisation stage, laying out what data types would be used, and how, as well as considering the end user of the visualisation, and what their needs are. The following breaks down the first three of the four layers of the framework.

Domain situation:

The target user is someone who might be familiar with using navigational applications in their everyday life, but is not an expert in the geovisualisation field. They are also unlikely to know what a point cloud is, and/or have never seen one. Their goal when interacting with the visualisation will be to better understand the environment around them, specifically, seeing what different types of spaces are in their vicinity and where, and what spaces are and are not visible to them from their present, real-life location. They will be treated as a pedestrian user of a space (able to transcend space boundaries, and viewing the world from that eye height). Potential questions that this user could have, include:

- “What type of space is situated behind that wall or that door?”
- “Where is my real-life current location situated in this [digital twin](#) environment?”
- “What features visible to me in this visualisation, are not visible to me in real life?”

Data Abstraction:

The intention of the visualisation will be to show the user the environment around them, providing them with information on the types of spaces that there are (outdoor, indoor, etc.), and whether or not these are a part of their isovist at their present location. The data input is point clouds, which will be used directly in the visualisations, and so it follows that the data type is “position” (a location in space for each point in the point cloud). There is also an “attribute” data type for each point, namely to what ‘space’ it belongs, and the Boolean “attribute” data type of whether or not the point is visible. The ‘space’ and ‘visible/occluded’ attribute types are categorical, that is, there is no order to the different attribute values – ‘outdoor’ is not deemed to possess more or less value than ‘semi-indoor’ for example.

The data set type is “geometry” (for understanding, in non-spatial visualisations, the data set type will most often be a table or a network). The data points will be plotted in space in the same formation as their real-world counterparts.

Task Abstraction:

The task abstraction layer focuses on *why* the user is looking at the visualisation: What kind of knowledge are they trying to gain? Brehmer and Munzner ([2013](#)) created a typology of visualisation tasks that was introduced in [section 2.5](#), and it includes the question of ‘why’ a task is being performed by the user of the data visualisation. When the user is “analysing” a data visualisation, they are either “consuming” the visualisation, or “producing” results using the visualisation. In this project, the user will be “consuming,” and more specifically, they will be wanting to “discover” knowledge (rather than “enjoy” it, or “present” their findings, which are the other options).

When the user moves on to the action of “searching” for information in order to consume the data and answer their question(s), there are four options for them, according to Brehmer and Munzner ([2013](#)). In this project, all four manners of searching are possible, though “locate” and “browse” are more likely options if a user familiar with the visualisation is wanting to orient themselves:

- **Lookup:** Target is known, location is known

- E.g. The user understands the space and has linked digital twin elements with real-world equivalents, but wishes to know more about an object or a space, such as the ‘space-type,’ or any additional attribute that has been associated with the points
- **Locate:** Target is known, location is unknown
 - E.g. The user knows what kind of building they are looking for within the visualisation, but does not yet know where it is. They will be cross-referencing their physical environment for clues as to where the building might be shown within the visualisation
- **Browse:** Target is unknown, location is known
 - E.g. The user knows where the building they are looking for should be within the visualisation, but does not know what the building looks like. They will be matching up real-world features with the visualisation, such as the shape of a façade, in order to identify the desired building
- **Explore:** Target is unknown, location is unknown
 - E.g. The user does not know what their own location is within the visualisation, nor what the visualisation is showing them, and needs some time to adjust and understand what the visualisation channels are representing. This is where many users are likely to begin their interaction with the visualisation

Now that the user has found their target and its location, they will want to query the data. If the user performed a “lookup” or “locate” action, then they will likely query attributes of the target, such as the ‘space-type,’ whether it is visible or occluded, or any additional attributes that have been assigned to the points in the point cloud. If the user performed a “browse” or “explore” action, then

The more objects of interest that the user is interested in, the quicker a user will progress to the “compare” and “summarise” actions. A user who is interested in the environment as a whole, as opposed to one particular building, will be wanting to compare the different spaces, e.g. *will this door or that door lead me to the semi-outdoor courtyard?*, and be forming a summary of the entire space, e.g. what percentage of the space around them, is outdoor or indoor?

3.5 Combining Space Classification and Visibility Analysis

The crucial moment of this project is the tying together of two separate avenues of research: combining the space classification (outdoor, semi-outdoor, semi-indoor, and indoor) with the visibility analysis (visible or occluded) that was performed on the point clouds.

Multiple types of software were considered for this task, but few point cloud processing software (both open-source and commercial) have enough customisation options for everything that is required here. Previously mentioned in [subsection 1.5.2](#), [Rhinceros](#) (Robert McNeel & Associates [2022](#)) together with [Grasshopper](#) (Rutten [2022](#)) have the capability of producing isovists. However, when trialling the potential for doing so with point clouds, there was the shortcoming that *Grasshopper* is needed to generate isovists, but it does not recognise point clouds as a data type (though *Rhino3D* does). In order to use the isovist functionality of *Grasshopper*, each point within a cloud would need to be stored separately as an instance of a class, greatly increasing the memory and processing power required of a computer for larger point clouds. Some plugins have attempted to rectify this shortcoming by providing point cloud functionality within *Grasshopper*, but those discovered were either obsolete, or incompatible with the operating system in use.

Ultimately, *CloudCompare* was chosen to render the final visualisations. A description of the process, as well as the results, can be viewed in [section 4.4](#).

Chapter 4

Results

4.1 Space-Type Classification

The mobile phone scans that had been gathered from four different locations (see [section 3.2](#)) were first loaded into CloudCompare ([2022](#)) and cleaned. This involved employing the **Statistical Outlier Removal Filter** (under **Tools** → **Clean**) to remove any points too far away from neighbours. The intention of this step was to ensure that the data being worked upon was as accurate a representation of the real world as possible, by excluding points that represented unwanted debris and particles picked up by a “too-sensitive” LiDAR scan, or points with insufficient numbers of neighbours at the outer edges of the scanned area. The *CloudCompare* default values were used (six points used for mean distance estimation, and standard deviation multiplier threshold of one). The filter rejects points further away than the average distance that all points are to their (in this case, six) closest neighbours, plus a multiple of the standard deviation (in this case, one).

The new point cloud tended to contain over 95% of the original points, and could then be subsampled using the **Subsample** tool in order to remove superfluous points. The main purpose of this step was to aid in reducing future visualisation processing times. There are multiple subsampling techniques available, but for this project, the **Space** method was selected, which results in a point cloud where every point is at least a user-specified distance away from every other point. The reason for choosing this method, rather than the **Random** subsampling method for example, is that LiDAR scans will tend to have a higher density of points closer to the scanner location. Usually, this results in many points representing the ground around the scanner, which does not add much in the way of valuable information (far fewer points would still easily form a distinct plane, both in the mind’s eye, and when using plane detection software). The **Space** method results in a more even distribution of points throughout the entire space. The minimum space value is dependent on the point cloud, and chosen heuristically, but in the case of all of the mobile scans, was 2 cm. This resulted in approximately 30% of the cleaned points being retained.

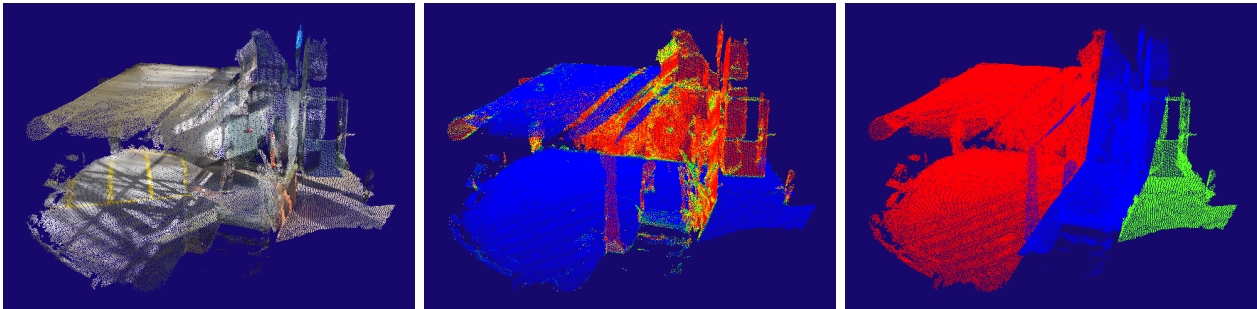
Then the clouds were manually classified into different ‘space-types,’ by selecting different planes using a polygon selection tool within the **Segment** function. Given that this method of classification is prone to human error, it meant that some points were likely either excluded or incorrectly classified on the boundaries, but that the vast majority of points were correctly classified into one of the four space-types, which was deemed good enough for this proof of concept project.

One quick and simple way in which a rough plane segmentation can be generated in order to guide the manual classification process, is via the geometric feature computation tool (**Tools** → **Other** → **Compute Geometric Features**). This clearly distinguishes most ceilings and floors, from walls. Now using the **segment** tool, the point cloud is manually split into individual parts (sub-clouds). Then in order to add a space attribute for each point, a constant scalar field was added to every sub-cloud

with the same name, and assigned a unique integer (`Edit` → `Scalar Fields` → `Add SF`):

- Outdoor: 0
- Semi-Outdoor: 1
- Semi-Indoor: 2
- Indoor: 3

These sub-clouds were then merged into a new point cloud, and the “SpaceType” attribute was retained, with each point within the cloud having an integer assigned between 0 and 3.



(a) The cleaned point cloud of the “Pedestrian Car Park Entrance” location, with points displaying their RGB values

(b) Performing a rough plane segmentation on the points using *CloudCompare*’s `Compute Geometric Features` tool

(c) Manual classification of points into space-types: outdoor (green), semi-outdoor (blue), and indoor (red)

Figure 4.1: Point cloud space-type classification process in *CloudCompare*

See [Figure 4.2](#) for the cleaned point clouds of each of the four point clouds created during the data collection stage, as well as the point clouds being segmented into different space-types.

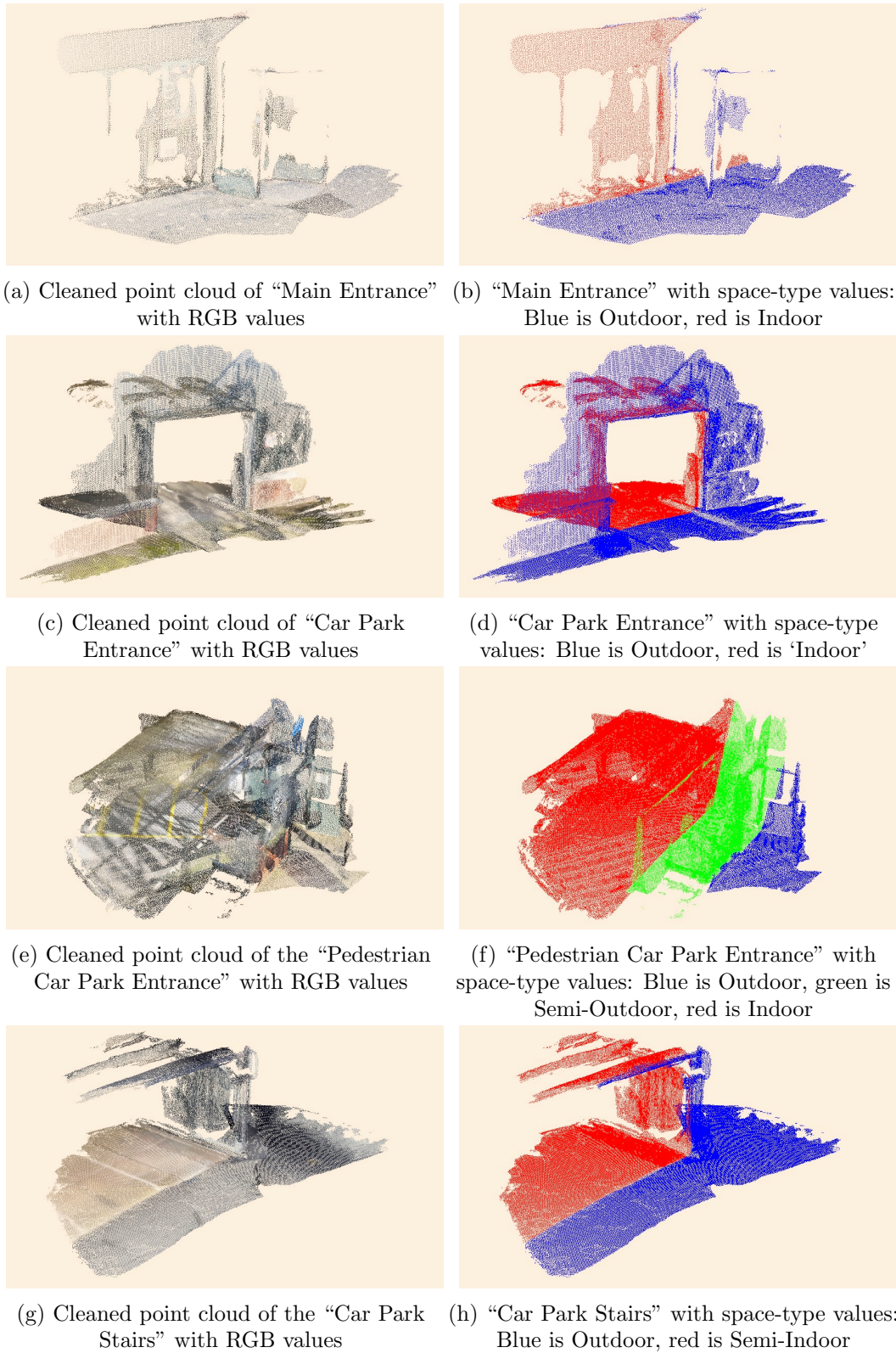


Figure 4.2: Images in figures [4.2a](#), [4.2c](#), [4.2e](#), and [4.2g](#), show four different space boundary areas as cleaned point clouds, with each point displayed in its RGB value. Figures [4.2b](#), [4.2d](#), [4.2f](#), and [4.2h](#) show four corresponding points clouds where the points are assigned a colour based on the ‘space-type’ they were assigned during manual classification

4.2 Visibility Analysis: Hidden Point Removal

CloudCompare's (HPR) operator (Plugins → Hidden Point Removal) was performed on the four scanned point clouds already seen in section 4.1. Although the clouds were fairly small (only a few hundred thousand points, rather than millions), an octree level of 16 was selected. The highest possible octree available was 21, but this resulted in too many false negatives, meaning that visible points near the visible-occluded boundaries were removed. This can also happen when the radius of the imaginary sphere around which the point cloud is flipped, is too small (see Figure 2.4 for a visual explanation of the spherical flipping). The ‘halos’ of empty space around the car park entrance’s clusters of red outdoor points, seen in Figure 4.4b, demonstrate the surplus of removed points produced by the algorithm near the lines of sight boundaries.

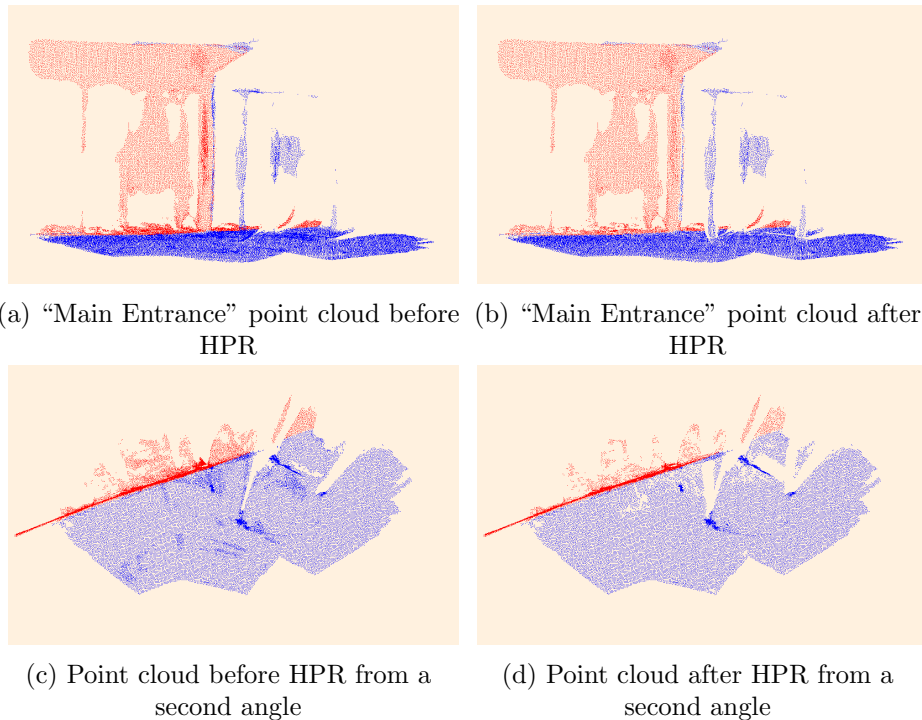


Figure 4.3: Before and after HPR tool from *CloudCompare* is used upon the “Main Entrance” point cloud. Figures 4.3c and 4.3d help show which occluded points were removed

Both Figure 4.4 and Figure 4.5, showing the “Car Park Entrance” and “Pedestrian Entrance” point clouds respectively, are clear examples of HPR’s capabilities. In the images showing the point clouds after HPR from an angle, the ground areas have missing points, where walls have obstructed the user’s view of these portions of the ground.

Figure 4.6d shows how a small cluster of points that likely represents a negligible object of no encumbrance to a pedestrian, can demand significant attention from a viewer who is looking at a post-HPR image. The straight belt devoid of points, cutting through the point cloud, jumps out at the viewer due to the contrast between the points and the background’s colour. This can have positive benefits, depending on the situation; a user with limited mobility might realise there is a stump or something which would provide enough of a barrier to a wheelchair, and they could opt to avoid this area, and take another route instead. A meshing of these surfaces would have most likely eliminated the cluster, and the user would never have seen this feature.

