TECHNICAL UNIVERSITY OF MUNICH
TUM School of Engineering and Design
Photogrammetry and Remote Sensing

Change detection in point clouds of urban street spaces using fuzzy spatial reasoning

Joachim Alexander Gehrung

Dissertation

2022

TECHNISCHE UNIVERSITÄT MÜNCHEN
TUM School of Engineering and Design
Photogrammetry and Remote Sensing

Change detection in point clouds of urban street spaces using fuzzy spatial reasoning

Joachim Alexander Gehrung

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr.-Ing. Christoph Holst
Prüfer der Dissertation:	1.	Prof. Dr.-Ing. Uwe Stilla
	2.	apl. Prof. Dr.-Ing. Claus Brenner Leibniz Universität Hannover

Die Dissertation wurde am 17.06.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 12.09.2022 angenommen.

Abstract

The high population density in urban areas causes a variety of interactions and thus events, which can have an impact on the cityscape. A notable proportion of this interaction also takes place in urban street spaces and manifests itself in form of changes. These may have a direct impact on road safety or be of interest to sustainable urban development. Various measurement methods such as manual descriptions, images or 3D sensor measurements in form of point clouds as well as a variety of carrier systems such as helicopters, drones or mobile mapping vehicles can be used to record a street space.

In this work, changes in urban street spaces are detected based on point clouds of mobile mapping vehicles. Based on a literature review conducted on the topics of change detection, volume-based environment representation and fuzzy reasoning, research gaps emerge that lead to the contributions of this thesis on the following three core aspects: (i) Representation of spatial information subject to uncertainties and contradictions through a generic evidence grid and the drawing of conclusions based on it, (ii) change detection on a spatially noisy, uncertain, and inconsistent data set; and (iii) metrics for the quantitative assessment of the properties of a generic evidence grid. When comparing two epochs, it is not possible to assume the same visibility of the environment due to occlusions as well as deviations of the recording trajectories. Therefore, ray tracing is used to determine the observed free space. Fuzzy logical reasoning enables the determination of changes based on the resulting occupancy information, as well as the subdivision of the latter by object classes. Fuzzy sets are used for information fusion; fuzzy measures are utilized to quantify the degree of ignorance, uncertainty, and contradiction in the available evidence.

A mobile mapping system equipped with two LiDAR sensors was used for data acquisition. The two data sets recorded at different times of the year contain various types of facades as well as people, traffic participants and vegetation in the vicinity of the TU Munich campus. The results of the experiments show that information with a resolution of 0.4 m can still be represented with an F1 score above 0.9. Uncertainties and inconsistencies can be compensated almost completely by choosing an appropriate membership function and by means of suitable defuzzification methods. Confirmed environmental elements could be detected with an F1 score of 0.93, and changed elements with an F1 score of 0.89. Qualitative experiments have shown that in addition to large changes such as scaffolding, small changes such as Christmas decorations and closed shutters can be reliably and comprehensively identified.

Kurzfassung

Die hohe Bevölkerungsdichte in urbanen Gebieten führt zu einer Vielzahl an Interaktionen und damit zu Ereignissen, welche einen Einfluss auf das Stadtbild haben können. Ein nenneswerter Anteil dieser Interaktion findet auch in urbanen Straßenräumen statt und äußert sich in Form von Änderungen, aus denen sich direkte Auswirkungen auf die Straßensicherheit ergeben können oder die für die nachhaltige Stadtentwicklung von Interesse sind. Zur Erfassung des Straßenraums können diverse Messmethoden wie etwa manuelle Beschreibungen, Bilder oder auch 3D-Sensormessungen in Form von Punktwolken sowie verschiedene Trägersysteme wie Hubschrauber, Drohnen oder Mobile Mapping Fahrzeuge herangezogen werden.

In der vorliegenden Arbeiten werden Änderungen im urbanen Straßenraum anhand von Punktwolken mobiler Mapping-Fahrzeuge detektiert. Auf Basis einer durchgeführten Literaturrecherche zu den Themengebieten Änderungsdetektion, volumenbasierte Umgebungsrepräsentation und unscharfes Schließen ergeben sich Forschungslücken, die zu den Beiträgen dieser Arbeit zu den folgenden drei Kernaspekten führen: (i) Darstellung von räumlichen, mit Unsicherheiten und Widersprüchen behafteten Informationen durch ein generisches Evidenzgitter sowie das darauf basierende Ziehen von Schlussfolgerungen, (ii) Änderungsdetektion auf einer räumlich verrauschten, unsicheren und widersprüchlichen Datengrundlage und (iii) Metriken zur quantitativen Bewertung der Eigenschaften eines generischen Evidenzgitters. Da beim Vergleich zweier Epochen auf Grund von Verdeckungen sowie von Abweichungen der Aufnahmetrajektorien nicht von derselben Sichtbarkeit der Umgebung ausgegangen werden kann, wird mittels einer Strahlverfolgung der beobachtete Freiraum bestimmt. Unscharfe logische Schlussfolgerungen ermöglichen auf den daraus resultierenden Belegungsinformationen aufbauend die Bestimmung von Änderungen sowie die Unterteilung von letzteren nach Objektklassen. Zur Informationsfusion werden unscharfe Mengen eingesetzt. Zur Quantifizierung des Grads des Unwissens, der Unsicherheit sowie des Widerspruchs in den vorliegenden Evidenzen finden unscharfe Maße Anwendung.

Zur Datenaufnahme wurde ein mit zwei LiDAR-Sensoren ausgestattetes Mobile Mapping System eingesetzt. Die zwei zu unterschiedlichen Jahrenzeiten aufgenommenen Datensätze enthalten verschiedene Typen von Fassaden sowie Personen, Verkehrsteilnehmer und Vegetation im Umfeld des Campus der TU München. Die Ergebnisse der Experimente zeigen, dass Informationen mit einer Auflösung von 0,4 m noch mit einer F1-Score von über $0,9$ repräsentiert werden können. Unsicherheiten und Widersprüche lassen sich durch die Wahl einer entsprechenden Membership-Funktion sowie mittels geeigneter Defuzzifizierungsverfahren nahezu komplett ausgleichen. Bestätigte Umgebungselemente konnten mit einer F1-Score von $0,93$ detektiert werden, veränderte Elemente mit einer F1-Score von $0,89$. Qualitative Experimente haben gezeigt, dass sich neben großen Änderungen wie etwa Baugerüsten auch kleine Änderungen wie Weihnachtsdekoration und geschlossene Fensterläden zuverlässig und umfassend ermitteln lassen.

Contents

List of Abbreviations

Abbreviation	Description	Page
DEM	Digital elevation model	5
ECEF	Earth-centered, earth-fixed	26
ENU	East, north, up	26
GIS	Geographic information system	10
ICP	Iterative closest point	67
LiDAR	Light detection and ranging	3
LRU	Least recently used	28
LWIR	Longwave Infrared	70
MLS	Mobile laser scanning	3
ROC	receiver operating characteristic	59
SLAM	Simultaneous localization and mapping	4
TLS	Terrestrial laser scanning	3

List of Figures

List of Tables

1 Introduction

1.1 Motivation

1.1.1 Why is change detection in urban street spaces required?

Urban environments make up a significant part of people's daily environment. This is particular true for street spaces, as they are relevant to the day-to-day business of companies, administrations, emergency services and drivers of autonomous vehicles. Detecting changes in urban street spaces is of interest to research, businesses and government agencies alike. However, a structured overview of possible fields of application requires categorization. Hebel [2012] organized the goals pursued by the various change detection methods into three time scales, the transitions of which, however, are deliberately indicated fluidly, as a clear subdivision is not easily possible. A categorization in *long-term, medium-term and short-term* changes is suggested. This form of subdivision has the advantage that it bundles applications with similar intentions and objectives.

The category of *long-term* changes deals with a period of time that is in the range of years. Effects of such forms of changes are of interest for urban development in an administrative sense and relate to topics such as the development, cultivation and usage of urban areas. Here the detection of changes is required to detect new construction, renovation and demolition of buildings and roads [Murakami et al., 1999]. This explicit awareness about changes is a precondition for recognizing and understanding trends in the development of urban areas. Knowledge of such trends can then be used to support infrastructure planning tasks, helping to improve factors such as environmental friendliness and sustainability in urban development concepts.

Changes within a *medium-term* period include time intervals of a few days, weeks or months. A much-studied application here is change detection in building construction (see also Tuttas [2017], Xu & Stilla [2021], Huang et al. [2022]). Change detection is required here to check the extent to which construction is on schedule, i.e. which construction phases are ahead or behind it. Explicit knowledge of medium-term changes is also required for civil protection, especially when it comes to assessing damage after severe natural disasters such as storms or floodings [Michel et al., 2012]. The field of autonomous driving has grown in popularity in recent years. Change detection is required here to extract robust features for vehicle navigation and automatically update feature maps (see also Schachtschneider & Brenner [2020], Berrio et al. [2019]).

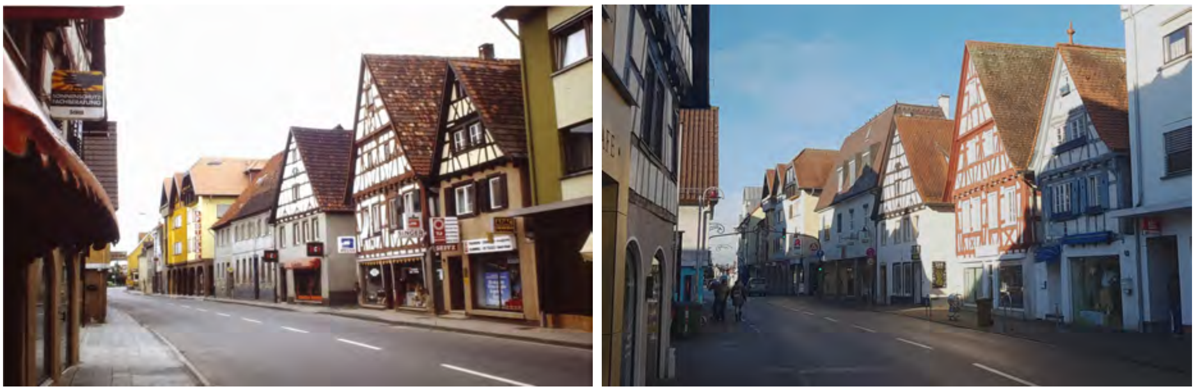
Change detection in the *short-term* time interval in required to be aware of short-term actions and to handle spontaneously occurring natural events. One example is moving object detection, which is used to identify moving people and objects. This information is important for autonomous driving, as dynamic environmental elements must be identified in order to control the vehicle safely (see also Asvadi et al. [2016a,b]). Spontaneous events mainly fall into categories such as disasters or accidents. Examples include the collapse of the Morandi Bridge in Genoa in August 2018 and the collapse of the Historical Archive in Cologne in March 2009. In these cases, change detection is required for the immediate assessment of the situation in terms of damage analysis. The insights gained here can be incorporated into the decision-making process for disaster relief.

