MSc thesis in Geomatics for the Built Environment

Exploring a pure landmark-based approach for indoor localisation

Oscar Timothy Willems

2017

TU Delft

EXPLORING A PURE LANDMARK-BASED APPROACH FOR INDOOR LOCALISATION

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

Master of Science in Geomatics for the Built Environment

by

Oscar Timothy Willems

November 2017

The work in this thesis was made in the:

3D geoinformation group
Department of Urbanism
Faculty of Architecture & the Built Environment
Delft University of Technology

Supervisors:	Prof.dr. Sisi Zlatanova (TU Delft - Department of Urbanism)
	Ir. Edward Verbree (TU Delft - Department of OTB)
Co-reader:	Dr. Pirouz Nourian (TU Delft - Department of Design Informatics)

ABSTRACT

Humans interact more and more with their environment through technology, recent decades have seen a huge increase in the need for and availability Location-Based Services (LBS). Recently landmarks have gotten a renewed interest in the field of LBS) , although already a quite old phenomenon. In both the outdoor and indoor environment they are being used to enrich existing services such as navigation, but not used as the basis for a technique or service.

The indoor environment relies heavily on building specific and less-scalable sensor-based localisation techniques (such as Wi-Fi and Bluetooth), alternatives to sensor-based are becoming a necessity and would be a welcome addition. The exploration and development of landmark-based approaches for indoor localisation is something that can extend the field of geomatics and LBS. This research investigates if a pure landmark-based approach works for indoor localisation and which characteristics of landmarks can be exploited. This is achieved by developing a conceptual framework that explores how a landmark-based indoor localisation would work from an artificial point of view. A Minimal Viable Product (MPV) is implemented to evaluate if a landmark-based approach works and what needs to be improved or considered in future studies

Starting from an artificial test case, the MPV to achieve indoor localisation is implementing and evaluated using a manually digitised real-world and more complex test case. The fundamental principle of landmark-based localisation is that through the observation of landmarks within the (indoor) environment a user's location is obtained because the visibility and location of landmarks are known. The workflow to go from an observation to a location is by 1) calculating the visibility/isovist area of each landmark, 2) interpret the observations into a combination of landmarks, 3) intersect the visibility of all landmarks in the observation, 4) refine the location based on relative landmark constellations, and 5) follow-up with questions on potentially visible landmarks to improve location further

One of the key giveaways of this research is that approach for indoor localisation a landmark-based is feasible, principles and techniques exist (or are being developed), it is only a matter of setting them up in the right order and format them to work, and connect input with the researched process and use them for LBS driven applications. Future work on the subject of landmark-based localisation and LBS is connecting with existing spatial standards, extend the principles into the 3rd dimension, and integrate more aspects of landmark salience.

Keywords: Landmarks, indoor environment, location-based services, localisation, SQL, PostGIS.

ACKNOWLEDGEMENTS

I would first like to thank my thesis supervisor Dr Sisi Zlatanova of the 3D Geoinformation Department at the Faculty of Architecture and the Built Environment of the Delft University of Technology. The door to Sisi Zlatanova office (or rather Skype connection) was always open whenever I ran into a trouble spot or had a question about my research or writing. She consistently allowed this paper to be my work but steered me in the right the direction whenever she thought I needed it.

I would also like to acknowledge Ir. Edward Verbree of OTB Department (Research for the built environment) at the Faculty of Architecture and the Built Environment of Delft University of Technology as the second supervisor and reader of this thesis, and I am gratefully indebted to him for our heated discussions and his very valuable comments on this thesis.

Also, I would like to credit Dr.ir. Pirouz Nourian of Design Informatics Department at the Faculty of Architecture and the Built Environment of the Delft University of Technology for his challenging but valuable comments as co-reader of this thesis.

Finally, I must express my very profound gratitude to my parents, family and my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Author *Oscar Timothy Willems*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACRONYMS

AOA Angle of Arrival

BIM Building Information Model

DEM Digital Elevation Model

BK CASE Faculty of Architecture (Bouwkunde) case

GIS geographical information system

GNSS Global Navigation Satellite System

GPS Global Positioning System

LBS Location-Based Services

MPV Minimal Viable Product

MPVS Minimal Viable Products

NFC Near Field Communication

PLPGSQL PostgreSQL's Procedural Language

POI Point-of-Interest

QGIS Quantum GIS

RFID Radio-Frequency IDentification

RSS Received Signal Strength

RTOF Round-trip Time of Flight

UML Unified Modeling Language

1 | INTRODUCTION

Landmarks have gained renewed interest within the field of Location-Based Services (LBS) both in the outdoor and indoor environment. Existing localisation and navigation services are extending to incorporate landmarks to improve the usability of the technology and increase the interaction of users with the context (rather than just staring at a dot on a screen and follow instructions). (Graser [2017], Ramirez et al. [2012]) are examples of researchers that try to incorporate landmarks into the LBS, notably each with their definition of what a landmark is – points of interest (Point-of-Interest (POI)) in OpenStreetMap, a monument along a route or an object recognisable for firefighters.

Continuing on the path of renewed interest in landmarks, finding applications and fundamental principles could be the next leap forward in extending the field of indoor (LBS), by asking questions such as how influential a role can landmarks play in the indoor environment, and could landmarks extend existing techniques and principles?

1.1 BACKGROUND

In ancient times before (Global Positioning System (GPS)) and (Global Navigation Satellite System (GNSS)), sailors used their relative position to the stars to determine locations on open seas and set a course. Roman strategists drew maps with context to give a sense of location and bearing to know what is north and south, left and right, and how to march toward and across the battlefield. In the ages before smartphones and navigation systems, humans navigated based on context, where relative positions (in front of, next to, behind, above) were linked to identifiable objects, known as landmarks (church, the baker, a park, etc.), to determine where one was within the context, and then navigation based on instructions that only used these objects/landmarks, see ‘A brief history of navigation’ by Sobel [1998]. A precise location in Cartesian space (x, y, z) was after all impossible due to the lack of positioning systems and paper maps, the best approximation of location within space was ”in proximity of a certain object (i.e. church)” or ”roughly x amount of miles/hours away from the closest landmass”. See Figure 1.1 for the difference between location and position.

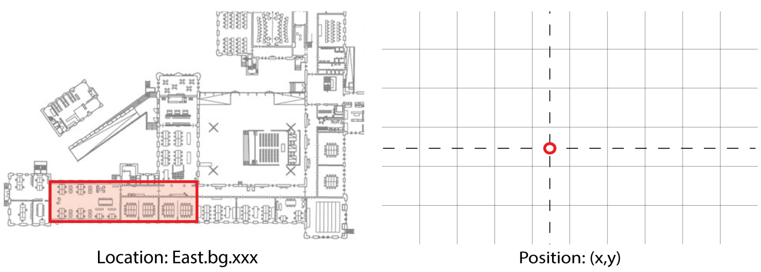


Figure 1.1: An example of a location within Faculty of Architecture (TU Delft) and a position within a coordinate reference system (CRS).

With the emergence of localisation methodology (i.e. the example of one’s location relative to an object such as a church) and positioning technology (i.e. lati-

Figure 1.2: Example of a lack of signal in a (unknown) public space.

tude, longitude coordinates of a user/object) the way of interacting with the context change, humans have gone global, explore unfamiliar areas through technology. Nowadays, humans depend a lot on (smartphone) applications that automate processes to save time and effort. Localisation and navigation tools are among the most used tools when leaving the house to go somewhere and to get around, but due to the reliance on these services humans are starting to lose their navigational skills making them rely even more on navigation technology according to McKinlay [2016].

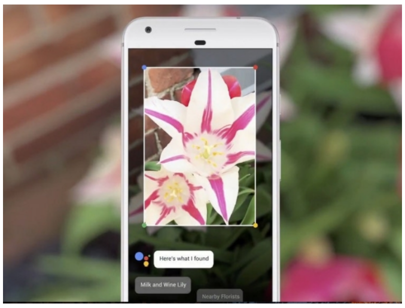
When talking about the indoor environment there are actually two 'indoor' environments: the physically enclosed indoor environment within buildings/structures (i.e. shopping malls, museums and the underground metro) and the outdoor environment where existing GNSS systems do not function adequately (i.e. close to buildings where GNSS signals are blocked or delayed by multipath effects, or inside buildings) such as is described in the research into indoor GPS signal strength by Peterson et al. [1997].

Real-time positioning services are essential for LBS since these are becoming more ingrained in human society, such as navigation services and information services (Hightower and Borriello [2001], Pahlavan et al. [2002] and Muthukrishnan et al. [2005]). The services started outdoors where the needs of military, professionals and consumers were greatest. The technology was developed to be sufficient for the outdoor environment, but now these services are also expected to be always available and reliable both indoors and outdoors, day and night. Some of these environments are not compatible with existing technology, i.e. the challenge of localisation in the 'indoor environment', where GPS/GLONASS/etc. positioning systems do not function adequately or are not precise enough, and navigation becomes quite the challenge. There is an increase in public areas that are considered outdoors (out of the home and office) but are technically indoors, i.e. shopping malls, underground infrastructure: public buildings with a roof. Performing outdoor activities in these indoor environments can pose a challenge to GPS signals to for example navigation, gaming and GPS-based fitness, see Figure 1.2.

1.2 PROBLEM STATEMENT

The gap between localisation and navigation in indoor compared to outdoor environments has resulted in number of techniques that bridge the gap: applications that use triangulation, trilateration of users based on Bluetooth, Wi-Fi, or other sensor beacon approaches (comparable to the GPS) but on a more local level, Chowaw-Liebman et al. [2010]), the fingerprinting of the (static) indoor environment based on signal strengths of (Wi-Fi) beacons (Xiao et al. [2011] and Chan and Sohn [2012]), or cameras and image (QR-codes) recognition to localise users within the indoor environment (Vo and De [2016]). Many of these techniques are applied to a select set of indoor environments since these are techniques that are costly and engineered specifically for a location.

A challenge in the field of localisation and navigation is to find a method that doesn't depend on expensive indoor sensor networks, and that is not location specific, i.e. by using commonly available features of the environment and exploiting existing human/technological capabilities. With the emergence of new technologies that range from being able to recognise features in the surrounding environment (i.e. Google Lens, Conditt [2017]), to optic sensors in autonomously driving cars that provide real-time object tracking and recognition (i.e. Tesla's autopilot), to added reality layers with augmented reality technology (Google Glass), see Figure 1.3. It is now possible to use previously impossible techniques in daily challenges. By using advanced image processing and recognition, there are new ways to conquer the urban (indoor) environment, and specifically our mode of interaction and navigation within this environment.



(a) Google Lens technology, source: Google

(b) Tesla's Autopilot with sensors, source: Tesla

(c) Google Glass with augmented reality, source: Google

Figure 1.3: Technological advances for interaction with context.

In the case of being lost within the 'indoor' environment due to a lack of GNSS signals, a lack of pre-engineered GNSS alternatives such as Wi-Fi, Bluetooth, or RFID beacons, positioning oneself within an unfamiliar environment becomes near impossible. What if you could use what you see within that same unfamiliar environment to localise yourself and then navigate to where you are heading? With the demand as mentioned earlier for LBS within the indoor environment, exploring new techniques and principles that exploit that what is present in the environment. This idea is novel in the indoor LBS, using just landmarks to localise and position oneself, and from thereon navigate the environment using the same landmarks.

1.3 RESEARCH QUESTION

To address the challenge of indoor localisation (and navigation) using the existing context, rather than rely on an engineered system of beacons, the main research question for this thesis is:

How can a pure landmark-based approach achieve adequate indoor localisation?

To answer the main research question, the capabilities of the landmark-based approach are investigated, by developing a proof of concept for landmark-based indoor localisation:

1. *Within the indoor environment what can be considered a landmark for localisation (and navigation) and how does object salience plays its part?*
2. *Which landmark parameters are most salient for indoor localisation?*
3. *How can a landmark be semantically and geometrically modelled in a database schema? And what is this schema's influence on the visibility calculation?*
4. *Which localisation principle shows potential for a pure landmark-based approach?*
5. *Can the relative position of a subject in relation to landmarks be used to improve localisation?*
6. *Does the number, and constellation of landmarks influence accurate localisation?*

1.4 RESEARCH RELEVANCE

Within the field of localisation and positioning there is already much work done in positioning centred around high precision systems (GNSS but also beacons such as Wi-Fi, Bluetooth and Near Field Communication (NFC)) due to the availability and market of these systems, however, the field of contextual localisation and navigation has not seen as much interest, both for the outdoor and the indoor environment. The indoor and often open-space environment have their own added challenges. Solving (or the attempt of solving) the challenge of (open-space) indoor localisation and navigation based on context without a network of sensors that is engineered specifically for a location, could be the next leap for providing the same services indoor that are already commonly relied upon outdoors. Renowned researchers Stephen Winter and Kai-Florian Richter recently (2017) emphasised the need for indoor localisation and navigation independent of sensor-based technologies, finding new ways using different concepts and a using state of the art imaging techniques is required to surpass the current roadblock with sensor-based technologies Winter et al. [2017].

Landmarks have been incorporated into existing techniques and principles to generate additional value. However, they have not been purely used in either localisation or navigational principles. Exploring a low-cost generic framework for indoor localisation by combining outdoor principles with the indoor context using a pure landmark-based approach is, therefore, a novelty and has the potential to contribute to the field of LBS.

1.5 OBJECTIVES AND SCOPE

The objective is to explore the possibilities of a landmark-based indoor localisation, this is step one from where landmark-based indoor location-based services can grow, and that can eventually blossom into indoor navigation, LBS, however, need a location first. Therefore, this research focusses on developing a proof of concept for indoor localisation based on landmarks and writing a future work recommendation for landmark-based location-based services, and specifically navigation. The research has two (artificial) test cases that represent a real-world case in all its possibilities and variations. The starting point is a case with landmarks within the indoor environment, in different constellations, where there is a floor plan of the indoor environment with obstacles. This research will not delve into the data acquisition and the abstraction of the real world into a spatial standard (CityGML, IndoorGML, Building Information Model (BIM), etc.). This research is also not about finding out which spatial standard is best suitable for a landmark-based approach for indoor localisation. This research focuses on trying to find a way to localise a user based on landmarks. The modelling of the interaction between human and computer is only limited within the thesis. This research considers that the human/device can communicate with the system in a one-on-one interaction about the visibility of recognisable objects (the user and system communicate with the same object identifiers). The lack of sensors and system means that a precise measure of angles or distances is not possible. The input observations are conceptualised and explored to fit the principle of landmark-based indoor localisation. This is research into landmarks, and how they can be made to work in existing indoor localisation principles or if a new principles have to be conceptualised, Figure 1.4 shows how abstraction, indoor localisation and positioning, and location-based services (including navigation) are tied together. The focus of this research is on solving the challenge of indoor localisation, to be the first step for landmark-based navigation.

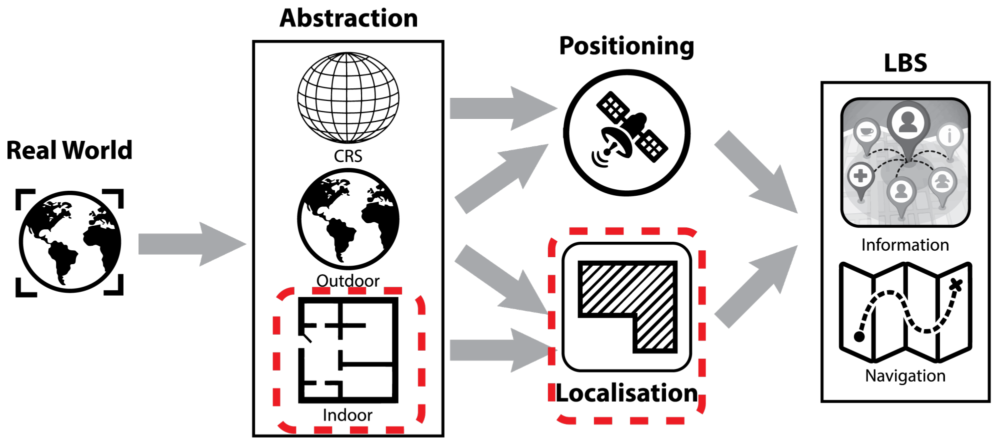


Figure 1.4: The scope of the research: focus on indoor localisation.

1.6 READING GUIDE

In the next chapters, this thesis delves into the related works (Chapter 2), that discusses the indoor environment, landmarks, localisation and navigation. Then the research approach, methodology (Chapter 3, and conceptual framework (Chapter 4) are explained showing the steps into solving the challenge of indoor localisation. Then the implementation and results of the conceptual framework show how and what is done to prove that the conceptual framework works (Chapter 5 and Chapter 6). So finally, the research questions of this thesis are answered in conclusion along with the scientific discussion and recommendations for future work (Chapter 7).

2 RELATED WORK

This chapter provides insight in prior research into landmarks (Section 2.1), the indoor environment (Section 2.3), (indoor) localisation and positioning (Sections 2.4, 2.5), and (indoor) navigation (Sections 2.6, 2.7).

2.1 LANDMARKS

For context-aware application, there has to be a definition of objects that make up the context, in history and in literature a commonly accepted definition of a landmark is an object that visibly stand out from the surroundings, in the early days of sailing it could be celestial constellations, lighthouses, and land features, see Sobel [1998]. Roman generals would draw maps with features that stood out within the landscape. Tourists get around cities by following skyline outlining structures, i.e. the Eiffel Tower in Paris, or the Big Ben in London. Classically landmarks are significant structures that are highly visible.

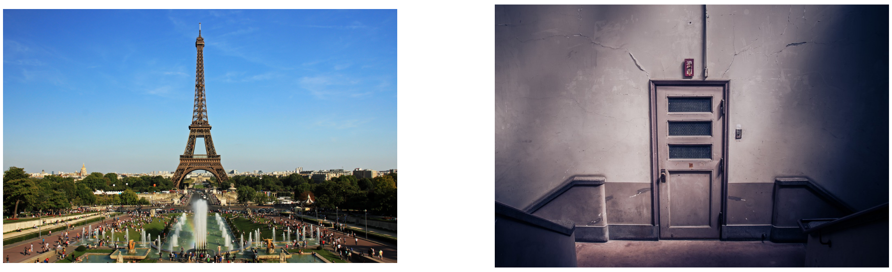
Golledge [1999] determines two critical roles of landmarks for wayfinding: 1) helping in organising space by providing categories to pinpoint objects within the fabric of to the environment, and 2) serve as objects within navigation where a decision can be made by functioning as directional cues and wayfinding clues. In this sense, landmarks can be anything that helps a subject in wayfinding, from the tallest Eiffel Tower to the smallest pebble on the road (if it is a bright colour and stands out from other pebbles).

Kai-Florian Richter highlights the potential of landmarks after comparing a multitude of approaches using landmarks, he establishes that landmarks play an important part in the visual communication of (indoor) space, and sees the emerge of user-generated information (such as OpenStreetMap) and services that work through images, like Flickr, as stepping stones for a commercial use of landmarks Richter [2013].

Landmarks can come in all sizes, shapes, and appearances: Ramirez et al. [2012] talked about heat images from a thermal scanner to orient oneself in the building, Anacta et al. [2017] describes landmarks as recognisable places that humans can (abstractly) draw when having to explain the context on a hand-drawn map to a stranger, Vasardani et al. [2017] labels landmarks within an indoor environment as *"to either not belong to the set of usual building structural features and furnishings, or if they do, then their properties (and perhaps functions) deviate from the prototypical ones of the set"*, Presson and Montello [1988] determines the minimal definition of an (elusive) landmark as any distinct object that is noticed and remembered, for an example see Figure 2.1.

2.1.1 The salience of landmarks

Stankiewicz and Kalia [2007] defines the visibility and recognisability of objects as saliency. Caduff and Timpf [2008] expressed saliency as three-valued vector, consisting of perceptual salience, the exogenous potential of an object to attract visual attention, cognitive salience, how well an object supports the orientation of a user when interacting with the context, and contextual salience, to what extent an object can be used for tasks to interact with the environment. Klippel and Winter



(a) Eiffel Tower source: WikiCommons (b) A distinguishable door source: Pixabay

Figure 2.1: Two objects that could be labelled as landmarks.

Table 2.1: Kattenbeck's assessment of landmark salience Kattenbeck [2017]

Construct	k	Example
Overall salience	3	To what extent does this object draw your attention?
Visual salience	15	Intensity, tone, size, location
Cognitive salience	6	To what extent is the current use of the object obvious?
Structural salience	4	How easy is it for you to refer to this object in a route description?
Visibility in Advance	4	To what extent can one easily refer to this object from afar?
Prototypicality	3	To what extent does this object represent your impression of such objects?

[2005] complements salience with structural salience that determines how easily landmarks are conceptualised cognitively or linguistically in wayfinding or localisation.