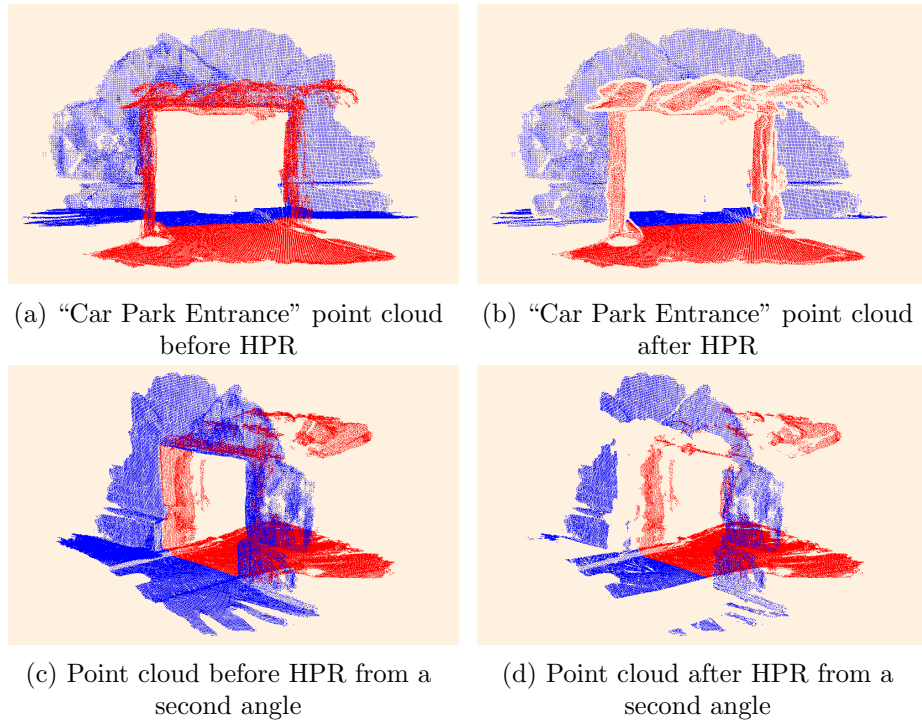


Figure 4.4: Before and after **HPR** tool from *CloudCompare* is used upon the “Car Park Entrance” point cloud. Figures **4.4c** and **4.4d** help show which occluded points were removed

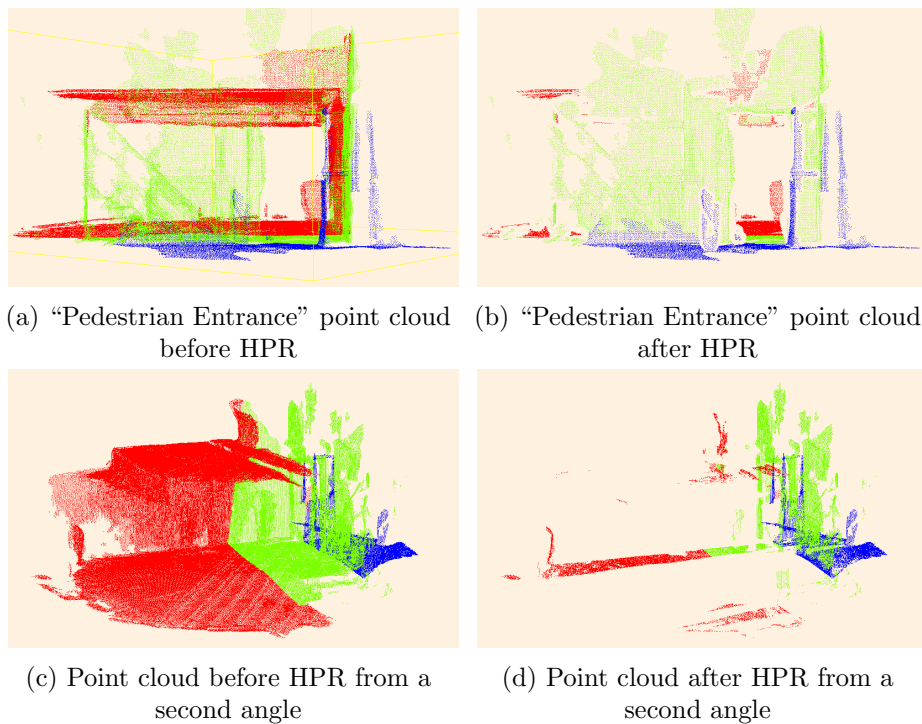


Figure 4.5: Before and after **HPR** tool from *CloudCompare* is used upon the “Pedestrian Entrance” point cloud. Figures **4.5c** and **4.5d** help show which occluded points were removed

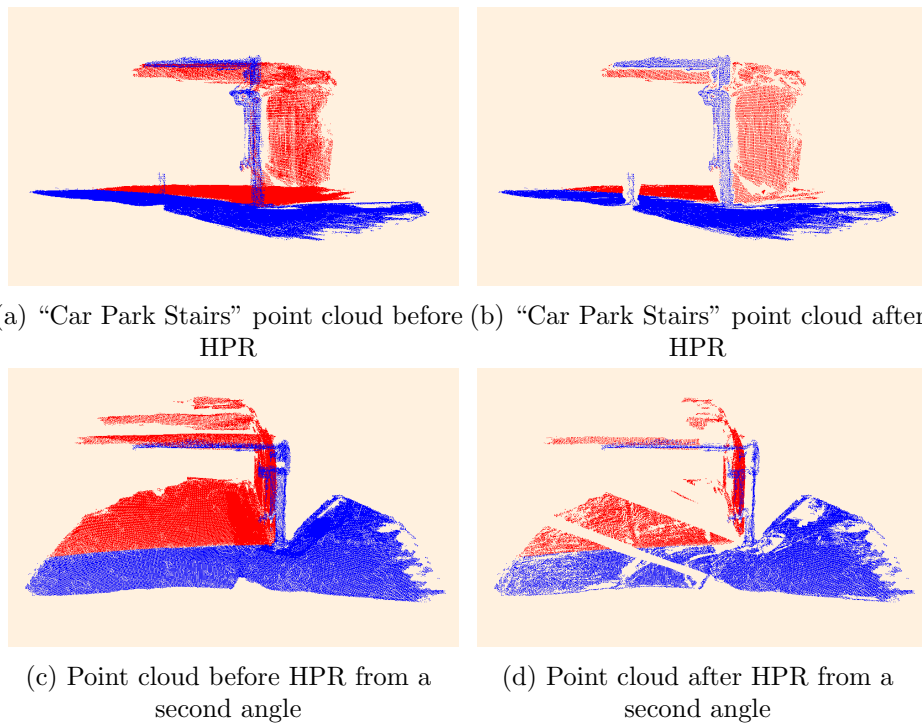


Figure 4.6: Before and after [HPR](#) tool from *CloudCompare* is used upon the “Car Park Stairs” point cloud. Figures [4.6c](#) and [4.6d](#) help show which occluded points were removed

4.3 Visualisation Designs

The most viable designs that had been roughly sketched and annotated on paper, were transferred into digital images, in order to more clearly communicate the designs' features, and compare and contrast the designs with one another.

There were four main types of visualisation that were taken on to the development phase, which can be differentiated as follows:

I **Floor Plan:** A traditional floor plan was the main foundation upon which the geovisualisation was built

II **Doorways:** A floor plan where the visualisation of the doorways is a key feature

III **Walls:** The walls dividing different spaces were the prominent features of these visualisations

IV **Isovist:** The concept of an isovist was taken more literally, with visibility polygons being a prominent design feature

Most of these different design types had at least two different iterations, usually involving a black and white, "greyscale," version. The design elements incorporated into the visualisations included use of colour, texture, and symbology. The primary focuses of these designs are on depicting where the space boundaries are, what different types of space there are (indoor, semi-indoor, and semi-outdoor), and which spaces are visible and which are occluded.

Using a mock-up of a space that contained a visible and occluded version of each of the three major space boundaries that this project deals with (see [Figure 4.7](#)), the different designs were all applied to this same space with the same angle.

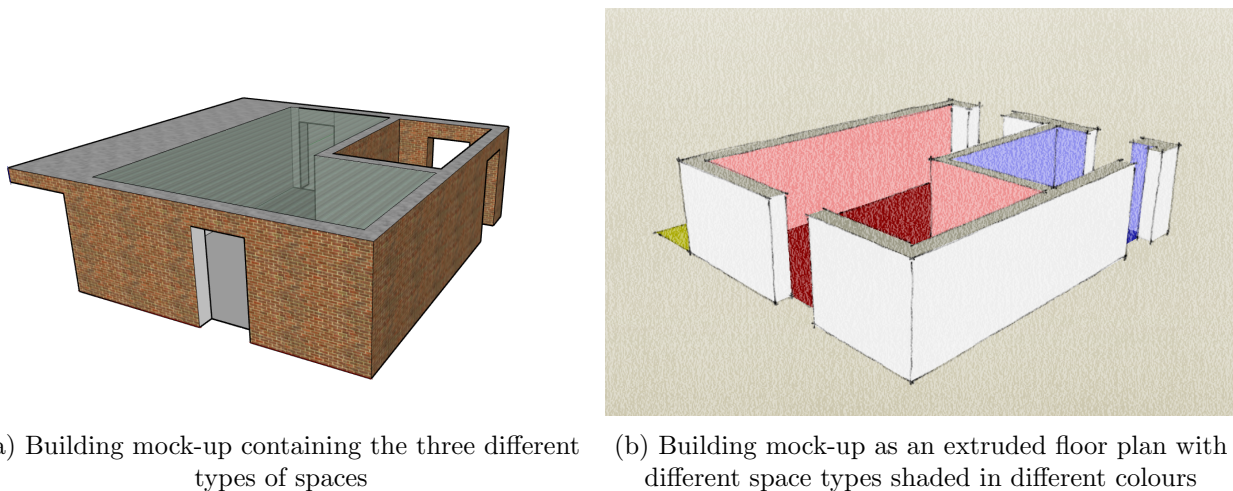


Figure 4.7: The [CAD](#) building used to demonstrate the design concepts of the various visualisations

4.3.1 The "Floor Plan" Designs

Four different iterations were made of the visualisation concept where the main design feature was using buildings' floor plans.

"Floor Plan Thresholds"

The first floor plan design iteration contains the following elements, as seen in [Figure 4.8](#):

- The ground surfaces of the different spaces are all filled in different hues (indoor: red, semi-indoor: yellow, semi-outdoor: blue), and also possess different textures (indoor: smooth, semi-indoor:

noisy, semi-outdoor: streaked), to ensure that distinction between different space types is not dependent on a single channel if a user is visually impaired or a screen displays the colours inaccurately

- The areas of the non-outdoor spaces that are visible from the user’s viewpoint, are shaded in a more saturated version of that space’s characteristic colour (indoor: red, semi-indoor: yellow, semi-outdoor: blue)
- The doorways or openings along the space boundaries are indicated by lines that cover the space boundary threshold (indoor: filled, semi-indoor: dotted, semi-outdoor: dashed). The lines are also coloured according to the characteristic colour of the space being entered
- Thresholds that are not visible to the viewer, are depicted with a less thick line
- The outdoor space, walls, doors, etc. are all omitted from the design

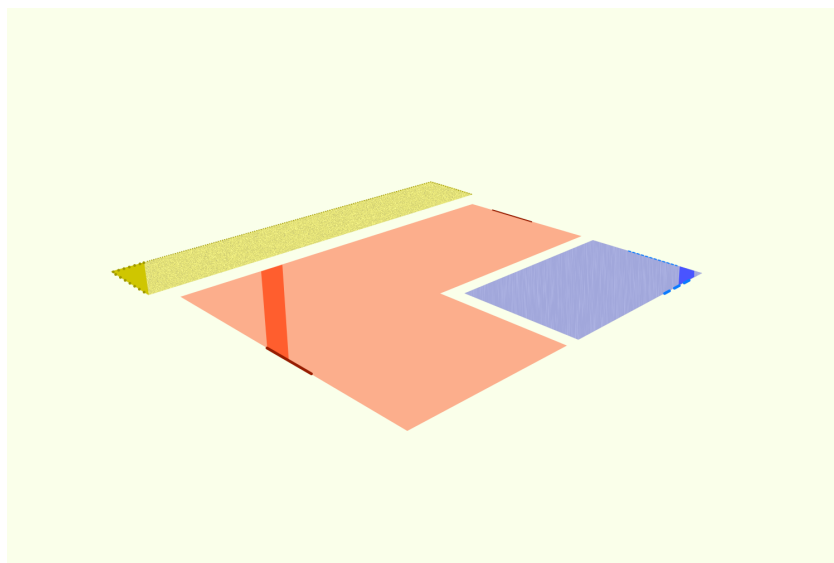


Figure 4.8: The “Floor Plan Thresholds” design where lines indicate locations of space boundary thresholds

The concept behind this design is that the user need not see the extent of the isovist in the outdoor space, as it will tend to occupy the majority of the area surrounding them, and distract from the primary information that needs to be conveyed (which has already been listed above, but to reiterate: space boundaries, space types, and visible versus occluded spaces). The isovist spaces that belong to space types other than “outdoor” are highlighted so that the user can quickly cross-reference their view in real space with the digital twin, and know to which space types those areas belong. As the user changes position, so should the areas that the polygons inhabit, so that with some initial experimentation it then ought to be fairly self-explanatory to the user that the lighter hued spaces are occluded spaces. The lines depicting the areas along space boundaries where the user can cross, are not as easily discernible, especially the thinner lines for the occluded entranceways. In this design, the isovist and non-isovist spaces are visualised together, and directly interact with one another as the user changes position.

“Floor Plan Enclosures”

The second floor plan design iteration contains the following elements, as seen in [Figure 4.9](#):

- The ground surfaces of the different spaces are all visualised in the same way as in the first visualisation (design [4.3.1](#)), including that the isovist spaces are shaded in an intenser hue

- The boundaries of the spaces are enclosed by a line, with any doorways or openings along the space boundaries being the only areas where the lines do not run (indoor: red and filled, semi-indoor: yellow and dotted, semi-outdoor: blue and dashed)
- There is no distinction made between boundaries that are and are not a part of the isovist

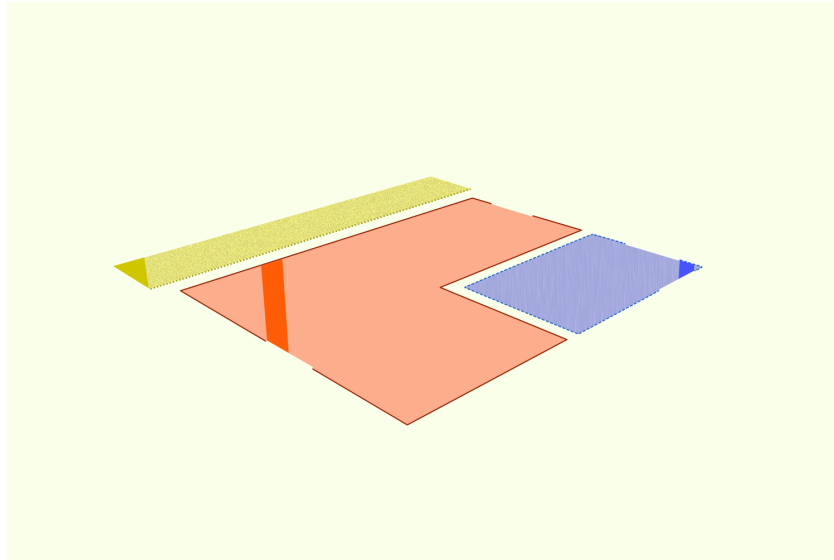


Figure 4.9: The “Floor Plan Enclosures” design where space boundaries that cannot be traversed, are bordered

This design is an attempt at combatting the weakness of the first design where the threshold lines do not draw enough attention to the places where the user can cross into another space. By enclosing the different spaces with a border, it should be intuitively clear that these are not areas that can be bypassed, and a lack of a boundary line is likely to be more noticeable, as well as intuitively suggest that there is an entranceway at that location.

“Floorplan Enclosures (greyscale)”

The third floor plan design iteration is simply a greyscale version of the second design, and can be seen in [Figure 4.10](#)

The idea behind a greyscale version is to remove visual clutter, and help the user focus more on the isovist spaces. There is now an even greater distinction between isovist and non-isovist space, and there are less keys for the user to have to remember. The downside is that the only visualisation channel being used to differentiate the different space types from one another, is the texture used.

“Floor Plan Markers”

The fourth floor plan design iteration contains the following elements, as seen in [Figure 4.11](#)

- The ground surfaces of the different spaces are all visualised in the same way as in the greyscale visualisation ([4.3.1](#)), including that the isovist spaces are shaded in saturated hues (indoor: red, semi-indoor: yellow, semi-outdoor: blue)
- The locations of space boundaries are indicated by circular markers. The circles have two colours to indicate which two types of space form the boundary. The bright green hue is indicative of outdoor space making up a part of the boundary space. The circles are also bordered differently according to the type of space that the user can enter at that location (indoor: filled, semi-indoor: dotted, semi-outdoor: dashed). The diameter of the circles indicates the breadth of the entranceway

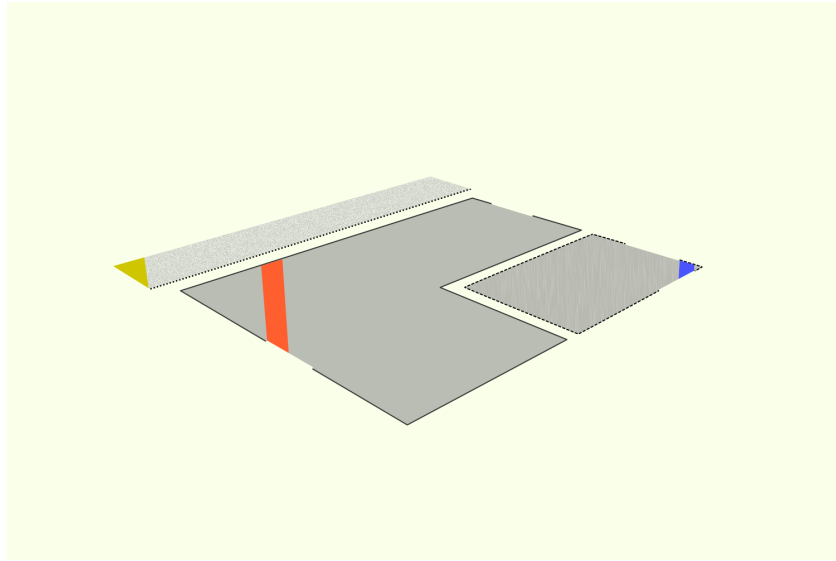


Figure 4.10: The same “Floor Plan Enclosures” design as [section 4.3.1](#), incorporating greyscale

- Boundaries that are not a part of the user’s isovist, are given less saturated circles that also have lighter, thinner borders

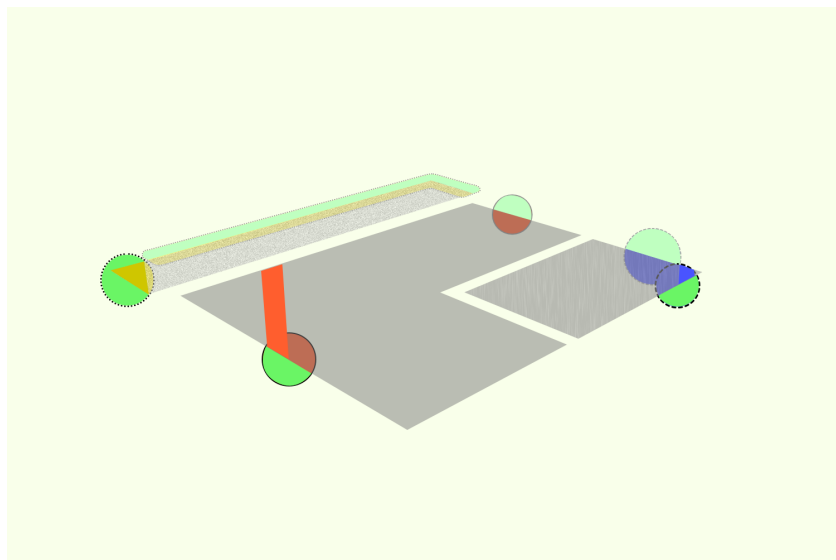


Figure 4.11: The “Floor Plan Markers” design where shapes (here, circles) indicate the locations of space boundary thresholds

This design really draws attention to the locations of boundaries, given that the spaces are all in greyscale, which allows the bright green of the circles to ‘pop’.

4.3.2 The “Entranceway” Designs

Two different iterations were made of the visualisation concept where the main design feature was visualising buildings’ entranceways (most often, doorways).

“Doors”

The first entranceway design iteration contains the following elements, as seen in [Figure 4.12](#)

- The ground surfaces of the different spaces are filled in the same way as in visualisations [4.3.1](#) and [4.3.1](#), with different hues (indoor: red, semi-indoor: yellow, semi-outdoor: blue), and different textures (indoor: smooth, semi-indoor: noisy, semi-outdoor: streaked), to ensure that distinction between different space types is not dependent on a single channel if a user is visually impaired or a screen displays the colours inaccurately
- The doorways or openings along the space boundaries are indicated by outlines that follow the shape of that particular entranceway. These lines are in a dark red hue, and filled with a semi-transparent grey tone
- The non-outdoor spaces that are visible from the user’s viewpoint, are not explicitly visualised, but the parts of entranceways that are a part of the user’s isovist, are shaded in a highly-saturated green hue
- Thresholds that are not visible to the viewer, are depicted with a less thick line
- Explicit visualisation of the outdoor space is omitted from the design

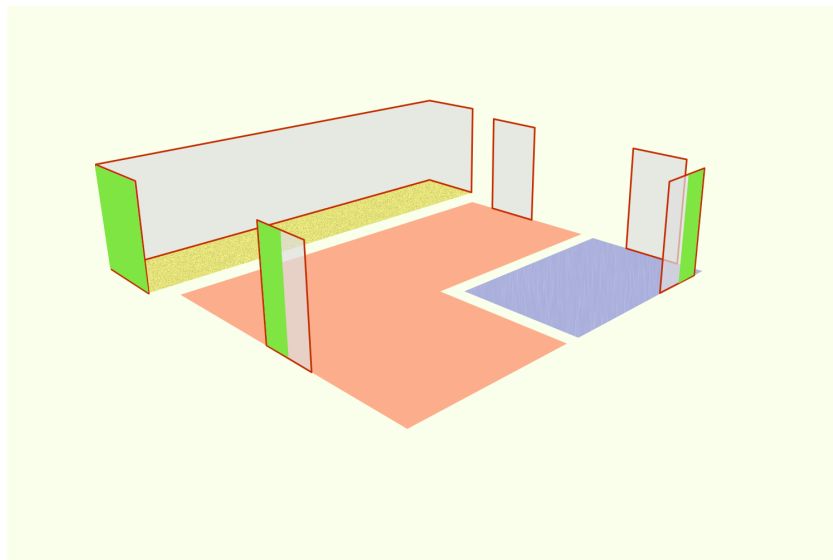


Figure 4.12: The “Doors” design where the shapes of the doorways indicate the locations of space boundary thresholds

The advantages of this design are that all entranceways are immediately obvious to the user (similar to visualisation [4.3.1](#)), and that there is a clear distinction between visible and occluded entranceways. It also has more of a “3D element” than all of the visualisations in [subsection 4.3.1](#), though it still omits the walls to avoid what might be considered unnecessary clutter. The visualisation of the ground to convey occluded spaces might be the same as in designs [4.3.1](#) and [4.3.1](#), but because there is no indication as to what ground *is* visible, it is highly likely that the user would not understand that these lightly shaded areas represent occluded spaces. Thus, visualising walls could have an added benefit.