As the above examples show, the motivations for the use of change detection in urban street spaces are diverse. What all of the above approaches have in common is a specific goal. However, it may be of interest to capture not only specific changes, but as many changes as possible. An example for this is a city survey, in which changes in the urban environment are first recorded and then evaluated by a viewer. In context of security-related applications, for example, a security company might want to be notified of all changes that could indicate break-ins or damaged fences. If a method is supposed to detect a particular type of change, e.g. in vegetation, an appropriate model must be chosen and its parameters optimized. The advantage of a generic change detection method, however, is precisely that no problem-specific model is necessary. Accordingly, it is not limited to a specific type of change.

1.1.2 Challenges of change detection

Sensor systems and acquisition concepts

In general, change detection in urban street spaces requires measurements of the environment from at least two points in time. In the following, these are also referred to as *epochs*. These measurements can for example result from photographs of a location, as shown in Figure 1.1, but also from other sensors. The two example images illustrate that changes of different magnitudes coincide in an urban environment. Large-scale changes evident in the images include modifications of long-term stable structures such as building facades and roofs. Installed trash cans, erected streetlights and parked vehicles represent changes of moderate magnitude. Minor changes include mounted street signs as well as replaced mailboxes and chimneys.



(a) (b)

Figure 1.1: An example of changes in an urban street space. a) A photo taken in 1988 showing the main street of Sinsheim in Germany, b) the same location in 2021. Approximately 40 small and large changes can be identified between the two images. The pictures were kindly provided by Klaus Gehrung.

The choice of a sensor system or combination of sensor systems for change detection depends on the intended area of application (see e.g. Hinz [2008]). The requirements for a general change detection method are that both small and large changes can be detected. Therefore, the acquired measurement data must represent the environment in such detail that even small structures can be detected. For this purpose, it is advantageous to record measurements from as close as possible and with as little occlusion as possible. In addition, it is advantageous if the sensor system enables large areas to be measured quickly and cost-effectively. This is particularly useful when change detection is to be used in a disaster relief situation.

In many of the applications mentioned in Section 1.1.1, mobile mapping has been established as an acquisition strategy. Here, the sensor is either attached to a backpack, vehicle or boat and can thus easily be used to record large areas [Vaaja et al., 2011]. This approach is similarly accurate to the stationary *terrestrial laser scanning (TLS)*, where the latter is used, for example, for the reconstruction of building facades, building models or the forest inventory (see also Pu & Vosselman [2009], Liang et al. [2016], Chizhova [2019]). The use of a mobile mapping platform is particularly beneficial when a large area is to be surveyed in a short time and with high spatial resolution. Other acquisition techniques involve airborne capture using helicopters or airplanes and are used for the large-scale 3D reconstruction of urban areas and city models (see also Poullis [2013], Dorninger & Pfeifer [2008]). These methods also deliver quite good results, but have a lower resolution than the above-mentioned recording techniques. From the perspective of the required resolution, ground-based systems meet the requirements better than airborne systems.

In addition to the acquisition strategy, the used *sensor principle* plays a role as well. Under this term, both the physical principle of measurement and the methodology for obtaining information from the measurement data are combined. Mobile mapping systems that utilize *light detection and ranging (LiDAR)* can create detailed, high-resolution 3D measurements of the environment [Borgmann et al., 2021]. Photogrammetry represents another common method that has been successfully applied to create dense point clouds (see also Pollefeys et al. [2008], Gallup [2011] and Cavegn & Haala [2016]). Even though both methods have their advantages and disadvantages, they meet the requirements demanded above. In this work, a mobile mapping system with LiDAR sensors was used to collect the data required for the change detection.

Sources of error

The most significant error sources for a Mobile Laser Scanning (MLS) system are listed below. The list results from direct hands-on experience of the author, who was involved in the construction of a measurement vehicle. It is sorted according to the influence of said sources, starting with the lowest influence:

- ❑ **Mechanical deformation**: Deformations of kinematic chains are caused by vibrations of the rotating laser scanners or insufficiently reinforced sensor mounts.
- ❑ **LiDAR-specific sources of error**: A multitude of different errors can occur here, such as discretization errors of the distance, reflections on windows and crosstalk from other sensors. In extreme cases, a wet surface can reflect the laser light away from the sensor [Lindenbergh & Pietrzyk, 2015].
- ❑ **Intrinsic and extrinsic sensor calibration errors**: Intrinsic calibration errors occur when the translational and rotational alignment of a sensor element with respect to the sensor has not been determined exactly. Extrinsic calibration errors occur when the transformation between the vehicle and sensor coordinate systems has not been determined with sufficient precision.
- ❑ **Synchronization**: The individual sensors and the navigation system must be synchronized with one another. This is the only way to know the time stamp for each individual measurement, otherwise the point cloud would be distorted even at low travel speeds.
- ❑ **Navigation system**: This category includes a number of factors that are related to navigation, such as noise in the inertial sensors, wheel slippage when using a wheel encoder and shadowing of the GNSS signal. The type and parameterization of the Kalman filter can also have an influence.

Mechanical deformations are negligible if the sensor carrier is designed appropriately. LiDAR-specific errors can be greatly reduced using heuristics and the challenges associated with sensor synchronization can be solved using technical measures. As far as sensor calibration is concerned, even small angular errors have a big impact. An angular misalignment of 0.1 ° can cause a displacement of approximately 20 cm over a measuring distance of 120 m, which leads to a distortion of the point cloud. Localization errors can be partially compensated for by post-processing of the navigation data and by applying registration methods such as SLAM. Together with calibration errors, localization errors represent the main source of uncertainty in the information derived from the measurement data of a mobile mapping system. The uncertainties must be taken into account in the change detection method.

Conflicting evidence

In addition to the sensor-related sources of error presented in the last section, certain environmental factors can lead to conflicting evidence. This in turn results in contradictions in the conclusions derived from the measurement data. An example of an influencing factor are *moving objects* like pedestrians, vehicles or flags, in short everything that moves during recording. Due to the movement of an object it occurs that an area is measured as occupied at one time and free at another time. Another example is *vegetation*. Here, the laser pulses sometimes pass through the free space between the foliage, sometimes they hit the foliage itself. Prior work has shown that spatial data structures such as occupancy grids are prone to contradictions due to discretization errors [Gehrung et al., 2017]. The effect occurs in areas where measuring rays pass along surfaces at a shallow angle. In order for a change detection method to handle contradictions, a form of representation is required that can explicitly represent the latter.

1.2 State-of-the-art

1.2.1 Change detection in point clouds

Singh [1989] defines change detection as the process of identifying differences in the state of an object or phenomenon by observing it at different times. A comprehensive overview of the field is given in Lindenbergh [2010] and Lindenbergh & Pietrzyk [2015]. Lindenbergh [2010] further differentiates between *change detection* and *deformation analysis*. Approaches of the first category answer the question about changes in a binary way. It is determined whether the scene as a whole or a specific part of it has changed. Approaches of the second category, deformation analysis, quantify the degree of change. In the further course of this work, only change detection will be discussed. Methods for change detection in point clouds can be divided into the four categories *point-*, *ray-*, *voxel-* and *object-based*. Some more complex approaches may also utilize techniques from more than one of these categories.

Point-based change detection

The point-to-point comparison, also called the *surface difference*, is the most direct way of detecting changes between two point clouds. Basgall et al. [2014] realized this by calculating differences between LiDAR and stereo-photogrammetric point clouds using a simple subtraction method. Kang et al. [2013] calculated the point-to-point distances using the Hausdorff distance. This approach was chosen to avoid issues related to local density variations.

Multiple octree-based approaches for storing the point clouds of multiple epochs were proposed by Girardeau-Montaut et al. [2005]. Changes were indicated by comparing point distances between epochs voxel by voxel, with a change occurring whenever the distance is above a noise threshold. Empty voxels in one epoch and changes or shifts in the local normal of the points within a voxel were seen as further indications for changes. The changes themselves were determined in a more accurate way by considering the neighbor voxels within the octree.

Xu et al. [2013] utilized a *3D Surface Separation Map*, which encoded the distance of a point to the nearest fitted plane from another epoch. Changes identified by said map were classified into different categories using a rule-based classifier. This method required prior knowledge of the scene in order to create the classifier. Du et al. [2016] proposed an automatic method for building change detection utilizing LiDAR data and aerial images. Both data sets were co-registered using the ICP algorithm. Changes were indicated by height difference and grey-scale similarity and a graph-cut method was applied to further optimize detection results by utilizing contextual information. Murakami et al. [1999] proposed an approach based on *digital elevation models (DEM)* for change detection based on ALS data. A surface representation was generated by projecting the individual measurements onto a two-dimensional grid. Changes were detected by subtracting an older representation from a newer one. A comprehensive discussion of DEM-based methods to detect changes was published by Matikainen et al. [2010].

Change detection methods based on point-to-point distances are sensitive to variations in point density. Due to the proximity to the recorded object, the effect is greater with mobile laser scanning than with airborne laser scanning. This category of techniques does not reveal occlusions, since no free space information is considered.

Ray-based change detection

If the recording position of the sensor is preserved, a point cloud can also be viewed as a set of rays. A ray is the way along the laser pulse propagation path, including the measured surface point. This allows not only to consider occupancy information implied by a measured surface point, but also the free space between the sensor and said point. No assumption can be made for the area behind the measuring point along an extension of the beam. The consideration of free space also allows an informed statement about occlusion. This is important, because otherwise an occlusion may be mistaken for a change.

Zeibak & Filin [2008] used a *visibility map* in form of a panorama depth map, which encoded the distance between the individual scene points and the laser scanner based on spherical coordinates. Applying a threshold allowed to divide the individual scene points of two recordings into the categories *change*, *no change* and *occlusion*. The process is fast and easy to implement, but cannot be applied between different points of view.

Hebel et al. [2013] applied a ray-based procedure for change detection in ALS point clouds. A three-dimensional grid structure was used to determine all rays in the neighborhood for an arbitrary point in space. The degree of change was calculated by combining the belief functions based on the Dempster-Shafer theory of evidence of all rays involved. This allowed to determine whether new measurements confirmed or contradicted the information provided by older measurements. Xiao et al. [2015] proposed a similar approach based on mobile laser scanning data that further improved the method with a point-to-triangle distance-based technique to conduct direct consistency evaluation on points.

Ray-approaches allow detailed statements about changes in the environment, but since each ray has to be considered individually, these techniques are challenging in terms of runtime for the large amounts of data produced by mobile laser scanning.