Markus Kattenbeck empirically measured the salience of objects, by developing a survey to assess the salience of a landmark. Kattenbeck's definition of salience consists of five parts: visual salience (what visual features does an object has), prototypicality (to what extent represents an object its original conception), the visibility in advance (how identifiable is it when approached), structural salience (the ease it may be incorporated in a route description), and the cognitive salience (to what degree it can evoke prior knowledge about the object) Kattenbeck [2017]. To test these levels of salience Kattenbeck developed a survey that assesses a landmark's salience according to Table 2.1.

2.1.2 The use of landmarks

This subsection discusses several researchers that are working on implementing landmarks in existing technologies. They provide theoretical frameworks as well as practical applications for landmarks.

Raubal and Winter [2002] and Klippel and Winter [2005] in collaboration with Stephen Winter of The University of Melbourne have extended the *'wayfinding choreme theory'*, a formalised theory for wayfinding to also integrate landmarks into route direction with the formalisation of *'the salience of landmarks as dominant objects in route knowledge and route directions'* as the recognition of the added value of landmarks within wayfinding and *'the conceptualization of wayfinding actions in relation to landmarks, i.e., the integration of landmarks in the formal specification of a conceptual route language'*, which describes the role of landmarks within instructions. Caduff and Timpf [2008] agrees with this vision by stating that landmarks are the principal facilitators for the interaction between user and context.

Research from Graser [2017] into landmarks resulted in an exciting addition to traditional network-based (car) navigation, where landmarks were used to add context to a route, and have directional clues not just on XY location and praxic instructions (i.e. turn left in 100 metres), but also have taxonomic instructions that use the visibility of landmarks within the context. Graser as one of the key figures behind open source GIS applications (OSGeo's QGIS) used objects stored in OpenStreetMap as landmarks and based on their characteristics enrich navigational instructions, to make users more aware of their environment when navigation (rather than staring at a smartphone screen when the 100 metres have passed).

Researchers, led by Ramirez created the ad-hoc application called Landmarke that provides firefighters with a navigational support system, which ties wayfinding and orientation within the context together by using landmarks to help in getting around Ramirez et al. [2012]. Orientation within the environment was achieved by using, for example, thermal cameras to be able to locate hot surfaces, have firefighters recall object that stood out of the context to determine where they had last seen their colleagues. Landmarks were used to determine the position based on their orientation in regards to the fire-fighters.

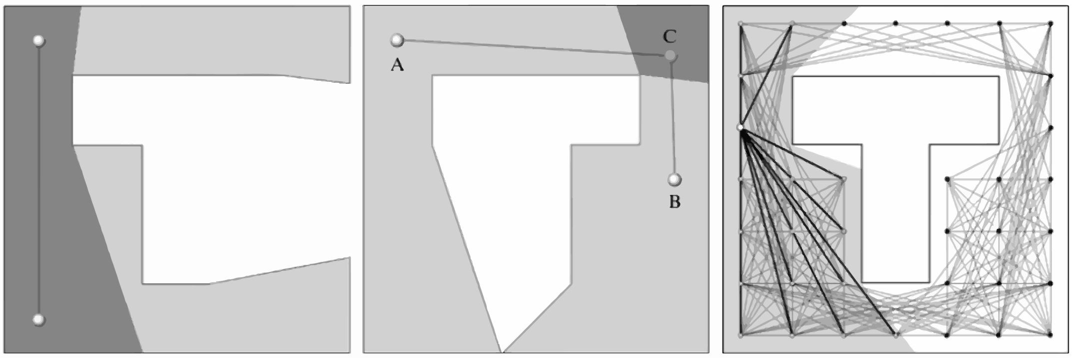
Brunner-Friedrich and Radoczky [2005], Michon and Denis [2001] note that landmarks play a vital role in orientation and reorientation within the (urban) environment, by being able to describe one's location in relation to landmarks. Stankiewicz and Kalia [2007] then continues to state that when landmarks are used in navigational instructions, users can remember them better, which helps with the orientation within the context.

All research into landmarks uses the salience, and specifically the visibility, of landmarks in the generation of instructions or to familiarise the user with its context. Most of the research use landmarks as an added element to an existing technique, to improve the familiarity of the environment and to provide added value.

2.2 VISIBILITY AND VIEWSHED ANALYSIS

Visibility is considered an important aspect for the placement of a cultural object in the urban landscape. Visibility can be from the perspective of a user with a Boolean visibility that tells if the user can see an object/location (clear line-of-sight), or from the perspective of an object/location that tells in which region the object/location is visible, either Boolean or a fuzzy visibility (which areas see part of an object/location or the whole object/location). Traditionally viewshed analysis was used on Digital Elevation Model (DEM) to find out if two locations have a clear line of sight (no visibility blocking elevation) Kim et al. [2004], viewshed analysis tools are available in most geographical information system (GIS) software. An isovist (first coined by Tandy [1967]), or viewshed, is the area in a spatial environment directly visible from a location within the space. Isovist analysis and isovist fields use the technique of ray-casting, by casting a ray at set angles or to the boundaries of surrounding objects a field of visibility is created Benedikt [1979], see Figure 2.2 for an example of isovist analysis. Turner et al. [2001] extended isovist and isovist fields with visibility graphs, as shown in Figure 2.3, that show the first order visibility (A) where two locations are both visible from one another, second order visibility (B) as an area from where A and B are both visible but from A you cannot see B, and the final visibility graph of which locations are visible from one another. Later Turner and Penn [1999] continued on the path of making isovists syntactic and Batty [2001] is an example of a researcher that used isovist analysis for analysis of the urban morphology.

Figure 2.2: Isovist analysis on T-shape from different locations Batty [2001].



(a) First-order visibility relationships between isovists.

(b) Second-order visibility relationships between isovists.

(c) Example of a first-order visibility graph, showing the pattern of connections for a simple configuration.

Figure 2.3: The second-order graph is just a 'flattened' first-order graph: A is linked directly to B, rather than through C as it would be in the first-order case Turner and Penn [1999].

2.3 INDOOR ENVIRONMENT WITH OBJECTS AND OPEN SPACES

Peterson et al. [1997] defines the indoor environment either as physically enclosed buildings and structures or as areas where GNSS services (GPS/GLONASS) fail to provide an adequate location, when there is no signal at all for a location or if the position is not accurate enough (position tells you are outside while you are inside, or if the position could be hundreds of meters off). The field of robotics and automated mapping have a clear definition of (indoor) open-space environments within navigation and mapping. Yamauchi's research in the automatic mapping and navigation of robots distinguishes between open-space environments and unexplored space. Robots perceive the open-space environment as the connectivity or arrangement of open regions with accurately mapped obstacles (usually these obstacles are mapped in advance by humans) Yamauchi [1997]. This perception of an open-space environment within robotics research combined with common understanding of open-space result in the following definition: *open-space is defined as a (series of) bounded/enclosed geometric open region(s), optionally containing obstacles.*

For the indoor environment, there are spatial standards that outline the abstraction and storage of (indoor) objects, such as CityGML's Level of Detail 4 (LOD4), the storage of spaces and connected spaces in IndoorGML for navigation, or BIM from the field of (architectural) engineering. These models have in common a formalisation of describing objects and spaces, but each has a specific purpose, some more appropriate for location-based services some more for storing information for contractors and city planners, Kim et al. [2014] proposed an integration of CityGML

LOD4 and IndoorGML for an optimal use of indoor spaces. Figure 2.4 shows the representation of space in CityGML LOD4 and IndoorGML.

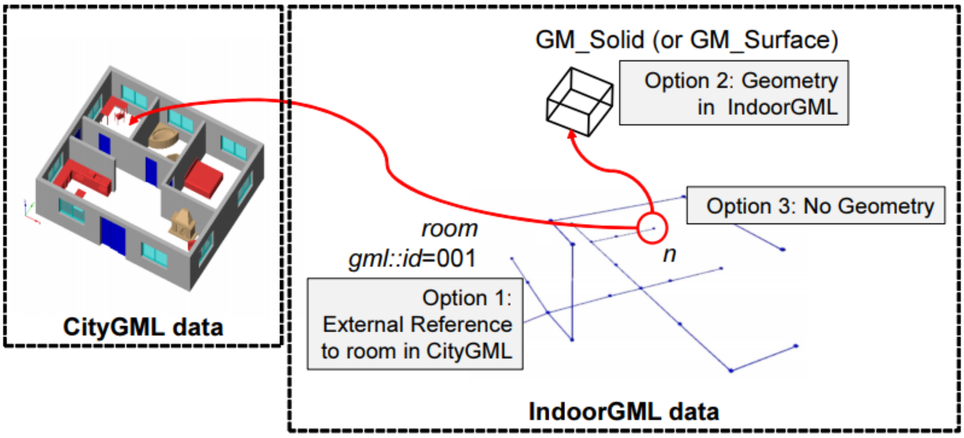


Figure 2.4: Cell representation (space) in CityGML LOD4 and IndoorGML.

2.3.1 Objects within the indoor environment

The indoor environment contains a mixture of objects, ranging from the core elements of a building (walls, doors, windows and floors), to objects that temporarily occupy space (like cars, pop-up stands, furniture), to objects that are constantly moving. Kummerle et al. [2009], Groh et al. [2014] and Ibisch et al. [2013] distinguish these environments as static and semi-static environments with possibly moving objects. The static environment is an environment where the only objects are static objects (objects set in stone) and with moving objects. Semi-static environments are environments where beside static objects semi-static objects occupy space, objects that might remain for longer periods of time (such as cars or pop-up stands), keeping track of these objects (mapping) or flagging them for their semi-static nature could improve LBS. To filter out moving objects, Litomisky and Bhanu [2013] used a temporal adjustment that takes moments in time and cross-reference if space is occupied during those moments.

Diakité et al. [2014] worked on automatically identifying and labelling objects from GIS or BIM sources. The objects from these sources were mostly part of the static environment, but could sometimes be temporal features (such as furniture and temporary structures).

2.4 LOCALISATION AND POSITIONING

Localisation, and in extension also positioning, is the field that looks into trying to accurately pinpoint the location of subjects in the (urban) context, by using sensors and localisation systems. The difference between localisation and positioning is that localisation tries to sufficiently provide a user's location on a map, around a landmark, or within the context or an area for providing LBS. Positioning (like the P in GPS, Global Positioning System) aims for a location with x,y coordinates in a coordinate reference system (CRS) or Cartesian plane, according to researchers like Gezici et al. [2005]. The difference between location and position is shown in Figure 2.5.

Werner [2014] distinguishes three types of positioning/localisation approaches based on their working principle:

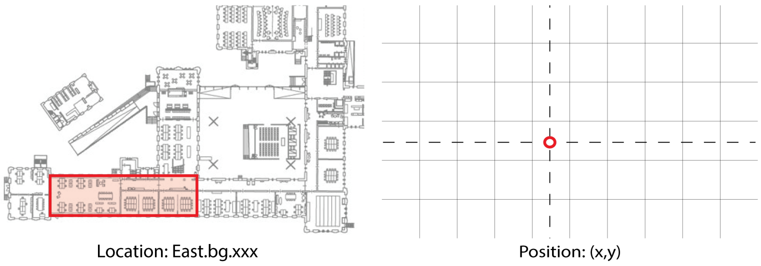
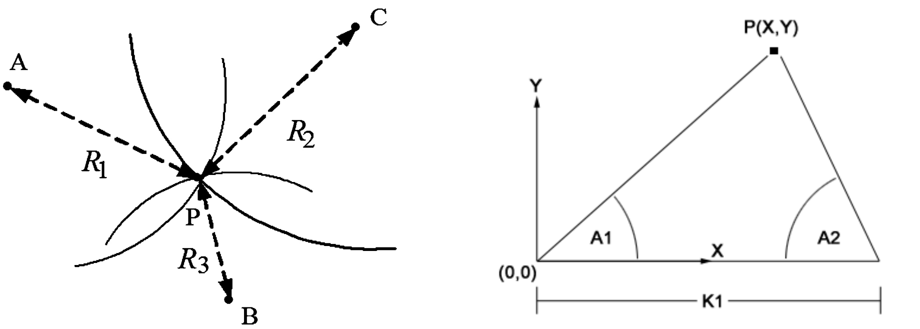


Figure 2.5: An example of a location within Faculty of Architecture (TU Delft) and a position within a CRS.

1. 1) *"Terminal-based positioning in which the mobile device calculates position without depending on some infrastructure"*
2. 2) *"Terminal-assisted positioning in which the positioning is distributed between infrastructure elements and the mobile device"*
3. 3) *"Terminal-free positioning in which a mobile device is located while the mobile device is passive"*

Terminal-assisted or system-based localisation uses a network of transmitters and receivers to send signals to the receiver (usually carried by the subject) and either through a one-way connection (in the case of GPS) or a two-way connection (localisation through (cellular) beacons). Depending on the LBS some degree of precision is required, different techniques and systems can provide locations that range from the level of precision of roughly hundreds of meters, or high-level precision of a few centimetres. Liu et al. [2007] surveyed wireless systems and distinguished several ways to position objects/user, systems that rely on the time it takes for a signal to travel back and forth between transmitter and receiver (Round-trip Time of Flight (RTOF)), **tof!** (**tof!**) or Received Signal Strength (rss) for a trilateration of a position, and wireless systems that use Angle of Arrival (AOA) for triangulation, see Figure 2.6. An example of trilateration is GPS, by measuring the distance to 3 satellites through TOF and correcting the time bias with a 4th satellite. An example of triangulation can be the AOA of a signal at two cellular towers.



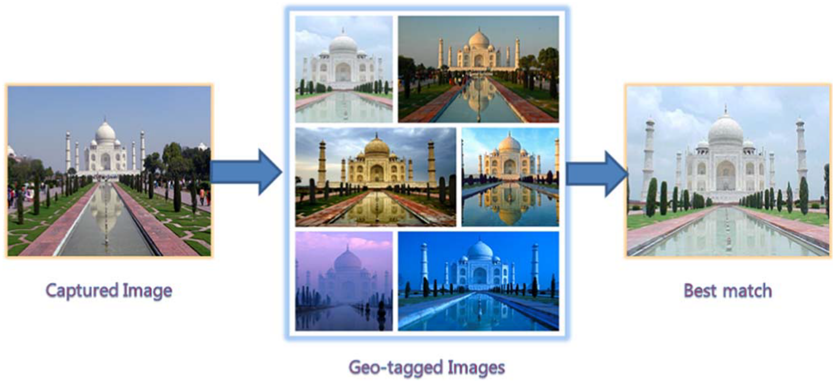
(a) Positioning based on **tof!**/RTOF measurements (trilateration), Liu et al. [2007].

(b) Positioning based on AOA measurement (triangulation), Liu et al. [2007].

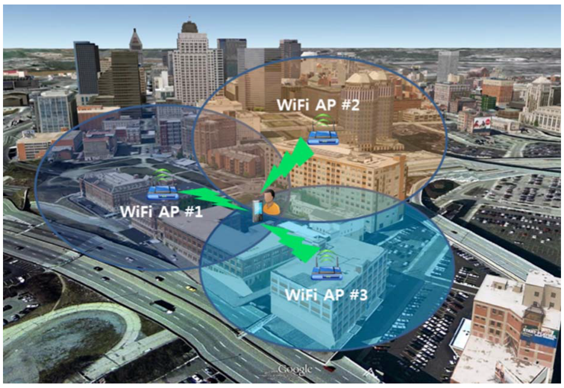
Figure 2.6: Principle of indoor positioning using cellular systems

A third localisation principle is that of fingerprinting, where a signature is matched against a geo-tagged signature. The geo-tagged signature is either one of three main fingerprint types: a visual fingerprint (image/picture signature), signal fingerprint (home or office Wi-Fi signal), or motion fingerprint (signature of gyroscope or accelerometer). Based on the fingerprint where the measured/observed signature is matched to a known geo-tagged signature a location can be obtained Vo and De

[2016]. The principle of fingerprinting maps the environment at a precise moment in time (see Figure 2.7), if the environment changes the fingerprints should be updated. Fingerprinting can, therefore, pose a problem for environments that are frequently changing or are continually shifting; these might get outdated too soon and keeping them up-to-date may prove costly.



(a) Matching an image to known geo-tagged images



(b) Matching a signature of signals to known fingerprints

Figure 2.7: Examples of outdoor fingerprinting Vo and De [2016].

The user-based (offline) localisation uses information presented to the user (through physical maps, an app or service) and let the user or the user's device use this information to localise oneself/itself, the user uses visual clues in the context to find a location on a (paper) map or the device can identify objects in the environment (QR codes, through object recognition, etc.).

2.5 INDOOR LOCALISATION

Curran et al. [2011] claims that *"The development of **rlsts!** (**rlsts!**) has become an important add-on to many existing location-aware systems. While GPS has solved most of the outdoor RTLS problems, it fails to repeat this success indoors"*. It is therefore important to look at localisation methods for the indoor environment. The three discussed methods of localisation (Section 2.4) behave differently in the indoor environment. The indoor environment operates on a smaller scale, resulting in that accuracy and precision have a more complex interaction.

The outdoor principle of triangulation and trilateration using cellular beacons can be equally applied indoors through Wi-Fi, Bluetooth or Radio-Frequency IDentification (RFID) beacons. Fingerprinting can be implemented indoors by mapping the rss of the different beacons. The environment can then be subdivided into regions with a precise fingerprint of rss, Chowaw-Liebman et al. [2010] described indoor fingerprinting, and Xiao et al. [2011], and Chan and Sohn [2012] for example applied this method to a Wi-Fi-based wireless system, which resulted in a method called Wi-Fi fingerprinting, each area has a unique combination of signal strengths for each beacon. Figure 2.8 shows an example of a radio map of the fingerprint of one Wi-Fi access point. By combining different radio maps of various access points, fingerprints are obtained.

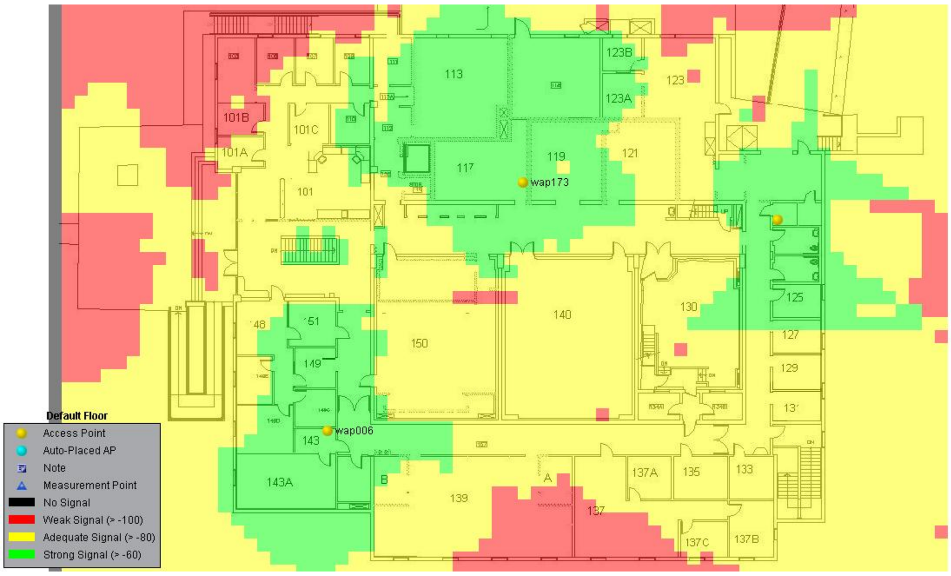


Figure 2.8: An example of a radio map of a Wi-Fi access point (Warner, J. [2008]).

2.6 NAVIGATION (PATH-/WAYFINDING)

Getting from A to B in the urban context is a daily activity, where the concept of wayfinding and navigation are often interchangeably used. According to Montello navigation can be called *"the coordinated and goal-directed movement of one's self (one's body) through the environment"* Montello [2005]. He proposed the conceptualisation of navigation as existing of two components: *wayfinding*, localising oneself and the destination in the urban context and picking a route to take; and *locomotion*, which describes the movement of one's self (one's body) in the direction of the intended destination while avoiding obstacles and barriers.

Although wayfinding and navigation are commonly used interchangeably, Montello sees a sharp distinction; wayfinding is only "the goal-directed and planned travel of one's self around an environment in an efficient way" without explicitly denoting how to move physically Montello [2005]. Within wayfinding, Redish elaborates on three types of wayfinding applicable for humans (random navigation, route navigation and locale navigation) Redish [1999] based on animal research into ways an animal can find a platform.

1. *Random navigation*: If the subject has no information about the location of the destination within the context, it must search randomly for it.