“Doors (greyscale)”

The second entranceway design iteration is simply a greyscale version of the first design, [4.3.2](#), and can be seen in [Figure 4.13](#)

Again, the intention of a greyscale version is to remove visual clutter, and help the user focus more on identifying which doorways are and are not visible to them.

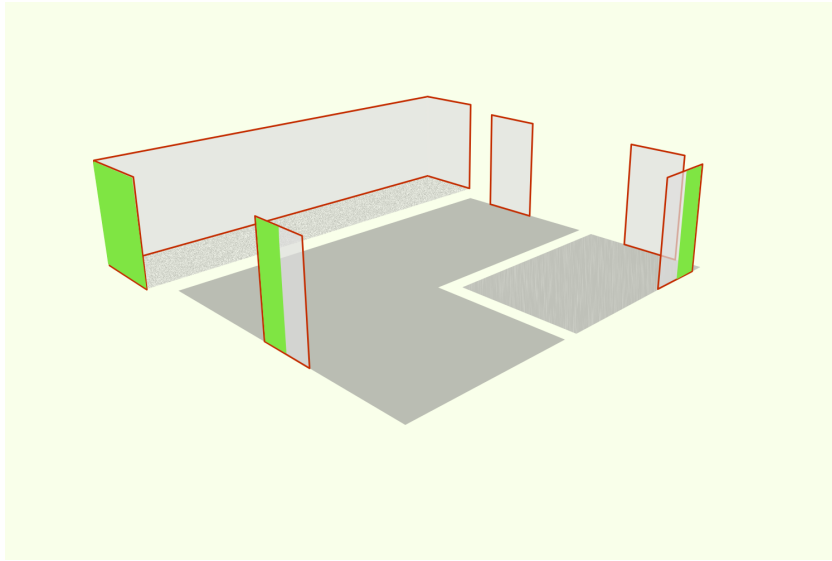


Figure 4.13: The “Doors (greyscale)” design where the shapes of the doorways indicate the locations of space boundary thresholds

4.3.3 The “Walled” Designs

A design that erred more on the side of the traditional concept of a 3D isovist, showcased the walls that straddled space boundaries.

“Walls”

The first walled design iteration contains the following elements, as seen in [Figure 4.14](#):

- Each exterior wall is shaded in a colour to indicate the type of space contained behind the wall (indoor: red, semi-indoor: yellow, semi-outdoor: blue). These surfaces are also given a texture to aid with distinguishing them from one another (indoor: smooth, semi-indoor: noisy, semi-outdoor: streaked). For the semi-indoor areas, the bounding top surface is visualised, as walls are not always present in semi-indoor spaces
- The isovist areas within these spaces are shaded in a more saturated hue of that space’s characteristic colour, but non-isovist ground is completely excluded
- The doorways or openings along the space boundaries are indicated by lines that cover the space boundary threshold (indoor: red and filled, semi-indoor: yellow and dotted, semi-outdoor: blue and dashed)
- Thresholds that are not visible to the viewer, are depicted with a less thick line

Though this design does try to mitigate the common 3D model weakness of information overload by removing ground surfaces, occluded walls, and the volumetric aspects of walls, it is still a fairly literal representation of the space. There is use of symbology for indicating thresholds, but these might be overlooked due to the significant amount of space being taken up by the walls.

4.3.4 The “Isovist” Designs

In the previous designs, the concept of the isovist has been fairly loosely used, with a distinction made between how the visible and occluded spaces are visualised. A traditional isovist visualises all of the visible space from a given point. These designs introduce a classic visualisation method for 2D isovists from a 3D viewpoint, with some design tweaks.

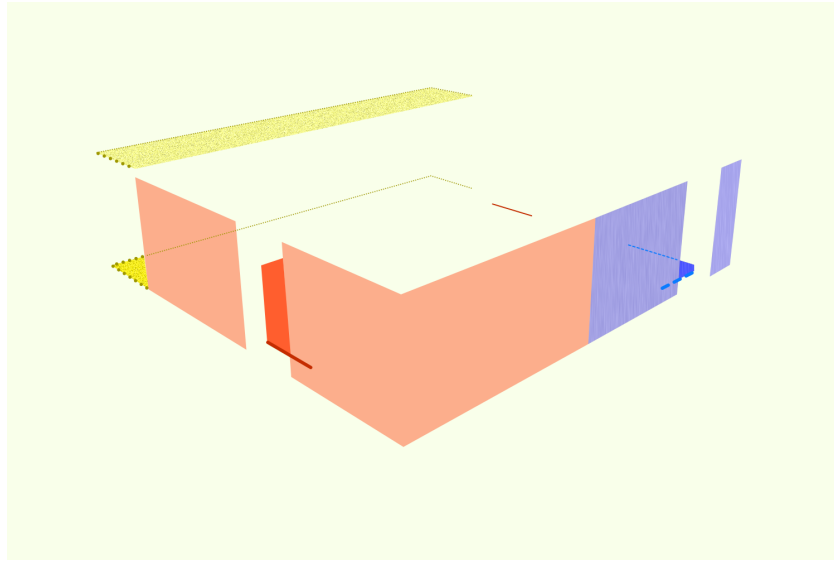


Figure 4.14: The “Walls” design where the visualisations of walls convey the type of space that the walls are occluding

“Isovist Paths”

The first isovist design iteration traces paths from the user’s position (which is now also visualised), to the different visible entrances. These paths are a part of the user’s isovist, and the rest of the outdoor area within the isovist is ignored. This design contains the following design details, as can be seen in [Figure 4.15](#):

- The non-outdoor spaces are visualised in much the same way as design [4.3.1](#).
 - The ground surfaces of the different spaces are all filled in different hues (indoor: red, semi-indoor: yellow, semi-outdoor: blue), and also possess different textures (indoor: smooth, semi-indoor: noisy, semi-outdoor: streaked), to ensure that distinction between different space types is not dependent on a single channel if a user is visually impaired or a screen displays the colours inaccurately
 - The areas of the non-outdoor spaces that are visible from the user’s viewpoint, are shaded in a more saturated version of that space’s characteristic colour (indoor: red, semi-indoor: yellow, semi-outdoor: blue)
 - The doorways or openings along the space boundaries are indicated by lines that cover the space boundary threshold (indoor: filled, semi-indoor: dotted, semi-outdoor: dashed). The lines are also coloured according to the characteristic colour of the space being entered
 - Thresholds that are not visible to the viewer, are depicted with a less thick line
- Light grey, filled-in polygons are formed between the user’s position and the various visible entrances. The rest of the outdoor isovist is not visualised

The paths should help the user quickly identify their position in relation to the entrances, and as they move, these paths will change shape as a result, as well as disappear if the entrances become occluded.

One drawback to this design is that the visualisation does not account for ‘invisible’ obstacles when drawing the pathways. Given that the entrance is a part of the user’s isovist, there is no physical barrier between the two, as far as the visualisation is concerned. But there might be a road without a crossing between the user and an entrance, and this could pose a safety risk if the visualisation encourages the user to take the most direct path.

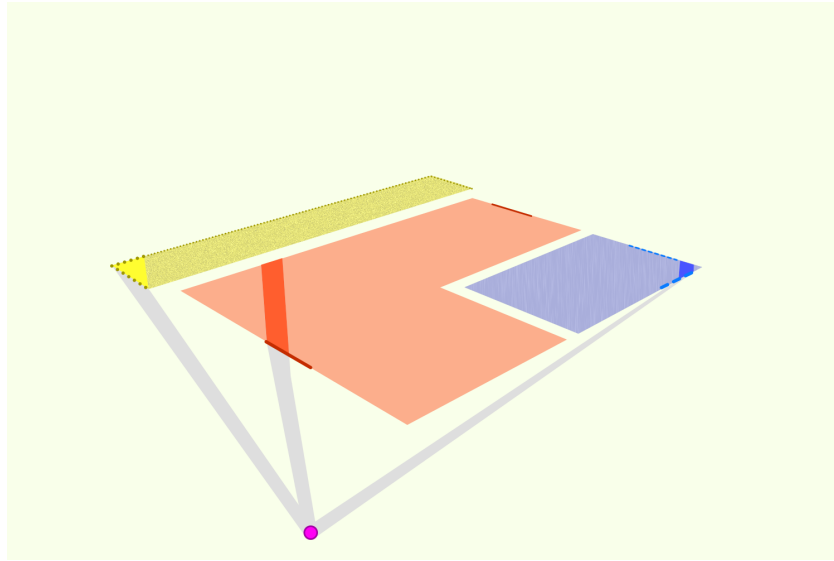


Figure 4.15: The “Isovist Paths” design, where the direct pathways towards entrances are visualised

“Isovist Outdoor Area”

The second isovist design iteration still shows the paths from the user’s position to the different visible entrances, but these paths are all coloured in the destination space’s characteristic colour. The rest of the outdoor area that makes up the user’s isovist is now also visualised in a light grey colour that fades away at the periphery of the user’s field of vision. In this sense it is a more traditional isovist representation than the previous design. A 360° isovist is not given, instead one with an angle of 120° is shown, corresponding to the extent of a human’s peripheral vision. This will require not only knowing a user’s position, but also their orientation, which is relatively simple to gauge with mobile phones’ position sensors. By centring the 120° isovist in front of the user, it also becomes easier for the user to quickly connect real world objects with digital twins (they need not look behind them in order to orient themselves). This design contains the following design details, as can be seen in [Figure 4.15](#):

- The non-outdoor spaces are visualised exactly as in design [4.3.4](#).
- Light grey, filled-in polygons are formed between the user’s position and the various visible entrances. The rest of the outdoor isovist is not visualised

An element of this design that could be confusing, is how the entirety of the path polygons are filled in the colour of the destination space’s characteristic colour. In all other designs, these colours are only used within the spaces to which they correspond, whereas here the colours cover outdoor space. For this reason, multiple iterations of this design have been made with different shading styles (see designs [4.17](#) and [4.18](#)).

Additional Isovist Designs

Two more subtle design iterations include a greyscale version of design [4.15](#), and a version of the “Isovist Outdoor Area” concept (design [4.16](#)), where the paths are only outlined in the colour of the space type to which they lead, with no fill colour. The green background can be seen underneath these polygons, indicating that it is still a part of the outdoor space.

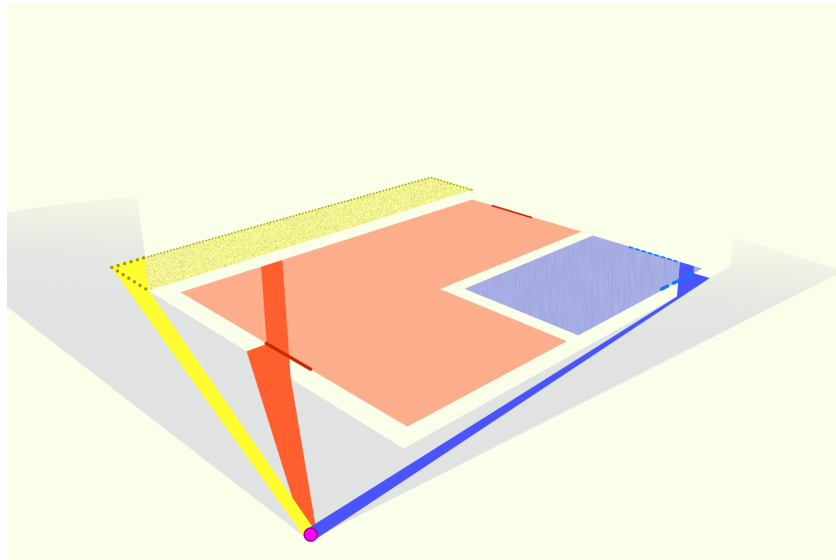


Figure 4.16: The “Isovist Outdoor Area” design, where the entire isovist is visualised, along with paths towards entrances

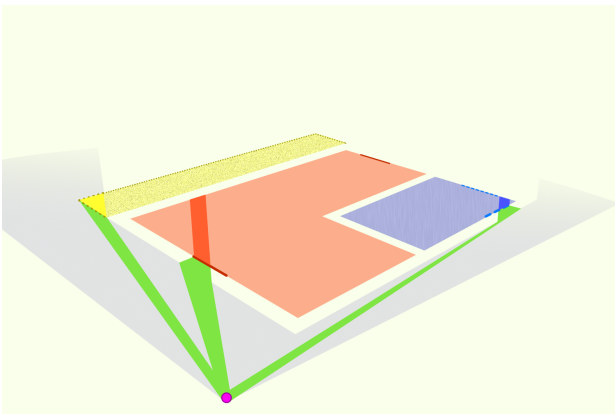


Figure 4.17: The “Isovist Outdoor Area with Green Paths” design

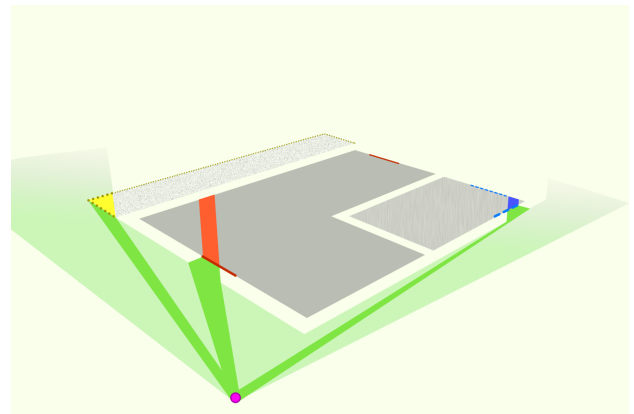


Figure 4.18: The “Isovist Green Outdoor Area” design with non-outdoor spaces in greyscale

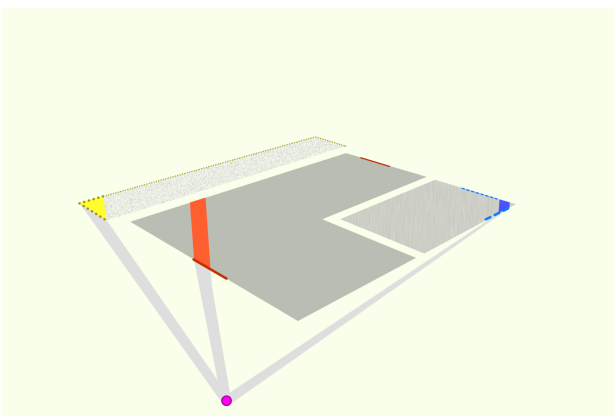


Figure 4.19: The “Isovist Paths (greyscale)” design

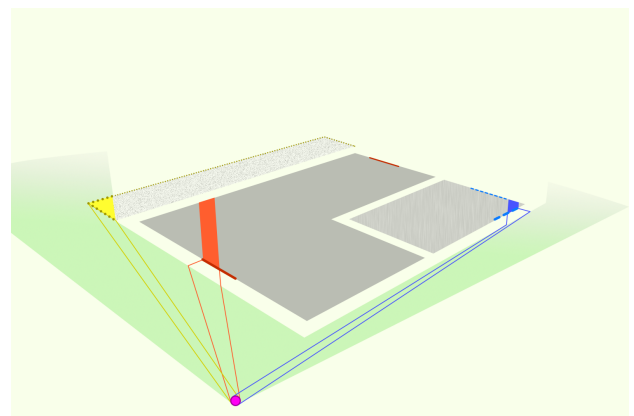


Figure 4.20: The “Isovist Paths Outline” design with green outdoor area

4.4 Data Visualisations

Combining the space classification and the visibility analysis performed on the point clouds to create the designs from the previous section (section 4.3), resulted in the data visualisations shown in this section.

These visualisations were all completed within *CloudCompare*, and predominantly made use of the point clouds directly to communicate information about the space. The input data for each viewpoint were a point cloud that classified all the points into different space types ((semi-)outdoor and (semi-)indoor) in the form of a scalar field, and a point cloud that showed which points are visible from the relevant vantage point. The second point cloud also contained the space type as an attribute. These two point clouds were then segmented into the relevant constituent parts in order to aid the application of different visualisation techniques upon subdivisions of the space. For example, both point clouds were separated into points that represented the ground, and points that represented walls, ceilings, or other architectural features. These point clouds were subsequently divided separate clouds for each of the space types.

The geometric primitives of points, lines, and surfaces (areas in 3D space that have no thickness), were all used as visualisation marks within various visualisations; all visualisations use points as a primitive, Figure 4.21b shows a visualisation employing lines, and figures 4.21e and 4.21h demonstrate use of surfaces as a primitive. Contour plotting and meshing of points was performed in order to create the relevant polylines and polygons.

The visualisation channels that were used upon the geometric primitives included:

1. Point hue, saturation, and luminance
2. Point size
3. Polyline hue, saturation, and luminance
4. Polyline thickness
5. Opacity
6. Lighting and shading

The order of effectiveness of different visualisation channels as described in section 2.5, guided the process of visualisation. In order of effectiveness, categorical attributes are best encoded by spatial region, hue, motion, and shape. Given that this is a geovisualisation where the spatial attributes of all data points are explicitly plotted in a 3D space, the spatial region encoding cannot be used. It was thus most appropriate to use the hue channel to encode the categorical attribute

Each visualisation was rendered from two viewpoints in order to better compare and contrast the way in which the visualisation changes with the user's viewpoint. Figure 4.21 shows all the visualisations of the car park point cloud from a viewpoint at eye height (160 cm) a few metres away from the car park entrance. Due to the ground lacking much visible depth in the first viewpoint, the second viewpoint (Figure 4.22) was raised to 2 m above the ground, in order to better discern the differences between the individual visualisations. This is also more in line with the initially proposed research, as it is intended that the visualisations produced, provide a third-person view, rather than a first-person view. The rest of the viewpoints for the other point clouds are all from a third-person view.