Change detection based on 2D and 3D grids

The methods listed here require a transformation of the point clouds into a two- or three-dimensional representation. The latter is either a grid, an octree or a set of voxels. Changes are found by comparing the occupancy of cells or voxels for different epochs.

Pagac et al. [1998] utilized a two-dimensional occupancy grid that represented the environment of an autonomous vehicle and was intended for the purpose of navigation. Sonar sensor readings were integrated into said grid by the Dempster-Shafer inference rule. Wolf & Sukhatme [2004] used a similar approach for SLAM-based robot navigation that utilized two occupancy grids to represent static and dynamic parts of a scene. The state of occupancy was specified as either free, unknown or occupied. The comparison of different states of occupancy allowed to conclude if some entity that was previously considered static was moved.

The idea of dividing the environment into static and dynamic elements was also used by Azim & Aycard [2012]. They utilized conflict search on an occupancy grid in order to determine dynamic elements. These were classified and tracked using a global nearest neighbor technique.

The methods mentioned here do not meet the requirements of this work regarding the handling of uncertainties and contradictions.

Object-based change detection

Change detection methods focus on high-level structures such as objects, segments or clusters as basic units for detecting changes. Vosselman et al. [2004] utilized laser scanning data to detect building changes in a two-dimensional map. A segmentation and filter step was used to identify bare earth points and extract object points. The latter were then classified as buildings or vegetation based on surface roughness, segment size, height, color and first-last pulse difference. Changes were identified by comparing classified segments with the building objects on the map.

Aijazi et al. [2013] classified point clouds into the object categories *permanent* and *temporary*. A similarity map derived from an evidence grid was used, inter alia, to detect natural or human-made changes in an urban environment, which were then analyzed using cognitive functions of similarity. These results were then utilized to progressively modify and update a 3D map. Voelsen et al. [2021] segmented point clouds of an urban environment using a region growing algorithm and then used points of each segment to create an occupancy grid. This was then used to assess the temporal behavior of the extracted clusters.

Huang et al. [2022] detected changes of construction sites using photogrammetric point clouds by considering both geometric and semantic changes. In a first step, semantic changes were detected using an occupancy-based change detection method inspired by the work of Hebel et al. [2013]. In a next step, geometric changes were considered to check consistency and detect conflicts. The usage of semantic information made the method application-specific. Hirt et al. [2021] proposed an approach for detecting changes in trees. A three-stage process consisting of trunk extraction, tree separation and crown expansion was used to extract tree instances from point clouds. These instances were then combined with geometric changes determined using an occupancy grid.

Voelsen et al. [2021] described a multitemporal analysis based on 14 epochs of mobile mapping LiDAR measurements used to derive a *static reference map*. This map contained all elements of the environment that had not changed over time. Changes corresponded to non-static objects or static objects that existed only temporarily. The point cloud of each epoch was first segmented by region growing. The segments were then classified by a random forest into the five static classes *facade*, *pole*, *fence*, *traffic sign* and *vegetation* as well as the three dynamic classes *vehicle*, *bicycle* and *person*. All static objects from all epochs were then organized using a voxel grid and combined with a visibility analysis in order to differentiate between free space and occlusion. Based on this information, the suitability of each voxel for being part of the static reference map was classified. This avoided eliminating occluded voxels or keeping segments such as scaffolds that were only temporarily present. The algorithm proposed by Voelsen et al. [2021] is a good example of a hybrid approach, as it utilizes techniques from several of the categories described here to classify change detection methods.

Changes derived from objects are usually more descriptive than changes in surface or volume, but require prior knowledge about the scene. Therefore, object-based techniques are less suitable for generic use and more for applications in special fields. In addition, change detection results are highly dependent on the performance of the classifier used to detect scene objects.

Moving object detection and removal

Moving object detection and removal is an area of research that shares similarities with change detection, but usually takes place in a shorter time frame. In the context of this work, the term *moving object removal* is used whenever change detection is applied in the current measurement run, that is, in the same data set. The different approaches can be divided into categories, with representatives of the categories being discussed below. The topic is extensively researched in the field of computer vision, especially in relation to object tracking in 2D images. However, since the focus of this work is on mobile LiDAR data, the most relevant topics from a methodological point of view are the above mentioned *change detection* and the *detection and tracking of moving objects (DATMO)*. The latter is often found in the context of robotics, autonomous driving and mobile mapping.

A frequently cited method for detecting change is that of Underwood et al. [2013]. In the proposed approach, differences between two point clouds are determined by placing one of the two in a spherical grid that has its origin in the sensor position. For the other point cloud, it is then checked whether or not one of its points shares a grid cell that is already occupied. If this is the case, the distance and angle between the two points are checked. If both are above a certain threshold required to suppress the influence of noise, then an object has appeared. In order to identify disappeared objects, the roles of both point clouds are swapped and the process is repeated. The method is well suited for quickly and efficiently finding differences between two point clouds and can be made more robust by forming clusters after removing the ground plane.

A method that explicitly focuses on removing moving objects was presented by Schauer & Nüchter [2018]. The authors also referred to this approach as the *peopleremover*. The method utilized an occupancy grid and can be used for mobile LiDAR data as well as for stationary terrestrial data. Each voxel in the grid contained a set of identifiers from each scan with at least one measurement point within the voxel. The reason for storing references instead of occupancy was to be able to abort the voxel traversal early. Furthermore, it allowed to avoid self-intersections as well as to achieve sub-voxel accuracy. Based on said data structure, all voxels that intersect with the line of sight of the sensor and contain a non-empty set of scan identifiers were determined and classified as *dynamic*. Voxel traversal was stopped once a voxel with the same identifier as the current scan was encountered. A point was removed if it falls into a voxel that is marked as dynamic. The peopleremover-method produces visually appealing results, is faster than Underwood's approach in terms of runtime and comparable with respect to the F_1-score.

The method proposed by Asvadi et al. [2016a,b] was developed within the field of autonomous driving. Due to real-time constraints, the approach was trimmed to speed, which is why free and unknown space were not taken into account. After removing the ground plane, the remaining part of the LiDAR scan was voxelized by quantizing the measurements. For each voxel, the number of measuring points within were counted. The basic assumption was that more measurement points accumulate in the voxels occupied by static structures than in those occupied by dynamic objects. Based on this occupancy grid, a two stage approach was applied. The first stage provided a rough estimation of static and dynamic voxels by applying a simple subtraction mechanism. The second stage refined the results by analyzing the 2D histogram computed from the output of the first stage. Since no free space was considered, the approach had to make a number of assumptions about the environment and also employed several heuristics in order to cope with occlusion.

Lee et al. [2020] examined the topic of moving objects removal from a mobile mapping point of view. Moving objects needed to be removed from the 3D scans in order to create a clutter-free image of the environment. The authors noted that a moving object looks different than a static instance of the same object, so detecting one in the accumulated point cloud is not an option. Instead, the object is detected in camera images and the pose obtained in this way is used to remove it from the individual LiDAR scans. The challenge with segmentation-based approaches like this is that only object classes for which a classifier exists can be removed. Also, it requires a second sensor in form of a camera and a known relative pose between it and the LiDAR sensors.

As shown, there is an overlap between the detection and removal of moving objects and change detection. The former represents an important tool in the pre-processing of LiDAR measurements. However, none of the above methods is suitable to account for uncertainties and inconsistencies in information derived from measurement data.

1.2.2 Spatial data representation

Volumetric spatial data structures are typically used to map location-based information to a linear memory. Addressing is based on a multidimensional search key. This is usually done with the intention of providing efficient, location-based access to a data set. Volumetric data structures can be categorized as *occupancy grids*, *elevation maps* and *octree-based representations*. A comprehensive overview of the subject is provided in [Xu et al., 2021].

Occupancy grids

Occupancy grids are a specific form of evidence grids that represent the degree of occupancy for a region of space with given boundaries. Elfes [1989] proposed an approach for indoor mapping that utilized a 2D grid for storing occupancy information. The state of each cell was represented by a probability that can be interpreted as either free, occupied or unseen space. The grid was constructed using ultrasonic sensors and allowed for an accurate representation of the environment, despite the wide aperture angle of the used ultrasonic sensors. Borenstein & Koren [1991] proposed a technique called histogramic in-motion mapping (HIMM). It used rapid in-motion sampling based on ultrasonic range finders to create a pseudo-probability distribution over the environment. The content of each cell corresponded to the level of evidence that an obstacle is present. The system was designed for real-time robot navigation and enabled quick reactions in terms of obstacle avoidance. Roth-Tabak & Jain [1989] used a 3D grid generated from distance measurements. Uncertainties were not taken into account.

For all the approaches listed here, the boundaries of the represented area must be known. Furthermore, the grid structure resolves homogeneous and heterogeneous areas with the same resolution, which leads to an unnecessarily high memory consumption.

Elevation maps

Herbert et al. [1989] proposed two-dimensional grid structures in which each cell stored a height value. They referred to this data structure as *elevation maps* or *2.5D maps*. The approach cannot represent overhanging structures such as trees, bridges or underpasses. Therefore, it is only applicable in cases where a surface representation of the environment is sufficient. An example where this is sufficient would be the contribution to the scene understanding tasks of driver assistance and autonomous systems proposed by Pfeiffer & Franke [2010]. The approach used so-called *Stixels*, rectangular sticks of a certain width that limit the free space in front of the vehicle. It was based on the assumption that objects are located on the ground and have an approximately vertical pose with a flat surface.

A number of extensions have been published that intended to increase the descriptiveness of elevation maps. Triebel et al. [2006] and Pfaff et al. [2007] proposed several surfaces per cell. Gutmann et al. [2008] suggested to use multiple cell classes in order to describe different structures. The disadvantage of 2.5D maps is that they do not represent a volumetric representation, but rather a discretization along the height. This severely limits their informative value, which makes them unfit for tasks that require an exact representation of a complex environment. As a possible solution to this, Ryde & Hu [2010] suggested to store a list of voxels in each grid cell. Although this representation was volumetric, no distinction between free and unobserved volumes was made. Dryanovski et al. [2010] suggested to store a list of both occupied and free voxels per cell. Douillard et al. [2010] presented a hybrid approach that utilized elevation maps to describe the background, while foreground objects were represented by high-resolution voxel structures. The above approaches are suitable for representations where three-dimensional information related to the ground plane plays a central role.

Octree-based environment representations

The advantage of octrees over grid-based data structures is that their resolution can be adapted locally to the data. Homogeneous areas can be represented with a lower resolution than areas with a high information density. The usage of octrees for the representation of spatial information was suggested by Meagher [1982]. Wilhelms & Van Gelder [1992] extended this approach to deal with the storage of binary information such as occupancy information. An example application for octree-based approaches is that of Surmann et al. [2003]. This approach used an octree as a generic and quick-to-generate environment representation, which was used for the navigation of a robot.