2. *Route navigation*: Route navigation can be thought of as chaining sequences of taxonomic and praxic sub-strategies, where the subject can learn to associate a direction with each sensory view and sequence tasks.

   a) *Taxonic navigation*. The subject can find a landmark toward which it can always navigate. For example, if the landmark is visible, it can simply "walk towards the landmark".

b) *Praxic navigation*: The subject can execute a constant motor program. For example, the subject follows a fixed distance with a fixed orientation (walk 100 metres in a straight line westbound or turn left in 100 metres), it can use the praxic navigation to reach the destination.

3. *Locale navigation*: The subject can learn the location of the destination relative to a constellation of cues. It can learn a map on which the location of the destination is known. If it knows both its location and the location of the destination in the same coordinate system, then it can plan a path from one to the other.

What is interesting is that 'locale navigation' is the way humans usually read maps if they do not have access to their precise real-time location: find oneself, find the destination and find clues in between to get there. If humans have access to an exact real-time location, they tend to use route navigation with praxic navigational instructions (go left in 100 metres, continue for 50 metres straight and you have reached your destination). Route navigation using taxonomic navigational instructions would be to use the context to navigate a user, even though the exact location of the user is not known to the user.

Richter developed a framework that would apply the concepts of Redish into strategies on navigation through the environment based on context, by asking the question "how humans find their way" with the answer planning the route and following it Richter and Klippel [2004] or getting constant updates. So for navigation, it is crucial that the subject/the user/the human has a sense of its location, has an awareness of the context, knows where the destination is and can find clues and follow cues. Moreover, depending on localisation techniques, a taxonomic or praxic approach to route navigation results in successfully reaching a destination.

Chun & Kim filed for a patent in 2003 that describes a now commonly used traditional vehicle navigation system as a *"navigation apparatus has a satellite signal receiver and a mileage calculator, which determines a current location of a vehicle based on a satellite signal and a mileage of the vehicle to its destination, and transmits information about the vehicle location from a mobile terminal to a base station transceiver system (BTS) periodically"* Chun and Kim [2003] where nowadays the BTS is computationally powerful enough and compact enough to fit on your car's dashboard (car navigation) or in your hand (smartphones).

2.7 INDOOR NAVIGATION

Within the indoor environment, there is the added challenge of not having a set network to navigate on, compared to the availability of outdoor navigation networks for cars and other road users. The open-space with obstacle characteristic of the indoor environment creates a chaotic and cramped environment where objects and destinations are visibly blocked on a very local scale, and where navigation becomes an art of dodging obstacles, moving around barriers and anticipate moving objects (i.e. humans, cars or ROVs). Mortari et al. [2014] lists several reasons why indoor navigation is such a challenge: *"positioning is not very accurate, users can freely move between the interior boundaries of buildings, navigational network generation may not be easy and straightforward due to the complexity of indoor space configurations"*.

Using traditional navigation methods in the open space and indoor environment requires that the environment is prepared appropriately for these methods, they rely on an edge-node network as opposed to just using the open spaces. Usually, the indoor space is therefore subdivided into areas through which a subject can move and information about how to go between these spaces. There are several methods developed that deal with the subdivision of space, Krūminaitė and Zlatanova [2014] and Xu et al. [2016] proposed a methods for subdivision of space that would "provide the most optimal path and guidance." by subdividing space "into navigable and non-navigable areas considering human perceptions of the environment and human behaviour [...] by applying a constrained Delaunay triangulation", see Figure 2.9. Zlatanova et al. [2013] proposed a framework that can do an *"automatic subdivision of indoor space and on-the-fly creation of a grid or irregular network."* where space is tiled rather than subdivided by context. Later this model is extended by Sithole and Zlatanova [2016] by including different types of placement (i.e. XYZ and room number), attributing resources and agents with modifiers to determine access and usage, sub-space inherent modifiers from resources and agents, and where there is no distinction between obstacles and resources, but just resources with modifiers about access and usability (not accessible equals an obstacle).

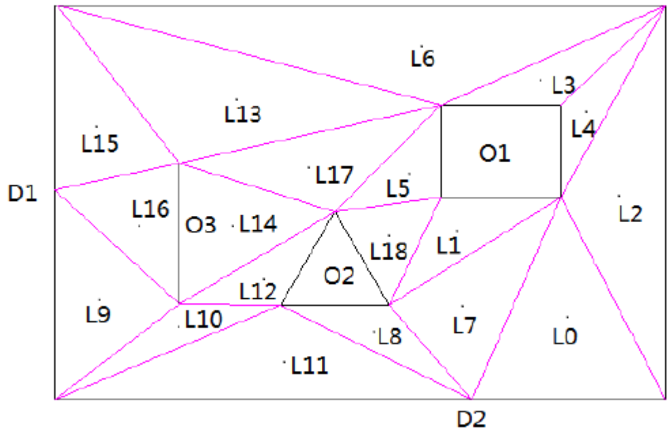


Figure 2.9: Subdivision of space using a constrained Delaunay triangulation, Xu et al. [2016].

The next step after having a subdivision of space is to get an edge-node network that supports network-based navigation and route calculation. One method of determining a path within the subdivided space is to calculate paths using the duality of the subdivided space, as you would on an edge-noded vehicle navigation network, as Xu et al. [2016] demonstrates in their framework: traversal between adjacent spaces is possible if the shared edge is not one of the edges from the constrained Delaunay (i.e. walls or obstacles), calculating the shortest route on this duality-graph provides a route, by simplifying this route afterwards, a more appropriate human route is the results, see Figure 2.10.

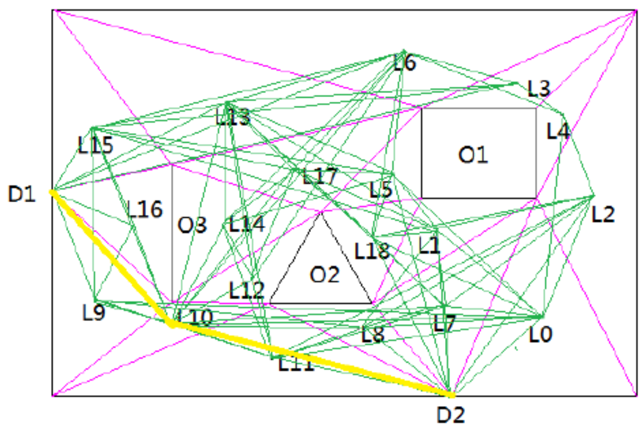


Figure 2.10: Duality-graph from subdivision of space, and simplified shortest path, Xu et al. [2016].

3 RESEARCH APPROACH AND METHODOLOGY

This chapter contains an overview of the selected research approach and methods as a starting point for the conceptual framework to give insights in the challenge of the exploratory research into proving indoor localisation using just landmarks is viable and can be put into a framework. Then it continues to describe the thoughts and ideas that emerged when theorising a case (3.1) for indoor localisation using a pure landmark-based approach and the resulting conceptual framework (4).

3.1 METHODOLOGY

When trying to answer the research question whether a pure-landmark based approach for indoor localisation is possible, a good start would be to look how this technique would ideally work in practice, then identify the inputs, processes and systems that are required and figure out how these can be made to work, in other words, prove indoor localisation of a user is possible/feasible.

Figure 3.1: The indoor environment with 'salient' objects (room names, furniture, people).

In the ideal case, users are getting around in the indoor environment based on the objects(landmarks) that they can see/distinguish in their surroundings. The only thing a user should have to do is describe the surroundings and have a system tell the user where he/she is or ask follow-up questions if the system can not locate the user. Where the observations could be that a user can see a particular set of landmarks or the locations of a landmark relative to another landmark. Figure 3.1

shows an example of the indoor environment of the Faculty of Architecture and the Built Environment (Delft University of Technology), in this example there are a few objects that stand out from the context: The letter P (signalling room P), with some difficulty the letter Q (for room Q), is visible, the letter R (signalling lecture hall R), the orange carpet (useful if it is the only place where they have an orange carpet), the observation that there are posters, or that you can see a particular poster, the presence of a clock and some high chairs, or even a person (agent) in a chequered shirt. All these objects/agents combined might make this location unique within the building. If building management knows the location of all these objects and agents, they could be used to determine one's location if you describe what you see. The picture also contains objects (room numbers) that are not readable from the viewpoint of the user, and they are too small, objects should, therefore, have information about when it is visible (distance or angle).

If this ideal case is broken down into its constituent parts (input, processing tools, output). There are a few components that are required: firstly the user communicates his **obervations** with the system. The system must be able to **interpret observations and translate it into action**, in this case, observations which landmarks are visible, and the relative locations of landmarks. For a system to understand observations, the observations require being according to a **machine-readable formal grammar**.The system must know **where each landmark is located** and **where each landmark is visible**. To **calculate where each landmark is visible** the systems must also know or be able to **where there are obstacles that impair visibility**. After interpreting the observations, the system should also be able to **cross reference or match the visibility** of various landmarks that are observed by the user. Given there might be **observations that talk about the relative location of one landmark to another landmark**, the system should provide **tools that narrow down location based on relative location**. With as a result a **user's location**. The visibility of landmarks could be considered as a **fingerprint or a functional region**.

The method of indoor localisation is shown in Figure 3.2, that shows where each bold component belongs and how the elements relate to each other in a workflow.

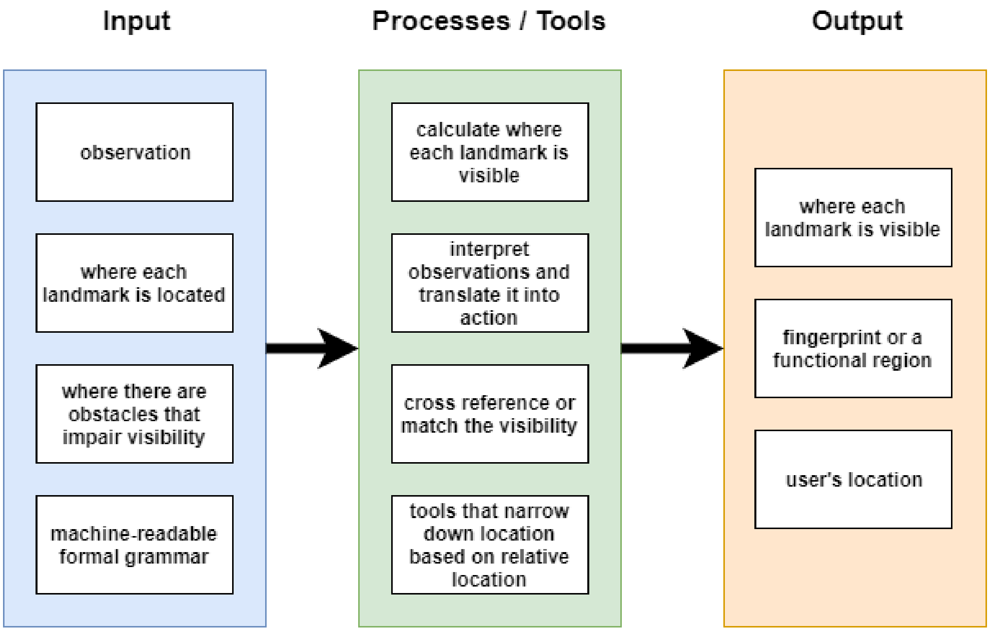


Figure 3.2: The ideal case written into a method

3.2 RESEARCH APPROACH

Due to the exploratory nature of the pure landmark-based approach for localisation, the research method that is applied is that of an iterative process of cyclic nature that stacks and re-evaluate the results. The total challenge of landmark-based indoor localisation is split into small stand-alone parts that are theorised, formalised and implemented into a Minimal Viable Product (MPV). Each of these small research parts can result in a set of definitions, concepts/guidelines, implementations/code, and a set of specifications, limitations and pre-conditions after evaluation. Figure 3.3 shows the flow of the research, where each step is either formalised for input or output (blue and orange) or theorised and implemented for the tools (green). After each component is evaluated, it can invoke changes in previous steps or new starting points for steps ahead.

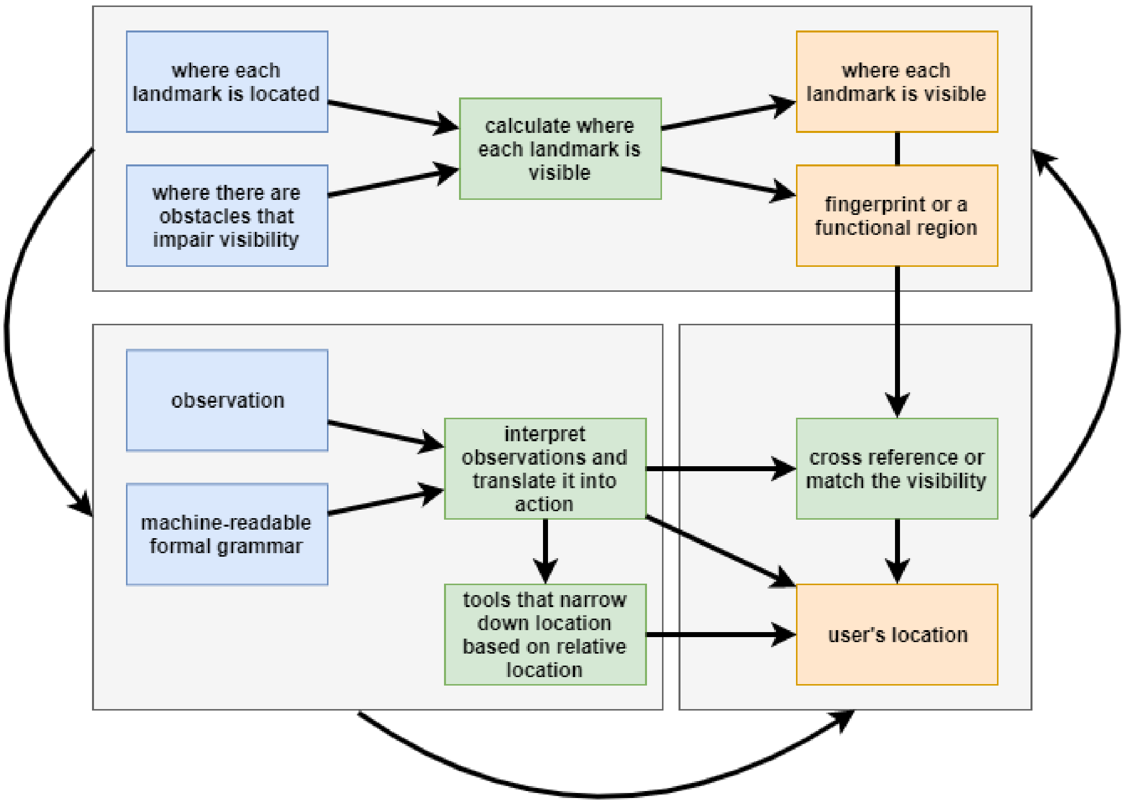


Figure 3.3: Cyclic research approach in order to develop a conceptual framework

Each iteration is driven by one of the research sub-questions, some questions are just about identifying and defining what's inside, and outside of the framework, other questions only go up to the conceptual phase within the iteration, and there are iterations where the concepts are also implemented. At each step of the iteration, the components are evaluated and re-evaluated if necessary.

Each research sub-question deals with a few aspects of the whole challenge, by weighing these aspects ideas are developed, these ideas are put in a schematic conceptual principle, this principle is implemented using code, and the conceptual principle is evaluated through simulation.

Due to the sequential nature of the implementation of a MPV, the product is first made to take a primary input and generate an output; then it is expanded to include more advanced conceptual principles that refine the product. Advanced feature implementations enhance the interactivity, interdependence, and the capabilities.

4 CONCEPTUAL FRAMEWORK

Starting from the case where the indoor environment consists of a floor plan, landmarks in different constellations, and obstacles, where a user describes his surroundings via observations, it is assumed that taking exact measurements of either distance or angle is not possible, but that the user can identify objects/landmarks. So through just a description of what a user/device can see in the surroundings, a location can be determined. So through the described relative location of one landmark compared to another or interaction with the user with follow-up questions, the location can be refined. The framework proposed in this chapter is a conceptualisation and exploration of obtaining this location from observations: it the describes the process of indoor localisation using a pure landmark-based approach.

The conceptual framework is implemented (see Chapter 5) into a MPV, to test and prove that the conceptual framework works, with two test cases. And any results, ideas, problems, limitations and thoughts are then discussed and analysed in Chapter 6. The conceptual framework that allows the indoor localisation of users through observations of their surroundings is established upon a set of definitions (Section]refsec:defintions) and a structured workflow that consists of a pre-processing step (Section 4.2) that prepares the system to accept observations, and the processing step (Section 4.3) where the user is actually localised through observations with an extension of on-the-fly adjustments to improve/refine the location.

4.1 DEFINITIONS WITHIN THE CONCEPTUAL FRAMEWORK

This part outlines the concepts and definitions that make up the conceptual framework. The conceptual framework outlines a workflow and a set of restrictions. The following definitions are assumed in each part of the workflow. And Figure 4.1 shows a Unified Modeling Language (UML) diagram of the entities.

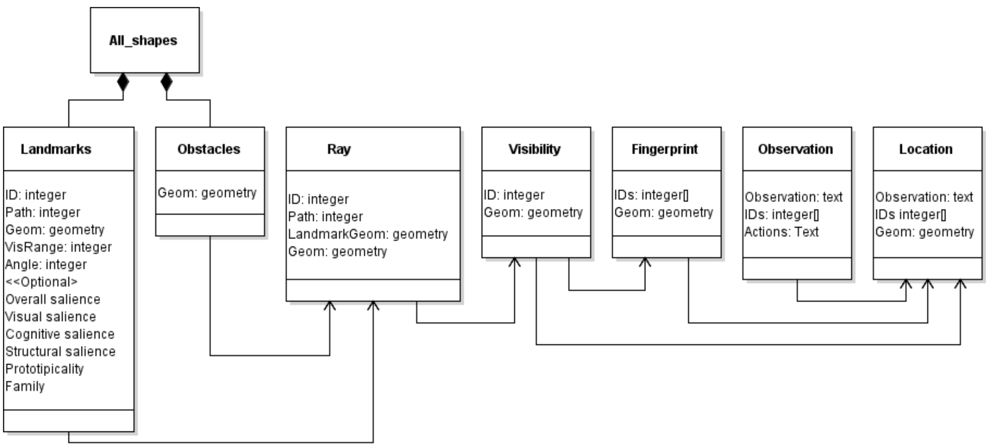


Figure 4.1: UML diagram of classes used within the framework.

4.1.1 Landmarks

An indoor landmark could be any objects, agent (person) or resource, that is visible from one to tens of meters and further, and that is distinguishable from its context/surroundings. Besides the fact that an object is labelled a landmark with a certain level of salience, additional landmark parameters are the level of uniqueness, the recognizability, the visibility in advance, and optionally an angle at which a linear landmark is visible, Table 4.1 shows the salience parameters of a landmark.

Table 4.1: Salience parameters of landmarks, and their use in the framework

Salience parameter	Description	Use within framework
Uniquenes	The ID of a landmark is determined if there is only a single instance of a landmark or if the landmark is part of a family of objects (i.e. furniture with a shared ID) or if the landmark is part of a subgroup within a family (i.e. chairs with a shared ID).	Unique or shared ID (integer)
Visual salience	Description of size, color, shape, physical appearance that might influence uniqueness, visibility in advance or cognitive salience.	-
Visibility in advance	Parameters that specify from where to where a landmark is visible.	Visibility range (meters)
Prototipicality	Whether an object belong to a family (through id or a description column), determines the uniqueness, affects the cognitive salience, if a chair doesn't look like a chair it's hard to identify.	-
Structural salience	To what extent can the landmark be used to provide navigational instructions (out of scope).	-
Cognitive salience	To what extent can a user refer to a landmark, this offers potential for machine-learning, by looking at which landmarks are most often referred to in order to provide better interaction with a user.	-

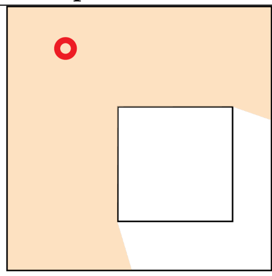
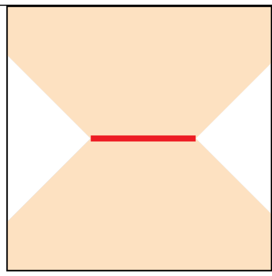
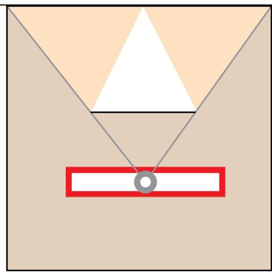
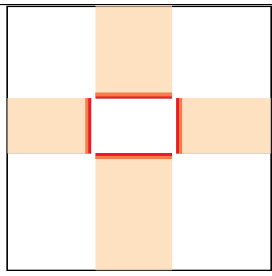
A landmark can be represented as a point, line or polygon (or polyhedron in 3D), depending on how it is visibility/isovist field behaves. The dimensionality of the geometries determines how visibility is calculated. For visibility/viewshed/isovist analysis, the geometry of landmarks should be represented as points by creating points along linear/area objects to account for the visibility of the whole line/area. Landmarks can also be obstacles.