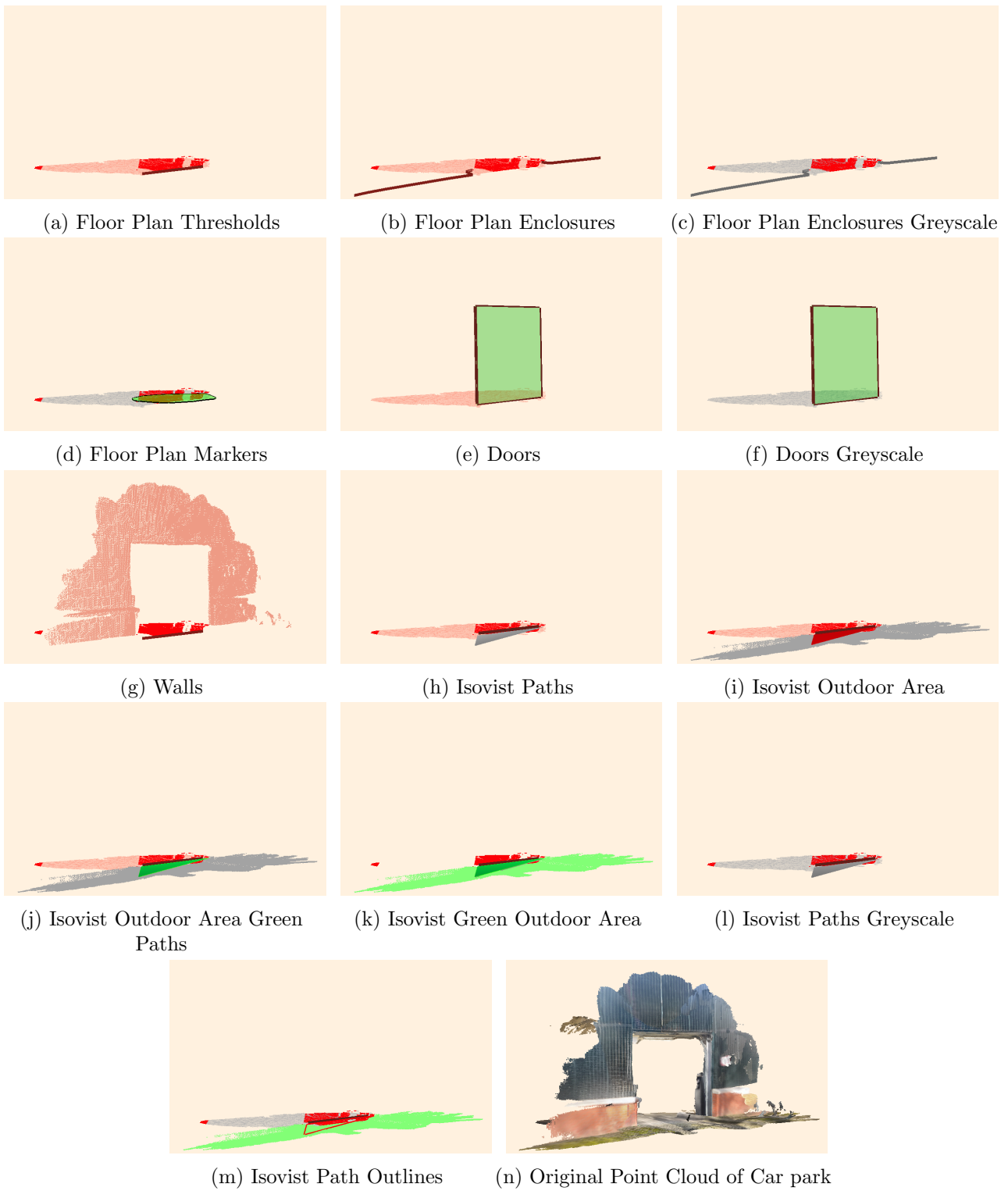


Figure 4.21: All data visualisations for the car park entrance data set from viewpoint 1

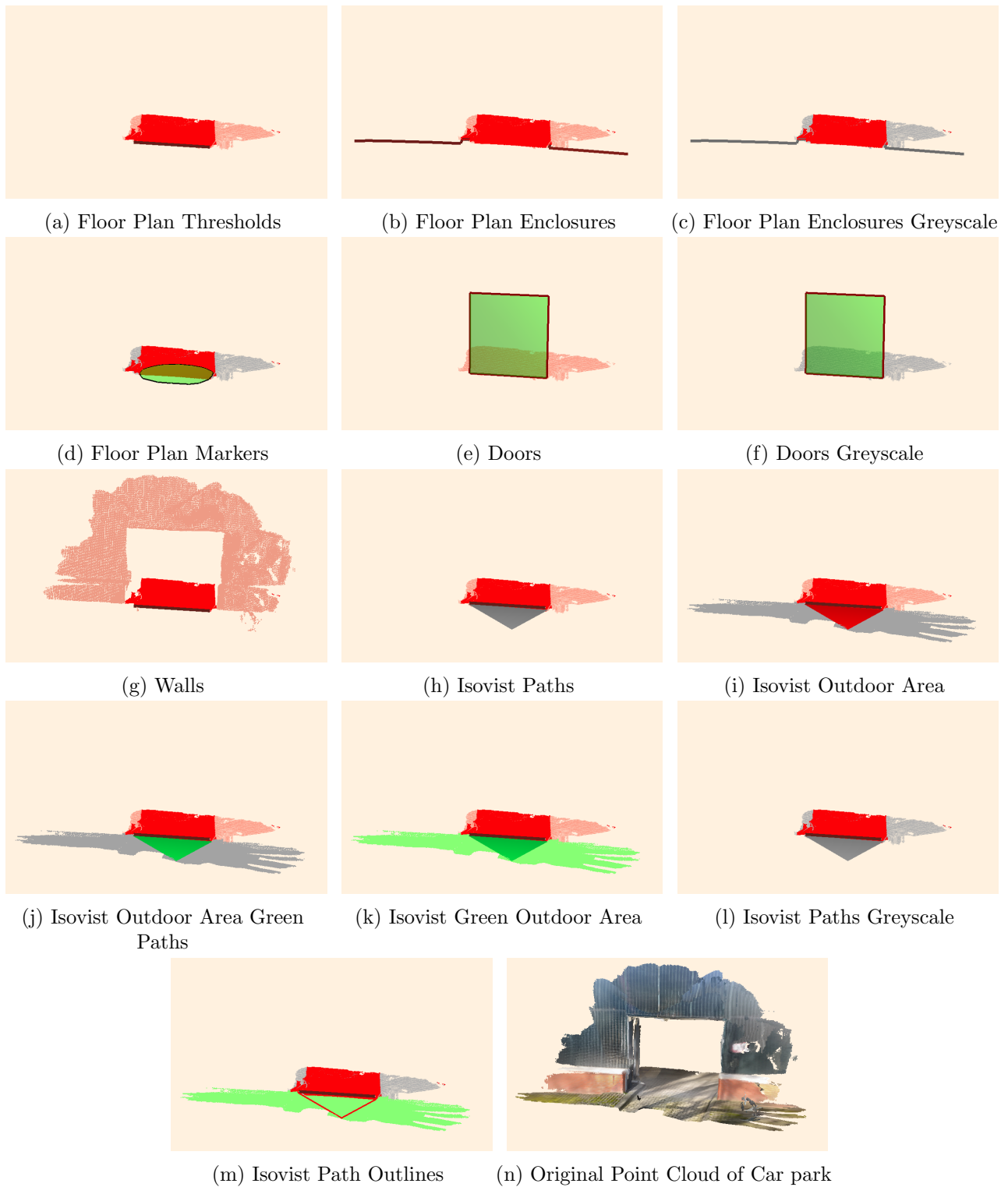


Figure 4.22: All data visualisations for the car park data set from viewpoint 2

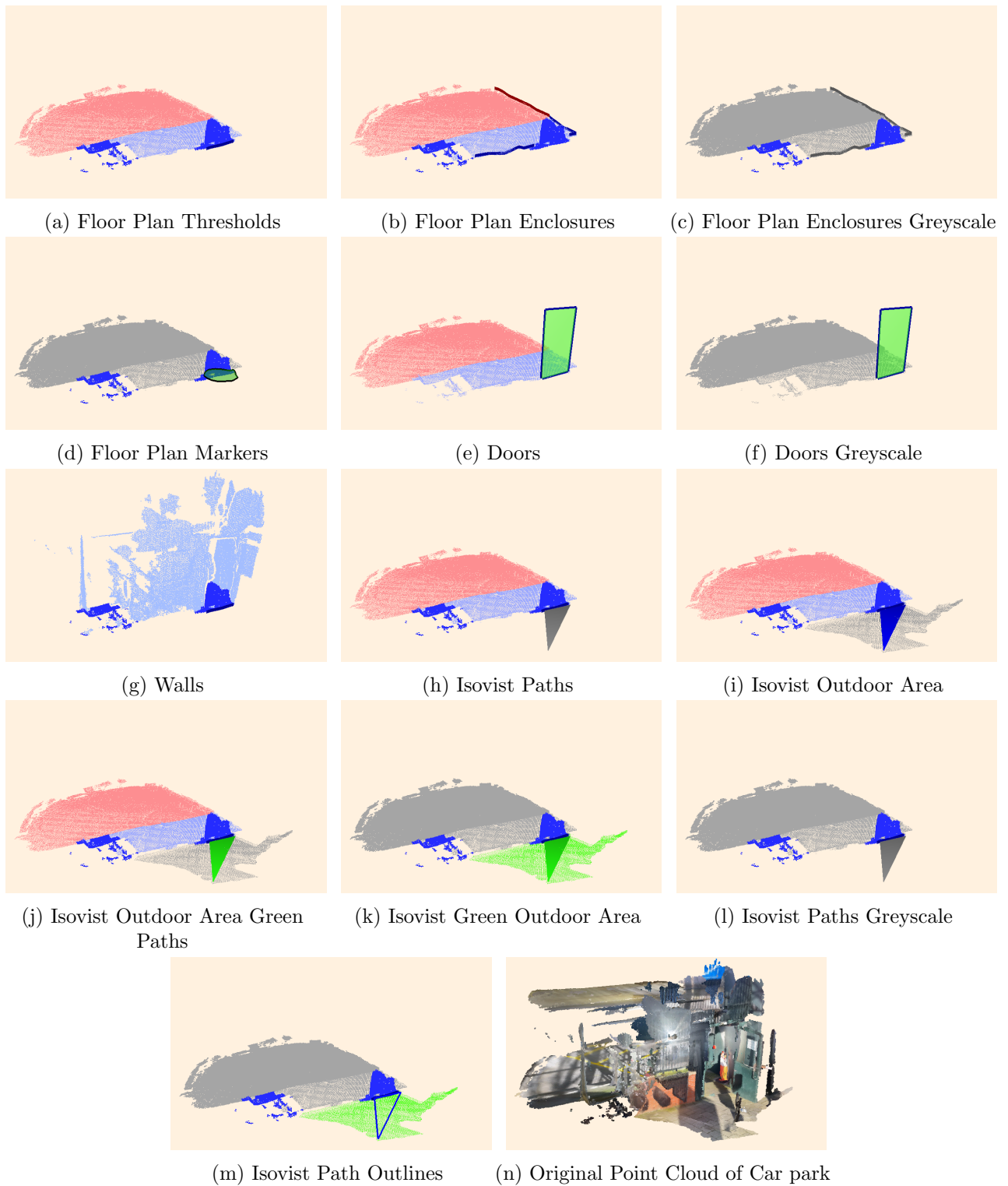


Figure 4.23: All data visualisations for the pedestrian entrance data set from viewpoint 1

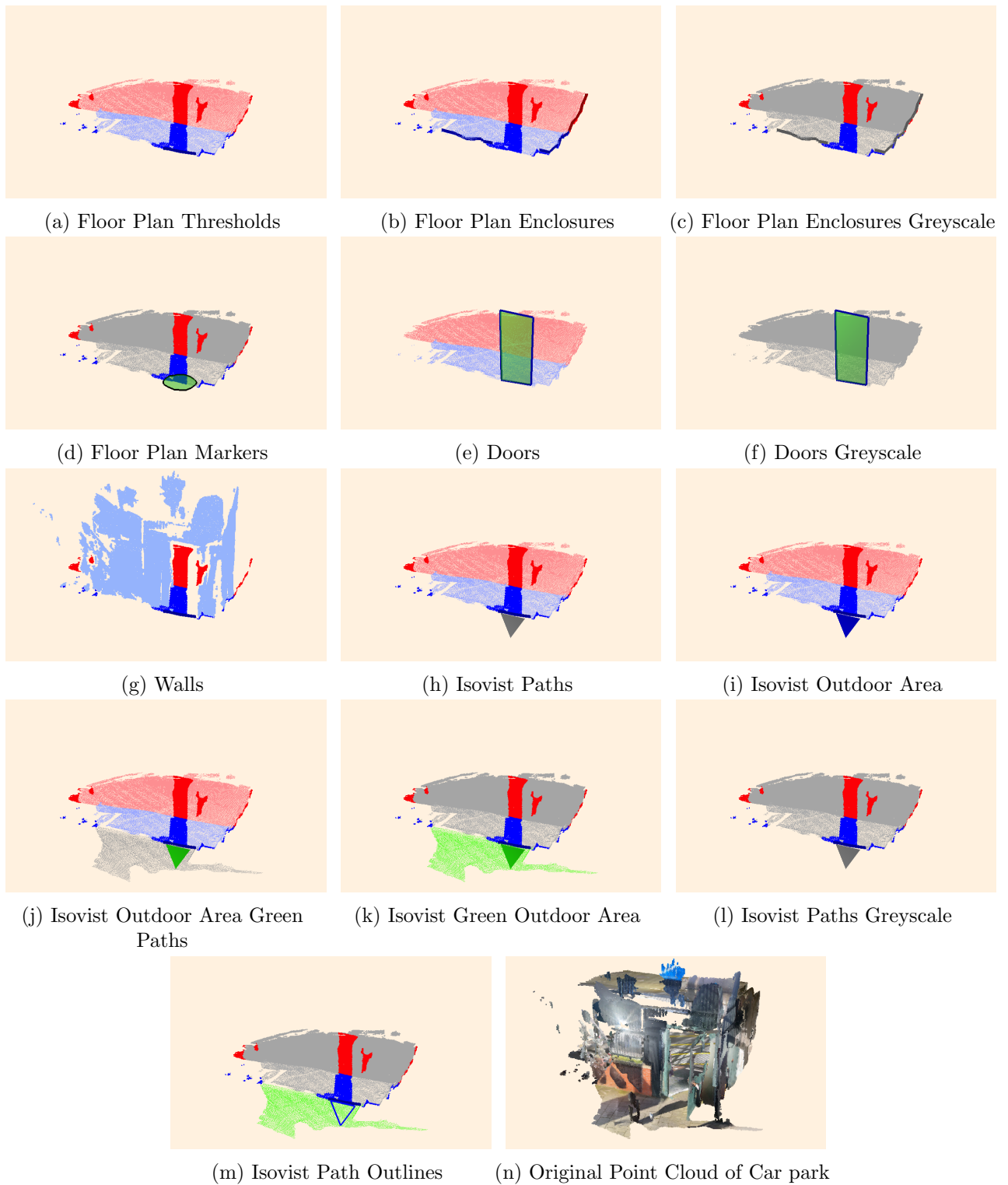


Figure 4.24: All data visualisations for the pedestrian entrance data set from viewpoint 2

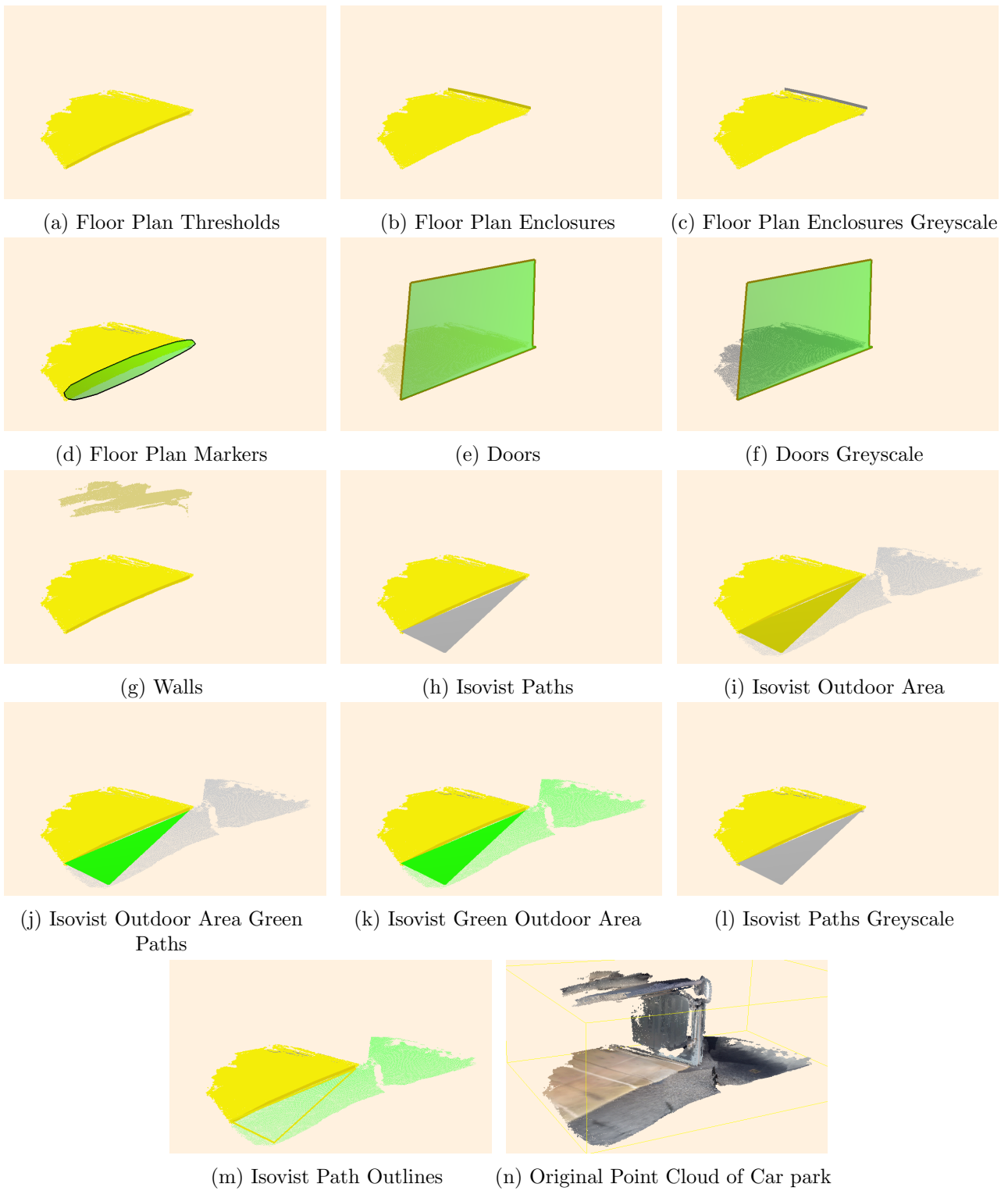


Figure 4.25: All data visualisations for the car park stairs data set from viewpoint 1

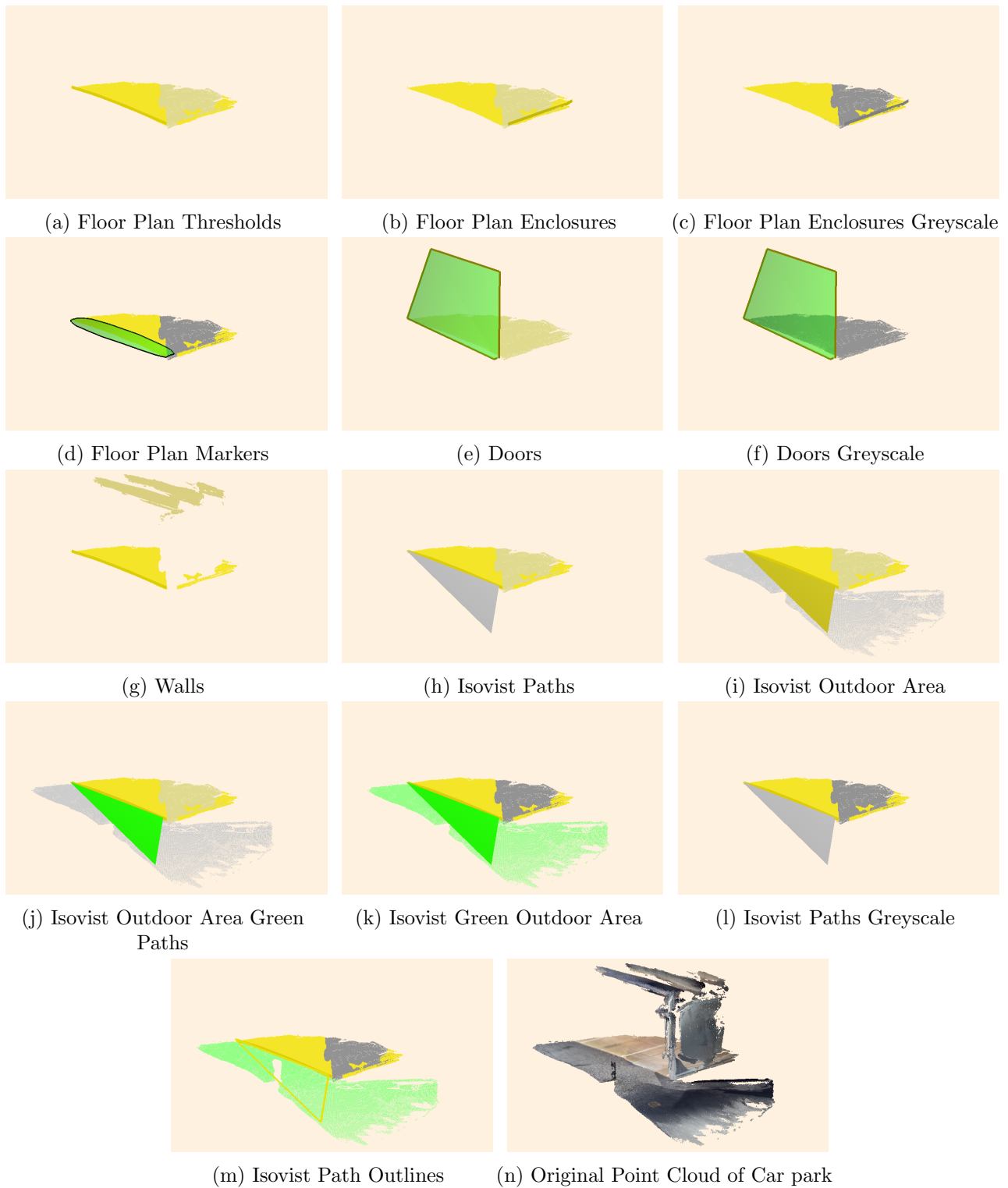


Figure 4.26: All data visualisations for the car park stairs data set from viewpoint 2

Chapter 5

Discussion

The 13 different types of data visualisation for each viewpoint of each point cloud, as seen in [section 4.4](#), are all stylistically similar: they are minimalistic in design, and consistent in their use of visualisation techniques on the same features of the point clouds. All the visualisations visualise the point clouds directly, but only display the most relevant sections of the point clouds, which are intended to be of help to a viewer. Point clouds' negative association with 'clutter' most often comes to light when an entire point cloud is visualised, but here the selective choice of which aspects to visualise, avoids this cluttered sensation.