Payeur et al. [1997] used octrees to expand the two-dimensional occupancy grid mentioned above into three dimensions. In the course of this, a probabilistic approach to modeling free and occupied space was presented. Another approach based on octrees called *Deferred Reference Counting Octree* was introduced by Fairfield et al. [2007]. The sharing of subtrees between several octrees and a sophisticated update mechanism allowed octrees to be copied quickly, which turned out to be an important property with regard to particle filter SLAM. A maximum likelihood approach was periodically used for compression and child nodes with the same status were pruned.

The probabilistic octree-based occupancy grid referred to as *OctoMap* proposed by Hornung et al. [2013] is based on the works of Fairfield et al. [2007] and Moravec [1988]. One of the essential extensions was the use of limited confidences for the almost loss-less compression. Furthermore, multi-resolution queries and the problem regarding representations with exaggerated confidence were also addressed. The *log-odd* notation introduced by Moravec [1988] doesn't keep track of evidences for and against the information, but instead stores the evidence using a ratio of both. This doesn't allow to distinguish between the case of *no evidence at all* and the case of *contradicting evidence*. A limited statement about the degree of uncertainty associated with the evidence can be made, but no statement about the *degree of ignorance*. Also, none of the above representations is designed to represent arbitrary spatial information.

1.2.3 Fuzzy spatial representations and inference

Fuzzy logic is a form of many-values logic that is employed to handle the concept of partial truth [Vilém Novák, 1999]. Its ability to describe concepts such as vagueness allows the handling of error-prone measurement data. Fuzzy logic is used, among other things, to represent location-based information. The use of fuzzy logic to represent location-based information can be divided into two categories, *fuzzy spatial representations* and *fuzzy spatial-semantic relations*.

Fuzzy spatial representations

The approach presented by Tunstel [1995] is similar to the probabilistic approach of Moravec & Elfes [1985]. A fuzzy two-dimensional occupancy grid was utilized to help a vehicle navigate in an unsafe environment. A numerical value in the interval $[0, 1]$ in each grid cell described the degree of occupancy as the degree of belonging to an element of a fuzzy set. Furthermore, the grid contained additional information such as the slope of the ground in order to additionally support the navigation. Information could then be derived using if-else rules. The approach was used to simulate an abstract map of the environment, which was then used to demonstrate the path planning for a mobile robot.

Ning et al. [1993] presented an approach that was able to reason about uncertainty in terrain description. The environment was not subdivided into a regular grid, but into contiguous regions. The boundaries of the latter were defined by fuzzy half-planes, which were represented by linear inequalities with triangular fuzzy numbers as coefficients. The terrain was classified into a set of fuzzy regions, which resulted in an uncertainty map where the weight of each region was interpreted as a linguistic variable. This allowed reasoning about terrain features under uncertainty, which in turn was applied to do path planning in uncertain terrain.

Hofmann et al. [2013] utilized fuzzy theory in a spatial way in order to solve the problem of vehicle cross-country mobility. Addressing factors with regard to thematic and spatial vagueness, the work combined multiple sources of information about terrain reliefs, vegetation cover, soils and soil cover, weather and climate, hydrology, built-up areas and road networks into a cost map. The map was then used to make a statement about the passability of terrain. The system was integrated into a geographic information system (GIS).

Robinson [2003] reviewed many of the major aspects of fuzzy set theory and its linkage to geographic information systems, thereby showing the widespread use of fuzzy logic in this field of research. The author discussed several topics and their applications in relation to GIS systems, including membership functions, fuzzy clustering, basic logical operations, fuzzy relations as well as fuzzy control systems. The need to deal with uncertainty was named as one of the main reasons for adapting fuzzy logic in the GIS community.

Zhang et al. [2017] proposed an approach that utilized the fuzzy analysis of airborne LiDAR data to determine rainforest boundaries. Variables derived from LiDAR data were used to calculate membership function values for both rainforest and non-rainforest. Confusion index values were derived to illustrate the transition zones between both classes.

The above approaches bind a semantically interpretable information to a spatial area, but represent only application-specific information. A representation of arbitrary information is not intended, generic combination rules are not provided.

Fuzzy spatial-semantic relations

Dutta [1991] used fuzzy logic to represent spatial constraints between landmarks given imprecise or incomplete information about them. The information can be both quantitative (e.g. the chair is 2 feet from the desk) and qualitative (e.g. the chair is near the desk) in nature. The resulting framework allowed reasoning about relative positions and motion of objects in response to spatial queries about such relationships.

Similar approaches have been proposed in computer vision. Krishnapuram et al. [1992] presented a unified methodology to characterize properties and spatial relationships of object regions in images, which play an important rule in scene understanding. Rosenfeld [1998] reviewed work on the fuzzy topology and geometry of image subsets and discussed a variety of properties, as well as some applications of these concepts in image processing and analysis. A similar report was published by Freeman [1975], which discussed various approaches for encoding spatial relationships between objects in an image. Psychological investigations related to human image encoding were also examined and parallels were drawn.

The above approaches deal with the fuzzy representation of spatial relations, but between abstract semantic elements. An application of these methods would therefore require that first abstract elements such as objects are extracted from the data. Generic spatial information described by the means of fuzzy logic can thus not be represented.

1.3 Objectives and contributions

The top-level goal of this work is to develop a method for the robust and accurate detection of changes in urban street spaces from mobile mapping point clouds. Emphasis is placed on keeping the change detection method as generic as possible so that as many different categories of changes can be detected. The following research questions, which have not yet been addressed or fully answered in the current state-of-the-art, are considered:

I To what degree can the influences of vagueness and contradiction in information derived from mobile LiDAR measurement data be reduced and what properties of method and data support this?

II With what accuracy is it possible to automatically detect changes in an urban street space based on vague, contradicting and spatially blurred mobile mapping point clouds and does it allow the use of the method in the context of applications such as city surveys and security applications, either supportive, or fully automatic?

III What metrics are appropriate for evaluating the spatial representation developed in this work and what are their limitations?

Based on the analysis of the change detection techniques as well as spatial data representations presented in the previous section, the research questions shall be answered by pursuing the following specific objectives in this thesis. From a methodological point of view, these can be divided into three groups.

1.3.1 Representation and inference of spatial data

Dealing with occlusions is required for detecting changes in urban street spaces, as they occur in large numbers due to the high concentration of objects and people in a confined space. This requires information on free space, because without that, it is not possible to distinguish between occlusions and changes. According to the current state-of-the-art, there are two categories of methods capable of representing free space. Ray-based methods such as the ones presented by Hebel et al. [2013] and Xiao et al. [2015] are able to model the free space very accurately. With methods of this type, challenges such as neighborhood search and comparison of the measurement rays of two epochs arise, the influence of which is then reflected in the runtime complexity. The other category involves voxel-based approaches such as *OctoMap* presented by Hornung et al. [2013]. However, state-of-the-art volumetric representations do not meet the requirements of the problem at hand. Most voxel-based approaches use the probabilistic evidence description developed by Fairfield et al. [2007] and Moravec [1988]. Such approaches do not represent evidence directly, but only through the ratio of evidence for and against the information at hand. Consequently, no statement can be made as to whether enough evidence is available to draw an informed conclusion. The chosen form of representation also means that *conflicting evidence* caused by moving objects and vegetation cannot be handled, since a distinction between *no evidence* and *conflicting evidence* is not possible.

The detection of changes requires a way to perform inference based on the chosen form of representation. If the information is represented by a generic evidence grid, an equally generic inference, for example based on Boolean logic, can be build upon it. This can also be used to combine the results with additional information, which makes it possible to answer complex questions. Furthermore, results can be filtered, e.g. according to object classes, which allows the advantages and disadvantages of the method to be examined in more detail. Such a form of generic evidence grid with incorporated inference has not yet been developed according to the current state-of-the-art. In summary, the tasks related to the generic representation of uncertain and conflicting spatial information required for change detection can be stated as follows:

❑ Define a methodological approach to represent arbitrary spatial information that allows to (i) handle uncertainty, (ii) is able to describe contradictions, and (iii) enables a statement about the degree of ignorance.

❑ Develop an inference mechanism based on the above methodological approach that allows logical reasoning in accordance with Boolean logic.

A solution to the above tasks provides the foundation for the development of a methodical approach to change detection that is capable of dealing with measurement data that is subject to uncertainty.

1.3.2 Change detection

Given a representation and inference mechanism as motivated in the last section, the complexity of the change detection task decreases, as the handling of the associated uncertainties and contradictions has already been taken care of. In addition to the noise associated with an information, however, there remains another disruptive factor to be compensated for, namely the noise on the localization of said information. Such an error can occur due to residual errors in the boresight calibration and the registration of the measurement data. Once this is compensated, the actual change detection is little more than applying the inference mechanism. One possible solution for this is motivated by classical two-dimensional image processing. If the information is spatially blurred with a radius larger than the actual spatial deviation, this reduces the occurring false positives at the expense of detection sensitivity. In the context of image processing, such an operation can be realized by a Gaussian filter [Forsyth & Ponce, 2012]. In the case of convolutional neural networks, a similar operation is used for feature extraction [Scherer et al., 2010]. In summary, the tasks with respect to change detection can be formulated as follows:

❑ Definition of a methodical approach for change detection based on the aforementioned spatial representation, capable of dealing with the uncertainty and contradictions inherent to MLS data. This also includes the correct handling of areas where no statement can be made due to a lack of observations.

- Adaptation of the above methodology to address not only uncertain information, but also spatially blurred information resulting from sources of error such as calibration and registration errors.

Fulfilling these tasks not only allows to robustly process changes affected by uncertainty, but also to take into account the aspect of spatial fuzziness caused by certain error sources. This provides all the techniques necessary for the main objective of this work.

1.3.3 Evaluation metrics

The results of change detection are encoded within the above-mentioned method for spatial representation. However, in order to evaluate the results, they must be compared with a ground truth. Since the data used in this work is based on point clouds, the same point clouds are used to carry the ground truth. The comparison can be made in several ways. First, the representation containing the results can be directly compared to the ground truth. Second, the representation can be converted to a point cloud before comparison. Third, the ground truth can be encoded in the same representation as the change detection results. The literature refers to numerical error measures such as *precision*, *recall*, and *F1-score*, which are based on a confusion matrix. These error measures were chosen as a starting point for the development of custom metrics capable of evaluating the above representation. The tasks in regard to the development of the metrics required for evaluation are:

- Development of a metric for the comparison of the represented information to a ground truth. This is required in order to make an informed statement about the strengths and weaknesses of the representation.
- Development of a metric for the comparison of the represented information with a ground truth after the former has been extracted from the representation. Thus, the overall performance of the representation can be evaluated, although systematic effects are not excluded.
- Development of a metric for the comparison of the represented information and ground truth in the context of the above representation. This is necessary to evaluate the performance of the representation, while excluding any systematic effects due to the nature of the representation.