Table 4.2 shows which geometric representations should be used in which cases, including an example of these objects. Point objects can be used as computationally light representations of objects that are visible 360 degrees and where the size of an object does not influence the way visibility can be modelled.

4.1.2 Obstacles

For localisation obstacles are objects that block the visibility of a landmark/user, such as walls. For navigation obstacles are objects that block the walkable path. Obstacles can also be landmarks

Table 4.2: Geoemtric representation of landmarks

Geometry	When to use	Example
Point	If an object is 360 degrees visible and size doesn't matter	
Line	If an object is only visible from front and/or backside	
Polygon	If the size of an object is sufficiently large that it affects visibility calculation	
Set of lines	If the size of an object is sufficiently large and the sides are visible from a certain angle	

4.1.3 Observations

The observations that the user inputs into the system have to be machine-readable observations that follow a formal grammar, for the system to be able to interpret them. A syntax and trigger words are part of that grammar, i.e. the observation separator can be the word 'AND' or the semicolon ';', and the observation is structured like 'I see landmark . . . ' where the dots are filled with a number or identification code. Through the machine-readable syntax, the system knows what's in the observation, if it is a series of observations or just one single observation, and if the first, second, or third word in the observations refers to the landmark.

4.1.4 Visibility

Visibility is calculated for points representing landmarks using, for example, the approach of ray-tracing. Ray-tracing calculates visibility by tracing a line from a landmark to the corner of an obstacle and looking past it, then tying the points into a non-intersecting polygon. After the visibility is estimated, adjustments of the visibility can be made in case of additional semantics (distance an object is visible, front side/back side), or according to the geometry following rules for linear landmarks.

Viewshed or isovist analysis is calculated for point locations, however, when calculating the visibility of objects, the geometric representation also plays an important part. The dimensionality of an object (0D point, 1D line, 2D surface and 3D volume) determines in which plane an object is visible. Figure 4.2 shows a 2D surface with text (size 10 cm by 50 cm by 0 cm thickness) at different angles (the side facing 0 degrees, 5 degrees, 10 degrees. 25 degrees, and front facing 90 degrees). Looking at the object from the side this particular object has a thickness of 0 cm by 10 cm, which results in a line with thickness 0 that is not visible. From an angle, the object starts to have a dimension in the plane and starts to get readable. This is because objects are only visible on the plane that is perpendicular to the plane they have a dimension (size). In 3D space, a 2D surface is not visible from the side and only readable from the front and the back side (and partly at an angle). In a 2D space, a 1D line element is visible on the perpendicular plane of where the line has a dimension (size). Geometrically modelling objects for visibility depends on how an object is visible if an object is only visible from a specific position. Geometrically an object can be modelled as a point, line, polygon or volume.

Figure 4.2: 2D object in 3D space and it's visibility at different angles to highlight that 2D objects are not visible from the side.

4.1.5 Fingerprint/functional regions

A fingerprint could be two things: a full subdivision of space and each (set of) area(s) has a unique visibility pattern, or the fingerprint could be the area of each (combination of) landmark(s), resulting in areas that are also divided into smaller areas if more landmarks are visible. The fingerprints can be considered functional regions that are identifiable through which landmarks are visible.

4.1.6 Location/position

The location is the region that is left after matching the fingerprint with a set of visible observation-derived landmarks and the refinement of the location through the extra information described in the observations.

4.2 PRE-PROCESSING: PREPARING THE SYSTEM FOR LOCALISATION

In the pre-processing (the step to prepare the system for localisation through observations) the input shapes that are mapped to landmarks and obstacles have to be geometric features with attributes about whether they are a landmark, obstacle, or both and can additionally contain semantics that describes the salience of the landmark (and thus how easy it is to identify and refer to), the uniqueness (is it a re-occurring object or a one-of-a-kind) and/or any additional information about the visibility of the object (from how far is it visible), or the angle from which the landmark is visible in case of a linear landmark.

When the input shapes match the database schema the input shape is split into obstacles and landmarks, then the obstacles and landmarks are homogenized into the correct representations (obstacles are polygon objects with an interior and landmarks are represented as a set of points that approximate the landmark for viewshed analysis), followed by the estimation of each landmark's area that is visible through ray-tracing, and finally these visibility areas are then cross-referenced into a fingerprint of each combination of landmarks, to save calculation time for the on-the-fly localisation. See Figure 4.3 for the steps in the pre-processing workflow.

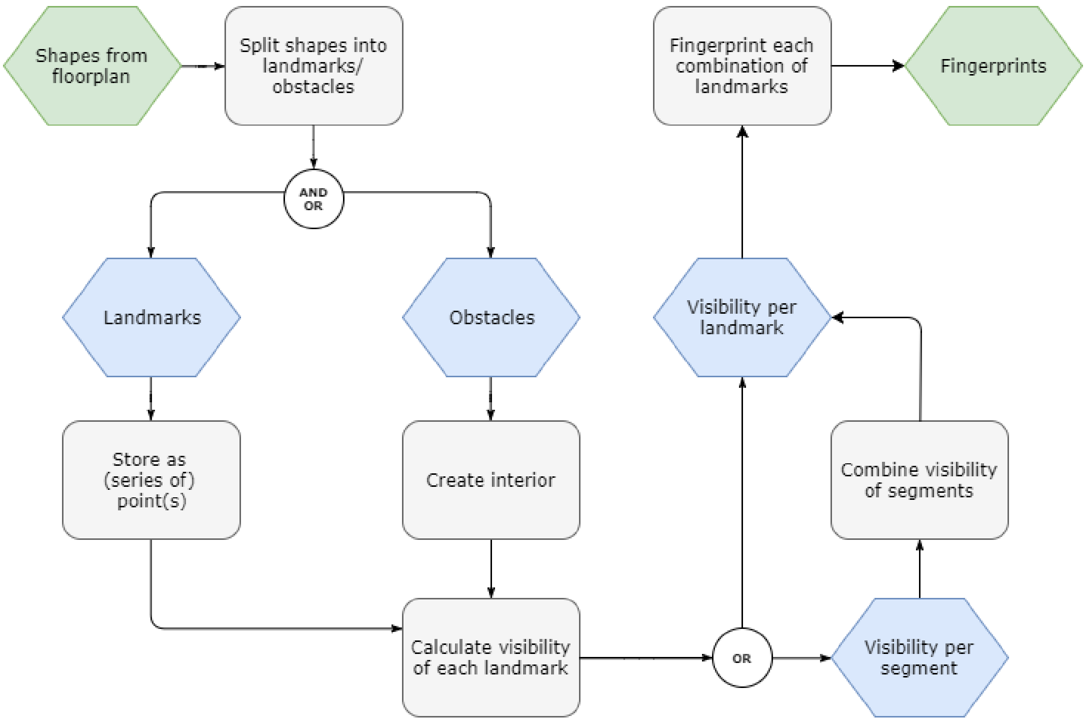


Figure 4.3: Pre-processing workflow to generate visibility and fingerprints.

PARSE TO SYSTEM SHAPES This step takes a set of parameters and map the table name, column names (id column, geometry column, type column, etc.), and case-specific key values, and the SRID of the location shapes to the parameters of the system shapes database, so that further steps know which columns to use.

SPLIT SHAPES Here the objects/shapes are split into obstacles and landmarks based on the label that denotes that an object is a landmark see Figure 4.4. During the parsing of the input table to the system table the value that marks an object a landmark is used. Landmarks are the shapes where the key value of landmark is in the column of the shape type, and obstacles are the shapes that don't have the landmark key value. Table 4.3 shows the input and output of this step. Landmarks have additional information about salience, as well as a unique or shared ID.

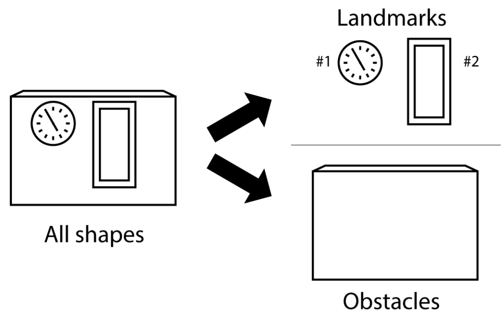


Figure 4.4: Split shapes into landmarks and obstacles and generate ID for landmarks.

Table 4.3: Input and output of 'split shapes'

Input	Output
Shapes (point, line, polygon)	Obstacles (line, polygon)
	Landmarks (point, line, polygon)

LANDMARKS: SUBDIVIDE INTO SEGMENTS AND REPRESENT AS POINT This step checks the geometry of the landmark. If the landmark is a point it is outputted as a point; no further actions are needed. If the landmark is a linear object (line or a polygon), the algorithm breaks each shape down into its linear sub-parts (polygon into lines), and each sub-part is split into segments with a maximum length according to user-specified key value for the max segment length. The endpoints of these segments are then used to represent the landmark with an 'id' and 'path', see Figure 4.5 for an example of a subdivided line, Table 4.4 shows input and output.

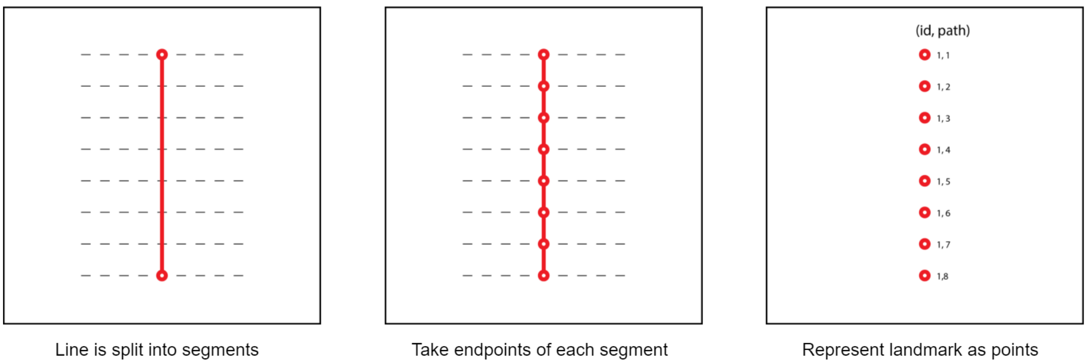


Figure 4.5: Steps involved in subdividing a line and represent the landmark as a point.

Table 4.4: Input and output of subdividing landmarks

Input	Output
Landmarks (point, line, polygon)	Landmarks (point(s))

OBSTACLES: CREATE INTERIOR Obstacles need to have an interior for more exact topological relationships and intersection later on in the algorithm. To get an interior for each obstacle, the geometry type of each obstacle is assessed. If the geometry type is a line, the line is buffered with a minimal distance (default value 0.1). If the geometry type of the obstacle is a (multi)polygon, the rings of the polygons are dumped (getting the original polygon along with new polygons for both the islands and holes), the linear ring representation (treated as a line) of these polygons is buffered. If the geometry type of the obstacle shape is a not linear (i.e. point), it technically is not an obstacle. Table 4.5 shows input and output.

Table 4.5: Input and output of creating an interior

Input	Output
Obstacles (line, polygon)	Obstacles (polygon)

RAY-TRACING Following the approach of Section **??** for the calculation of visibility, rays are cast from each landmark to all the corner points/nodes of the context (collections of polygons that represent obstacles). Then to 'look' past the corner, the rays are extended beyond the corner using the diagonal length of the context's bounding box (calculate the angle of the ray, translate/move the corner point with a vector of angle and distance of the max diagonal). Figure 4.6 shows the steps involved in ray-tracing, Table 4.6 shows the input and output.

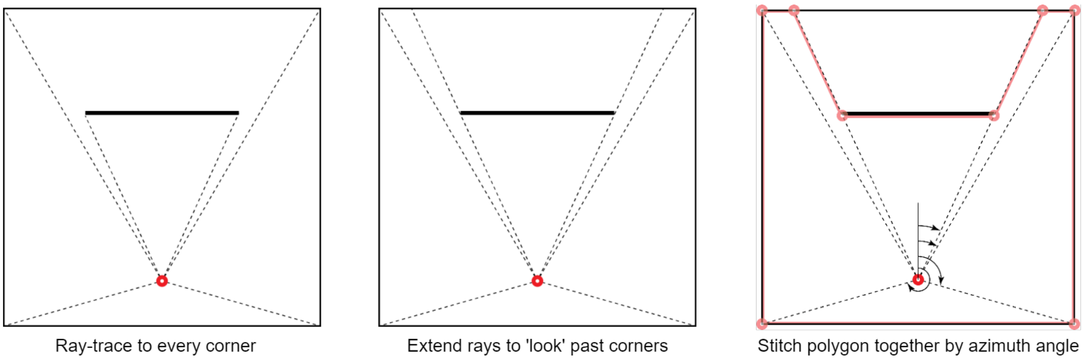


Figure 4.6: Raytracing of visibility of landmarks

Table 4.6: Input and output ray-tracing

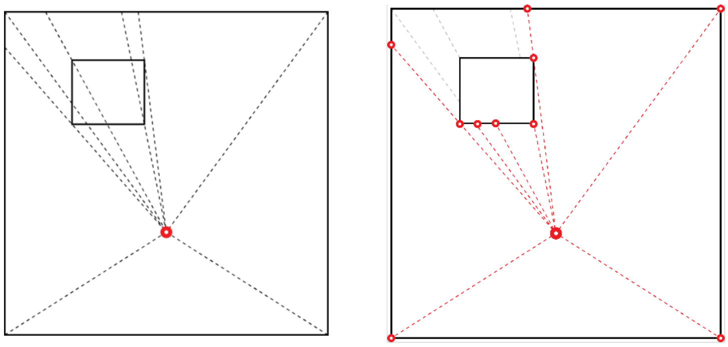
Input	Output
Landmarks (point)	Rays (line)
Obstacles (polygon)	

SUBTRACT CONTEXT FROM RAYS This step makes a Boolean difference of the context on the rays. The rays are cut in multiple linear parts, and only the part that is connected to the landmark is kept, see Figure 4.7 and Table 4.7 with input and output.

Table 4.7: Input and output of subtracting the context from rays

Input	Output
Rays (line)	Rays (line)
Obstacles (polygon)	

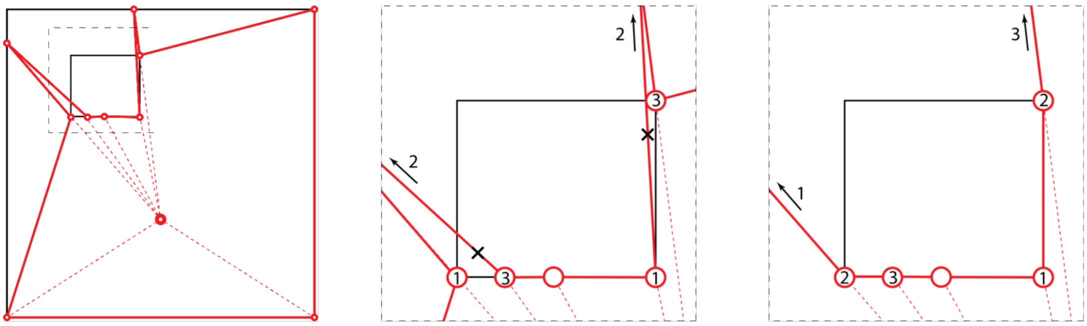
STITCH RAY ENDPOINTS INTO VISIBILITY POLYGON BY ANGLE Here the endpoints of the rays are sorted by angle and stitched together into a polygon. When two points with the same angle occur (in the case where corners are extended), the stitched polygon may intersect the context. At locations where a line segment intersects with the context two points must be flipped, the order of the points is then determined by whether the order will not make the newly formed line intersect



Rays after ray-tracing   Keep parts that originate from landmark

Figure 4.7: By subtracting the context from the rays, the results are visible points only.

with the context, Figure 4.8 shows this flip procedure. After the polygon is stitched together, all collinear points are removed through a simplification with a tolerance of zero (if points are collinear). Table 4.8 shows the input and the output. Visibility polygons of a landmark are also the fingerprint of the visibility of that one landmark.



Problem with same angle   Zoom in: problem with same angle   Fix problem by flipping 1-2 / 2-3

Figure 4.8: By stitching the points together a polygon that represents the visibility is created.

Table 4.8: Input and output stitching ray endpoint into visibility polygon

Input	**Output**
Rays (line)	Visibility (polygon)
	Fingerprint (polygon)

LANDMARK: COMBINATION Before the step of cross-reference/fingerprinting, a combinatorial algorithm can be run on the ids of the landmarks that gives every theoretical combination of landmark ids. If there are landmarks A, B and C, it results in combinations A, B, C, AB, AC, BC, and ABC.

CROSS-REFERENCE/FINGERPRINT Together with the combination step combinations of visible landmarks can be pre-calculated, to speed up the on-the-fly localisation, by pre-calculating all combination of visibility. The pre-calculated theoretic combinations of landmarks are used to do an intersection of each combination of landmarks. The algorithm loops through the combination and intersects polygon A with polygon B, and the intersection of AB are intersected with C, and so on. When the resulting geometry is empty (i.e. there is no intersection, so the combination of landmarks is not visible), the combination is scrapped from the list. Otherwise, if a combination results in a valid polygon, this polygon is stored as a fingerprint with its corresponding combination. Figure 4.9 shows the steps to get from landmarks to visibility, to fingerprint. Table 4.9 shows the input and the output.

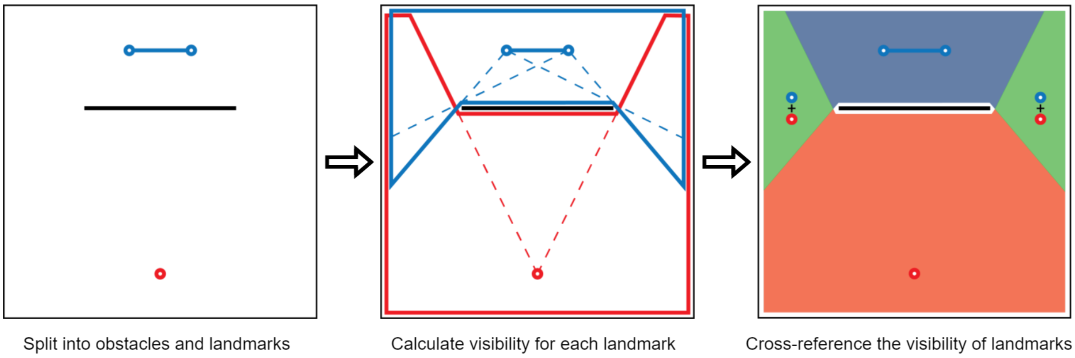


Figure 4.9: Pre-processing products that result in fingerprints.

Table 4.9: Input and output of fingerprinting

Input	**Output**
Visibility (polygon)	Fingerprint (polygon)

4.3 PROCESSING: LOCALISE A USER THROUGH OBSERVATIONS

The processing step, that actually localises a user through observations, first has to interpret the observation(s) that is/are fed into the system, then it queries the pre-calculated fingerprint that matches with the visible landmarks of the observation(s), and finally it tries to refine the location by querying the user with a follow-up question (i.e. can you also see C?) or using additional information contained within the observations (i.e. A is left of B). See Figure 4.10 for a complete workflow of on-the-fly localisation.

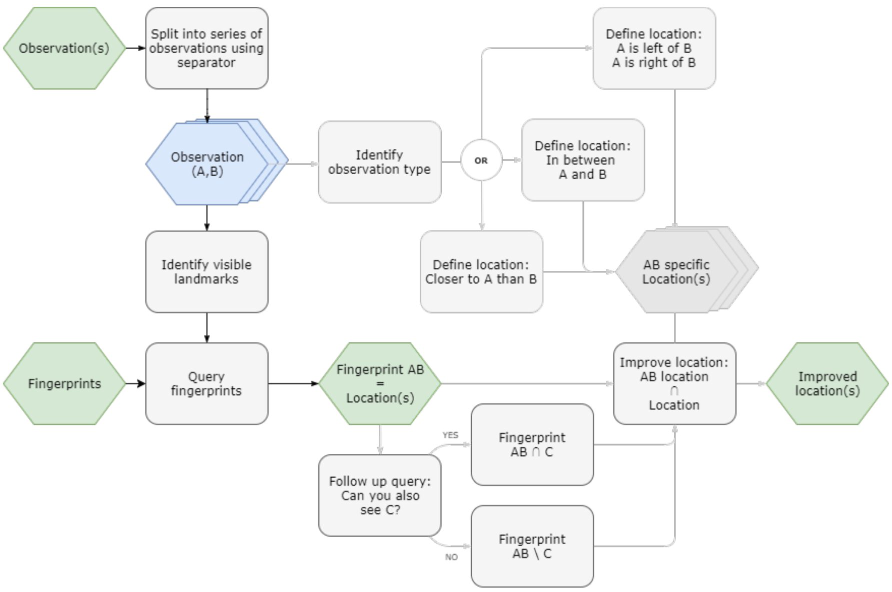


Figure 4.10: Processing workflow of localising a user through observations.