Though there is a total of 13 different visualisations, many are subtle variations of the same concept. In this discussion, the six principle different designs will be examined:

1. ["Floor Plan Enclosures"](#)
2. ["Floor Plan Markers"](#)
3. ["Doors"](#)
4. ["Walls"](#)
5. ["Isovist Paths"](#)
6. ["Isovist Paths Outline" design: Paths are outlined](#)

The results in [section 4.4](#) that showcase these designs as data visualisations are all the figures labelled (b), (d), (e), (g), (h), and (m), respectively.

[Figure 5.1](#) contains the results of these six designs when applied to the car park entrance point cloud from viewpoint 2.

The 13 requirements laid out in [Appendix A](#), are as follows:

1. Visualisation must be designed with a **layman** in mind as the end user
2. Updating the visualisation view must involve minimal **latency**
3. Must not demand too much **processing power** from device
4. Must be **spatially dynamic**: If the user changes the virtual camera position, the isovist updates accordingly to reflect the new view
5. (Optional:) Must be **temporally dynamic**: The visualisation can change with incoming, real-time data being collected
6. User must be able to quickly link physical environment features with digital twins

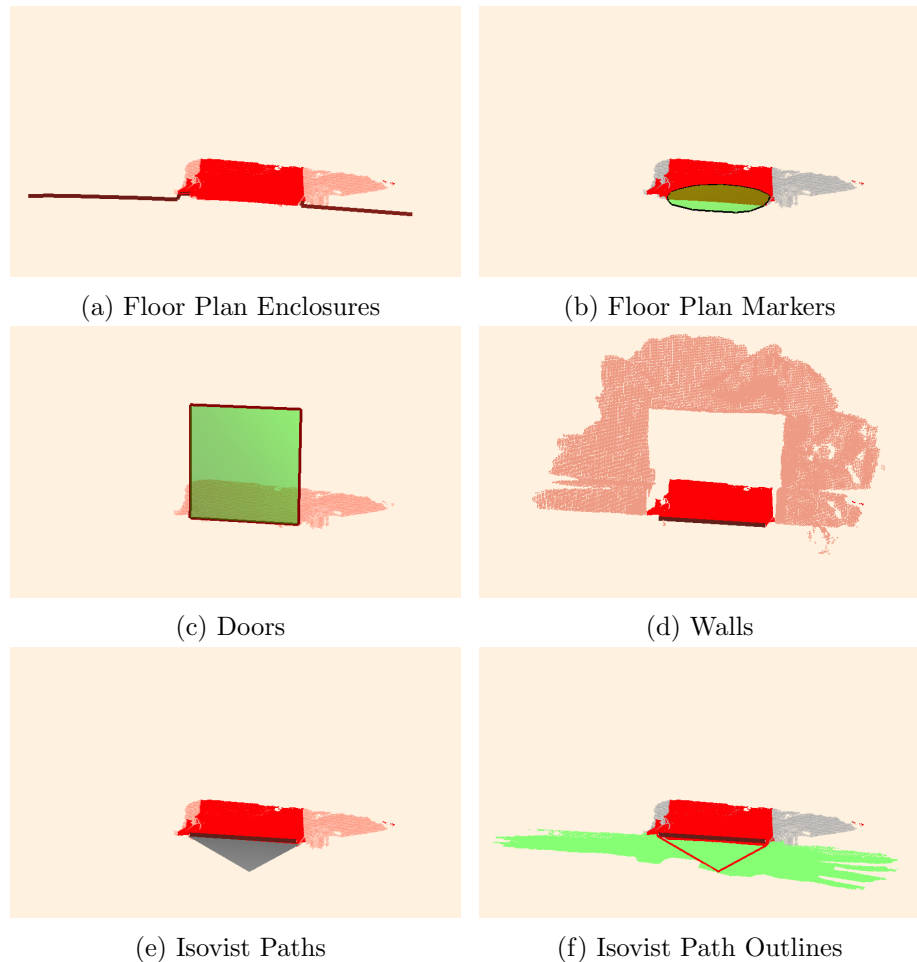


Figure 5.1: The six most differing data visualisations, modelled by the car park point clouds from viewpoint 2

7. Positions of **boundaries** between different types of spaces must be easily identifiable
8. The following three different types of space boundaries must be clearly distinguishable from one another: outdoor-indoor, outdoor-semi-indoor, and outdoor-semi-outdoor
9. The '**non-isovist**' space must be visualised in a way that clearly distinguishes it from the isovist space
10. (Optional:) If time allows, visualisation should be able to incorporate multiple storeys into one view
11. Must be visually **minimalistic** to avoid visual clutter
12. Must follow design best practises
13. Should not be too dependent on a single visualisation channel

The relevant requirements to reflect upon in the separate discussion sections for each different style of visualisation, are:

6. User must be able to quickly link physical environment features with digital twins
7. Positions of **boundaries** between different types of spaces must be easily identifiable

9. The ‘**non-isovist**’ space must be visualised in a way that clearly distinguishes it from the isovist space
11. Must be visually **minimalistic** to avoid visual clutter
12. Must follow design best practises
13. Should not be too dependent on a single visualisation channel

A discussion of each of the six different visualisations with regard to the above six requirements, is laid out in the following sections. The other requirements are relevant for the all the different visualisations, and are reflected upon here.

Layman in mind as the end user:

All of the designs should be comprehensible to a layman; someone who uses navigational applications on their mobile phone, but is not explicitly familiar with the field of geovisualisation, nor has much (or any) experience with point clouds. The designs are all intelligible due to their fairly literal interpretation of 3D space. A user can recognise familiar surfaces like walls and doors. However, the pointillistic style of the visualisations, inherent to point clouds, could throw some users off at first. They might be more attuned to seeing meshes representing smooth surfaces. The use of data visualisation best practices also ensures that the visualisations are as intuitive as possible, but user testing would likely need to be carried out in order to definitively rank the visualisations in terms of ‘most friendly for a layman.’

Minimal latency:

Due to needing to manually generate visualisations for each separate view within the *CloudCompare* environment, there has not been the possibility to test how quickly the visualisation can update to reflect the user’s view as they move around in the real world. However, it was possible to calculate how long certain processing steps took to complete, and the cleaning and subsampling steps for the point clouds all took well under one second to perform. Performing the **HPR** algorithm with an octree level of 16 on each of the (cleaned and subsampled) point clouds also took less than one second:

- Time to perform HPR on “Car park entrance” point cloud: 0.691 s
- Time to perform HPR on “Pedestrian entrance” point cloud: 0.853 s
- Time to perform HPR on “Car park stairs” point cloud: 0.892 s

This means that were the point clouds to be loaded into an environment where it is possible to automatically update the view whenever the user’s position changes, it should be feasible to expect relatively quick updates to the visualisation, without an inhibiting amount of latency for the user. The RAIL performance model (Google Developers [2020](#)) states that “a frame should be produced within 10 milliseconds” for animations, and that “beyond 1000 milliseconds (1 second), users lose focus on the task.” These times stated above straddle the border of being perceived as too slow, but given that the user should be comparing their real environment with the visualisation, they will not be solely focused on wanting to watch an ‘animation’ on their screen. With much larger point clouds, it can be expected that the time for generating new views, will increase considerably.

Processing power:

The processing power required of a computer is fairly minimal because point clouds are efficient data types, but as a point cloud grows larger, so does the processing power required. Whether or not a mobile phone application will be able to generate the visualisations will be very dependent upon

the size of the point cloud, and would need to be tested before developing a final version of these visualisations for end user use.

Spatially dynamic:

The visualisations designed here are spatially dynamic in the sense that different positions result in different visualisations. Visualisations from different positions can be compared and contrasted in order to gain more understanding about a space. However, within *CloudCompare* it was not possible to render automatic updates to the visualisations, whenever the view changed – they had to be manually recalculated.

(Optional) Temporally dynamic:

None of the visualisations experienced new data streaming in while the visualisation process was already in progress. A temporally dynamic feature was ultimately not developed during this research, nor would it have been possible within the *CloudCompare environment*. A different software would need to be found in order to make these visualisations temporally dynamic.

Boundary types distinguishable:

Different hues were used to distinguish different boundary types: red for a transition into indoor space, blue for a transition into semi-outdoor space, and yellow for a transition into semi-indoor space. Green was reserved for any visualisations requiring the visualisation of the outdoor space, though most designs did not explicitly visualise all the points contained in the outdoor space. It is undesirable that such a reliance is placed upon a single visualisation channel, even if hue is the most effective channel for visualising categorical variables (after position in space, but that is not applicable for these spatial visualisations), but other channels like texture and point size were already being used to distinguish the visible points from the occluded points. Additionally, the feature of the Enclosures design where semi-indoor boundaries are depicted with a dotted line, and semi-outdoor boundaries with a dashed line, was not possible in *CloudCompare*, as it is not a graphic design software, but a point cloud comparison software. Further visualisation channels could be explored in other software that caters for more customisation.

Optional: Multiple Storeys:

This was not explored due to time-constraints meaning that other research was prioritised.

5.1 “Floor Plan Enclosures” Visualisation Evaluation

The “Floor Plan Enclosures” design (Figure 5.1a) effectively makes clear the location of the non-traversable boundary through use of the bold boundary line, and highlights the indoor area that is a part of the viewer’s isovist, while distinguishing it from the non-isovist area.

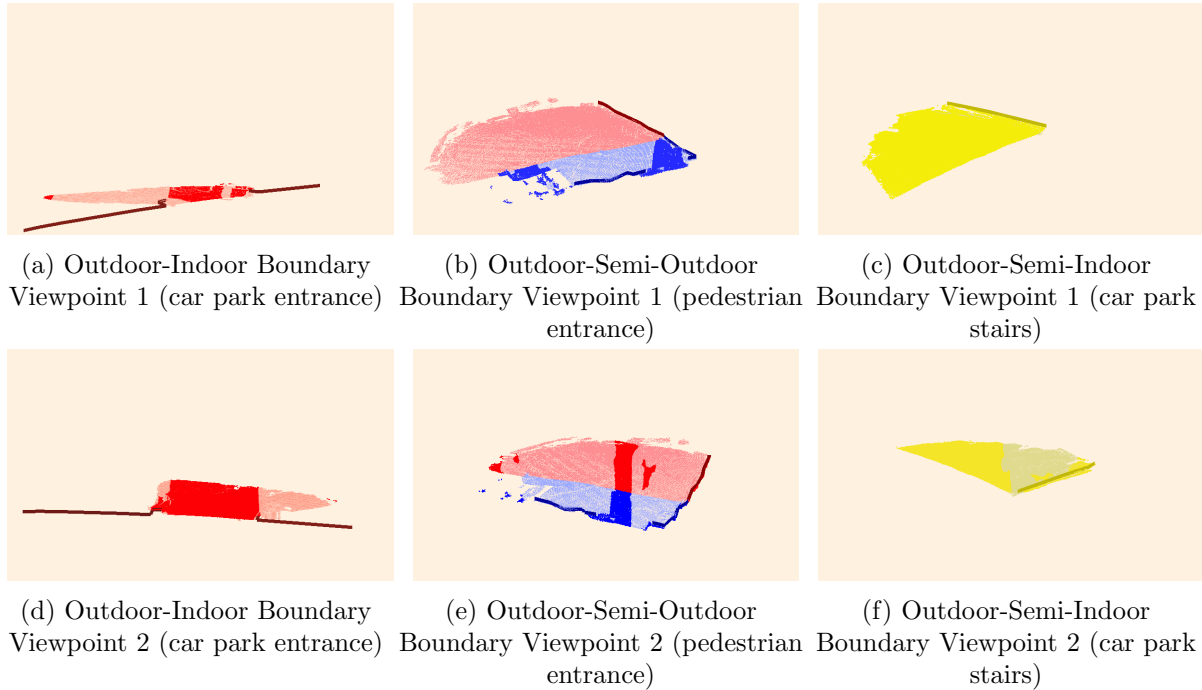


Figure 5.2: The “Floor Plan Enclosures” visualisations for each space boundary type (Outdoor to Indoor, Outdoor to Semi-Outdoor, and Outdoor to Semi-Indoor), from two differing viewpoints

Requirement	Fulfilled?
3.a. Physical Environment/Digital Twins	Yes: clear distinctions between spaces using multiple visualisation channels
3.b. Boundaries identifiable	Yes: gaps in enclosure marking makes this explicit. thick polylines also draw attention to the boundary
3.d. Isovist/Non-isovist distinguishable	Mostly: clear contrast between more and less saturated points, except in semi-indoor (yellow)
4.a. Visually Minimalistic	Yes
4.b. Design Best Practises	Yes: most effective channels used for the most important information (space-type)
4.c. Multiple visualisation channels	Yes: hue, saturation, point size

Table 5.1: An evaluation of the “Floor Plan Enclosures” visualisation, based upon set requirements

5.2 “Floor Plan Markers” Visualisation Evaluation

The “Floor Plan Markers” design (Figure 5.1b) draws attention to the locations of space boundaries by use of a mesh marker drawn using the surrounding points along the space boundary.

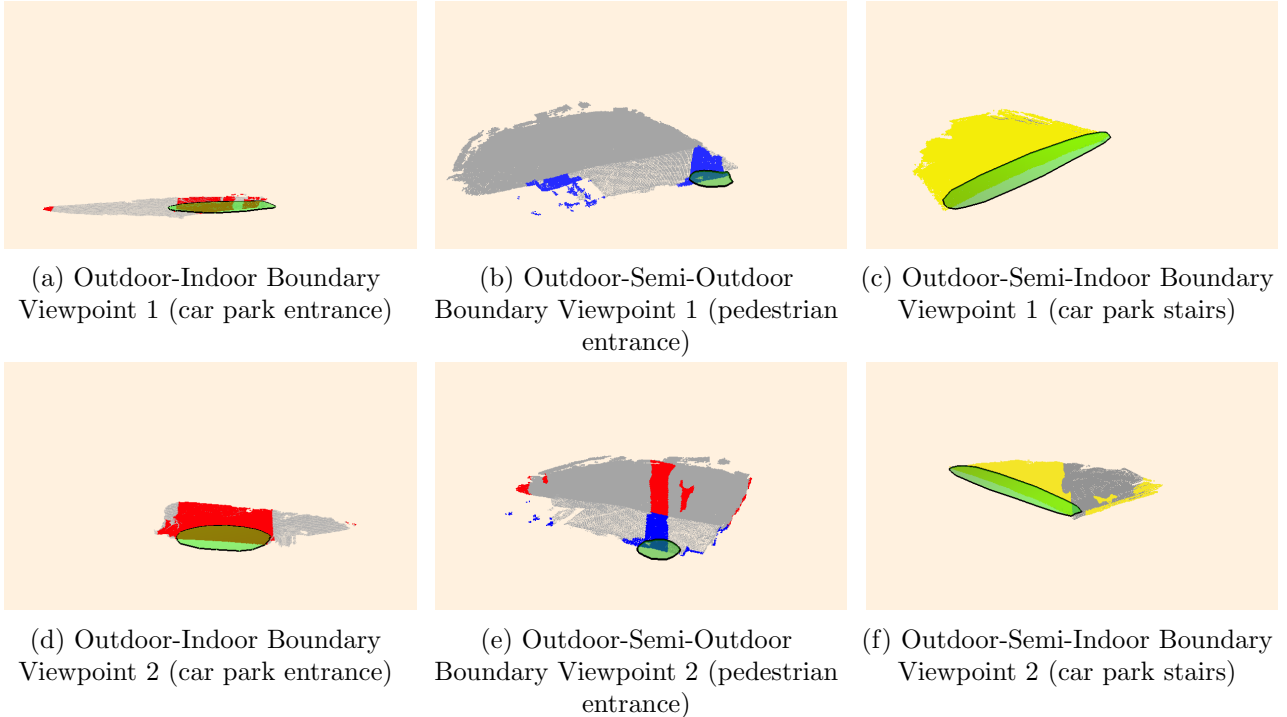


Figure 5.3: The “Floor Plan Markers” visualisations for each space boundary type (Outdoor to Indoor, Outdoor to Semi-Outdoor, and Outdoor to Semi-Indoor), from two differing viewpoints

Requirement	Fulfilled?
3.a. Physical Environment/Digital Twins	Mostly: markers that do not exist in real life, might throw user off at first. They will need to realise that this is a more symbolic visualisation
3.b. Boundaries identifiable	Yes: markers draw a lot of attention to the boundaries
3.d. Isovist/Non-isoivist distinguishable	Yes: grey background of points allows coloured points (visible) to jump out at user
4.a. Visually Minimalistic	Yes: though another visualisation technique for the markers is possible, potentially avoiding the use of more colour
4.b. Design Best Practises	Yes: most effective channels used for the most important information (space-type and boundary location)
4.c. Multiple visualisation channels	Yes: hue, point size, texture. Also polyline and mesh marks

Table 5.2: An evaluation of the “Floor Plan Markers” visualisation, based upon set requirements

5.3 “Doors” Visualisation Evaluation

The “Doors” design (Figure 5.1c) focuses on communicating the location of the boundary in a recognisable form of a typical doorway. The translucent element allows the points beneath to remain visible.

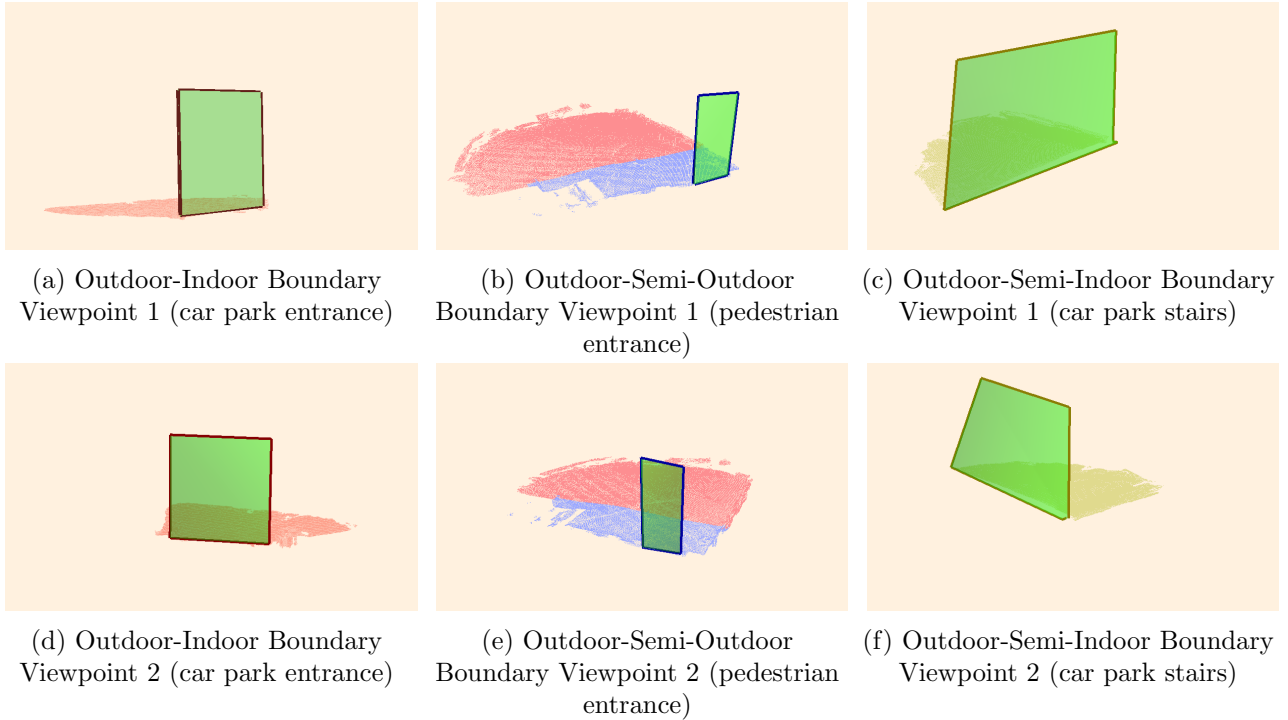


Figure 5.4: The “Doors” visualisations for each space boundary type (Outdoor to Indoor, Outdoor to Semi-Outdoor, and Outdoor to Semi-Indoor), from two differing viewpoints

Requirement	Fulfilled?
3.a. Physical Environment/Digital Twins	Yes: doorways are recognisable elements in the real-world space, allowing for quick matching
3.b. Boundaries identifiable	Yes: doors facilitate this greatly
3.d. Isovist/Non-isovist distinguishable	No: a visible doorway indicates that surrounding points could be visible, but this is implicit
4.a. Visually Minimalistic	Yes: only doorways and different hues for the different space types are used
4.b. Design Best Practises	Mostly: Only colours used to differentiate space-types, should used another visualisation channel, like texture
4.c. Multiple visualisation channels	No: Only colour is used, along with a mesh for the doorways

Table 5.3: An evaluation of the “Doors” visualisation, based upon set requirements

5.4 “Walls” Visualisation Evaluation

The “Walls” design (Figure 5.1d) draws attention to the space boundary, visualising points along the walls that constitute the boundary. Gaps in the point clouds implicitly visualise the locations where users can transcend the boundaries.