Meeting these objectives allows the methodology for change detection in urban environments proposed in this work to be quantitatively evaluated and analyzed for advantages and disadvantages.

1.4 Structure and organization

This work is structured as follows. Chapter 2 presents an overview over the theoretical background of fuzzy logic and inference. In Chapter 3, fuzzy spatial reasoning is introduced, i.e., the methods for information fusion as well as for quantifying the degree of ignorance, uncertainty, and contradiction. Chapter 4 explains how fuzzy spatial reasoning is used for change detection. Chapter 5 presents metrics for comparing fuzzy representations to each other or to a ground truth in form of a point cloud. Chapter 6 presents the experiments used to examine the capabilities of spatial reasoning as well as the change detection procedure that is based on it. The data set used for evaluation and the associated ground truth are also discussed here. Chapter 7 presents the results of the evaluation and their analysis. Chapter 8 contains the discussion with respect to the three research questions of this work, namely the performance of fuzzy spatial reasoning, its suitability for change detection and the significance of the evaluation results. Chapter 9 finalizes this work by presenting the conclusion and outlooks for future works.

2 Basics

This chapter briefly outlines the basic concepts and principles of *fuzzy reasoning*. First, an overview of the field is provided and a number of fundamental terms are defined. The fuzzy sets used to represent evidence and the corresponding logical operations are introduced. Subsequently, the fuzzy measures used for the analysis of the fuzzy sets are presented. These basics are then utilized to introduce *general fuzzy reasoning* as well as the *evidence-based fuzzy reasoning*, which is based on the former. The chapter concludes by outlining how an evidence distribution can be interpreted using a fuzzy measure.

2.1 About fuzzy reasoning

Fuzzy reasoning is a theoretical framework with multiple degrees of freedom [Klir & Folger, 1987]. Weisbrod [1996] discussed two specialized forms of fuzzy reasoning that do not have any degree of freedom. One is called *possibilistic reasoning*, the other one *evidence-based reasoning*. Compare Figure 2.1 for an overview.

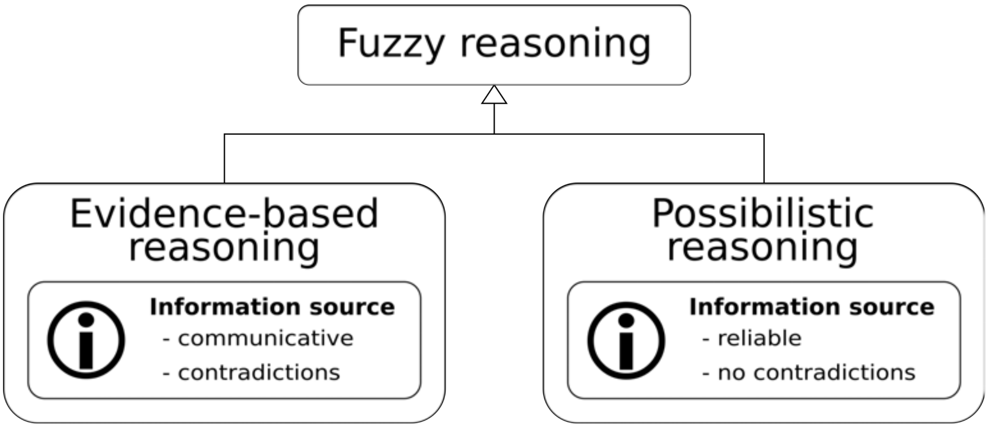


Figure 2.1: Comparison of evidence-based reasoning and possibilistic reasoning. Both are special forms of fuzzy reasoning based on different sources of information.

Possibilistic reasoning is based on a so-called *reliable source of information* that only forwards information that is considered completely reliable, but possibly incomplete. Said information must contain little or no contradictions. The basic idea is to gradually reduce confidence in those statements that are excluded by the information source. Evidence-based reasoning utilizes a so-called *communicative source of information*. It communicates whatever it deems credible in an unfiltered manner, even if these statements contain contradictions. In such a case, significantly more information is available than with a reliable source of information. However, this information does not always have to be accurate. The underlying idea is to collect information that supports the respective statements. This approach of reasoning also allows information to be ignored if it turns out to be incorrect.

Possibilistic and the evidence-based reasoning are *complementary* to each other. This can be seen, among other things, from the handling of the fuzzy set. In case of possibilistic reasoning, the fuzzy set of potential results decreases with increasing knowledge (formation of intersections). In case of evidence-based reasoning, the fuzzy set increases (formation of union).

Modern sensor systems are to be regarded as a communicative source of information due to the sources of error mentioned in Section 1.1.2. Therefore, the focus of the theoretical basics imparted here is on evidence-based reasoning.

2.2 Vagueness and uncertainty

The concept of fuzziness comprises two concepts that describe important properties of information. These concepts are called *vagueness* and *uncertainty*. According to Weisbrod [1996], vagueness describes the conscious or unconscious lack of precision. Klir & Folger [1987], on the other hand, associate vagueness as the difficulty of making sharp or precise distinctions in the world. In other words, vagueness is always present when no sharp borders can be drawn.

A vague statement therefore is considered to be fuzzy because vague boundaries were drawn in its formulation, for example through the use of abstraction. Linguistic terms are usually vague, since the concepts described by them are usually not explicitly defined. An example of this is the statement *The witness observed a tall person*. By referencing the concept of a tall person, the possible height of said person is reduced to an interval with implied but not clearly defined limits. The use of such a concept can, but does not have to, presuppose ignorance. Weisbrod [1996] describes a coarse abstraction like the one mentioned as fundamental for processing complex knowledge. Vagueness is formalized through the use of a *fuzzy set*.

Weisbrod [1996] equates the concept of uncertainty with admitting a lack of knowledge. To follow up on the above example, the sentence *It was dark, the witness is not entirely sure of the height of the person* proves to be a statement fraught with uncertainty. The height of said person is therefore a fuzzy variable and only uncertain statements can be made about its value. Uncertainty is formalized through the use of *fuzzy measures*. Klir & Folger [1987] equate fuzzy measures with *ambiguity*, that is an unspecified choice between two or more alternatives. However, the above example shows that ambiguity addresses a concept that is comparable to uncertainty.

The transition between the concepts of vagueness and uncertainty can be viewed as smooth. Vagueness does not automatically imply uncertainty, since a vaguely defined statement (e.g. about a person's height) can be made despite the fact that certain knowledge about the height is present. The focus on fuzzy information is particularly useful when intermediate and final results can be calculated exactly, but pretend to have an unjustified precision.

2.3 Fuzzy sets

Large parts of the terms and definitions on the subject of fuzzy sets used today were coined by Zadeh [1965]. The fundamental idea can already be found in the publications of Black [1937]. The concept of the fuzzy set was probably first formulated in French as *ensemble flou* [Menger, 1951]. Fuzzy sets are a generalization of the classical, Boolean set algebra. Instead of a definitive assignment of a variable to an element of a set, a gradual assignment is given.

2.3.1 Definition

Fuzzy sets are denoted by $\tilde{A}$. The characteristic function in Equation 2.1 indicates the degree to which element u from universe $\mathcal{U}$ belongs to the fuzzy set $\tilde{A}$.

$$\mu_A(u) : \mathcal{U} \rightarrow [0,1] \tag{2.1}$$

The specific nature of the function is problem-specific and does not follow any formal framework. Rather, the only requirement is that it roughly describes the affiliation to the fuzzy set. A triangle function is often used.

2.3.2 Union, intersection and complement

In order to handle fuzzy sets, generalized fuzzy set operations are required. The definition of the operations *union* and *intersection* is based on so-called *s- and t-norms* (see Equations 2.2 and 2.3).

$$\mu_{A \cup B}(u) \coloneqq s(\mu_A(u), \mu_B(u)) \tag{2.2}$$

$$\mu_{A \cap B}(u) \coloneqq t(\mu_A(u), \mu_B(u)) \tag{2.3}$$

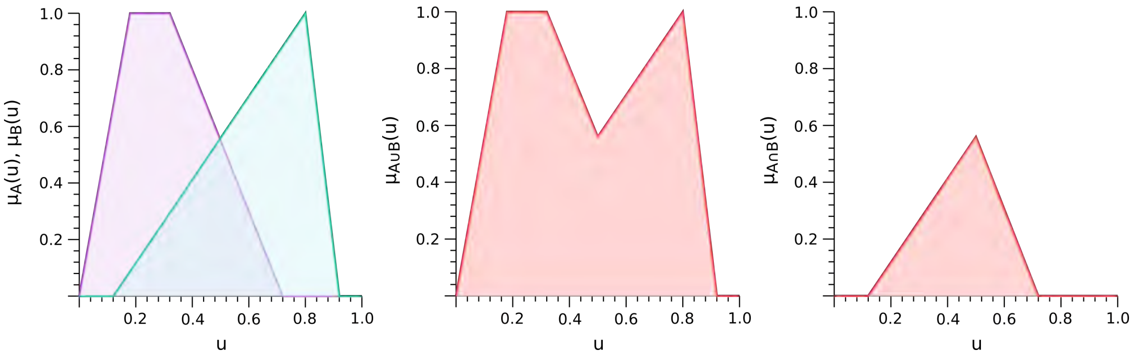
The fuzzy *complement function* is defined as

$$\mu_{A^c}(u) \coloneqq c(\mu_A(u)) \tag{2.4}$$

The functions for the realization of union, intersection and complement function have to fulfill certain formal properties which are explained in detail in Weisbrod [1996].

2.3.3 Norm pairs

There are different families of s- and t-norms, with each s-norm always having an associated t-norm. Using DeMorgan's law and the complement function, an s-norm can be converted into the t-norm of the respective family and vice versa. This is also referred to as a norm pair.



(a) (b) (c)

Figure 2.2: Fuzzy set operations based on s- and t-norms. a) Examples of two membership functions, b) union of both functions, c) intersection of both functions.

The most common norm pair are the *maximum- and minimum-norm*. Their special significance is due to the fact that they are the only norm pair that does not include additional, problem-specific knowledge. This can be, for example, a mutual reinforcement of membership functions linked in disjunctive manner or a reciprocal weakening of membership functions linked in a conjunctive manner. Using the maximum- and minimum-norm, the intersection and union operations are formulated as shown in Equation 2.5 and 2.6.

$$\mu_{A\cup B}(u) \coloneqq \max\{\mu_A(u), \mu_B(u)\} \tag{2.5}$$

$$\mu_{A\cap B}(u) \coloneqq \min\{\mu_A(u), \mu_B(u)\} \tag{2.6}$$

An example can be seen in Figure 2.2. The complement is implemented using the so-called intuitive complement function:

$$\mu_{A^c}(u) \coloneqq 1 - \mu_A(u) \tag{2.7}$$

It is also known as the standard complement.