INTERPRET OBSERVATIONS Before being able to localise a user, the observation (machine-readable) has to be interpreted by the system. The observation is a text fragment that is parsed into the system and based on an observation separator (a comma or a semicolon), a single line of text is read into multiple observations. Each observation follows a pre-set grammar so that from each observation the visible landmarks are filtered out. Also based on the additional information if an observation includes particular machine-readable observations (i.e. explaining the user's relative position to one or more landmarks) refinement actions are kept in reserve further on. Table 4.10 shows which machine-readable observations could be supported, and Table 4.11 shows the input and the output.

Table 4.10: Supported grammar for machine-readable observation

Observation	Visible landmarks	Refinement
I see A	A	
I see A and B (.. and C)	A, B (C)	
I see A left/right of B	A, B	refine: left/right (A,B)
I see A behind B	A, B	refine: behind (A,B)
I am in between A and B	A, B	refine: between (A,B)
I am closer to A than B	A, B	refine: closer (A,B)
I am touching A	A	refine: touch (A)
I am in functional area X		refine: area (X)

Table 4.11: Input and output of interpreting observations

Input	**Output**
Observations (text)	Landmarks (integer ids)
	Refinement actions (text)

LOCALISE USER While interpreting the observations the visible landmarks are stored. Then the next step is to match the fingerprint with the stored combination of visible landmarks (i.e. ABC or just A). The resulting output is a (multi)polygon with the location(s) a user can be, as shown in the conceptual framework (Figure 4.11). Sometimes the result contains multiple areas; either this could be narrowed down by the refinement, or it might trigger a follow-up query eliminate one of the multiple areas.

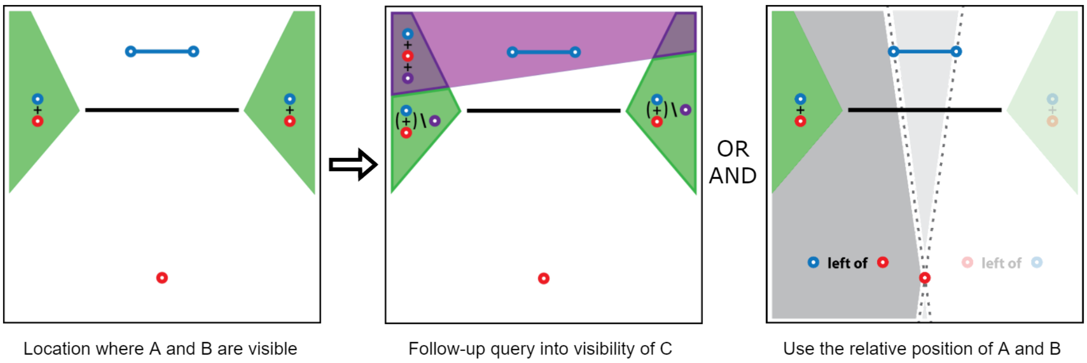
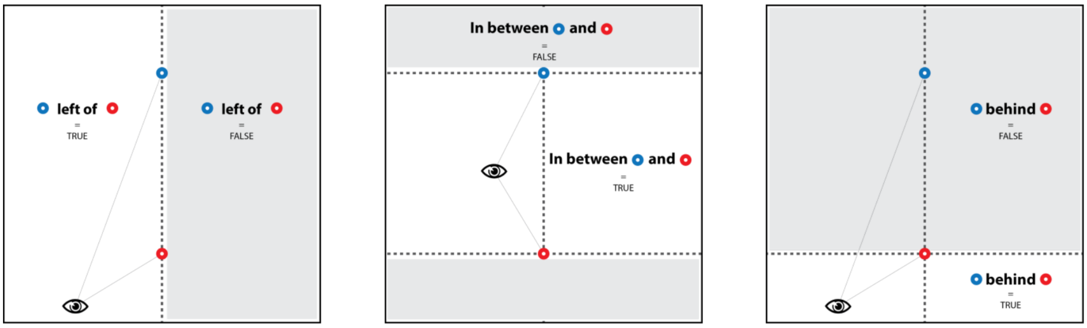


Figure 4.11: Localising a user with refinement and/or follow-up.

REFINE LOCATION This step takes the refinement triggers/actions from Table 4.10 and translate them into an on-the-fly calculated area, this area is then intersected with the initial location to narrow down the location potentially (see Figure 4.11). The fundamental principle of this step is to draw a line between the two landmarks that are compared ('I see A left of B') and either perpendicular to the line or an extension of the line draw a region, as can be viewed in Figure 4.12 or Figure 4.13. The result from this step is a used to narrow down the location. Table 4.12 shows the input and the output.

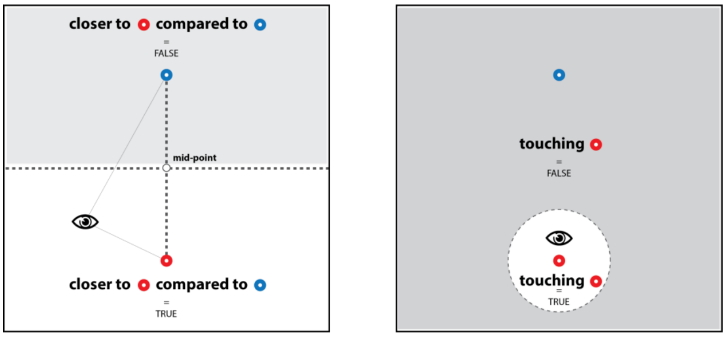
When observing the surroundings a user can use angle to describe whether a landmark is left or right of another landmark, if a user is standing in between two landmarks, or if one landmark is behind another landmark, this can be used to eliminate areas that don't fit this condition, Figure 4.12 shows both examples. Note that the observation always needs two landmarks since the location and orientation of a user is not known, just observing one object is to the left will not result in a location if the user's location is not known. However, it can be used to determine the orientation of the user after a location has been established.



I see A is left of B | I am in between A and B | I see B behind A

Figure 4.12: Using angle to refine location.

Through an estimation of the distance the user can also give clues regarding his location, if a user is closer to one landmark relative to another, or if a user is touching a landmark (there is no mistaking the distance to a landmark), see Figure 4.13 for examples.



I am closer to A compared to B | I am touching A

Figure 4.13: Using distance to refine location.

Table 4.12: Input and output of refinement of location

Input	**Output**
Refinement action (text)	Refinement location (polygon)
Landmarks (point)	Refined location (polygon)
Location (polygon)	

FOLLOW-UP QUESTION/QUERY The final refinement of the location is done by querying the user with a follow-up question (i.e. can you also see C?) or using additional information contained within the observations (i.e. A is left of B). So if the initial observation is also visible in a combination that includes an additional landmark that the user has not observed, a follow-up question can be asked if it can also see an additional landmark. If the user can see this other landmark, the location is narrowed down further (the user initially saw A and B, and after follow-up confirms it can see C, location AB becomes location ABC). If the user cannot see the other landmark, its area can be subtracted from the initial location (location AB, but not C results in a Boolean difference of C from AB). The result from this step is a narrowed down location. Figure 4.11 shows the products of the steps in the workflow, Table 4.13 shows the input and output.

Table 4.13: Input and output of follow-up step

Input	**Output**
Location (polygon)	Question/Query (text)
Visibility (polygon)	Improved location (polygon)
Fingerprint (polygon)	

5 IMPLEMENTATION

This chapter provides insights into the implementation of the conceptual framework (4)for indoor localisation using a pure landmark-based approach, it explains to what extent this framework is implemented and could be implemented further on, and the results from the research iterations (as described in 3.2) are presented in the chapter analysis and discussion (Chapter 6).

The components of the conceptual framework are implemented entirely in a series of Minimal Viable Products (MPVs) in the form of functions (see Table 5.1. Each component is described as a (set of) function(s) and where applicable pseudo-code is provided to show how it could be implemented using code.

The next parts highlight which tools and libraries and programming language are used, what data model is used to process the data, which artificial cases are used, and how the conceptual framework is implemented through which functions and methods.

5.1 TOOLS AND LIBRARIES

In order to get an artificial case into the PostGIS environment of PostgreSQL FME was used to get a .DWG/.DXF (AutoCad) file into PostGIS. Then the implementation is done entirely in PostgreSQL using PostgreSQL's Procedural Language (PLPGSQL), the extensions PostGIS (to add spatial capabilities), and intarray library (to be able to sort and reverse integer arrays). Quantum GIS (QGIS) was then used to visualise the results from the algorithm written in PostgreSQL/PLPGSQL. Figure 5.1 shows the flow and order of tools.

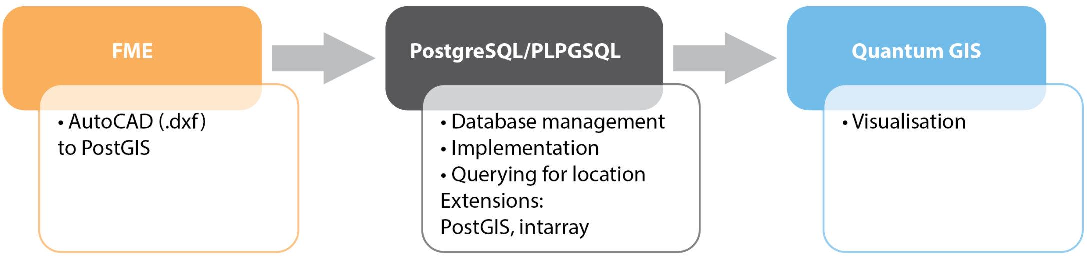


Figure 5.1: Tools used for the implementation of the conceptual framework.

5.2 ARTIFICIAL CASES

The implementation used two 2D cases: an artificial test case and an artificial 'real world' case using the east wing of the Faculty of Architecture and the Built environment (Delft University of Technology). When creating the artificial test case, all efforts were into creating a case that accounts for all possible variations of both obstacles and landmarks that would be present when using a real-world case, without having too complex a case, see Figure 5.2. Different geometry for landmarks and obstacles ensures that the implemented algorithm can deal with unexpected variances if geometry and shapes change, i.e. walls represented as lines, as concave polygons and as closed lines without thickness. The landmarks are represented as both points and lines, where linear landmarks also cover the possibility of polygon representations of a landmark. The artificial case was used to get a minimal viable product to achieve indoor localisation, and then the MVP was tested using a manually constructed (therefore artificial) real-world case (existing building) with the challenges of complex floorplans and the (inconvenient) location of salient objects (landmarks), see Figure 5.3.

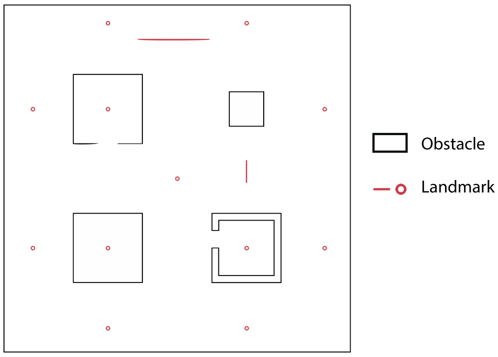


Figure 5.2: Artificial case to create MVP.

Figure 5.3: Faculty of Architecture and the Built Environment case to test MVPs.

5.3 ALGORITHM

The implemented algorithm exists of a series of functions that all execute a part of the total algorithm and one function that runs the whole implementation when asked to initiate or take in observations to get a location. The flow of algorithms and functions can be seen in Table 5.1.

Table 5.1: Functions, their flow and dependencies

Main algorithm	Main function	Subfunctions	Utilities
MainAlgorithm	Visibility	Subdivide	
		CreateInterior	
		VisibilityPolygon	
		LineVisCorrection	
	FingerprintSet	Combination	
	Localise		
	RefineLocation	InfiniteLine	Clockwise
		PerpendicularLine	Clockwise
	FollowUp		

5.3.1 Main algorithm

MAINALGORITHM(TASK,TASK INPUT,TABLE NAME, TABLE COLUMNS, KEY-VALUE) The main algorithm is the algorithm that sets all the wheels in motion, this is the algorithm that is used to initialise the system, and which the user uses to localise one's self, Table 5.2 shows which tasks, which inputs are supported, and which function are called. Algorithm 5.1 and 5.2 shows the steps and invocation of the main algorithm.

Table 5.2: Input, output and available tasks of Main Algorithm

Task	Input	Called function	Output
Initialise (checks for 'ini%')	tablename, column names, landmark value, SRID	Visibility, (FingerprintSet*)	Tables: landmarks, obstacles, rays, visibility, (fingerprints)
Localise (checks for 'loc%')	observation	Localise, RefineLocation, FollowUp	Table: location
...	...	...	
*FingerprintSet can only be invoked if the system can handle the amount of landmarks in the combinatorial method			

MAIN ALGORITHM: INITIALISATION The initialisation Algorithm 5.1 starts up the visibility function and takes the parameter of the database that contains the shapes of the indoor environment and starts the algorithm that calculates the visibility of landmarks and if applicable create the combinations of landmarks in the fingerprintSet step.

Algorithm 5.1: Task of initialisation and visibility calculation of main algorithm

```
--/ Main Algorithm: Initialisation
IF task ILIKE 'ini%' THEN
    --/ Parse the input parameters into visibility function to
     ↪ initate and calculates visibility
    CREATE TABLE 'visibility' AS (Visibility(...))
    --/ FingerprintSet function calculates every combination of
     ↪ landmarks and calculates geometry of every combination
    CREATE TABLE 'fingerprints' AS (FingerprintSet(...))
```

MAIN ALGORITHM: LOCALISATION After the initialisation the main algorithm is ready to process observations into locations, a user feeds an observation in accordance with the machine-readable grammar and the system finds a location, refines the location (if the observations allows for that), and asks a follow-up questions on other visible landmarks (Algorithm 5.2).

Algorithm 5.2: Task of localisation and refinement of main algorithm

```
--/ Main Algorithm: Localisation
IF task ILIKE 'loc%' THEN
    IF TABLE EXIST 'visibility' THEN
        CREATE TABLE 'location'
        --/ example observation = 'I see 1, I see 2 left of 4'
        LocationPolygon = Localise(observation)
        --/ example observation = 'I see 2 left of 4'
        RefinementPolygon = RefineLocation(observation)
        --/ Union of location based on what landmarks are
         ↪ visible and on which region fits condition of
         ↪ refinement
        LocationPolygon = LocationPolygon ⋃ RefinementPolygon
        --/ Look for additional landmarks to refine location
         ↪ and query the user with follow-up question
        QueryText = FollowUp(observation,location)
    ELSE
        RAISE WARNING 'Run initialisation first!'
    END IF
END IF
```

5.3.2 Visibility function

VISIBILITY(TABLE NAME,TABLE COLUMNS, KEY-VALUE) The visibility function accounts for the majority of the algorithm, after parsing the input parameters (table name, table columns, and key values) (Algorithm 5.3 this function creates the system table that is used to split the shapes into landmarks and obstacles (Algorithm 5.4 and homogenise the geometry (Algorithms 5.5 and 5.6). Then it applies to do a ray-tracing of the visibility for all landmarks, the endpoints of the rays are stitched together, and visibility polygons (isovist fields) are created. Some of the operations are done internally, and some operations are done through more complex functions, Table 5.3 shows these called functions and their role.

Table 5.3: The input, called functions and the role of the functions within the Visibility algorithm

Input	Called functions	Role/result of function
All shapes	Subdivide	Represent landmarks as a set of points to estimate the total visibility, linear landmarks are subdivided according to the max segment length.
Segment length	CreateInterior	Obstacles are homogenised into polygons with an interior, this is essential for the ray-tracing step that test for a topological relationships.
Angle	VisibilityPolygon	When the rays are cast and the endpoints collected, this function stitches the points together by angle and corrects self-intersection.
	LineVisCorrection	When the visibility polygons is created, it is corrected with specific geometric and semantic corrections.

VISIBILITY FUNCTION: PARSING PARAMETERS During Algorithm 5.3 a table is created that takes the table name, column names, and key-values, and map them onto the all-shapes system table. This table is then used for the rest of any algorithm.

Algorithm 5.3: Parsing the input parameters and create system shapes table.

```
--/ Visibility Algorithm: Parse parameters
table name                    = IN table_string text
id column                   = IN id_column text
geom column               = IN geom_column text
object-type column     = IN type_column text
vis-range column         = IN visrange_column text
angle column                = IN angle_column text

--/ Construct query with string replace
EXECUTE 'CREATE TABLE/VIEW 'all_shapes' AS (
        SELECT % id,% geom,% object_type,% vis-range,% angle
    ↪ FROM % table)' % (...)
```

VISIBILITY FUNCTION: SPLITTING OBJECTS Algorithm 5.4 splits the objects, based on the landmark value in the object type column it is either put in the landmark table or in the obstacle table. While getting the landmarks, they are split into point objects that are required for the ray-tracing step. Obstacles are homogenised into polygon objects that have an interior, and this is a requirement for the topology operations of the ray-tracing step.

Algorithm 5.4: Splitting the objects into landmarks and obstacles.

```
--/ Visibility Algorithm: Split objects
landmark-value = IN landmark_string text

--/ If an object in the all_shapes table got object type label
    ↪ landmark, select it
CREATE TABLE 'landmarks' AS (SELECT * FROM all_shapes WHERE
    ↪ object-type = landmark-value)
--/ If an object in the all_shapes table not got object type
    ↪ label landmark, keep only the geometry
CREATE TABLE 'obstacles' AS (SELECT geom FROM all_shapes WHERE
    ↪ NOT object-type = landmark-value)
```

SUBIDIVIDE SUBFUNCTION: REPRESENT A LANDMARK AS A (SET OF) POINT(S) Algorithm 5.5 takes the geometry of a landmark and subdivides linear landmarks in a set of points that together make sure the visibility that is calculated during the ray-tracing covers the whole landmark.

Algorithm 5.5: Subdividing linear landmarks into a set of points to better estimate the visibility.

```
--/ Subdivide subfunction
IN landmark geometry, segment-length float,
--/ If geometry is a point, no need to subdivide
IF geometry = 'Point' THEN
    RETURN [geometry]
--/ If geometry is not a point, break geometry down into
 ↪  smallest lines (from point n to point n+1)
ELSE
    pointArray = []
    FOR EACH line IN geometry LOOP
        line-length = length(line)
        subdivide-ratio = 1 /
         ↪  roundup(line-length/segment-length)
        r = subdivide-ratio
        --/ Create point on the line at ratio and add to points
         ↪  of pointArray
        WHILE r <1 LOOP
            point = InterpolatePoint(line,subdivide-ratio)
            pointArray = pointArray + point
            r = r + subdivide-ratio
        END LOOP
        --/ After point that are interpolated add first and
         ↪  last point
        pointArray = pointArray + StartPoint(line) +
         ↪  EndPoint(line)
    END LOOP
END IF;
RETURN pointArray
```

CREATEINTERIOR SUBFUNCTION: HOMOGENISE THE OBSTACLES AS POLYGONS Algorithm 5.6 assesses the type of the geometry and takes the appropriate action. For points, the output geometry is null, since a point cannot be an obstacle. Other geometry types are buffered with a marginal radius to ensure that the obstacle geometry has an interior. Polygons can get a special treatment based on the case that is worked, if polygons are solid objects they can be returned as solid objects, if polygons represent hollow objects, the rings should be taken and buffered as if they are lines.

Algorithm 5.6: Ensuring an interior for obstacles based on geometry type.

```
--/ CreateInterior subfunction
IN obstacle geometry
--/ If geometry is a point, it's not an obstacle
IF geometry = 'Point' THEN
    obstacle geometry = null
--/ If geometry is a line
IF geometry = 'Line' THEN
    --/ Create a buffer around object to ensure it has an
    ↪  interior
    obstacle geometry = Buffer(geometry,0.01)
IF geometry = 'Polygon' THEN
    --/ Take the rings of the polygon as lines and buffer the
    ↪  lines
    rings = rings(geometry)::lines
    obstacle geometry = Buffer(rings,0.01)
ELSE
    obstacle geometry = null
END IF
RETURN obstacle geometry
```

VISIBILITY FUNCTION: RAY-TRACING Algorithm 5.7 takes the point representations of the subdivided landmarks along with the context (obstacles) that is within the visibility range of the original landmark (point, line, polygon representation) and create a point from every landmark to every corner point of the context and extended beyond (to look past a corner). Algorithm 5.8 demonstrates the subtraction of the context from the raw rays, and the retrieval of those parts of the ray that originate from the landmark. After the rays are finalised, they are used as input in the subfunction VisibilityPolygon, that stitches the endpoints of the rays into a visibility polygon (Algorithm 5.9.