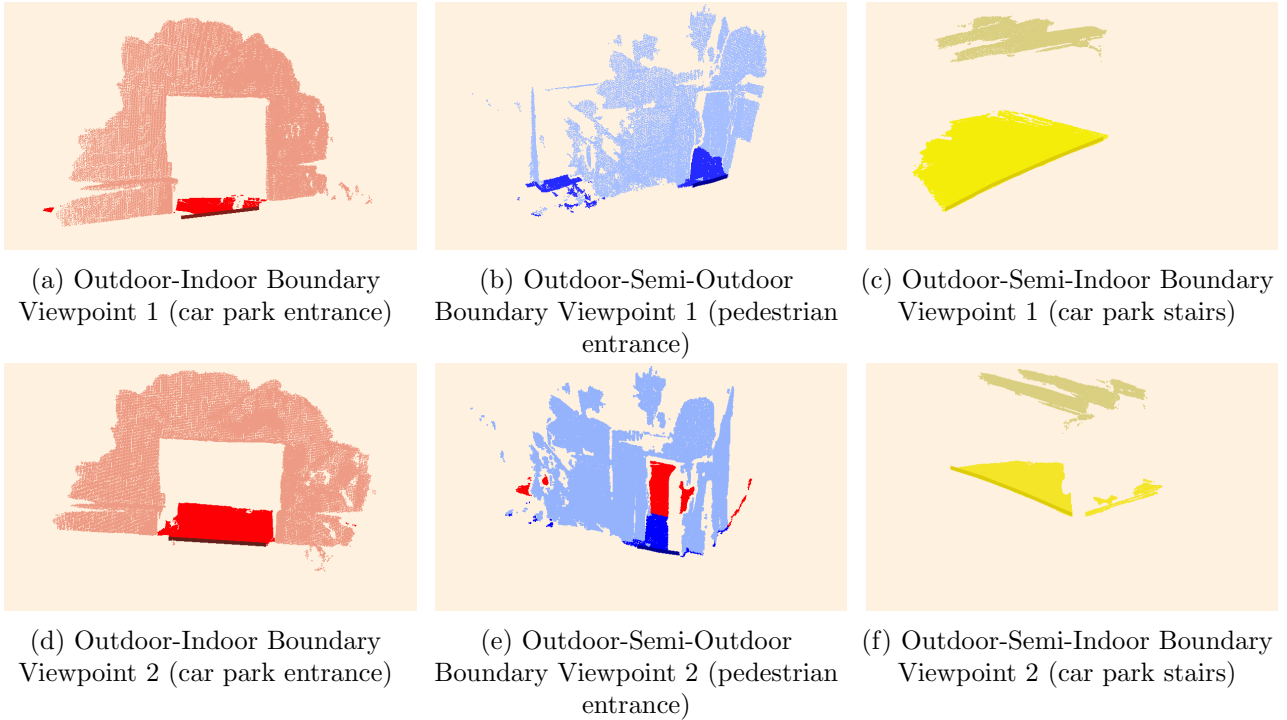


Figure 5.5: The “Floor Plan Markers” visualisations for each space boundary type (Outdoor to Indoor, Outdoor to Semi-Outdoor, and Outdoor to Semi-Indoor), from two differing viewpoints

Requirement	Fulfilled?
3.a. Physical Environment/Digital Twins	Mostly: visualising along the z -axis helps user understand that there is a 3D environment, but the walls can be unclear at times (Figure 5.5b)
3.b. Boundaries identifiable	Yes: walls draw attention to the boundary locations, but it can be difficult to discern where the entrance is, depending on the point cloud
3.d. Isovist/Non-isoivist distinguishable	Yes: saturation contrast of points facilitates this
4.a. Visually Minimalistic	Mostly: only using points is commendable, but complex or badly recorded point clouds, can cause confusion about what is being visualised (see Figure 5.5e)
4.b. Design Best Practises	Mostly: colours are utilised well, but some visualisations can feel cluttered
4.c. Multiple visualisation channels	Yes: hue, saturation, point size, along with polylines

Table 5.4: An evaluation of the “Walls” visualisation, based upon set requirements

5.5 “Isovist Paths” Visualisation Evaluation

The “Isovist Paths” design (Figure 5.1e) is where .

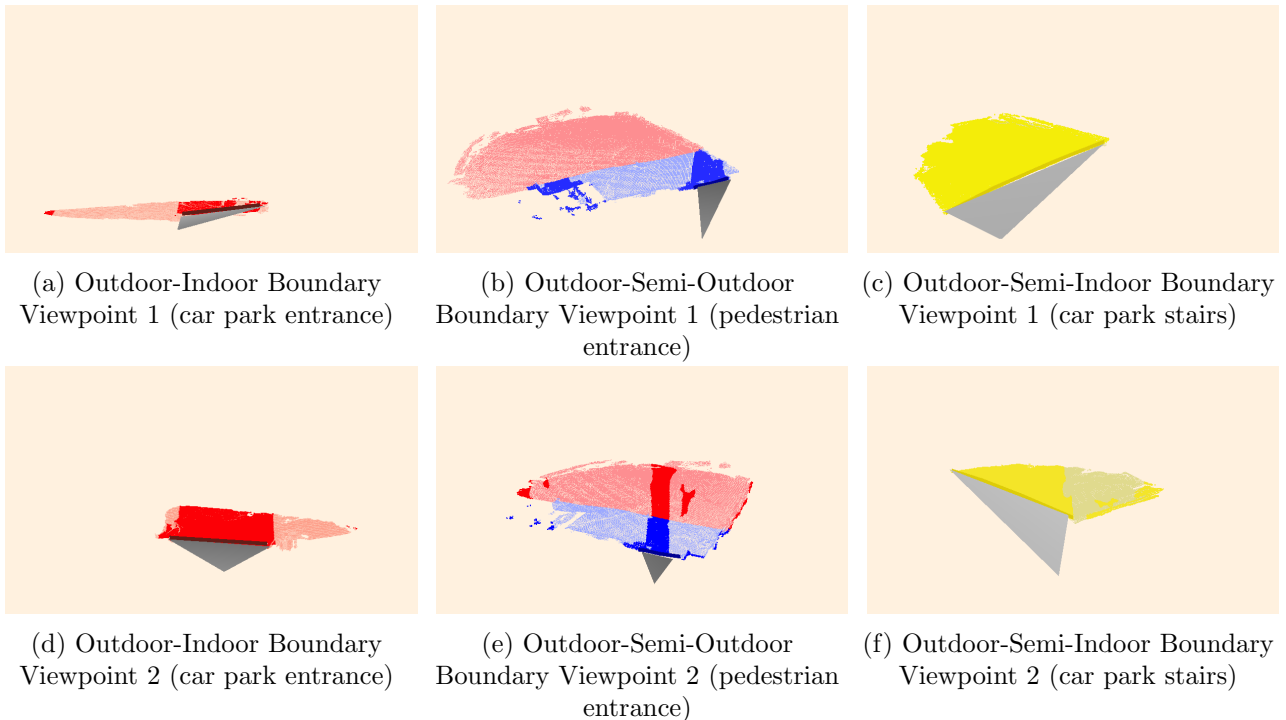


Figure 5.6: The “Isovist Paths” visualisations for each space boundary type (Outdoor to Indoor, Outdoor to Semi-Outdoor, and Outdoor to Semi-Indoor), from two differing viewpoints

Requirement	Fulfilled?
3.a. Physical Environment/Digital Twins	Maybe: user testing would verify whether isovist visualisation here help in quicker feature recognition
3.b. Boundaries identifiable	Mostly: the isovists guide the eye towards the boundary location, but the flow between the outdoor isovist and the visible space that continues beyond it, is too harmonious
3.d. Isovist/Non-isovist distinguishable	Yes: all isovist sections are given far more saturation, though the yellow of the semi-indoor space needs improvement
4.a. Visually Minimalistic	Yes: only points and a single mesh and polyline are used
4.b. Design Best Practises	Mostly: hue is used, but consider using a different channel to better distinguish outdoor isovist, e.g. texture or transparency
4.c. Multiple visualisation channels	Yes: hue, saturation, point size, along with mesh and polylines

Table 5.5: An evaluation of the “Isovist Paths” visualisation, based upon set requirements

5.6 “Isovist Paths Outline” Visualisation Evaluation

The “Isovist Paths Outline” design (Figure 5.1f) visualises the borders of isovists outdoors in colours corresponding to the space-type they lead towards. The only one of the six designs that visualises points comprising the outdoor ground space.

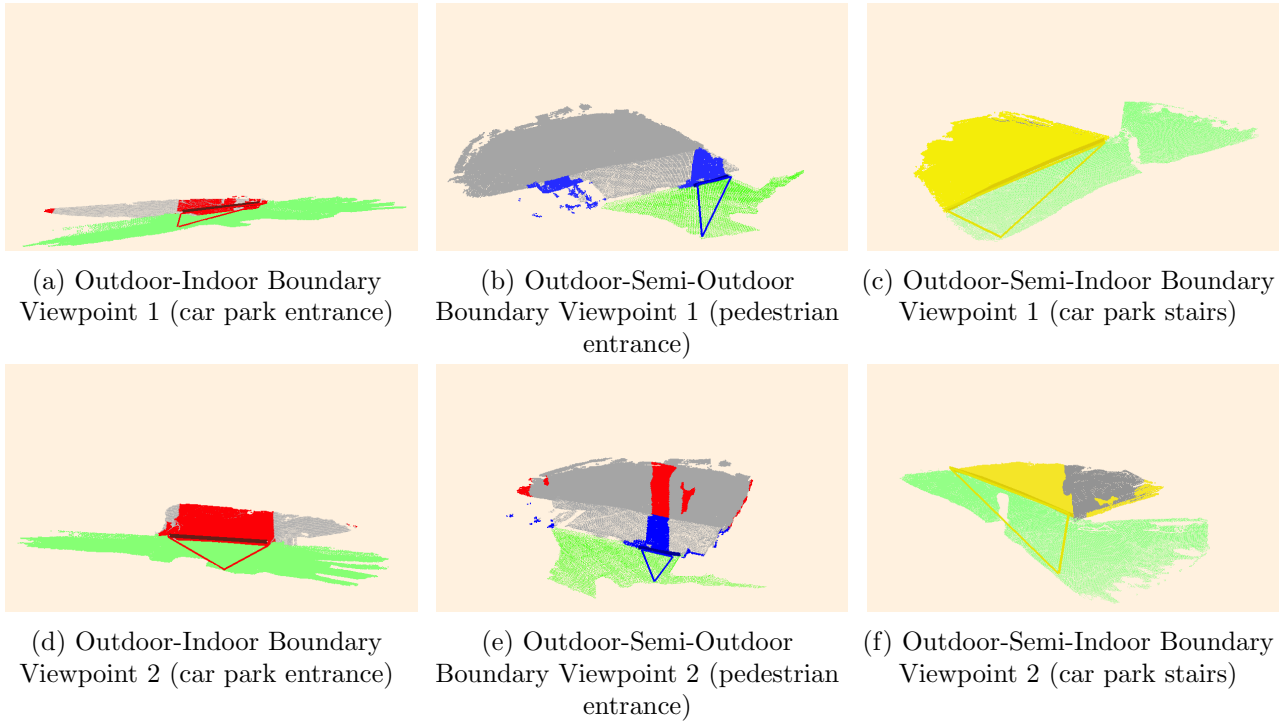


Figure 5.7: The “Isovist Paths Outline” visualisations for each space boundary type (Outdoor to Indoor, Outdoor to Semi-Outdoor, and Outdoor to Semi-Indoor), from two differing viewpoints

Requirement	Fulfilled?
3.a. Physical Environment/Digital Twins	Mostly: using green is indicative of outdoor space, but visualisation has many colours of equal saturation in use, confusing the eye
3.b. Boundaries identifiable	Mostly: outlines are usually easy to spot, with the exception of the yellow semi-indoor isovist on the bright green backdrop
3.d. Isovist/Non-isovist distinguishable	Partially: grey-coloured occluded points allow focus on the saturated points, but multiple colours of equal saturation are in use, making a hierarchy difficult to identify
4.a. Visually Minimalistic	Partially: reducing the colours used could make visualisation simpler
4.b. Design Best Practises	Yes: hue is used as the primary channel for categories
4.c. Multiple visualisation channels	Yes: hue, saturation, point size, and polyline marks

Table 5.6: An evaluation of the “Isovist Paths Outline” visualisation, based upon set requirements

5.7 Limitations

The designs have potential to be effective at communicating the nature of a transitional space with the user, but the quality of the point clouds used, limits the extent to which the designs can display their strengths. Implementing these designs upon more complete point clouds would be beneficial, both in order to provide the user with more comprehensive views, as well as be able to test functionality such as latency with a larger point cloud.

These visualisations are still very much in their infancy when considering that the ultimate intended usage is within a navigational application, and the fact that the visualisations are not yet entirely spatially dynamic, nor do they have the capacity to accept incoming live data. However, there is potential with respect to the minimalistic and efficient manner in which these designs communicate information about 3D space to the viewer, and some thorough user testing could shed light onto the persisting weaknesses of the designs, so that they can be developed into optimal renditions of the concepts first shown in this work.

One shortfall in the evaluation of the designs, is not easily being able to determine the effectiveness of the “Isovist” designs (subsection 4.3.4), because of the short physical range of the point clouds. If the point clouds extended all the way back to the viewer’s position, then the visualisation of the “isovist path” would be elongated, and due to that, more likely to be interpreted as a path by the user, who would allow the eye follow the polygon towards the space boundary. As it stands, the isovist paths are mostly irregular triangles (see Figure 4.21h and Figure 4.22h as examples).

Chapter 6

Conclusions

This thesis was an exploration into the viability and suitability of applying elements of isovists as a form of spatial representation, onto point cloud data for the purpose of aiding understanding around outdoor space boundaries.

The main research question encapsulates the above: “To what extent could point clouds be successfully used to visualise third-person view isovists of, and occluded spaces surrounding, outdoor space boundaries, in a way that helps users of the visualisation better understand these environments?”

The concept behind this thesis was imagining a near-future in which multiple users (passively) create point cloud scans of their surroundings using their mobile phones, and that this data is assembled into a cohesive data set. This data set would then be transformed into visualisations that differentiate between visible and occluded spaces from a certain viewpoint, as well as provide information on the different types of spaces that surround the user. Further attributes for the point cloud could be easily added, providing only more information on, and understanding of, the space.

Firstly, the initial research questions will be re-addressed, and answers to these questions will be given. These answers should together form a general understanding of the extent to which the main research question has been addressed, and the general conclusion should encapsulate this.

6.1 Addressing the Research Questions

As detailed in [section 1.5](#), the sub-research questions that were formed in order to help answer the main research question, are as follows:

1. How can point cloud data be best classified into types of spaces and space boundaries, as well as classified as ‘visible’ or ‘occluded,’ in the context of a third-person view?
2. What visualisation techniques can be used upon a point cloud to visualise different types of spaces within and outside of an isovist, and how can the space boundaries be made clearly distinguishable?
3. Which visualisation techniques are the most successful when evaluated against a list of requirements?
4. What does the visualisation of isovists from a third-person view offer, that other visualisation techniques of outdoor space boundaries do not, and does visualising the non-isovist area have added value?

The next four subsections deal with each sub-research question individually, discussing to what extent they were answered, and what conclusions were derived from them.

6.1.1 RQ 1: Point Cloud Classification

This research’s classification technique involved manually classifying point clouds into four different types of spaces; ‘Indoor,’ ‘Semi-Indoor,’ ‘Semi-Outdoor,’ and ‘Outdoor,’ with the aid of *CloudCompare*’s `Compute Geometric Features` tool to better identify space boundaries). The manual approach was suitable for this research, as the point clouds used were fairly uncomplicated – usually containing only two or three different space-types – and small in terms of point count. Therefore manual classification did not take long, nor was there much of an opportunity for significant errors to be made. (Semi-)automatic classification was not particularly suitable for this project for the same reasons. The time spent developing an automatic classification of different space-types would have outweighed all time spent on manual classification, and there probably would have been a greater opportunity for incorrectly classified points, due to an algorithm’s inability to make sense of imperfect or incomplete point clouds. However, if this research were to be applied to a wider context where multiple users are all wishing to view larger sets of (near) real-time data, (semi-)automatic classification methods would certainly need to be developed in order to realistically maintain such data visualisations.

During this research, much time was initially spent on attempting to define what a space boundary is, and to classify these thresholds separately from the remaining data, in order to be able to directly call upon these areas to apply different visualisation techniques. However, with LiDAR scans often containing only information about one side of an architectural element such as a wall, this ultimately proved unsuccessful, and instead the boundaries between different space-types remained implicit within the classified point clouds. Contrary to initial expectations, this did not affect the process of building the visualisations as much as had initially been anticipated. This could be partly due to the types of visualisations that were designed, and that were different designs to be made, it might be necessary to explicitly call upon the boundaries themselves. Again, the fact that the point clouds being used were small in nature, meant that it was manageable for the limited number of space boundaries to be manually defined during the visualisation process. Appropriate points that constituted a boundary could be selected, with which to draw a line or a polygon.

Classifying the point clouds into visible and occluded areas proved a greater challenge than the classification of space-types, with typical methods such as using thicker ray casting rays being dismissed fairly early on, due to the inaccuracies that often result from such techniques. Other methods that involve transforming a point cloud into a mesh or a voxelised space, were initially accepted as the only viable alternative, and work was done to build a voxel traversal algorithm for ray casting. However, these alternatives were never considered optimal, as they required an additional transformation step to be carried out upon the point cloud, which would inevitably slow down the visualisation generation process. Another method for performing visibility analysis involved the calculation of point normals, but during the early stages of exploration, it was soon discarded upon the discovery of Katz, Tal, and Basri (2007)’s work. Katz, Tal, and Basri (2007) showed how an algorithm that inverted a point cloud and extracted the points on the inverted point cloud’s hull, could determine visibility without need for the above methods (see section 2.4 for a more detailed explanation). This algorithm was employed to a relative degree of success upon the point clouds within the *CloudCompare* program, though changes in camera position could not result in automatic updates to the subset of visible points within this environment, despite the research claiming the algorithm to have the capability for such use.

6.1.2 RQ 2: Visualisation Techniques

Working towards answering this research question was the core focus of this project, and involved researching good data visualisation practices, writing a `List of Requirements`, and developing a variety of viable designs to prepare for the visualisation building process. Due to the “third-person view” requirement of the topic, the designs retained a very evident connection to 3D space rather than branching out to more abstract visualisation concepts, and thus the potential for a higher level of creativity was slightly stifled. The designs did not differ wildly from one another because they all used the ground

layout of the point cloud space as a starting point from which to build up a visualisation. However, the objective of the visualisations was to help aid a user in better understanding an environment. To hark back to Brehmer and Munzner (2013)'s "Typology of Visualisation Tasks" (subsection 3.4.2), the intended user for this project is someone who is wishing to "consume" the visualisation (rather than "produce" information with it) by "discovering" knowledge, instead of "presenting" or "enjoying". More symbolic visualisations set outside of a 3D model of space, could have been appropriate for a user who wants to "enjoy," letting them see their world in a new light, and encouraging them to think more creatively about the space.

The designs also needed to incorporate a method for distinguishing between the spaces that were included and excluded from the user's isovist at a given viewpoint. The definition of what counted as a visualisation of an isovist was sometimes interpreted fairly broadly. For example, in the "Floor Plan" visualisations (subsection 4.3.1) the viewer's isovist is implicitly indicated by visualising visible and occluded boundaries differently, thus implying that a user who can see the visible boundaries, can also see the adjoining boundary walls. Many of the visualisations ignored the ground of outdoor areas in order to avoid clutter, and help the user focus on the space boundaries themselves, even though this ground area is indisputably a part of any viewer's isovist. This means that not all of the visualisations fully visualised the differences between visible and occluded space, and thus did not entirely fulfil the brief. However, due to this, insights were gained into which spaces are strictly necessary to visualise, and which are not, for a user to still obtain the same information. Some sections of isovists were rendered superfluous, but all visualisations did incorporate at least one element distinguishing the isovist from the non-isovist space.