2.3.4 Exclusive or

Utilizing the basic logical operations defined above, more complex operations such as the *exclusive or (XOR)* can be formulated:

$$\begin{aligned} \mu_{A\oplus B}(u) &\coloneqq t\Big(s\big(\mu_A(u), \mu_B(u)\big), c\big(t\big(\mu_A(u), \mu_B(u)\big)\big)\Big) \\ &\coloneqq \min\Big\{\max\big\{\mu_A(u), \mu_B(u)\big\}, 1 - \min\big\{\mu_A(u), \mu_B(u)\big\}\Big\} \end{aligned} \tag{2.8}$$

This operation is of importance for the change detection, since it allows to identify those elements of the environment that have not remained the same.

2.4 Fuzzy measures

2.4.1 Fuzzy variables

A mathematically exact and also intuitive explanation of the topic can be found in [Weisbrod, 1996]. The foundation of a fuzzy measure is the *canonical variable.* This is a form of random variable, the value of which should not depend on a random experiment. A canonical variable x is a representative for an element u from the universe $\mathcal{U}_x$. A *fuzzy measure* is described as a mapping that assigns a value within the unit interval to all subsets of a universe. In this context, all subsets are defined as the power set over the universe, thus $\mathcal{P}(\mathcal{U}_x)$.

$$U : \mathcal{P}(\mathcal{U}_x) \to [0, 1] \tag{2.9}$$

For the formal properties of such a fuzzy measure it is referred to Weisbrod [1996]. The semantic difference between uncertainty and vagueness can be illustrated, among other things, by the mapping function. While in case of vagueness the universe is mapped into the unit interval, in case of uncertainty the power set of the universe is mapped. A *fuzzy variable* (x, U) is defined as a canonical variable x in combination with a fuzzy measure, as it is shown in Equation 2.9.

2.4.2 Basic assignments

A *basic assignment* is a mapping such as the one defined in Equation 2.9 and considered to be a fundamental component of a fuzzy measure [Shafer, 1976; Klir & Folger, 1987; Wang & Klir, 1992; Kruse et al., 1993]. The basic assignment $m(A)$ reflects the confidence that a fuzzy variable x has exactly the value of the subset $A \in \mathcal{U}_x$. The value assigned in this way will be also referred to as the mass or evidence mass.

A basic assignment has the following properties. On the one hand, the empty set may only be mapped to zero. The reason for this is that the fuzzy variable x must always take on a value from $\mathcal{U}_x$. Furthermore, the sum of all evidence must be 1. All mass that cannot be assigned to a real subset of $\mathcal{U}_x$ is automatically assigned to the *certain statement*, therefore the subset of $\mathcal{P}(\mathcal{U}_x)$ that contains all elements.

In case $m(A) > 0$, one speaks of a *focal element*. The focal elements are the subsets of the universe $\mathcal{U}_x$ that the available evidence is focused on. There is complete ignorance when the only focal element is the certain statement.

2.4.3 Belief and plausibility measure

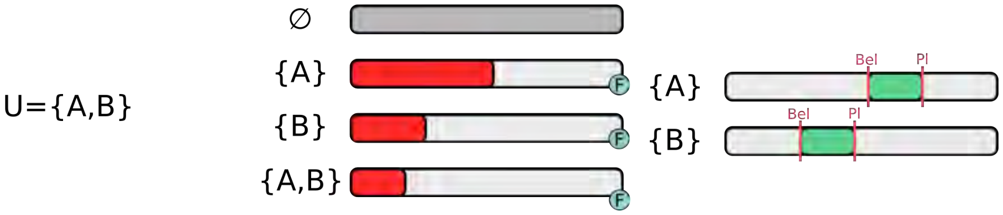
The two basic assignments used in this work are the *belief measure* and the *plausibility measure*. The belief measure $Bel(A)$ equals the mass of all evidences that support the statement $x \in A$:

$$Bel(A) := \sum_{B \subseteq A} m(B) \tag{2.10}$$

The plausibility measure $Pl(A)$ equals the mass of all evidences that do not contradict the statement $x \in A$:

$$Pl(A) := \sum_{B : B \cap A \neq \emptyset} m(B) \tag{2.11}$$

These evidences are all subsets $B \subseteq \mathcal{U}_x$ that overlap with A. The two mentioned measures are well-known in the context of the Dempster-Shafer evidence theory [Dempster, 1967; Shafer, 1976; Fine, 1977]. An example for a fuzzy measure can be seen in Figure 2.3. Evidence mass $m(A)$, the belief measure $Bel(A)$ and the plausibility measure $Pl(A)$ are in relation to each other. If one of them is known, the other two can be derived from it. The belief measure can at most be as large as the plausibility measure. If there is only evidence for all single-element elements of the power set $\mathcal{P}(\mathcal{U}_x)$, the belief and plausibility measure coincide and form a *probability measure*.



(a) (b) (c)

Figure 2.3: Example of a fuzzy measure. a) Universe over which the measure is defined, b) an exemplary fuzzy measure, c) the associated belief and plausibility. All focal elements are marked with a blue symbol.

2.5 General fuzzy reasoning

The basics of general fuzzy reasoning and evidence-based reasoning can be found in Weisbrod [1996]. A fuzzy set $\tilde{A}$ can also be interpreted as a *linguistic term*. x is $\tilde{A}$ is then referred to as *predicate*. x is a canonical variable, which in this context is also called a *linguistic variable* and stands for the vague quantification of an observation. As the terminology implies, fuzzy reasoning is rooted in natural language processing.

2.5.1 Generalized modus ponens

The *linguistic rule* describes a connection between two linguistic predicates. An example of such a linguistic rule is:

$$\textsf{IF } x \textsf{ is } \tilde{A} \textsf{ THEN } y \textsf{ is } \tilde{B}$$

To apply a rule to observations, the *generalized module Ponens* for linguistic predicates is required [Zadeh, 1973].

$$\begin{array}{lr} \textsf{IF } x \textsf{ is } \tilde{A} \textsf{ THEN } y \textsf{ is } \tilde{B} & (rule) \\ x \textsf{ is } \tilde{A}' & (fact) \\ \hline y \textsf{ is } \tilde{B}' & (conclusion) \end{array} \tag{2.12}$$

2.5.2 Implication

The implementation of the generalized modus ponens requires an *implication relation*. It is based on the *fuzzy relation* that maps a fuzzy set $\tilde{R}$ to a product space. It is described by the following membership function:

$$\mu_R : \mathcal{U}_1 \times \mathcal{U}_2 \times \cdots \times \mathcal{U}_k \to [0,1]$$

An *implication relation* is a suitable representation of the linguistic rule $[\tilde{A} \Rightarrow \tilde{B}]$. Both the two fuzzy sets $\tilde{A}$ and $\tilde{B}$ are defined via the power set of their respective universes. The implication relation is realized by the *implication function* $i(\mu_A(u), \mu_B(v))$. The choice of the function i represents a degree of freedom within the framework of fuzzy reasoning.

2.5.3 Composition

A *composition* connects a fact x is $\tilde{A}'$ in the form of a fuzzy set with a rule represented by an fuzzy relation, such as the implication relation. The fact is defined by the power set of the universe. The *s-t composition* is formed by the s- and t-norms, which results in the *max-min composition* [Zadeh, 1965]:

$$\begin{aligned} \mu_{P \circ Q}(a,b) &:= \underset{v \in V}{\mathrm{s}} \Big(t\big(\mu_P(u,v), \mu_Q(v,w)\big)\Big) \\ &:= \max_{v \in V} \Big\{ \min \left\{ \mu_P(u,v), \mu_Q(v,w) \right\} \Big\} \end{aligned} \tag{2.13}$$

2.5.4 Realization of the generalized modus ponens

The implication relation $[\tilde{A} \Rightarrow \tilde{B}]$ is applied to the fact $\tilde{A}$ by linking the fuzzy set $\tilde{A}'$ and the implication relation R' using the composition $\tilde{A}' \circ \tilde{R}'$.

$$\mu_{A' \circ R'}(v) \coloneqq \max_{u \in \mathcal{U}_x} \Big\{ \min \big\{ \mu_{A'}(u), \mu_{R'}(u, v) \big\} \Big\} \tag{2.14}$$

Inserting this into Equation 2.12 results into the *compositional rule of inference* [Zadeh, 1973; Hellendoorn, 1973]. The relation $\tilde{R}$ doesn't have to be an implication, it can be seen as a universal mechanism for evaluating fuzzy relationships between canonical variables.

2.5.5 Applying multiple rules

The topic of applying several rules is discussed in detail in Turksen & Tian [1993]. There are two options when applying multiple rules. A *reduction type inference* relies on minimum formation, which is suitable for reliable information. It is executed by linking the results of multiple rules with an intersection.

With the *expansion type inference* the processing of several rules is based on the union operation. It is suitable for unreliable information, which requires a careful, disjunctive connection [Lee, 1990; Turksen & Tian, 1993]. As already stated above, the sensor systems used are to be considered as rather communicative sources of information, which speaks in favor of expansion type inference.

It is also relevant whether all rules are applied first and then the partial results are aggregated or whether all rules are aggregated and the resulting rule is applied to the fact $\tilde{A}$. The former is called *local rule application*, the latter *global rule application*.

2.5.6 Multiple premises and consequences

The negation is a single-element premise that can be determined using the complement of the fuzzy set. In case of linguistic predicates that are concatenated with *and* respectively *or*, there are multi-element premises. In the further course of this work, there are only concatenations of predicates with an identical universe. Therefore, the concatenations of predicates with *and*-respectively *or*-operators takes place as defined in Equations 2.2 and 2.3.

2.6 Evidence-based reasoning

Evidence-based reasoning is a special case of fuzzy reasoning that assumes a communicative and therefore contradicting source of information. A detailed explanation of the topic as well as the formal background can be found in Weisbrod [1996]. The basic building block of evidence-based reasoning is the *evidence distribution*, which is defined as a mapping of the universe $\mathcal{U}$ into the unit interval:

$$\sigma_x : \mathcal{U}_x \rightarrow [0, 1] \tag{2.15}$$

For $u \in U_x$, the function $\sigma_x(u)$ indicates the degree to which the assumption $x = u$ is supported. Complete ignorance is expressed by $\sigma_x(u) = 0$, whereas $\sigma_x(u) = 1$ expresses precise knowledge. A linguistic predicate x is $\tilde{A}$ can be interpreted as an evidence distribution by specifying $\sigma_x \coloneqq \mu_A$. Rules can be implemented as a common evidence distribution $\sigma_{x,y}$. The composition of evidence distributions is then carried out as stated in Equation 2.14. The *Minimum- or Mamdani-relation* is used as the implication relation:

$$\mu_R(u, v) := \min\left\{\mu_A(u), \mu_B(v)\right\} \tag{2.16}$$

It does not matter whether *global or local rule application* is used when applying several rules, since a combination is determined by expansion type inference, which is in this case via the formation of the maximum. This ensures that the resulting evidence distribution contains the information of all the incorporated evidence distributions.