Algorithm 5.7: Ray-tracing from landmark point to obstacle points.

```
--/ Visibility: Ray-tracing
IN landmark geometry, landmark-original geometry, obstacle geometry,visrange integer
--/Create obstacles that are in close proximity to a landmark = within visrange
viewextent = Buffer(landmark-original geometry, visRange)
--/ Union obstacles and add the view extent buffer as shape
context = Obstacles ⋃ Obstacles
localcontext = (viewextent ⋂ context) ⋃ viewextent
--/ Take all the points of obstacle polygons
obstaclePoints = DumpPoint(localcontext)

rayArray = []
FOR EACH point IN obstaclePoints LOOP
    ray = Line(landmark geometry, point)
    --/ Extend the ray by adding a 3rd point to an existing ray
    extendedray = ray.AddPoint(Translate(point, cos(Azimuth(ray))*visrange, sin(Azimuth(ray))*visrange
    rayArray = rayArray + ray + extendedray
END LOOP
--/ Product is an array of rays per landmark
OUT rayArray
```

Algorithm 5.8: Boolean difference of rays with local context and filtering of rays that touch their landmark

```
--/ Visibility: Ray-tracing difference
IN landmark geometry, localcontext, ray geometry

--/ Only keep the rays where the unextended ray (initial ray) doesn't cross the interior of the context
raysArray = ARRAY(SELECT ray FROM rayArray WHERE NOT Crosses(Line(ray.point1,ray.point2), localcontext)
--/ Subtract the local context from the rays
raysArray = ARRAY(SELECTray \ localcontextFROMrayArray)
--/ Merge the rays and keep the part that has shared point with landmark
raysArray = ARRAY(SELECT Merge(ray) FROM rayArray WHERE Touches(ray,landmark geometry)

--/ Output is a set of rays that originate from the landmark, that extends past a corner, but stops at an obstacle
OUT raysArray
```

VISIBILITYPOLYGON SUBFUNCTION Algorithm 5.9 takes the rays as input, calculates the angle of each ray, sorts the rays by angle and stitches the visibility polygon together. This subfunction encounters rays with the same angle (in case of extended rays), and it makes sure that the constructed polygon does not intersect with the context or is a self-intersection.

Algorithm 5.9: Stitching the visibility polygon together and check/repair errors

```
--/ VisibilityPolygon subfunction
IN landmark geometry, ray geometry

--/ Calculate the azimuth angle of ray, sort the rays on azimuth angle, and take endpoint of ray
(azimuth,rays) = (Azimuth(ray),ray)
(azimuth,rays) = Sort(azimuth,ray)
(azimuth,endpoint) = (azimuth,EndPoint(ray))

--/ Stitch points together sorted by angle
line = []
FOR EACH angle, point IN (azimuth, endpoint) LOOP
    line = line + point
END LOOP
--/ Add first point to line to make it closed
polygon = Polygon(line + line[0])

--/ Check if resulting polygon is valid (not self-intersecting) and if it crosses the localcontext
IF IsValid(polygon) THEN
    IF Crosses(polygon,localcontext)
        FOR n, line IN enumerate(polygon) LOOP
            IF Crosses(line,context) --/ Identical angle raysArray
                swap point[n] with point point[n+1] or point[n-1]
            END IF
        END LOOP
        RETURN po lygon
    ELSE --/ polygon is valid and doesn't intersect with localcontext
        RETURN polygon
ELSE --/ polygon is not valid (self-intersecting)
    lineArray = lines(polygon)
    --/ If line 1 and 3 interset, it means the lines of line 2 must be flipped
    FOR n IN 1 .. length(lineArray) LOOP
            IF Crosses(line[n],line[n+2])
                flip line[n+1]
            END IF
    END LOOP
    RETURN polygon
END IF
```

LINEVISCORRECTION SUBFUNCTION Algorithm 5.10 will calculate the visibility of linear landmarks based on their geometric shape and angle when the landmark is recognisable/readable. The basic operation is to extend the linear feature and rotate the extension with the angle specified (both in the positive direction as negative). This V-shaped line then splits the visibility buffer based on the visibility range of a landmark. The part of the split buffer that has its centroid over the centroid of the line will be kept. If a point landmark is supplied, a buffer with the visibility range is returned

Algorithm 5.10: Visibility correction for linear landmarks, and visibility range for points

```
--/ VisibilityPolygon subfunction
IN landmark original geometry, visRange integer, angle integer

IF landmark = 'Point' THEN
    RETURN Buffer(landmark, visrange)
ELSE
    --/ For each line in the linear/polygon landmark, extend line on both ends, rotate the extension and split the buffer of the line with extended lines
    visibilityCorrection = []
    FOR EACH line IN geometry LOOP
        visibilitybuffer = Buffer(landmark, visRange)
        splitLines = []

        extendPoint = Translate(point1, cos(Azimuth(line.point2,line.point1}))* (visRange + 1),sin(Azimuth(line.point2,line.point1}))* (visRange + 1))
        extendLine = Line(point1,extendPoint)
        splitLines = splitLines + Rotate(extendLine,origin=point1,+angle) + Rotate(extendLine,origin=point1,-angle)

        extendPoint = Translate(point2, cos(Azimuth(line.point1,line.point2}))* (visRange + 1),sin(Azimuth(line.point1,line.point2}))* (visRange + 1))
        extendLine = Line(point2,extendPoint)
        splitLines = splitLines + Rotate(extendLine,origin=point2,+angle) + Rotate(extendLine,origin=point1,-angle)

        splitArray = Split(visibilityBuffer,splitLines)
        visibility = SELECT geom FROM unnest(splitArray) as geom WHERE Centroid(geom) = Centroid(line)
    END LOOP
    visbilityCorrection = visbilityCorrection + visibility
    --/ Output is the correction of the visibility based on angluar visibility and visibility in advance
    RETURN Union(visbilityCorrection)
```

VISIBILITY FUNCTION: FINALISE VISIBILITY POLYGONS Algorithm 5.11 is the last part of the visibility function, that outputs the visibility polygons into the visibility table. It takes the calculated visibility polygons from the VisibilityPolygon subfunction and intersects these with the visibility correction polygons of the LineVisCorrection subfunction. The resulting visibility polygons are the visual fingerprints of single landmarks.

Algorithm 5.11: Visibility correction for linear landmarks, and visibility range for points

```
--/ VisibilityPolygon subfunction
IN visibilityPolygon geometry, visibilityCorrection geometry
--/ Create a table that stores the intersection of the
  ↪ calculated visibility through ray-tracing with the
  ↪ visibility that is geometry based.
CREATE TABLE 'visibility' AS (
 SELECTid, visibilityPolygon ∩ visibilityCorrection)
```

5.3.3 Fingerprinting

FINGERPRINTING/FINGERPRINTSET AND COMBINATIONS: NO LONGER IMPLEMENTED
This step was about pre-calculating the combination of visible landmarks, by either recursively subdividing space until there are no overlapping polygons anymore and each visibility polygon contained a unique combination of landmarks (see Figure 5.4 or to calculate the union of the geometry of every combination of landmarks. None of this is implemented any longer, the recursive fingerprint algorithm got strange anomalies with certain landmarks and threw errors with topology relationships. The combination algorithm was unstable and only worked for small test cases with few landmarks. The calculated union of the fingerprint is now done on-the-fly in the processing phase, due to this the time to calculate a location has been increased from 10-15 milliseconds to 25-30 milliseconds, which is still incredibly fast, for localise function see Algorithm 5.12.

At the time of writing the not implemented recursive function can be found at https://stackoverflow.com/questions/15620529/postgis-recursive-intersection-between-polygons or in the Appendix at 18.

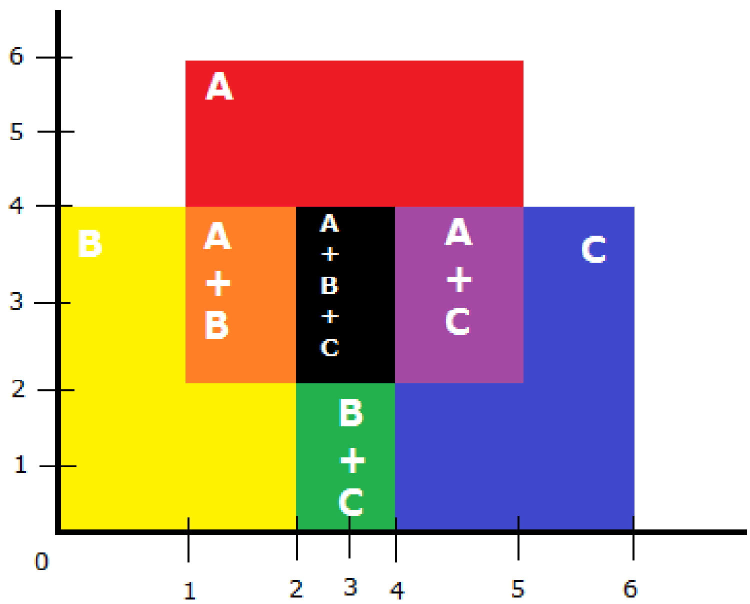


Figure 5.4: Principle subdividing space with unique combinations of landmarks.

5.3.4 Localise function

LOCALISE(OBSERVATION) The third function that the Main Algorithm invokes is the localise function. This localisation function takes in a (series) of observation(s) and interprets the syntax. From the observation, it extracts which landmarks are visible and then queries the visibility table and intersects the visibility polygons to get the area that is visible for a set of landmarks. Algorithm 5.12 has as input an observation and outputs a location.

Algorithm 5.12: Localisation through observations.

```
--/ Localise function, NOTE that PLPGSQL works with 1-based arrays
IN observation text --/ example: 'I see 1, I see 2 AND I can see 2 left of 3'
--/ First step is to split the string into single observations (if it contains multiple) and extract landmarks ids
observationArray = Split(observation, ','|'AND'|';') --/ split on comma, semicolon and 'AND'
landmarkIDs = []
FOR EACH observation IN observationArray LOOP
    wordsArray = Split(observation, ' ')
    IF observation ILIKE 'I see%' THEN
        landmarkIDs = landmarkIDs + wordsArray[3]
    IF observation ILIKE 'I can see%' THEN
        landmarkIDs = landmarkIDs + wordsArray[4] + wordsArray[7]
    IF obseration ILIKE .... THEN
        ...
    END IF
END LOOP

--/ Then retrieve the visibility polygons and intersect them
visibility = (SELECT geom FROM visibility WHERE id = landmarkIDs[1])
FOR n IN 2 .. array_length(landmarkIDs) LOOP
    visibility = visibility ⋂(SELECTgeomFROMvisibilityWHEREid = landmarkIDs[n])
    IF visibility IS NULL THEN --/the resulting intersection is empty
        EXIT/BREAK LOOP
    END IF
END LOOP
RETURN visibility
```

5.3.5 RefineLocation function

REFINELOCATION(LOCATION, OBSERVATION) The refinement of the location is done by interpreting the additional landmark relations that might be given in an observation, The syntax is understood, and the calculated location from localise is split based on the type of relation (Section 4.3 paragraph refinement contains the landmark relation). Algorithm 5.13 shows the interpretation of the observation, and it will draw an infinite line through the two landmarks of the relation or draw a perpendicular line at a given location (see Algorithm 5.14 and 5.15). When the refinement function has to check for left or right, it adds a 3rd point to the landmark line, and check the orientation (clockwise or counter-clockwise using the Clockwise utility function).

Algorithm 5.13: Refinment based on landmark relations.

```
--/ RefineLocation function, NOTE that PLPGSQL works with
 ↪  1-based arrays
IN location geometry, observation text --/ example: 'I can see
 ↪  2 left of 3, I am closer to 3 than 4'
--/ First step is to split the string into single observations
 ↪  (if it contains multiple) and extract refinement actions
observationArray = Split(observation, ','||'AND'||';') --/ split
 ↪  on comma, semicolon and 'AND'
--/ Create a polygon that is the extent of all shapes.
worldBox = Extent(all_shapes)
landmarkIDs = []
FOR EACH observation IN observationArray LOOP
    wordsArray = Split(observation, ' ')
    IF ANY IN wordsArray ILIKE '%left%' THEN
        landmark1 = wordsArray[4]
        landmark2 = wordsArray[7]
        --/ Infinite line subfunction, takes in two landmarks
         ↪  and the worldBox extent and creates a line that
         ↪  goes outside the worldBox
        splitLine = InifniteLine(landmark1,landmark2,worldBox)
        splitArray = Split(worldBox,splitLine)
        --/ split worldBox in two and test if first polygon is
         ↪  left of landmark 1 and 2 (i.e. clockwise)
        IF Clockwise(geom_1,geom_2,Centroid(splitArray[1]))
         ↪  THEN
                    refinePolygon = splitArray[1];
        ELSE --/ if splitArray[1] is not clockwise pick
         ↪  splitArray[2]
            refine_poly = splitArray[2];
        END IF;
    IF ANY IN wordsArray ILIKE '$closer%' THEN
        landmark1 = wordsArray[5]
        landmark2 = wordsArray[7]
        --/ Perpendicular line subfunction, takes in two
         ↪  landmarks and the worldBox extent and creates at a
         ↪  point that goes outside the worldBox
        splitLine =
         ↪  PerpendicularLine(landmark1,landmark2,worldBox)
        splitArray = Split(worldBox,splitLine)
        --/ split worldBox in two and test if first polygon is
         ↪  closer to landmark 1
        IF Distance(polygon,landmark1) <
         ↪  Distance(polygon,landmark2) THEN
                    refinePolygon = splitArray[1];
        ELSE --/ if other part of splt polygon is closer to
         ↪  landmark 1
            refine_poly = splitArray[2];
        END IF;
    IF obseration ILIKE .... THEN
        ...
    END IF
END LOOP
```

INFINITELINE The subfunction that creates an 'infinite' line (Algorithm 5.14) takes two points (landmarks) and calculates the mathematical coefficient (a) of the line through the two landmarks (Δ Y/Δ X) and fills in one of the points to get the b in y = ax * b. Then it will take the minX and maxX or the minY and maxY of worldBox (wBox) depending on the absolute value of coefficient 'a' (¡0.5 use minX,maxX, ¿0.5 use minY,maxY), this line is the output of the function. If aCoefficient is null, the infinite line is a vertical line.

Algorithm 5.14: Mathematical creation of 'infinite' line.

```
--/ InifiniteLine subfunction
IN lm1 geometry, lm2 geometry, wBox geometry

--/ Prevent division by zero
IFΔX! = 0THEN
aCoefficient = ΔY/ΔX
ELSE aCoefficient = null
END IF

IF @ aCoefficient <= 0.5 THEN --/ Shallow line
    point1 = Point(minX(wBox)-1,aCoefficient * (minX(wBox)-1) - aCoefficient * X(lm1) + Y(lm1))
    point2 = Point(maxX(wBox)+1,aCoefficient * (maxX(wBox)+1) - aCoefficient * X(lm1) + Y(lm1))
    infLine = Line(point1,point2)
ELSIF @ aCoefficient > 0.5 THEN --/ Steep line
    point1 = Point(minY(wBox)-1,aCoefficient * (minY(wBox)-1) - aCoefficient * X(lm1) + Y(lm1))
    point2 = Point(maxY(wBox)+1,aCoefficient * (maxY(wBox)+1) - aCoefficient * X(lm1) + Y(lm1))
    infLine = Line(point1,point2)
ELSE --/ IF aCoefficient is null, make vertical line
    point1 = Point(X(lm1),minY(wBox)-1)
    point2 = Point(X(lm1),maxY(wBox)+1)
    infLine = Line(point1,point2)
END IF
RETURN infLine
```

PERPENDICULARLINE The subfunction that creates a perpendicular line (Algorithm 5.15) takes two points (landmarks) and calculates the mathematical coefficient (a) of the line through the two landmarks (Δ Y/Δ X), apply the a1 * a2 = -1, getting the perpendicular a coefficient and then fill in one of the points to get the b in y = ax * b. Then it will take the minX and maxX or the minY and maxY of worldBox (wBox) depending on the absolute value of coefficient 'a' (¡0.5 use minX,maxX, ¿0.5 use minY,maxY), this line is the output of the function. If aCoefficient is null the perpendicular line is a horizontal line, if the a coefficient is zero, the perpendicular line is vertical.

Algorithm 5.15: Mathematical creation of perpendicular line.

```
--/ PerpendicularLine subfunction
IN lm1 geometry, lm2 geometry, wBox geometry

--/ Prevent division by zero
IFΔX! = 0THEN
aCoefficient = ΔY/ΔX
ELSE aCoefficient = null
END IF

IF aCoefficient  != 0  THEN
    pCoefficient = -1 / aCoefficient;
ELSIF aCoefficient = 0 THEN --/ if aCoefficient is horizontal
    pCoefficient = null --/ is vertical
ELSIF aCoefficient IS NULL THEN --/ if aCoefficient is vertical
    pCoefficient = 0 --/ is horizontal
ELSE pCoefficient = null;
END IF;

IF @ aCoefficient <= 0.5 THEN --/ Shallow line
    point1 = Point(minX(wBox)-1,aCoefficient * (minX(wBox)-1) -
      ↪ aCoefficient * X(lm1) + Y(lm1))
    point2 = Point(maxX(wBox)+1,aCoefficient * (maxX(wBox)+1) -
      ↪ aCoefficient * X(lm1) + Y(lm1))
    perpLine = Line(point1,point2)
ELSIF @ aCoefficient > 0.5 THEN --/ Steep line
    point1 = Point(minY(wBox)-1,aCoefficient * (minY(wBox)-1) -
      ↪ aCoefficient * X(lm1) + Y(lm1))
    point2 = Point(maxY(wBox)+1,aCoefficient * (maxY(wBox)+1) -
      ↪ aCoefficient * X(lm1) + Y(lm1))
    perpLine = Line(point1,point2)
ELSE --/ IF aCoefficient is null, make vertical line
    point1 = Point(X(lm1),minY(wBox)-1)
    point2 = Point(X(lm1),maxY(wBox)+1)
    perpLine = Line(point1,point2)
END IF
RETURN perpLine
```

5.3.6 Follow-Up function

FOLLOWUP(LOCATION, OBSERVATION) The follow-up step looks for visibility polygons that intersect with the location that is given, it will give all other intersecting landmark IDs. One way to sort the follow-up landmark is to sort them on how close they are to half the size of the initial location (to eliminate 50

Algorithm 5.16: Provide follow-up landmarks to improve location further.

```
--/ FollowUp function
IN observationIDs, location geometry

idsArray = ARRAY(SELECT ids
                FROM 'visibility' WHERE
                 ↪ Intersects(location,visibilityPolygon)
                ORDER BY @ Area(location) / 2 -
                 ↪ Area(Intersection(location,visibilityPolygon)))

RAISE NOTICE 'Can you see landmark %' %s
 ↪ (array_to_text(idsArray)
```

6 RESULTS AND ANALYSIS

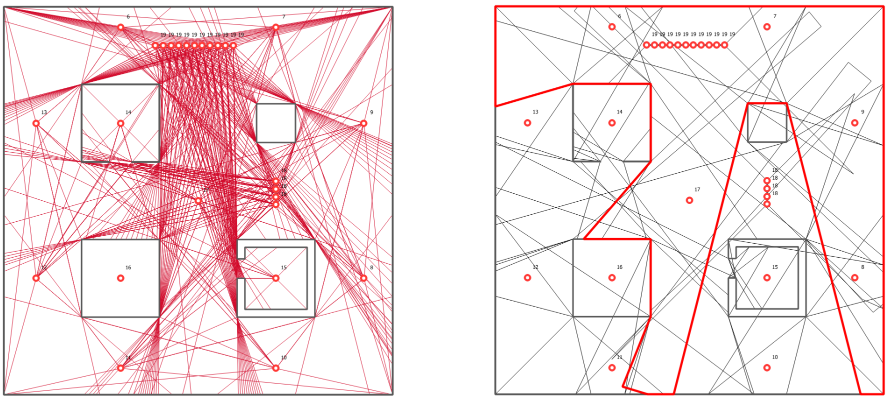
This chapter presents and discusses the results of the implementation (Chapter 5), and evaluates the results along with any specific problems or challenges that occurred while implementing the conceptual framework (Chapter 4).

The chapter is structured with first the results and evaluation of the two (artificial) cases (Section 6.1 and 6.2), then the general challenges of the implementation (Section 6.3), along with challenges of specific parts of the implementated algorithm (Section 6.4), and finally with the still open challenges (Section 6.5).