To answer the question of which techniques can be used for the visualisation of (non-)isovists, the methods employed in various visualisations shown in section 4.4 are as follows:

- **Saturation:** The points that were a part of the isovist were given a higher colour saturation than the points that were occluded
- **Point size:** The points that were a part of the isovist were given a larger point size, resulting in smaller gaps between points, and thus putting greater emphasis on the isovist space
- **Isovist Polygon:** A more traditional approach where a polygon was drawn to outline the part of the isovist that connects the user's position to a space boundary. Most of these polygons were filled with an opaque hue that provided additional information such as the type of space being entered. One design only showed the outline of a polygon (Figure A.1m), which left the outside space beneath visible.

Care was taken to ensure that all designs adhered to the List of Requirements as much as possible, and that elements like space boundaries were always thoughtfully visualised. In section 4.3, the intention of using multiple visualisation channels on each of the boundaries is described, in order to keep the visualisation accessible for visually impaired users. This also benefits users without visual impairments: if a screen is set at a low brightness level, different hues are more difficult to distinguish from one another, and having boundary designs with more divergent features from one another, will instantly draw attention to the fact that those features represent different types of boundaries. The channels chosen for visualising different space-types and boundaries were:

- **Hue:** Points were assigned different colour hues based on their space-type
- **Texture:** Supplementary to hue, different textures were used on the points clouds of different space-types
- **Polygons:** In the section 4.3.1 design, a circle was drawn (defined by the surrounding points of the transitional space) on the ground to indicate the location of the space boundary. In the section 4.3.2 design, polygons in the shape of the boundary, typically rectangular like doorways, were drawn (defined by the surrounding points). These stood out from the rest of the point

cloud, due to the smooth, continuous surface, and were also made translucent so that the user could see the space beyond the boundary

- **Walls:** In the [section 4.3.3](#) design, the points that constituted the wall that was a component of the isovist were displayed, in much the same fashion as a traditional 3D isovist

Additionally, ensuring that the visualisations remained accessible to any users with visual impairments or non-standard screen displays, was at the forefront of the design process, with multiple visualisation channels for single elements being used when possible (e.g. line weight, colour, and stroke type of space boundaries were all different for different space-types). When it came to producing the visualisation designs within *CloudCompare*, there was a limit to how many channels could be used to represent some of the space boundaries; it was not possible to draw dotted or dashed lines between different points.

Without a clear legend or an explanation, it is anticipated that new users will not understand which visualisation channels correspond to which type of space, as ‘space-type’ is a form of spatial information that is rarely visualised symbolically in navigational applications. Far more common are 3D projections of buildings without any addition symbology, to indicate that they contain volumes inside – see [Figure 6.1](#) for an example of how *Google Maps* does this within its mobile application.

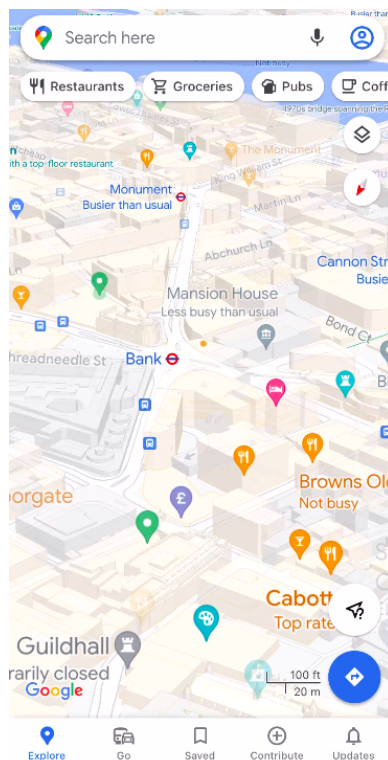


Figure 6.1: A screen capture of *Google Maps*’ mobile application method for depicting the 3D aspects of buildings

6.1.3 RQ 3: Evaluation of Techniques

Upon evaluating each final design against the [List of Requirements](#) in the [Discussion](#) section, the designs could be better contrasted with one another. The requirements that directly dealt with visualisation techniques were given a greater weighting because these requirements are more relevant to answering this research question about which visualisation techniques are the most successful.

To summarise what was stated in [chapter 5](#), the visualisations that fulfilled the most requirements definitively, can be found in the ["Floor Plan Enclosures" Evaluation](#), the ["Floor Plan Markers" Eval-](#)

uation, and the "Doors" Evaluation. They were the most successful because they most explicitly addressed the need to visualise the isovist space in a way that clearly distinguishes it from the non-isovist space. However, the more exciting visualisations with potential for further development can be seen in the "Walls" Evaluation, and "Isovist Paths Outline" Evaluation.

6.1.4 RQ 4: Reflection on the Use of Isovists

2D isovists are more prevalent than 3D isovists, especially outside of the architectural field, but they tend to be used more as a novelty gimmick for users to enjoy, rather than a practical tool for everyday life. The 3D isovist is a fairly well-known and used concept in environment analysis, but it has rarely been put to use in geovisualisations designed for end users of navigational applications. This is could be because they look more complex than 2D isovists, or are considered more appropriate for calculations of static view angles and volumes, rather than for providing useful information to navigators. A 3D isovist runs the risk of requiring a lot of space within a visualisation, because a common way of visualising 3D isovists is to translate the concept of a visible plane in a 2D isovist, into the total volume of space visible in 3D. The isovist is often depicted rather literally by a solid, 3D shape that encloses the entire space. The tactics employed in the visualisations created in this work, avoided the common pitfall of the isovist volume being directly visualised. Instead indirect visualisation methods were most often used to indicate which areas are visible and occluded. That a user can see all of the space between them and the next solid object along their line of sight, is assumed to be true, and it was therefore considered unnecessary to visualise this 'empty' space. One could say that visualisations such as in Figure 5.1a where only the points that constitute the ground area are being visualised, are simply 2D isovists visualised in a 3D environment. This technique of visualising the user's isovist is certainly legitimate, and it is speculated that the depth that such a 3D view provides is more intuitive than a 2D isovist, for a user trying to place themselves within that virtual space in their mind's eye. This theory could be proven or debunked by means of user testing.

Projecting the isovist onto the ground in a 3D view with an angle less than 360°, as in the "The Isovist" Designs, provides the user with the benefit of being able to orientate themselves quicker by immediately spotting which façades should be in their line of sight in real life. The isovists also guide their eyes to the various entrances in front of them, faster.

Depending on the main aim of a visualisation, including information about the user's isovist can be advantageous or distracting. If the main aim of a navigational application is helping a user understand their own position in relation to their environment, the isovist visualisation (as seen in "Isovist Paths") is helpful, as it allows for quicker orientation and matching of real-world objects to their digital twins. But if the main aim is to provide more information about other aspects of the environment, then the isovist might be less valued information that unnecessarily clutters the design, and thus implicitly visualising the isovist (like in the "The Walled" Designs) might be more appropriate.

In the realm of navigational applications, most will visualise a space while making no distinction between the visible and occluded space, giving users the task of understanding which features in the application, are also in front of them in real life. This can be seen in Figure 6.1, where many buildings are shown from a third-person view, but the majority will not be visible to a user on the ground. The antithesis to this, are the visualisations that simply ignore any occluded space, in the desire to keep the visualisation as 'clean' as possible. The visualisation of the non-isovist area can undoubtedly be helpful to a user, as it provides additional information about their surroundings that they cannot directly gauge from the real environment. The visualisations of occluded spaces created in this project only give information on their space-type, which might not seem like an overly useful function as in most cases it is reasonably clear what type of space is most likely to be contained behind a closed doorway. However, these fundamental visualisation concepts could be developed with more complex data, to containing more valuable information about occluded spaces. Examples of this could include what kinds of activities a space is being used for (is it a cultural building or one full of offices?), or

whether the space boundary is private or publicly accessible. Information on the accessibility of spaces could also be insightful, with information like how many stairs must be climbed beyond a particular space boundary, in order to reach a user’s desired destination, or the width of the smallest doorway en route. This would allow users with restricted mobility or parents with prams, etc. to choose in advance the entrance into a building that will take them on a path of least ‘resistance’ to their final destination, making their total journey right up until the end point, more comfortable.

To recapitulate, it has been argued that visualising the isovists of an area can be useful to a user, depending on their task, and that visualising the non-isovist area can provide helpful information to a user. Now the question remains as to whether or not the visualisations created in this project can be applied to a larger point cloud of a more complex environment with multiple space boundaries and space-types, while still retaining these advantages and remaining clear visualisations.

6.2 General Conclusion

From the results of this project, it was determined that the direct use of point clouds is a viable method for the data visualisations of transitional spaces. The well-documented drawbacks of point clouds did not hamper these visualisations too significantly:

- That point clouds can be considered cluttered was overcome by the exclusion of less relevant areas of the point clouds, as well as by the inherently minimalist design of all of the visualisations
- That impurities and other temporary obstacles in an environment can result in rogue points, was not an issue in this project. This is because fairly quiet areas were scanned, and because the point clouds were first cleaned and subsampled before data visualisation was performed with them. However, if the results of this project were regenerated using point clouds of busier spaces with doors that are opened and closed at different moments, this could cause issues for both the classification of points, and the generation of visualisations
- That a sparse point cloud cannot represent 3D space well was also not of concern, because the mobile phone application that scanned the environments was effective enough at generating decent point clouds. If not enlarged too much, or if the points were given larger radii, then the point clouds could be conceived as representing solid surfaces. In this sense, there was little difference between how a point cloud portrayed the space, and how a mesh would have done so

The method for manual classification of ‘space-types’ was successful within the context of this project, and the classifications did not cause any issues at the data visualisation generation stage. However, if this project were to be extended to larger regions and larger point clouds that contain many separate spaces, then it will be necessary to find an appropriate method of (semi-)automatic classification. Otherwise it will simply be too time-consuming to generate such visualisations. Especially if in the future there were a situation with multiple users who want to see the isovists of each other’s real-time positions, and are continuously feeding new data into the model by means of scanning with their own mobile phones.

The `HPR` algorithm from Katz, Tal, and Basri (2007) that was used within *CloudCompare* to classify point clouds as either ‘visible’ or ‘occluded’ from a given position, was promising in terms of the quality of the output, but needs improving if one wants it to perform quicker visibility analysis on larger point clouds from more viewpoints. Though the paper states that visibility does not have to be re-calculated when changing the field of view, the function for creating a convex hull around the point cloud is called separately for each viewpoint. Creating a vectorisation function might make the algorithm more efficient in this regard. For the visualisations in this project, 3D data (the point clouds) is always being used, but Katz, Tal, and Basri (2007) consider a “ D -dimensional sphere”. The MATLAB (The Mathworks, Inc. 2022) code provided in the paper for the `HPR` algorithm therefore uses the `convhulln` function. However, according to MATLAB’s help centre (The MathWorks, Inc

2022), “for better algorithm efficiency with 2-D and 3-D input, consider using the convhull function.” This is another option for potentially decreasing the run-time of the algorithm when using it for 3D point cloud visualisations.

Writing a list of requirements helped to frame the problem at hand by demanding deliberation over needs, wants, and constraints. It was also useful for avoiding previously demonstrated pitfalls in the context of data visualisation. More literature could have been consulted, and the list further developed, to ensure a fool-proof design and building process, were it not for the constraint of time. It was deemed better to move on to the practical task of experimenting with different visualisation software and techniques, rather than spending too much time on the theoretical.

Moving on to the more user-oriented conclusions, this work shows that the integration of space classification with visibility analysis within a data visualisation, can be successful. Though the resulting visualisations show two different attributes within single views, the visualisations are not cluttered, and the two different pieces of data are visualised distinctly enough from one another, using different visualisation channels. However, making the visualisations dynamic in a way that means they will update in real-time with the slightest movement of the camera angle, should help a user in more quickly understanding for what they can use the visualisation, and how they should interact with it. This is because if live updates are fed to the user while they move around, they will immediately see how the shapes and visualised channels of the visible and occluded spaces are changing as a result of their own position changing. They should be able to understand more quickly that they are being shown their own isovist of the space, than someone who must first choose to generate an updated visualisation from their new position.

These visualisations provide users with more information about transitional spaces. Traditionally, the indoors and the outdoors have been kept very separate within the field of geomatics and beyond, but in an ever-complex world, the lines are blurring between these different spaces. Because of this, geovisualisations have a difficult time in effectively communicating different kinds of spaces around a user, and often opt for strict divides between outdoors and indoors, ignoring semi-indoor, semi-outdoor, and more spaces in the process. The visualisations presented here also help a user in understanding where they can traverse the space boundaries in their environment, and were more attributes such as accessibility to be visualised, could be of even more benefit to users.

Though the Dutch Golden Age painters were already busy in the 17th century with trying to incorporate as many different space-types as possible within their paintings’ static views of everyday life, modern digital depictions of spaces are still struggling to do so. It has been the intention of this research to draw attention to the scarcity of geovisualisations that integrate different types of spaces within their views, and also explicitly visualise occluded spaces, and respond by providing an exploratory look at ways in which this could be tackled.

Chapter 7

Future Work

Following on from the discussion of the results, and the conclusion, some recommendations can be made for if the work completed here, were to be taken further.

First of all, a software capable of generating (near) real-time updates for the visualisations as the camera position moves, would be invaluable. This would provide clarity for whether dynamic visualisations of the isovists help the user to more quickly understand what the visualisation is representing. Following on from this, user testing on target users, using both eye-tracking technology and qualitative assessments, would provide more definitive answers to the conjectures drawn in this work from using a list of requirements. By seeing how a potential user interacts with and understands the visualisations, they can be tweaked accordingly, and the most ‘successful’ visualisation design might become identifiable. Developing the [HPR](#) algorithm as detailed in the [General Conclusion](#), could be of interest, as well as exploring additional means of improving upon this work.

Also catering for multiple users who are feeding real-time data into the model while scanning their environment, is an exciting prospect for these visualisations. Were multiple users able to see everyone’s positions, as well as everyone’s respective isovists, such a visualisation could have a variety of applications. Firstly, the all-too-well experienced situation where individuals want to find one another within a foreign environment. Rather than calling one another on a mobile phone and verbally describing where someone is in relation to surrounding objects and buildings, a user immediately knows what the other person can and cannot see. This avoids unnecessary communication about features in the environment that are of no use to the other individual if they cannot see them. Secondly, and in much the same vein, in an emergency situation where multiple first responders to a scene are being directed by a central commanding unit, this unit would be able to better conceptualise where their team members are located, and know how to direct them to the necessary location. This remote leader could group together individuals to work on a particular task after seeing that they are in proximity to one another, even if they cannot see the other, themselves.

This project only dealt with three kinds of space boundaries: the boundaries between ‘outdoor,’ and ‘indoor,’ ‘semi-indoor,’ and ‘semi-outdoor’. Looking at how to visualise boundaries between ‘indoor’ and ‘semi-indoor,’ etc. while keeping the visual dissimilarities great enough, would be interesting. Also considering whether there are more space-type classifications, and using various data about spaces together with or instead of ‘space-type,’ would help with seeing how adaptable this sort of visualisation is to different information sources. Introducing more building levels than one, would also be an interesting challenge.

Effectively communicating the level and type of accessibility of different spaces in all walks of life (not just geovisualisation), is something that has been disregarded for far too long. It is a pressing issue that affects many members of the population, including those who are wheel-chair bound, who have impaired vision, or who need to transport a pram or even a suitcase. Visualising the accessibility of a

space would suit this type of visualisation well, as point cloud data can be used to make calculations such as the width of a doorway, or used for identification of steps by machine learning algorithms. Showing a space's accessibility features would complement the information about isovists and non-isovists.

Appendix A

List of Requirements

A summary of all the requirements:

1. User Experience

- (a) Visualisation must be designed with a **layman** in mind as the end user
- (b) Updating the visualisation view must involve minimal **latency**

2. Software

- (a) Must not demand too much **processing power** from device
- (b) Must be **spatially dynamic**: If the user changes the virtual camera position, the isovist updates accordingly to reflect the new view
- (c) (OPTIONAL EXTRA) Must be **temporally dynamic**: The visualisation can change with incoming, real-time data being collected

3. Intuitiveness

- (a) User must be able to quickly link physical environment features with digital twins
- (b) Positions of **boundaries** between different types of spaces must be easily identifiable
- (c) The following three different types of space boundaries must be clearly distinguishable from one another: outdoor-indoor, outdoor-semi-indoor, and outdoor-semi-outdoor
- (d) The '**non-isovist**' space must be visualised in a way that clearly distinguishes it from the isovist space
- (e) (OPTIONAL EXTRA) If time allows, visualisation should be able to incorporate multiple storeys into one view

4. Design

- (a) Must be visually **minimalistic** to avoid visual clutter
- (b) Must follow design best practises
- (c) Should not be too dependent on a single visualisation channel

A.1 Justification of Requirements: User Experience

A.1.1 Visualisation must be designed with a layman in mind as the end user

Why:

A layman, someone who does not have experience in the geovisualisation field, is the intended end user. Knowing this helps the design process focus on making the visualisation as intuitive as possible, so that non-veterans of the geovisualisation field can immediately understand the environment being represented, decreasing time until comprehension (good for time-sensitive situations). It is assumed however, that the user has previous experience in using computers and mobile phones to perform navigational tasks, e.g. they are acquainted with Google Maps.

A secondary user might also be an external commander of an operation if this thesis adjusts its focus towards such a situation, in which case different design choices will need to be taken.

Evaluation Metric:

The extent to which this requirement is fulfilled, will be evaluated by assessing visualisation designs against the remainder of this list of requirements, seeing how many requirements they fulfill (some of the following requirements directly relate back to designing with a layman user in mind).

An optional, additional method for evaluation, would be to perform user testing, which could be done if this project allows the time and resources for it.

Literature:

Not Applicable.

A.1.2 Updating the visualisation view must involve minimal latency

Why:

Users tend to lose their attention for a task when the latency (or loading time), becomes too long. However, users are more lenient about loading times on mobile phones than computers, as it is a commonly experienced phenomenon.

Evaluation Metric:

If there is too much latency in the visualisation loading on a standard computer, this requirement will be deemed “unfulfilled”. This can be quantified using the RAIL performance model developed by *Google Developers* (Google Developers 2020):

1. Response times to user actions ought to be between 0 and 100 milliseconds
2. Animations are perceived as smooth if “60 new frames are rendered every second”, a.k.a. 16 milliseconds per frame, but as browsers require 6 milliseconds to render a frame, a frame should be produced within 10 milliseconds
3. Idle: “Beyond 1000 milliseconds (1 second), users lose focus on the task they are performing”; “Beyond 10000 milliseconds (10 seconds), users are frustrated and are likely to abandon tasks. They may or may not come back later”
4. Loading time should be under 1000 milliseconds for it to “feel part of a natural and continuous progression of tasks”

Literature:

The RAIL performance model, which has been used for determining the evaluation metrics here, is based upon Nielsen (1993)’s work, which in turn references Miller (1968) and Card, Robertson, and Mackinlay (1991) as forerunners with very similar standards for latency.

For actions that take longer than 50 ms to complete, feedback ought to be provided (e.g. a loading sign), but a ‘percent-done’ indicator should be reserved for operations taking longer than 10 s (Myers

1985).

A.2 Justification of Requirements: Software

A.2.1 Must not demand too much processing power from device

Why:

So that the visualisation can be updated in (near) real-time without the need for computers with unusually large CPUs/GPUs; the intended user will likely be using a mobile phone or regular computer to access the visualisation.

Evaluation Metric:

If there is too much latency in the visualisation loading on a standard computer, this requirement will be deemed “unfulfilled”. This can be quantified using the RAIL performance model as done above for requirement [subsection A.1.2](#).

Literature:

The same as in [subsection A.1.2](#).

A.2.2 Must be spatially dynamic

Why:

A spatially dynamic visualisation is adaptable to a user’s specific requirements. In this case, that is the position from which the user wants to view a scene.

It is also fundamental to isovists, as their purpose is to show how a scene differs between viewpoints, including different viewpoints from the same location.

Evaluation Metric:

This can be quantified simply by whether or not the design allows for the changing of viewpoints, according to the user’s desired position.

If an isovist path has been animated, there should be a high enough number of outlook points where the isovists are calculated, so that the differences between views from adjacent points are minimal. However, this must be balanced with the speed at which the visualisation can load — the more points at which isovists must be calculated, the longer the loading time for the visualisation.

Literature

Not applicable.