2.7 Relation between evidence distributions and fuzzy measures

Fuzzy reasoning allows both the vagueness and uncertainty of information to be taken into account. As already explained above, vagueness is represented by *fuzzy sets*, uncertainty by *fuzzy measures*. However, the connection between the two requires explanation.

In evidence-based reasoning, logical operations such as intersection and union are carried out on an evidence distribution. This is to be equated with a fuzzy set, which is therefore regarded as a knowledge base. The tool for formally assessing the completeness of this knowledge base are fuzzy measures. The fundamental idea is that the process of drawing sharp conclusions from vague information leads to statements fraught with uncertainty [Weisbrod, 1996]. For example, the attempt to conclude the exact height of a person from the statement that it is very tall only leads to an interval in which each element is represented with a certain degree of uncertainty. It does not matter here whether the information behind the vague statement was already fraught with uncertainty or has only become a vague statement through abstraction from a precise statement. An illustration of the issues discussed here can be found in Figure 2.4.

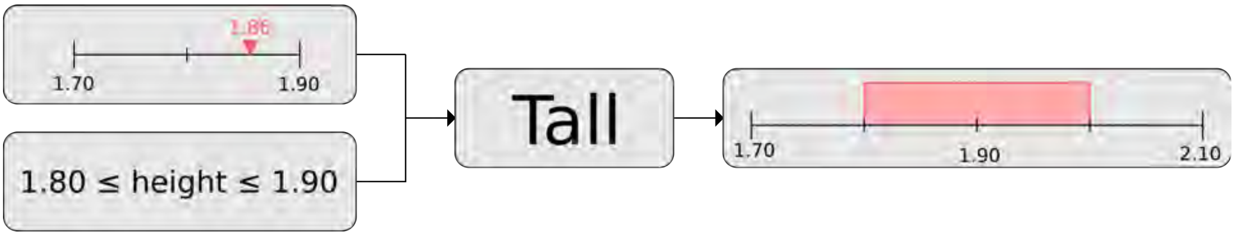
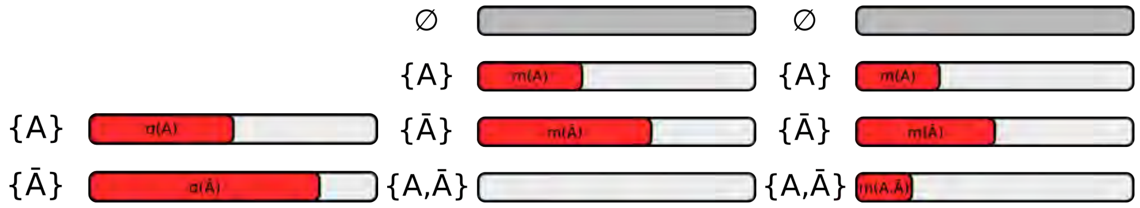


Figure 2.4: Example of a vague statement. left) Statement about the height of a person, given as both as a precise and an uncertain information, center) the derived vague statement, right) an interval fraught with uncertainty, resulting from an attempt to reconstruct the original information.



(a) (b) (c)

Figure 2.5: Relation between evidence distributions and fuzzy measures. a) Evidence distribution, b) associated fuzzy measure with simple normalization, c) associated fuzzy measure with height-based normalization.

Formally, this is done by equating an evidence distribution with a fuzzy set [Weisbrod, 1996]. This is possible because the trust placed in a statement by a basic assignment is equivalent to the support of the statement by the evidence distribution. However, it must be taken into account that a basic assignment must be normalized. The choice of the normalization factor is crucial. A simple approach would be to divide each element by the sum of the mass of all elements of the evidence distribution. However, this would cause that only the single-element elements are focal elements. This in turn would lead to the credibility and plausibility coinciding and forming a probability measure. Therefore, the height of the evidence distribution, which acts as a measure for the overall evidence, is used for normalization:

$$H(\sigma_x) \coloneqq \max_{u \in \mathcal{U}_x} \{\sigma_x(u)\} \tag{2.17}$$

As suggested by Weisbrod [1996], the normalization takes place via the following case distinction:

$$\forall A \subset \mathcal{U}_x : m(A) \coloneqq \begin{cases} \frac{H(\sigma_x)}{\sum_{u \in \mathcal{U}_x} \sigma_x(u)} \sigma_x(u), & \text{if } A = \{u\}, u \in \mathcal{U}_x \\ 1 - H(\sigma_x), & \text{if } A = \mathcal{U}_x \\ 0, & \text{else.} \end{cases} \tag{2.18}$$

The direct implication is that there is no longer any value that is fully supported. The mass lost as a result is assigned to the trivial statement $x \in \mathcal{U}_x$. As a result, the credibility and plausibility measures are different and therefore do not form a probability measure. An illustration of the difference between simple normalization and height-based normalization can be found in Figure 2.5. The scaling is compatible with both the inference mechanisms as well as with the concept of fuzzy sets. This is because the logical operations used are invariant to such manipulations [Weisbrod, 1996].

3 Fuzzy spatial reasoning

This chapter describes the basic principles of the *fuzzy spatial reasoning*, that was developed as part of this work. The representation of spatial data as well as the mechanisms for inference are explained. The basic idea of this technique is to map the distribution of evidences for and against a given information to a three-dimensional space. Fuzzy spatial reasoning is based on two components. The first component is a volume-based spatial data structure that links an area in three-dimensional space with a payload. The second component is said payload, which manages the evidence, performs the mathematical part of the inference and allows a statement about the *degree of certainty* and the *degree of ignorance* connected with the represented information.

The chapter is divided into three parts, with the first two parts covering the above mentioned components of fuzzy spatial reasoning. The third part introduces a selection of so-called spatial predicates relevant for this work and describes how they are generated. Spatial predicates are fuzzy representations such as occupancy grids. These form the building blocks which are combined by means of the inference mechanisms of fuzzy spatial reasoning.

3.1 Organization of large data quantities

LiDAR data is often available in large quantities, so organizational structures are needed to manage them. This section therefore describes the approach used to manage said data sets, as well as other resources such as the data structures for spatial fuzzy reasoning. This part is technical in nature and not directly related to spatial fuzzy reasoning, but needs to be explained to understand the overall concept.

3.1.1 Challenges of data management and processing

Mobile laser scanning produces large quantities of individual LiDAR measurements, the handling of which presents a number of challenges. First and foremost here is the organization of data. A measurement run of a few minutes can lead to billions of samples. A data set that is too large cannot be processed in one piece if it exceeds the maximum working memory of a computer. Efficient memory management techniques are therefore required so that data can be cached during processing. It must be taken into account that interactions with storage devices such as hard disks are orders of magnitude slower than interactions with the working memory.

Another issue is that for many applications, all measurements must be accessible for a given area. Due to the layout of a city, many streets have to be visited several times during the acquisition process. In order to access all measurements for one of the streets in question, the entire data set would have to be searched. Storing the data in the form of individual scans would therefore require additional processing each time the data is accessed. It is therefore necessary to store the data in such a way that such requests can be fulfilled without much computational effort.

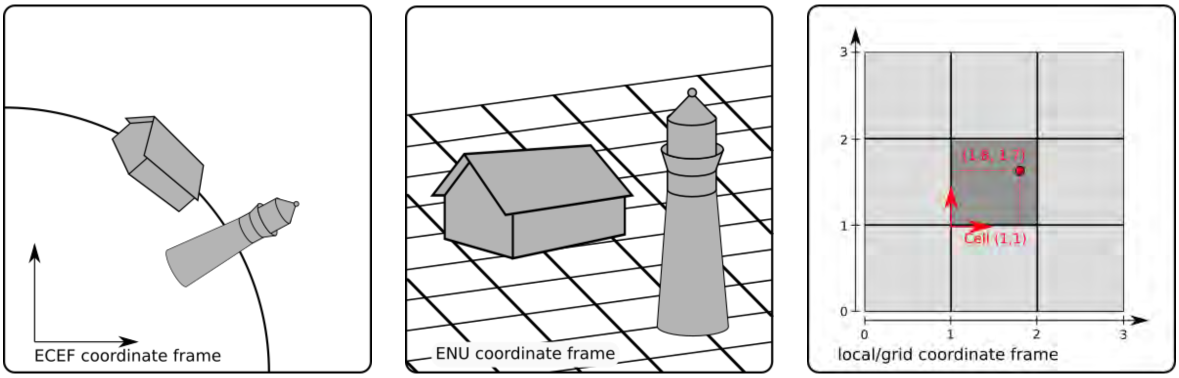
The possibility for parallel processing represents a third challenge. An algorithm can only be optimized up to a certain point. All other speedups arise only when the algorithm is instantiated multiple times and applied in parallel to the data at hand. While this does not reduce the computational effort as optimization does, it does drastically reduce the overall run time. The challenge is to find a way to parallelize a particular process, as this is not always possible. The alternative is to partition the underlying data. The latter has the advantage that the method itself does not have to be parallelizable at all. Next, it is explained how the data structure underlying the fuzzy spatial reasoning deals with the above challenges.

3.1.2 Data organization

The approach to organizing the data is based on previous work [Gehrung et al., 2016]. A simple but efficient way to overcome the above challenges is to split the data into small chunks. The details associated with this are explained below.

Local coordinate frame

An key cornerstone for data organization is the *local coordinate frame*, for which a Cartesian coordinate system was chosen. The use of an *earth-centered, earth-fixed (ECEF)* coordinate system was deliberately avoided because the tilt of the data in the coordinate system depends on the latitude (cf. Figure 3.1a). Instead, all measurements were transformed into a *local tangent plane coordinate frame*, that is into a right-handed *east, north, up (ENU)* coordinate frame. A specific point in the data set was chosen as the point of origin. This procedure ensures that the measurements are represented in a way that appears natural to the observer (cf. Figure 3.1b). This facilitates the interpretation of the measurement data.



(a) (b) (c)

Figure 3.1: Overview of the coordinate systems used. a) Symbolic measurement data appears skewed when observed in an ECEF coordinate frame, b) the same data after transformation into an ENU coordinate frame, c) the coordinate frame of a single cell embedded in the local/grid coordinate frame.