The implementation is built upon an artificial case aimed at getting the MPVs up and running, then the MPVs were tested, trialled and evaluated with the Faculty of Architecture (Bouwkunde) case (BK case).

6.1 RESULTS OF THE INITIAL ARTIFICIAL CASE

For the initial case, the system is prepared through an initialisation command, Figure 6.1 shows the rays that are cast to all corner points, and the visibility polygons (equals single landmark fingerprints) with the visibility polygon of landmark 19 highlighted.



(a) Ray-tracing results

(b) Visibility results

Figure 6.1: Results of the intialisation/pre-processing (runtime: 2.4 sec).

Then when the system is prepared to take in observation and localise a user, observations are fed into the system, three examples of commands that provide a location are displayed in code snippet 6.1 and the corresponding locations are shown in Figure 6.2. Observation #1 where landmarks 7, 8, and 17 are visible are marked in red, observation #2 where landmark 11, 12, and 13 are visible are marked in green, and observation #3 with landmark 18 and 14 visible is marked blue.

The initial artificial case works like a charm without any faults, anomalies, and above all runs fast. The initial case resulted in a working MPV, this **mvs!** (**mvs!**) is later evaluated with the BK case.

Algorithm 6.1: Inputting machine-readable observations into the algorithm

```
---/ Observation #1
SELECT MainAlgorithm('localise', observation:='I see 7, I see 8,
  ↪ i see 7 right of 17')
---/ Observation #2
SELECT MainAlgorithm('localise', observation:='I see 11, I see
  ↪ 12, i see 13')
---/ Observation #3
SELECT MainAlgorithm('localise', observation:='I see 14, I see
  ↪ 18')
```

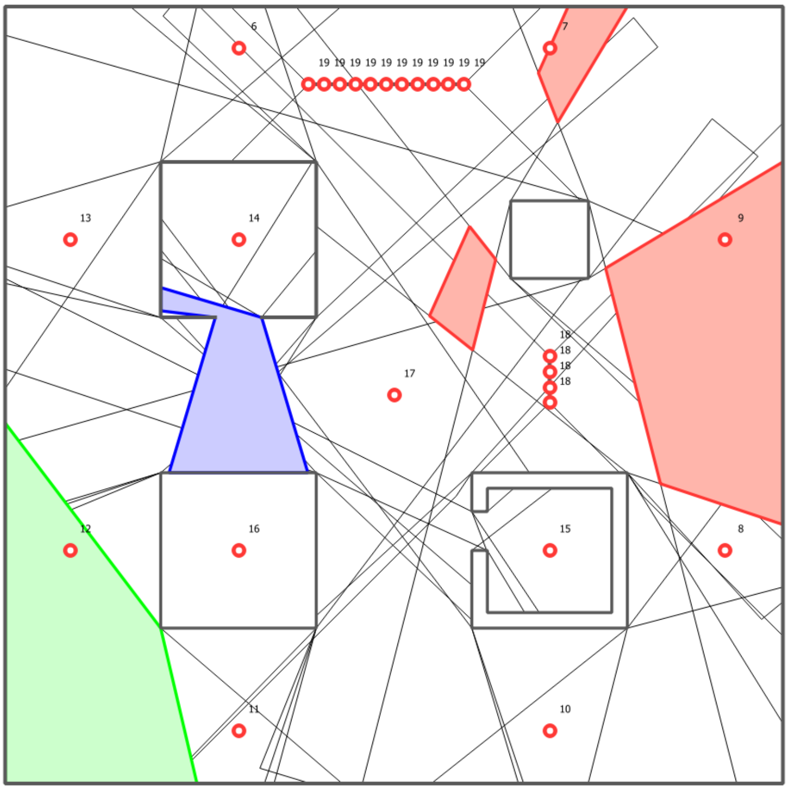


Figure 6.2: Resulting locations from observations

6.1.1 Performance of algorithm

The performance of the artificial case is evaluated with speed shown in Table 6.1 and statistics of the case are given in table 6.2. The initialisation step was a lot slower when the combination+fingerprinting step was included, and initialisation took 30 seconds compared to 2.2 seconds without, by removing the combination+fingerprinting step the speed of localisation went from 10-15 milliseconds to 25-30 milliseconds.

Table 6.1: Performance: speed

Task	Time	% of Time
Initialisation	2.2 secs	99%
Localisation	25 msec	1%

Table 6.2: Performance: stats

Object	Amount
Obstacles	5
Landmarks	14
Landmark points	27
Rays	509
Visibility polygons	13

6.2 RESULTS OF THE FACULTY OF ARCHITECTURE CASE

The BK case is more complex, has landmarks that are 'poorly' located in the environment (very close to a wall, or clustered together), or there was a lack of landmarks in a certain area. This case is meant to test and evaluate the MPV that was constructed for the initial case. The case also implemented the limited visibility range of landmarks, and these are picked randomly between 10 and 20 meters per landmark.

With minor adjustments to the implementation, the algorithms could be run for the BK case. However, some results of the algorithms are not as perfect as they were in the initial case. The ray-tracing succeeded as expected (Figure **??**), but the creation of the visibility polygon has a few shortcomings. When creating the visibility/isovist polygons out of the rays, where rays have the same angle, irregularities occur where the stitching of the points gives a self-intersecting polygon or edges that go through a wall. These irregularities are not stopping the initialisation from finishing but result in incorrect visibility and when localising a user at incorrect (overestimated) locations.

(a) Ray-tracing results

(b) Visibility results

Figure 6.3: Results of the intialisation/pre-processing (runtime: 9:30 min).

Where the test case had just random landmarks, the BK case has references to actual objects, the three example observations show how the user would interact with the system if object identification were integrated (out of the research scope). The corresponding locations of the observations are shown in Figure 6.4. Observation #1 where landmarks 79 (linear object: geomatics information banner), and landmark 80 (linear object: architecture information banner) are visible are marked in red, observation #2 where landmark 61 (agent point: lecturer), landmark 1 (point object: clock), and landmark 21 (linear object: room B) are visible are marked in green, and observation #3 with landmark 53 (EspressoBar), landmark 69 (BK-Expo), and landmark 70 (coffee corner) visible is marked blue.

6.2.1 Performance of algorithm

The performance of the BK case, for a resulution with subdivision of 1.0 and 0.5 meters, is evaluated with speed shown in Table 6.3 and statistics of the case are given in table 6.4.

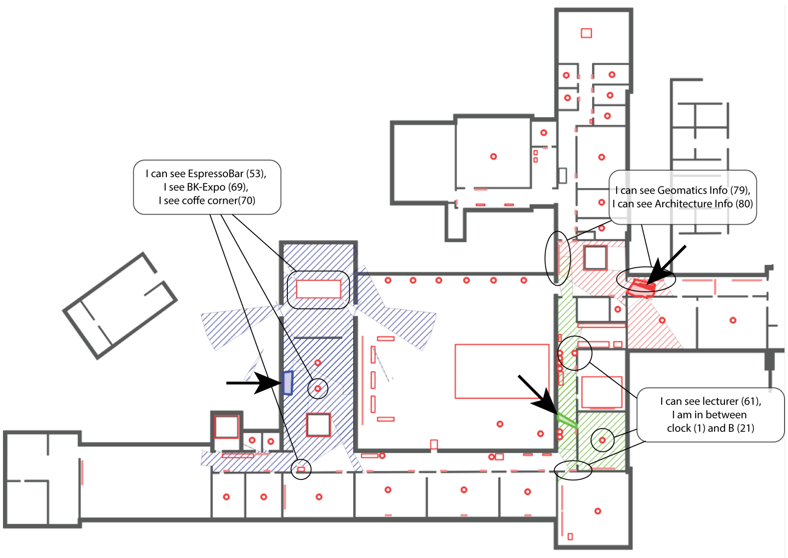


Figure 6.4: Resulting BK case locations from observations

Table 6.3: BK case: performance speed

Task	Time	% of Time
Initialisation (res: 1.0)	9:30 mins	99,9%
Initialisation (res: 0.5)	14:38 mins	99,9%
Localisation	25 msec	>0.1%

Table 6.4: BK case: performance stats (resolution 0.5)

Object	Amount
Obstacles	26
Landmarks	27
Landmark points	884
Rays	23231
Visibility polygons	117

6.3 GENERAL CHALLENGES WHEN IMPLEMENTING THE FRAMEWORK

This section talks about the general challenges that occurred during the implementation of the conceptual framework. These challenges were not specific to one part of the implementation but played a role throughout the whole process.

PRECISION One of the challenges with geometry is decimal precision. For various actions in the implementation, there is a check to relate objects (intersections) mathematically. Since the ray-tracing step involves extending lines beyond a corner based on a definite decimal angle, often the extended ray no longer passes through the corner. This result in incorrect intersections (there is no longer an intersection with the boundary of an obstacle, or there is an intersection with the interior). Figure 6.5 shows an example of a precision error when extending lines. A workaround this problem was to find the end point of the extended line and rather than make a line from the landmark to the endpoint, make two connected lines with a shared point at the corner. The same problem of precision happened when intersecting the visibility polygons into fingerprints. When having these slightly off corner lines, spikes and sliver polygons are the results of the intersections, providing errors of self-intersection and terminating the algorithm. A workaround was to constantly snap the vertices of polygons to the obstacles and making sure that the rays were as problem free as possible.

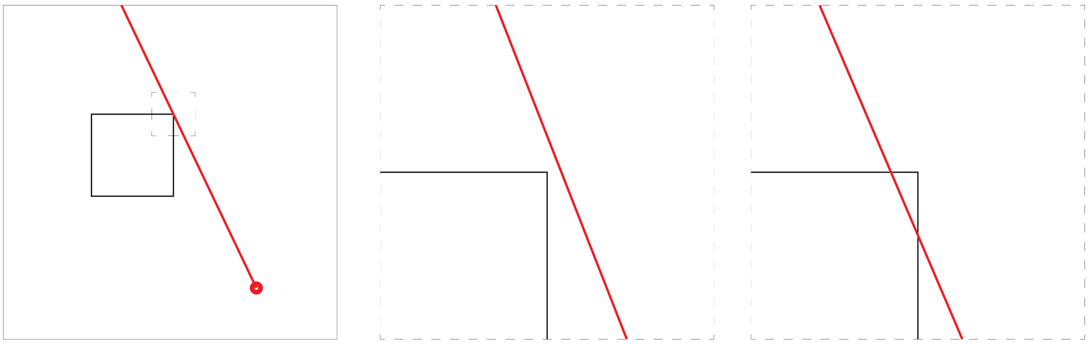


Figure 6.5: False appearance of ray going through corner

9-DEM INTERSECTION MODEL Related to the decimal precision challenge, there was also a challenge in topology relations. The implementation relies a lot on checking if objects are spatially related to one another(if they share a point on the boundary if they touch or overlap), due to the variety of objects, landmarks and in between products, the intersection has to deal with a lot of different variations. Sometimes an intersection works for one combination of geometries but fails for another combination, really making sure that geometry follows the rules and assumptions are therefore of the utmost importance. Figuring out workarounds for these issues is sometimes a temporary fix, sometimes a permanent one.

6.4 SPECIFIC CHALLENGES WHEN IMPLEMENTING THE FRAMEWORK

The research and implementation also had a few specific challenges to get from a conceptual framework to an implemented and working algorithm. This part highlights some of the challenges and solutions to specific parts of the implementation.

HAVING 2-DIMENSIONAL OBSTACLES When the rays are cast to all corners of all the obstacles it is hard to distinguish points that are on the inside of a corner or the outside if the obstacles are 1-dimensional linear objects. The easiest way to see if a ray goes past an object or against an object is for the object to have an interior; if an extended ray never touches the interior of an obstacle, it is not blocked. So obstacles that had no interior had to get an interior, no matter how small (besides the limitations of decimal precision), obstacles had to have one (by definition). Drawing a buffer around an obstacle proved a simple solution, however, this created a problem with objects that were simultaneously a landmark and an obstacle. Setting additional rules on the placement of landmark objects and obstacles would be a quick fix, but decreases the robustness of the method. For that purpose linear landmarks are for now not used also as obstacles to prevent this challenge. Another solution would be to also draw a slightly more extensive buffer around the landmarks that are also an obstacle to make them outside of the obstacle. This does create slightly more computationally heavy landmarks. Considering the performance of the algorithm, this approach only affects the pre-processing step and not the on-the-fly localisation. The BK case had two significant differences with the initial case, the obstacles are all polygons shapes, so there is no need to ensure obstacles have an interior, however, due to the complexity of the obstacles, the number of corner points present in the case was that much that the combinatorial function would run out of memory.

visibility approximation of landmarks One of the challenges of calculating the visibility of linear landmarks was that the landmark is visible along the full length of the line. Subdividing the line can compensate for this phenomenon. However, determining the correct length to split the line proves difficult. Ideally, the line is broken up into infinitely small segments. This, however, is computationally heavy and impossible. Finding the golden ratio of line segment length and computationally fast, is something that this research did not go in depth into, but a sufficiently short length was selected that would mostly fix the challenge of the linear landmark visibility. The current implementation also assumes that as long as one of the points of a linear landmarks is visible that the whole landmark is visible, it does not include a threshold of how much a linear landmark has to be visible for it to be visible (so-called fuzzy visibility). So finally there is the aspect that (linear) landmarks that are visible through a small gap between two walls are considered visible. The current implementations define visibility as the region that is not blocked by an object, even though that might mean that the landmark is only visible through a keyhole. On the other hand, it is better to assume a location of a user (even one highly unlikely) than excluding a location where a user actually is (if he is looking through that keyhole).

impact of geometry on visibility A points is assumed to be visible from all directions as a computationally light object, however depending on the dimensionality of an object, it is only visible in the dimensions perpendicular to where an object has a dimension (a line, for example, is only visible from its sides in 2D). Using these geometric constraints, the geometry of a landmark has to be adapted to fit with how the landmark is visible. Linear objects might be represented as polygons to be visible from all sides, or an object that had physical dimensions as a polygon might be represented as a line if it is only visible from the front or back. There might even be landmark specific obstacles that block certain parts of the visibility of a landmark (if an object has a front and not a back). An initial recommendation on how to model a landmark geometrically is provided in Chapter 4, 360-degree visibility, visibility of only a back/front side and the size should determine the right geometry.

6.5 STILL OPEN CHALLENGES

This section will discuss the limiting assumptions and shortcomings of the implementation, how each could be dealt with and where there is room for improvement.

using an existing dataset Due to the position of this research within the broader field of localisation, this research did not go into the compatibility of spatial standards (such as CityGML) and a landmark-based approach. The research assumed that there is a step that automates the conversion of the real world into a digital equivalent that can run the proposed framework (or that such technology will be developed in the short future). The artificial case is a case that was constructed from scratch by hand to best resemble a real-world case. Then a real-world case was manually drawn from floorplans of the Faculty of Architecture, these were used to test the MPVs, but still, lack some aspects of a real-real-world case. Then an existing floor map would have to be digitised even more meticulously since there is not a standard data format that can be used. The aim of this research was not to use an existing standard and see what could be done with that; instead, it went into researching a (new) method for localisation.

LANDMARK SALIENCE The implementation did not incorporate all aspects of salience. However, salience could have an added value in creating a hierarchy of objects that should be used over other objects, machine-learning could even help out in learning which objects are most referred to. The current implementation assumes the equal importance of all landmarks, but could relatively easily be extended to include salience. Salience plays an essential part in the interaction of a user with the context, so when asking about observations salience could and should be included. The implementation does cover the uniqueness, visibility in advance to better calculate the visibility.

REFINEMENT OF THE LCOATION One of the key principles of the conceptual framework that proved more challenging than initially estimated is the refinement of the location. Being restricted to the packages and libraries of the chosen tools and languages, this step needed many workarounds, hindering the process of implementation, examples are the lack of vector objects in PostGIS, and inconvenient translation (moving objects on a plane).

IMPROVING FOLLOW-UP QUERIES AND INTERACTION WITH USER When following-up on an observation several factors should be taken into account: can the user understand the questions that is asked to it by the system, can the user adequately answer the question in a way that the system understands it, and has one landmark prevalence over another (i.e. does salience make a difference in the interaction), now the focus was to try to reduce the area as much, instead can ask for more salient landmarks first.

USING JUST 2D RATHER THAN 3D Though out of the scope, the determination of the visibility of objects is a lot more accurate and lifelike in 3D, the 2D environment cannot distinguish between an object that partially obscures a landmark from a certain height, and 2D can only represent objects abstractly. The downside of a 3D environment is that it comes with its own set of challenges and problems and functions that seise to exist in the 3rd dimension, making visibility polyhedrons rather than polygons makes the process a lot more complicated. However, the used methods in this research could be extended to 3D by increasing the dimensionality of the methods, for example rather than casting a ray to the point of an obstacle in 2D, in 3D rays should be cast to the edges of obstacles and extended. 3D is the next-level step, but it is not impossible.

7 CONCLUSION AND RECOMMENDATION

The objective of this thesis is to explore whether a pure-landmark based approach can be used for indoor localisation. Where through observations of the environment the location of a user can be determined. Before rushing to the conclusion this chapter will give answers to the sub-research questions first (Section 7.1 to ultimately conclude with an answer to the main research question (Section 7.2. Then the conclusion is discussed and evaluated for the field of geomatics. And lastly, a recommendation is written that provides areas where improvements can be gained along with highlighting topics that need further research.

7.1 ANSWERS ON RESEARCH QUESTIONS

In this section answers will be provided for the sub-research questions that were proposed in Section 1.3 that form the basis for the answer to the main research question.

- *Within the indoor environment what can be considered a landmark for localisation (and navigation) and how does object salience plays it is part?*

  Following the definitions of various researchers outlined in Section 2.1, objects are landmarks if they are usable for navigational clues, are distinguishable from their surroundings and can be remembered and referenced to by users. However, this is the definition for navigation since there is no need to include them in directions, or remember, or reference them, for localisation objects only have to be distinguishable from their surroundings. Resulting in the definition posted in Section 4, that specifies that an "Indoor landmark could be any objects, agent or resource, that is visible from one to tens of meters and further, and that is distinguishable from its context/surroundings."

  Typical objects can be classified as (semi-)static. Two special cases of object here are agents, these are moving persons that can be considered moving/-dynamic objects, and resources, these are non-physical/dynamic objects like functional regions or destinations.

- *Which landmark parameters are most salient for indoor localisation?*

  As discussed in Section 2.1, Kattenbeck assesses the overall salience of landmark using five criteria (visual salience, cognitive salience, structural salience, visibility in advance and prototypicality), in Section 4 the salience of a landmark is extended with uniqueness. An essential type of salience is visual salience, and this determines the potential that an object is spotted/visible. After that prototypicality is most important, does the object look like it should look or typically looks? Then uniqueness is essential, is the landmark one-of-a-kind or is it a generic object, this is important when communicating with the user. And lastly cognitive and structural salience, these determine how easy a landmark is remembered and can be incorporated into navigation.

  Salience is commonly assumed as the defining factor in determining whether an object is labeled as a landmark in the first place, but because of the definition of an indoor landmark (Section 4), any object in the indoor environment is a **landmark to some extent**, creating levels of hierarchy is more important than deciding if an object is a landmark.

- *How can a landmark be semantically and geometrically modelled in a database schema? And what is this schema's influence on the visibility calculation?*

  Landmarks are objects that are identifiable in the surroundings, these observations are given to the system where the system has an identifier for each landmark or each category of objects (according to the definition Subsection **??**). 1) The level of uniqueness combined with prototypicality decides whether an object it gets a collective identifier from a group/family of objects or whether gets a unique identifier if it is a one of a kind or if it an odd representation (not prototypical) of an object. 2) Information on the overall salience should be stored to determine hierarchy for the interaction with users, including shape, size, color, etc. 3) Information regarding the visibility of a landmark should be stored, such as the visibility in advance of an object or at which angle a linear landmark is visible, these impact the visibility calculation (as demonstrated in the framework, Chapter **??**).

  The geometry of a landmark can be any of the joint OGC compliant geometry types that are supported (PostGIS supported geometry types for the implemented algorithm), such as points, linear or area geometry types. However, what should be noted is that when geometry dictates the visibility, landmarks should be represented in the geometry that fits the visibility behaviour of the landmark, rather than how the landmark looks/is represented in the real world. If a landmark is only visible on either side, consider using a line representation, if a landmark is visible 360 degrees, represent it as a point (see Chapter **??**, Table 4.2).