A.2.3 (OPTIONAL EXTRA) Must be temporally dynamic

Why:

This makes the visualisation remain relevant for any moment in time, and would be useful for emergency situations, as well as locations where there is less previously existing data about the space available.

Evaluation Metric:

This would require the use of certain software, potentially (akin to) a game engine, which is not necessarily the focus of this thesis, hence this requirement being an optional extra.

There should be the perception of continuity between (nearby) viewpoints. User testing could include questions about how smooth the transitions between different viewpoints feel, and how natural the updates of scenes with incoming data, feels.

Objective quantification could include recording the latency of updating the views, using the RAIL performance model.

Literature:

Supporting literature is the same as in [section A.1](#), requirement 1B.

A.3 Justification of Requirements: Intuitiveness

A.3.1 User must be able to link physical environment features with digital twins

Why:

The visualisation’s primary aim is to aid a user in understanding their environment, and it is expected that the user will interact with the visualisation in a similar way to how they would with a map; comparing the visualisation with their real-life surroundings, using key features found in both, to link and line up the digital space with real-life.

Evaluation Metric:

User testing could include using eye-tracking software to analyse the level of transition entropy and stationary entropy:

→ A high “stationary gaze entropy” means that equal visual attention is given to different Areas of Interest, which is indicative of a navigational task being easier to perform.

→ A high “transition gaze entropy” means that the user frequently moves their gaze between Areas of Interest, indicative of “exploratory visual scanning.”

Literature:

“Low stationary gaze entropy is indicative of task difficulty” (Kapaj, Lanini-Maggi, and Fabrikant [2021](#)). A method for “quantifying eye movement transitions between areas of interest” is developed in Krejtz et al. ([2015](#)).

A.3.2 Boundary locations must be easily identifiable

Why:

Again, a main aim of this project is to help users in understanding the types of spaces surrounding them, and knowing what type of space they are about to enter. In order to enter a space, they must be aware of where the boundary lies, and how the entrance might be accessible to them.

Evaluation Metric:

Qualitative feedback from user testing while they are interacting with the visualisation could be of use. One could ask the user to describe their experience and what they are doing as they navigate the visualisation. Or ask questions such as “what do you see?”, “what type of space lies beyond the boundaries/doorways visible to you from your current position?” or “how would you get to a semi-indoor space from your current position?”, can provide insights into whether the user interprets the visualisation correctly. These open-ended questions should not be polar questions (yes-no) or leading questions (they should not encourage a particular kind of answer).

Literature:

“A qualitative feedback method can enhance quantitative findings” (Drey, Rietzler, and Rukzio [2021](#)).

A.3.3 Different space boundary types must be clearly distinguishable

Why:

This is a fundamental aspect of this research project, as indicated in the title.

Evaluation Metric:

If there are distinct differences in how these three boundaries are visualised, this requirement will be fulfilled. Ideally, there should be more than one way in which the different boundaries can be distinguished from one another, such that the user is not dependent on a single visualisation channel.

E.g. only using colour values (hue) to differentiate boundary types, could exclude some colour-blind users from optimal interactions.

Literature:

Not Applicable.

A.3.4 ‘Non-isovist’ space must be clearly distinguishable from isovist space**Why:**

As above, this is a fundamental aspect of the research project that is discussed in the Introduction. The reasoning for wanting to visualise the non-isovist spaces (as opposed to only visualising the isovist), stems from the aim of wishing to provide more valuable information to a viewer about the spaces beyond their lines of sight. This thesis focuses on visualising the boundaries between outdoor and other spaces, supplying a user with more tools for navigating their environment.

Evaluation Metric:

Data visualisation best practices can ascertain whether or not a great enough distinction has been made between two different attributes, and qualitative feedback from users would again be of value for determining how real people perceive the visualisations.

Literature:

Not Applicable.

A.4 Justification of Requirements: Design**A.4.1 Must be visually minimalistic to avoid visual clutter****Why:**

As Kraak and Ormeling (2021) point out, visualising unnecessary information can lead to information overload. The visualisations should strive to convey all the information required of them, but no more. Additionally, the methods by which the designs are visualised should be as minimalistic as possible, drawing upon the best practices of symbology usage.

Evaluation Metric:

This requirement can be judged by means of a comparative analysis between different visualisation concepts.

Literature:

Kraak and Ormeling (2021).

Appendix A

All Data Visualisation Designs

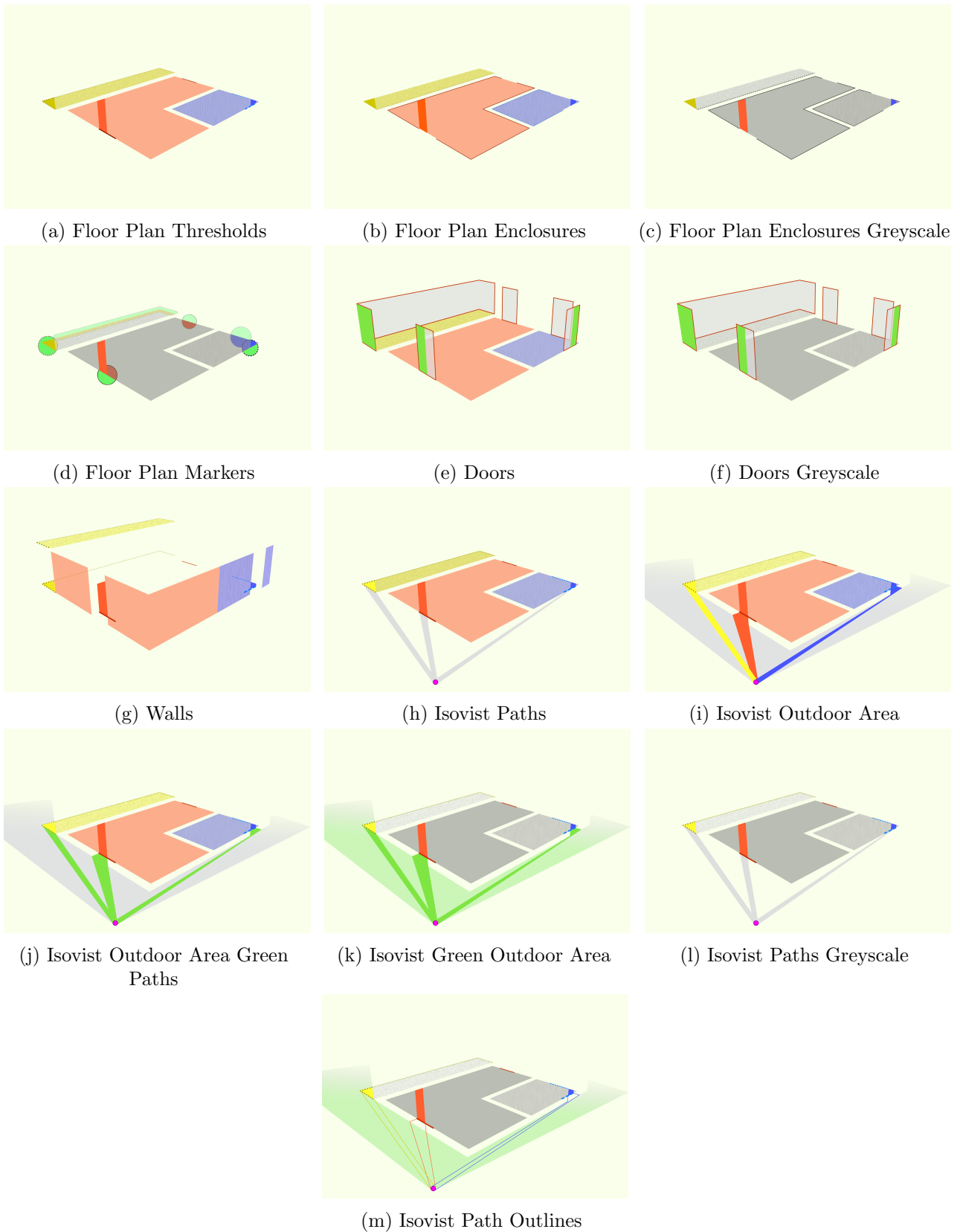


Figure A.1: Compilation of all data visualisation design concepts described in section 4.3

Bibliography

- Alattas, Abdullah. 2022. “The Integration of LADM and IndoorGML to Support the Indoor Navigation Based on the User Access Rights,” <https://doi.org/10.7480/abe.2022.05>.
- Bauer, Christina, Manuel Ullmann, and Bernd Ludwig. 2015. “Displaying Landmarks and the User’s Surroundings in Indoor Pedestrian Navigation Systems.” *Journal of Ambient Intelligence and Smart Environments* 7 (5): 635–657. <https://doi.org/10.3233/AIS-150335>.
- . 2016. “Evaluating Indoor Pedestrian Navigation Interfaces Using Mobile Eye Tracking.” *Spatial Cognition & Computation* (August). <https://doi.org/10.1080/13875868.2016.1219913>.
- Bertin, Jacques. 1967. *Semiology of Graphics*. 1st ed. Esri Press. ISBN: 1589482611.
- Brehmer, Matthew, and Tamara Munzner. 2013. “A Multi-Level Typology of Abstract Visualization Tasks.” *IEEE Transactions on Visualization and Computer Graphics* 19 (12): 2376–2385. <https://doi.org/10.1109/TVCG.2013.124>.
- Cantarero Navarro, Rubén, Ana Rubio Ruiz, Javier Dorado Chaparro, Felix J. Villanueva Molina, Maria J. Santofimia Romero, David Villa Alises, and Juan C. Lopez Lopez. 2021. “A Proposal for Modeling Indoor–Outdoor Spaces through IndoorGML, Open Location Code and OpenStreetMap.” *ISPRS International Journal of Geo-Information* 9 (3): 169. <https://doi.org/10.3390/ijgi9030169>.
- Card, Stuart. 2008. “Information Visualization,” 509–543. January. ISBN: 978-0-8058-5870-9.
- Card, Stuart, George Robertson, and Jock Mackinlay. 1991. “The Information Visualizer, an Information Workspace,” 181–186. January. <https://doi.org/10.1145/108844.108874>.
- CloudCompare. 2022. *CloudCompare*. www.cloudcompare.org.
- Couto, Marcelo, Cid Souza, and Pedro De Rezende. 2007. “An Exact and Efficient Algorithm for the Orthogonal Art Gallery Problem,” 87–94. November. ISBN: 978-0-7695-2996-7. <https://doi.org/10.1109/SIBGRAPI.2007.15>.
- Dalton, Ruth, and Nick Dalton. 2015. “The Problem of Representation of 3D Isovists.” The 10th Space Syntax Symposium (SSS10). July.
- Davis, Larry S., and Michael L. Benedikt. 1979. “Computational Models of Space: Isovists and Isovist Fields.” *Computer Graphics and Image Processing* 11 (1): 49–72. [https://doi.org/10.1016/0146-664X\(79\)90076-5](https://doi.org/10.1016/0146-664X(79)90076-5).
- Díaz-Vilariño, Lucia, L. González-deSantos, Edward Verbree, G. Michailidou, and Sisi Zlatanova. 2018. “From Point Clouds to 3D Isovists in Indoor Environments.” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-4 (September): 149–154. <https://doi.org/10.5194/isprs-archives-XLII-4-149-2018>.

- Drey, Tobias, Michael Rietzler, and Enrico Rukzio. 2021. "Questionnaires and Qualitative Feedback Methods to Measure User Experience in Mixed Reality." Yokohama, Japan, April. <https://doi.org/10.48550/arXiv.2104.06221>.
- ESRI R&D Center Zurich. 2018. *ArcGIS CityEngine*. V. 2018.1, September 18, 2018. www.esri.com/cityengine.
- Google Developers. 2020. "Measure performance with the RAIL model." Accessed May 5, 2022. <https://web.dev/rail/>.
- Kapaj, Armand, Sara Lanini-Maggi, and Sara Fabrikant. 2021. "The Influence of Landmark Visualization Style on Expert Wayfinders' Visual Attention during a Real-World Navigation Task." September. <https://doi.org/10.25436/E2NP44>.
- Katz, Sagi, Ayellet Tal, and Ronen Basri. 2007. "Direct Visibility of Point Sets," vol. 26. July. <https://doi.org/10.1145/1275808.1276407>.
- Khalid, Hammad, Emad Shihab, Meiyappan Nagappan, and Ahmed E. Hassan. 2014. "What Do Mobile App Users Complain About?" *IEEE Software* 32 (3): 70–77. <https://doi.org/10.1109/MS.2014.50>.
- Kim, Kyohyoun, and John Wilson. 2014. "Planning and Visualising 3D Routes for Indoor and Outdoor Spaces using CityEngine." *Journal of Spatial Science* 60 (November): 1–15. <https://doi.org/10.1080/14498596.2014.911126>.
- Kraak, Menno-Jan, and Ferjan Ormeling. 2021. *Cartography: Visualization of Spatial Data*. 4th ed. CRC Press. ISBN: 9780429464195.
- Krassanakis, Vassilios, Vasilis Mitropoulos, and Byron Nakos. 2013. "A cartographic approach of the process of map symbolization on gvSIG software." Valencia, Spain, January.
- Krejtz, Krzysztof, Andrew Duchowski, Tomasz Szmidt, Izabela Krejtz, Fernando Perilli, Ana Pires, Anna Vilaro, and Natalia Villalobos. 2015. "Gaze Transition Entropy." *ACM Transactions on Applied Perception* 13 (November): 1–20. <https://doi.org/10.1145/2834121>.
- Krukar, Jakub, Charu Manivannan, Mehul Bhatt, and Carl P. L. Schultz. 2020. "Embodied 3D isovists: A method to model the visual perception of space." *Environment and Planning B: Urban Analytics and City Science* 48:2307–2325. <https://doi.org/10.1177/2399808320974533>.
- Lanini-Maggi, Sara, Ian Ruginski, and Sara Fabrikant. 2021. "Improving Pedestrians' Spatial Learning During Landmark-Based Navigation with Auditory Emotional Cues and Narrative." September. <https://doi.org/10.25436/E26P43>.
- Liu, Zhenyu, Runnan Fu, Linjun Wang, Yuzhen Jin, Theodoros Papakostas, Xenia U. Mainelli, Robert Voûte, and Edward Verbree. 2022. "Game Engine-based Point Cloud Visualization and Perception for Situation Awareness of Crisis Indoor Environments," <https://doi.org/10.34726/1786>.
- Lv, Chenlei, Weisi Lin, and Baoquan Zhao. 2022. "Voxel Structure-Based Mesh Reconstruction From a 3D Point Cloud." *IEEE Transactions on Multimedia* 24:1815–1829. <https://doi.org/10.1109/TMM.2021.3073265>.
- Makri, Antigoni, Sisi Zlatanova, and Edward Verbree. 2015. "An Approach for Indoor Wayfinding Replicating Main Principles of an Outdoor Navigation System for Cyclists." *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XL-4/W5 (May): 29–35. <https://doi.org/10.5194/isprsarchives-XL-4-W5-29-2015>.
- McElhinney, Sam. 2022. *Isovist*. Accessed July 1, 2022. www.isovists.org.

- Miller, Robert B. 1968. "Response Time in Man-Computer Conversational Transactions," 33:267–277. Los Alamitos, CA, USA: IEEE Computer Society, December. <https://doi.org/10.1145/1476589.1476628>. <https://doi.ieeecomputersociety.org/10.1109/AFIPS.1968.149>.
- Munzner, Tamara. 2009. "A Nested Model for Visualization Design and Validation." *IEEE Transactions on Visualization and Computer Graphics* 15 (6): 921–928. <https://doi.org/10.1109/TVCG.2009.111>
- . 2015. *Visualization Analysis and Design*. 1st ed. A K Peters/CRC Press. ISBN: 9781466508910.
- Myers, Brad A. 1985. "The Importance of Percent-Done Progress Indicators for Computer-Human Interfaces." *ACM SIGCHI Bulletin* 16 (April): 11–17. <https://doi.org/10.1145/317456.317459>.
- Nakamura, Jun P. 2020. "Seeing Outside the Box: Reexamining the Top of Samuel van Hoogstraten's London Perspective Box." *Journal of Historians of Netherlandish Art* 12 (2). <https://doi.org/10.5092/jhna.12.2.3>.
- Nebiker, Stephan, Susanne Bleisch, and Martin Christen. 2010. "Rich Point Clouds in Virtual Globes – A New Paradigm in City Modeling?" *GeoVisualization and the Digital City, Computers, Environment and Urban Systems* 34 (6): 508–517. ISSN: 0198-9715. <https://doi.org/https://doi.org/10.1016/j.compenvurbsys.2010.05.002>.
- Nielsen, Jakob. 1993. "Usability Heuristics." In *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. ISBN: 0125184069.
- Nikander, Jussi, Juha Järvi, Muhammad Usman, and Kirsi Virrantaus. 2013. "Indoor and Outdoor Mobile Navigation by Using a Combination of Floor Plans and Street Maps." In *Progress in Location-Based Services*, edited by Jukka M. Krisp, 233–249. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-34203-5. https://doi.org/10.1007/978-3-642-34203-5_13.
- OGC. 2019. "IndoorGML: OGC Standard for Indoor Spatial Information." Accessed January 16, 2022. www.indoorgml.net.
- OpenStreetMap contributors. 2022. *OpenStreetMap*. www.openstreetmap.org.
- Peters, Ravi, Hugo Ledoux, and Filip Biljecki. 2015. "Visibility Analysis in a Point Cloud Based on the Medial Axis Transform." In *Eurographics Workshop on Urban Data Modelling and Visualisation*, edited by Filip Biljecki and Vincent Tourre. The Eurographics Association. ISBN: 978-3-905674-80-4. <https://doi.org/10.2312/udmv.20151342>.
- Petrasova, Anna, Brendan Harmon, Vaclav Petras, and Helena Mitasova. 2015. "Viewshed Analysis," 77–82. December. ISBN: 978-3-319-25773-0. https://doi.org/10.1007/978-3-319-25775-4_6.
- Puikkonen, Arto, Ari-Heikki Sarjanoja, Merja Haveri, Jussi Huhtala, and Jonna Häkkinen. 2009. "Towards Designing Better Maps for Indoor Navigation: Experiences from a Case Study." In *Proceedings of the 8th International Conference on Mobile and Ubiquitous Multimedia*. MUM '09. Cambridge, United Kingdom: Association for Computing Machinery, January. ISBN: 9781605588469. <https://doi.org/10.1145/1658550.1658566>.
- Robert McNeel & Associates. 2022. *Rhinoceros*. V. 7.14, January 10, 2022. www.rhino3d.com.
- Rutten, David. 2022. *Grasshopper*. V. Rhino: 7.14, January 10, 2022. www.grasshopper3d.com.
- Smit, Mels. 2022. "Deducing the Location of Glass Windows in 3D Indoor Environments," www.resolver.tudelft.nl/uuid:b1e20016-5e1c-4610-a951-6631983b7511.
- Stanford University Computer Graphics Laboratory. 1994. *Stanford Bunny*. Accessed July 6, 2022. <http://www.graphics.stanford.edu/data/3Dscanrep/>.

- The MathWorks, Inc. 2022. “convhulln.” Accessed July 30, 2022. www.mathworks.com/help/matlab/ref/convhulln.html.
- The Mathworks, Inc. 2022. *MATLAB: 2022a*. Natick, Massachusetts: The Mathworks, Inc.
- Toolbox AI, Inc. 2021. *Scaniverse*. www.scaniverse.com.
- Unity Technologies. 2021. *Unity*. V. 2021.2.1, November 5, 2021. www.unity.com.
- Xiong, Xuehan, Antonio Adan, Burcu Akinci, and Daniel Huber. 2013. “Automatic Creation of Semantically Rich 3D Building Models from Laser Scanner Data.” *Automation in Construction* 31 (May): 325–337. ISSN: 0926-5805. <https://doi.org/10.1016/j.autcon.2012.10.006>.
- Yan, Jinjin, Abdoulaye A. Diakité, and Sisi Zlatanova. 2019. “A generic space definition framework to support seamless indoor/outdoor navigation systems.” *Transactions in GIS* 23 (6): 1273–1295. <https://doi.org/10.1111/tgis.12574>.
- Yan, Jinjin, Sisi Zlatanova, and Abdoulaye A. Diakité. 2021. “A unified 3D space-based navigation model for seamless navigation in indoor and outdoor.” *International Journal of Digital Earth* 14 (8): 985–1003. <https://doi.org/10.1080/17538947.2021.1913522>.
- Zlatanova, Sisi, Liu Liu, George Sithole, Junqiao Zhao, and Filippo Mortari. 2014. “Space subdivision for indoor applications.” *Transactions in GIS*, <https://doi.org/10.13140/2.1.2914.2081>.