- *Which localisation principle shows potential for a pure landmark-based approach?*

  Triangulation needs 2 or more landmarks, trilateration requires 3 or more landmarks, fingerprinting needs 1 or more landmarks but could also give a location if no landmarks are visible (if you know where something is visible, the lack of visibility is also a location). Due to focus on a localisation method independent from sensors and therefore the assumed lack of measuring equipment, only the Boolean indication of whether a landmark is visible or not is available for the user. Fingerprinting is the only method that can localise using just the visibility of the environment to match an observation to a visual fingerprint. The user is able though to tell relative distances and rudimentary estimation of the angles between objects (left/right, in front or behind). These extra observations allow the fingerprinting method to be extended with adjustments based on rudimentary observations (see Figure 4.12 and 4.13).

  Fingerprinting is the preferred method for landmark-based localisation, due to its support of just visual observations. With the extended use of the location rather than position acquisition from triangulation and trilateration, the location from fingerprinting can be refined and improved.

- *Can the relative position of a subject in relation to landmarks be used to improve localisation?*

  As just mentioned, the relative location of the subject in relation to landmarks, can indeed help to improve the localisation, through narrowing down the area where a user could be estimated in. Knowing where a person is or is not, or what the user can see or not, the location can be narrowed down in the case of fingerprinting.

- *Does the number, and constellation of landmarks influence accurate localisation?*

The fingerprint in itself can be considered as a functional region according to the definition of fingerprint and functional region in Subsection 4.1.5. If the constellation of landmarks is such that the fingerprints are oddly shaped the constellation might be inadequate. The definition of 'odd', however, depends on the use of the functional regions and the user's calculated location. If through observations the user's location results in a small enough area to provide LBS (such as providing information or navigating a user around), the constellation works.

There is not necessarily a set of rules for a right or wrong constellation, besides the general requirement that enough landmarks are required to get a fully fingerprinted indoor environment. In some environments a constellation with most of its landmarks on the edges might work, while in another environment a more spread out constellation provides the best result. It is advisable to add landmarks to a constellation if certain areas are not evenly fingerprinted. In the BK case the landmarks were clustered or against the wall, this resulted in 'messy' fingerprints, a lot of overlap or

7.2 CONCLUSION

When assessing the potential of a pure landmark-based approach for indoor localisation from the sub-research questions, and looking at the implementation of the conceptual framework, the main research question can be answered:

How can a pure landmark-based approach achieve adequate indoor localisation?

Using the visibility of landmarks allows for the indoor environment to be fingerprinted, and the visibility of different landmarks can be cross-referenced (intersecting overlapping visibility). Knowing where a particular combination of landmarks is visible, and simultaneously knowing where each landmark is not visible, allows the localisation of a user, if the user can tell the system what it sees in its surroundings, or if the system asks the user if it can see a landmark and narrow down the location based on the answer.

The more the system knows about objects within the indoor environment, the more accurately the user can observe and describe the environment, and the better the system is able in interacting with the user, the more likely it is that a sufficiently accurate and precise location can be provided for LBS. This is also the step where machine-learning can prove to be invaluable, learning how landmarks are observed and used, can improve interaction.

The aim of this research is not to give the most accurate and precise location, but to provide a location that has to be specific enough that it can be used for orientation and navigation in the future. The fingerprint, considered as functional regions, could provide useful for navigation, where a user navigates between different fingerprints, from landmark to landmark, through the context, in an immersive manner.

The landmark-based indoor localisation approach of this exploratory research had the objective to look for a possible alternative for sensor-based indoor localisation. The newly proven approach paves the way for future researchers to exploit the power of landmarks and indoor localisation thoroughly, the potential of the core concepts are tested, and the field of localisation can add a new branch to its tree.

7.3 RECOMMENDATION

This research was of an exploratory nature, by proving that indoor localisation is achievable when using a pure landmark-based framework, it looked into the potential of landmarks, and it implemented ideas on how to use the context. However, the research still leaves room for improvement and starting points for new research. This section will discuss some of the recommendations on the potential improvements and will provide leads for future studies.

7.3.1 Improvement of the reserach

There are several components of the research that could be further implemented, and there are some concepts that can be incorporated.

- Using a dataset in a spatial standard and use that as a real-world case rather than an artificial case or the manually digitised 'real-world' case would make the proposed framework more relevant, by demonstrating that it can solve real-world challenges as a stepping stone to more widespread use and further research.
- Salience is mostly a tool to label objects as landmarks, for the indoor environment and localisation landmarks have a different meaning. Introducing 'traditional' components of landmark salience into the conceptual framework would flesh out their use better, so far the uniqueness of the objects is mainly modelled in the identifier of the landmarks. Additionally, a few rules for the visibility are explored (range and angle), but these could better be exploited for a hierarchy structure and a better user experience.
- The implementation of the location refinement is one of the first steps to try to use the full potential of a landmark-based approach for indoor localisation. By drawing inspiration from the proposed concepts of using a rudimentary estimation of angle and distance, these can be extended, and the relative location of a user to context could be improved.
- The interaction between system and user should be further implemented, taking into account a hierarchy of landmarks to give the user the optimal queries to improve location.
- Extending the conceptual framework to 3D would approach the real-world more closely, certain aspects of the indoor environment cannot be abstracted to 2D space, such as obstacles and the limitation of 'eye-level' visibility.

7.3.2 Future research

This research made one of the first attempts to use landmarks for indoor localisation. Further research could improve utilisation and incorporation of landmarks in existing methods, rather than add landmarks as a gimmick to existing methods.

Topics that could be further researched are:

- Automatic detection and labelling of landmarks in indoor environment models such as floor plans, CityGML and BIM to be used in landmark-based approaches.
- Context-aware navigation using landmarks as an alternative to expensive location engineered sensor networks.
- Research into the interactivity of the environment using landmark-based information services.

BIBLIOGRAPHY

Anacta, V. J. A., Humayun, M. I., Schwering, A., and Krukar, J. (2017). Investigating representations of places with unclear spatial extent in sketch maps. In *International Conference on Geographic Information Science*, pages 3–17. Springer.

Batty, M. (2001). Exploring isovist fields: space and shape in architectural and urban morphology. *Environment and planning B: Planning and Design*, 28(1):123–150.

Benedikt, M. L. (1979). To take hold of space: isovists and isovist fields. *Environment and Planning B: Planning and design*, 6(1):47–65.

Brunner-Friedrich, B. and Radoczky, V. (2005). Active landmarks in indoor environments. In *VISUAL*, pages 203–215. Springer.

Caduff, D. and Timpf, S. (2008). On the assessment of landmark salience for human navigation. *Cognitive processing*, 9(4):249–267.

Chan, S. and Sohn, G. (2012). Indoor localization using wi-fi based fingerprinting and trilateration techiques for lbs applications. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(4):C26.

Chowaw-Liebman, O., Christoph, U., Krempels, K.-H., and Terwelp, C. (2010). Evaluation of an indoor navigation approach based on approximate positions. In *2010 International Conference on Wireless Information Networks and Systems (WINSYS)*.

Chun, K. and Kim, N. (2003). Vehicle navigation network, apparatus and method for use in a mobile telecommunication system. US Patent 6,532,418.

Conditt, J. (2017). Google lens is a powerful, ai-driven visual search app.

Curran, K., Furey, E., Lunney, T., Santos, J., Woods, D., and McCaughey, A. (2011). An evaluation of indoor location determination technologies. *Journal of Location Based Services*, 5(2):61–78.

Diakité, A. A., Damiand, G., and Gesquiere, G. (2014). Automatic semantic labelling of 3d buildings based on geometric and topological information. In *Proc. of 9th International 3DGeoInfo Conference (3DGeoInfo)*. Karlsruhe Institute of Technology.

Gezici, S., Tian, Z., Giannakis, G. B., Kobayashi, H., Molisch, A. F., Poor, H. V., and Sahinoglu, Z. (2005). Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE signal processing magazine*, 22(4):70–84.

Golledge, R. G. (1999). Human wayfinding and cognitive maps. *Wayfinding behavior: Cognitive mapping and other spatial processes*, pages 5–45.

Graser, A. (2017). Towards landmark-based instructions for pedestrian navigation systems using openstreetmap. *AGILE 2017 Conference on Geographic Information Science, Wageningen, The Netherlands*.

Groh, B. H., Friedl, M., Linarth, A. G., and Angelopoulou, E. (2014). Advanced real-time indoor parking localization based on semi-static objects. In *Information Fusion (FUSION), 2014 17th International Conference on*, pages 1–7. IEEE.

Hightower, J. and Borriello, G. (2001). Location systems for ubiquitous computing. *Computer*, 34(8):57–66.

Ibisch, A., Stümper, S., Altinger, H., Neuhausen, M., Tschentscher, M., Schlipsing, M., Salinen, J., and Knoll, A. (2013). Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded lidar sensors. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 829–834. IEEE.

Kattenbeck, M. (2017). How subdimensions of salience influence each other. comparing models based on empirical data. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 86. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Kim, J.-S., Yoo, S.-J., and Li, K.-J. (2014). Integrating indoorgml and citygml for indoor space. In *International Symposium on Web and Wireless Geographical Information Systems*, pages 184–196. Springer.

Kim, Y.-H., Rana, S., and Wise, S. (2004). Exploring multiple viewshed analysis using terrain features and optimisation techniques. *Computers & Geosciences*, 30(9):1019–1032.

Klippel, A. and Winter, S. (2005). *Structural Salience of Landmarks for Route Directions*, pages 347–362. Springer Berlin Heidelberg, Berlin, Heidelberg.

Krūminaitė, M. and Zlatanova, S. (2014). Indoor space subdivision for indoor navigation. In *Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, pages 25–31. ACM.

Kummerle, R., Hahnel, D., Dolgov, D., Thrun, S., and Burgard, W. (2009). Autonomous driving in a multi-level parking structure. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3395–3400. IEEE.

Litomisky, K. and Bhanu, B. (2013). Removing moving objects from point cloud scenes. In *Advances in Depth Image Analysis and Applications*, pages 50–58. Springer.

Liu, H., Darabi, H., Banerjee, P., and Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080.

McKinlay, R. (2016). Technology: Use or lose our navigation skills. *Nature*, 531(7596):573–575.

Michon, P.-E. and Denis, M. (2001). When and why are visual landmarks used in giving directions? In *International Conference on Spatial Information Theory*, pages 292–305. Springer.

Montello, D. R. (2005). Navigation. In University, C., editor, *The Cambridge Handbook of Visuospatial thinking*, pages 257–294. Cambridge University Press.

Mortari, F., Zlatanova, S., Liu, L., and Clementini, E. (2014). " improved geometric network model"(ignm): a novel approach for deriving connectivity graphs for indoor navigation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(4):45.

Muthukrishnan, K., Lijding, M., and Havinga, P. (2005). Towards smart surroundings: Enabling techniques and technologies for localization. In *International Symposium on Location-and Context-Awareness*, pages 350–362. Springer.

Pahlavan, K., Li, X., and Makela, J.-P. (2002). Indoor geolocation science and technology. *IEEE Communications Magazine*, 40(2):112–118.

Peterson, B., Bruckner, D., and Heye, S. (1997). Measuring gps signals indoors. In *International Association of Institutes of Navigation. World congress*.

Presson, C. C. and Montello, D. R. (1988). Points of reference in spatial cognition: Stalking the elusive landmark. *British Journal of Developmental Psychology*, 6(4):378–381.

Ramirez, L., Dyrks, T., Gerwinski, J., Betz, M., Scholz, M., and Wulf, V. (2012). Landmarke: an ad hoc deployable ubicomp infrastructure to support indoor navigation of firefighters. *Personal and Ubiquitous Computing*, 16(8):1025–1038.

Raubal, M. and Winter, S. (2002). *Enriching Wayfinding Instructions with Local Landmarks*, pages 243–259. Springer Berlin Heidelberg, Berlin, Heidelberg.

Redish, A. D. (1999). *Beyond the cognitive map: from place cells to episodic memory*. MIT Press.

Richter, K.-F. (2013). Prospects and challenges of landmarks in navigation services. In *Cognitive and Linguistic Aspects of Geographic Space*, pages 83–97. Springer.

Richter, K.-F. and Klippel, A. (2004). A model for context-specific route directions. In *International Conference on Spatial Cognition*, pages 58–78. Springer.

Sithole, G. and Zlatanova, S. (2016). Position location, place and area: An indoor perspective. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 4:89–96.

Sobel, D. (1998). A brief history of early navigation. *Johns Hopkins APL technical digest*, 19(1):11.

Stankiewicz, B. J. and Kalia, A. A. (2007). Acquistion of structural versus object landmark knowledge. *Journal of Experimental Psychology: Human Perception and Performance*, 33(2):378.

Tandy, C. (1967). The isovist method of landscape survey. *Methods of landscape analysis*, pages 9–10.

Turner, A., Doxa, M., O'sullivan, D., and Penn, A. (2001). From isovists to visibility graphs: a methodology for the analysis of architectural space. *Environment and Planning B: Planning and design*, 28(1):103–121.

Turner, A. and Penn, A. (1999). Making isovists syntactic: isovist integration analysis. In *2nd International Symposium on Space Syntax, Brasilia*.

Vasardani, M., Timpf, S., and Schinner, P. (2017). Is a door a landmark?

Vo, Q. D. and De, P. (2016). A survey of fingerprint-based outdoor localization. *IEEE Communications Surveys & Tutorials*, 18(1):491–506.

Warner, J. (2008). Wi-fi reception communications building main floor. [Online; accessed October 27, 2017].

Werner, M. (2014). *Indoor location-based services: Prerequisites and foundations*. Springer.

Winter, S., Tomko, M., Vasardani, M., Richter, K.-F., and Khoshelham, K. (2017). Indoor localization and navigation independent of sensor based technologies. *SIGSPATIAL Special*, 9(1):19–26.

Xiao, W., Ni, W., and Toh, Y. K. (2011). Integrated wi-fi fingerprinting and inertial sensing for indoor positioning. In *IPIN*, pages 1–6.

Xu, M., Wei, S., and Zlatanova, S. (2016). An indoor navigation approach considering obstacles and space subdivision of 2d plan. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 339–346.

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151. IEEE.

Zlatanova, S., Liu, L., and Sithole, G. (2013). A conceptual framework of space subdivision for indoor navigation. In *Proceedings of the Fifth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, pages 37–41. ACM.

A RECURSIVE SUBDIVISION OF SPACE FOR FINGERPRINTING VISBILITY

StackOverflow: asked Mar 25 ’13 at 16:58 by StackOverflow user: Eggplant

I am trying to perform a recursive intersection between all the Polygons in a spatial Table, and obtain the resulting (multi)polygons and the information about every intersection for each of them.

An image (not really in scale) to explain it: Example

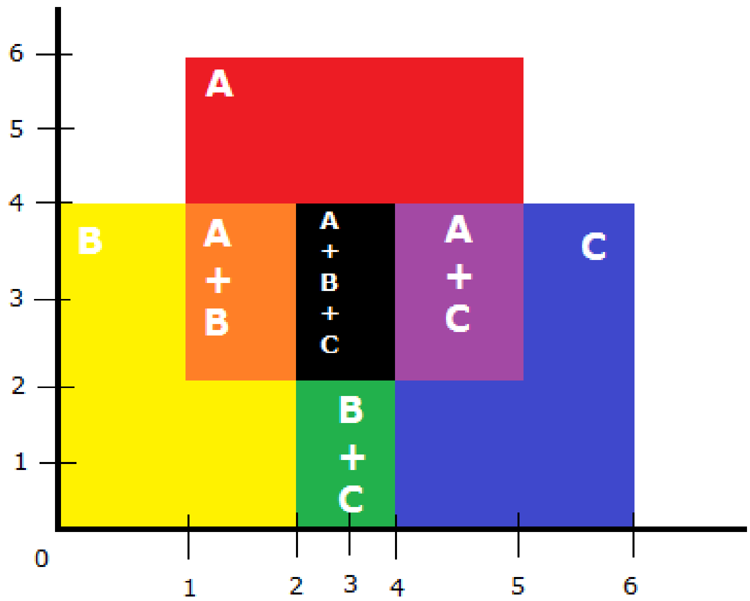


Figure A.1: Principle of resursively subdividing space with unique combinations of landmarks.

Let’s say there are A, B, C squares in a table. I would like to have A, B, C, A+B, A+C, B+C, A+B+C polygons in output, and I need to know that A+B is the intersection of A and B and so on.

Answered Mar 25 '13 at 23:32 by StackOverflow user: Jakub Kania

```
https://stackoverflow.com/questions/15620529/postgis-recursive-intersection-between (accessed: 28th of October 2017)
    WITH RECURSIVE
source (rownum, geom, ret) AS (
    SELECT row_number() OVER (ORDER BY name ASC), ST_Multi(geom), ARRAY[name] FROM test
),
r (rownum, geom, ret, incroci) AS (
    SELECT rownum, geom, ret, 0 FROM source
    UNION ALL
    SELECT s.rownum, ST_CollectionExtract(ST_Intersection(s.geom, r.geom), 3), (r.ret  s.ret), (r.incroci + 1)
        FROM source AS s INNER JOIN r ON s.rownum > r.rownum AND ST_Intersects(s.geom, r.geom) AND ST_Area(ST_Intersection(s.geom, r.geom)) > 0.5
),
result (geom, ret) AS (
    SELECT ST_Difference(ST_Union(r.geom),q.geom) AS geom, r.ret FROM r JOIN (SELECT r.ret,ST_UNION(COALESCE(r2.geom,ST_GeomFromText('POLYGON EMPTY'))) as geom FROM r LEFT JOIN r AS r2 ON r.ret<@r2.ret AND r.ret!=r2.ret GROUP BY r.ret) AS q on r.ret=q.ret GROUP BY r.ret,q.geom
)
SELECT geom, ST_Area(geom) AS area, ret FROM result ORDER BY ret
```

B P4 REFLECTION

- *The relationship between the methodical line of approach of the Master Geomatics and the method chosen by the student in this framework.*

  During this research, the focus will be on developing a proof of concept for indoor localisation based on landmarks and writing a future work recommendation for landmark-based navigation. The research will have an artificial test case that represents a real-world case in all its possibilities and variations. The starting point is a case with landmarks within the indoor environment, in different constellations, where there is a floor plan of the indoor environment with obstacles. The case considers that there is (existing) technology/human/device that through visibility can recognise objects (the user know the system's identifier of the object and is not able to measure angles or distances), where the input observations are conceptualised and explored to fit the principle of landmark-based indoor localisation. This will be research into landmarks, and how they can be made to work in existing indoor localisation principles or if a new principal has to be conceptualised, indoor localisation and indoor navigation are tied together where the combination of indoor localisation and indoor navigation results in a real-time/dynamic navigation. The focus of this research within the broader picture of indoor navigation will be on solving the indoor localisation.

- *The relationship between the conducted research and application of the field geomatics.*

  Within the field of localisation and positioning there is already a lot of work done in positioning centred around high precision systems (GNSS but also beacons such as Wi-Fi, Bluetooth and NFC) due to the availability and market of these systems, however, the field of contextual localisation and navigation has not seen as much interest, both for the outdoor and the indoor environment. The indoor and often open-space environment have their own added challenges. Solving (or the attempt of solving) the challenge of (open-space) indoor localisation and navigation based on context without a network of sensors that is engineered specifically for a location, could be the next leap for providing the same services indoor that are already commonly relied upon outdoors. Renowned researchers Stephen Winter and Kai-Florian Richter recently (2017) emphasised the need for indoor localisation and navigation independent of sensor-based technologies, finding new ways using different concepts and a using state of the art imaging techniques is required to surpass the current roadblock with sensor-based technologies (Winter et al. [2017]).

  Landmarks have been incorporated into existing techniques and principles to generate additional value. However, they have not been purely used in either localisation or navigational principles. Exploring a low-cost generic framework for indoor localisation by combining outdoor principles with the indoor context using a pure landmark-based approach is, therefore, a novelty and has the potential to contribute to the field of location-based services.

- *The relationship between the project and the broader social context.*

  Using the visibility of landmarks allows for the indoor environment to be fingerprinted through cross-referencing. Knowing where a particular combination of landmarks is visible, and simultaneously knowing where each landmark is not visible, allows the localisation of a user, if the user can tell the system what it sees in its surroundings.

The more the system knows about objects within the indoor environment, the more accurately the user can observe and describe the environment, and the better the system is in interacting with the user, the more likely it is that a sufficiently accurate and precise location can be provided for location-based services.

The aim of this research is not to give the most accurate and precise location, but to provide a location-specific enough that it can be used for orientation and navigation. The fingerprint, considered as functional regions, could provide useful for navigation, where a user navigates between different fingerprints, from landmark to landmark, through the context in an immersive manner.

The landmark-based indoor localisation approach of this exploratory research had the objective to look for a possible alternative for sensor-based indoor localisation. The newly proven approach paves the way for future researchers to exploit the power of landmarks and indoor localisation thoroughly, the potential of the core concepts are tested, and the field of localisation can add a new branch to its tree.

COLOPHON

This document was typeset using LaTeX. The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from André Miede.

TU Delft