Addressing scheme

Based on the local coordinate frame, an intuitive mapping of data to chunks can be done by tessellating the space into a regular, three-dimensional grid. A cell can therefore be addressed based on its position in the grid. This addressing scheme can be continued infinitely along all axes. A predefined bounding box, as is necessary with other approaches such as the one proposed by Hornung et al. [2013], is therefore not required.

Furthermore, a direct mapping between spatial coordinates within the local coordinate frame and the addresses within the grid is possible. The relationships between the local and grid coordinate frame as well as the one of a single cell are illustrated in Figure 3.1c. Converting from local coordinates p_{local} to grid coordinates p_{grid} is defined as

$$p_{grid} = \begin{bmatrix} \frac{1}{l} & & \\ & \frac{1}{l} & \\ & & \frac{1}{l} \end{bmatrix} p_{local}, \tag{3.1}$$

where l is the edge length of a cell in the regular grid. The places before the decimal point indicate the index in the grid, which can therefore be determined from a position in grid coordinates by rounding off. The inverse transformation, i.e. from grid coordinates to a point in the local coordinate frame, is defined as

$$p_{local} = \begin{bmatrix} l & & \\ & l & \\ & & l \end{bmatrix} p_{grid}. \tag{3.2}$$

Calculating a position in the coordinate frame of a grid cell is carried out by subtracting the cell address $\lfloor p_{grid} \rfloor$ from the position p_{grid} in grid coordinates. As octrees are a focal point of this work and will be addressed frequently in the further course, the term *grid cell* would only cause confusion. The same applies to the term *supervoxel*. Therefore, each grid cell is referred to as *3D tile*, due to its similarity to a tile as it is used in geodesy.

3D tile dimensions

It is also the aspect of octrees that has an influence on the size of a 3D tile. As for octrees, the space encompassed by one is successively divided into higher and higher resolution areas. The edge length of an octree cell is halved with each octree resolution level. Thus, the edge length of a 3D tile must be a multiple of the edge length of a voxel of the highest resolution level. A 3D tile should be small enough so that many of them can be kept in memory, but also large enough that the data is not too fragmented. A good tradeoff between the two requirements is 25.6 m, which is a multiple of the maximum resolution of 10 cm used in this work.

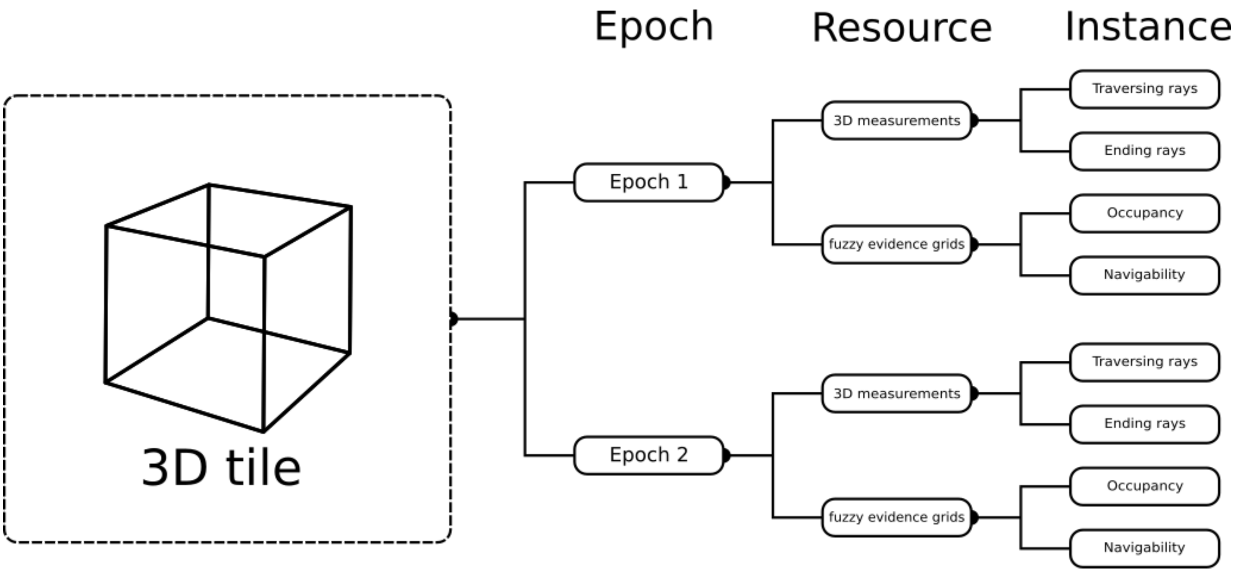


Figure 3.2: Organizational levels of a 3D tile. Data is organized at the level of *epochs*, *resources* and *instances*. The latter are specific versions of a resource for a specific point in time.

Levels of organisation

For the structured storage of data, there are several levels of organization in each tile (cf. Figure 3.2). The first one is the *epoch*. Since the focus of this work is on change detection, a distinction between different time periods is necessary. Each time period has its own pool of data. The next level of organization is the *resource*. It is subordinate to the epoch, which can have several resources. A resource comprises all data of a category and epoch for the area covered by the 3D tile. For example, a resource can be a series of 3D measurements or an octree-based *fuzzy evidence grid* as described in Section 3.2. It may happen that there are several versions of a resource per epoch. In this case one speaks of an *instance* of said resource. For example, there may be multiple fuzzy evidence grids describing properties of the underlying area such as *occupancy* or *navigability*. Epoch, resource and instance form a hierarchical structure that enables efficient access to the data stored in a 3D tile.

Memory management

The basic idea of memory management is to keep the 3D tiles that are still used in memory and to swap out all the others. If this happens in the background without an application being aware of it, this is called *transparent caching*. The *least recently used (LRU)* paradigm is used as a caching strategy. This means that the least used element is removed from the cache whenever it is full and an non-cached 3D tile is requested. If the least used element has been changed, it is written to a storage medium, otherwise it is discarded.

Resource filtering

Filtering is another way to save run time when loading 3D tiles. The idea is to filter data at the organizational levels illustrated in Figure 3.2. For this purpose, a whitelist describing the required epochs, resources and instances is passed to the cache. The cache then ensures that only listed data is loaded for each 3D tile. This not only reduces the required memory, but also drastically decreases the run time, as the loading time accounts for a significant part of it.

3.1.3 Measurement data import

The breakdown of the data into processable chunks requires that the measurement data is distributed across the 3D tiles. Here, not only the surface measurements but also the free space they traverse must be taken into account. The latter is located between the measured surface point and the sensor position. The combination of free space and surface point is therefore also called a *ray*. The rays must be distributed to the 3D tiles they traverse. For performance reasons, all rays that belong to a 3D tile are truncated at its boundaries. Since this means that it is no longer possible to distinguish between traversing rays and rays ending within the 3D tile, both categories of rays are represented by mutually exclusive sets of indices.

The process of distributing 3D measurement to the corresponding tiles is referred to as *measurement data import*. Each surface point is handled in combination with the corresponding sensor position, as mentioned above. Using the *ray casting* algorithm proposed by Amanatides & Woo [1987], the 3D tiles traversed by each ray are determined. This is done by deriving the address of the start tile using Equation 3.1 and then successively using the ray's direction to determine the following tiles up to the end of the ray. This process is illustrated in Figure 3.3. The rays are truncated to each of the traversed tiles using a technique proposed by Kay & Kajiya [1986]. This is a fast slab-based intersection test between a ray and an axis aligned bounding box as used in computer graphics.

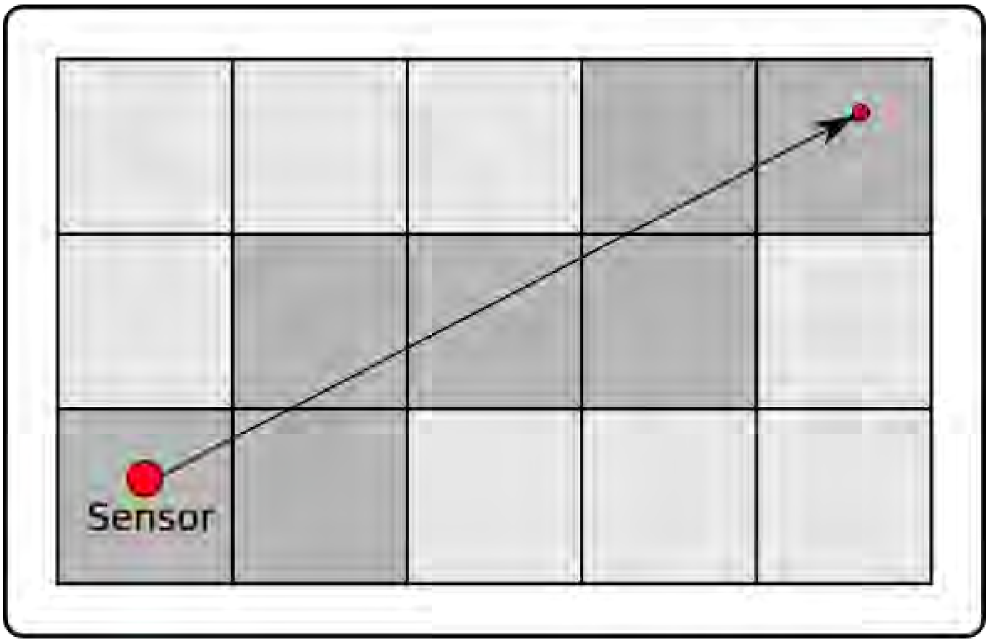


Figure 3.3: Distribution of rays on tiles. Raytracing is used to determine the tiles intersected by the ray, where the latter is mapped to each traversed tile.

3.2 Components of fuzzy spatial reasoning

This section describes *fuzzy spatial reasoning*, which is one of the main contributions of this work. First, a description is given of how the theory of fuzzy reasoning must be adapted. Based on this, the *fuzzy evidence grid* is defined and it mechanisms for logical inference are explained. Certain core elements are explained in detail. This includes the methods used to determine the degrees of uncertainty and ignorance, the membership function used to convert measurements to evidences and the defuzzification.

3.2.1 Concept for evidence representation

Fuzzy spatial reasoning utilizes a representation that stores evidence in three sequential layers. Each of them holds the same information, but in a different form. The three layers and their relationship to each other are shown in Figure 3.4. The bottom layer symbolizes the *raw data* that is to be integrated into the representation. In case of an occupancy grid, it is 3D measurement data. These are converted into a *fuzzy set*. For this an *membership function* which is described in detail in Section 3.2.6 is used. Fuzzy sets act as a middle layer and are used to logically combine information. For example, a representation of *changes in the environment* can be *and*-linked with a representation marking all *vehicles* in the scene in order to get the *all changes related to vehicles*.

In order to interpret a fuzzy set, it must be transformed into a *fuzzy measure*. These form the top layer that enables evaluation and interpretation. In addition to the *degree of certainty* regarding the available evidence, the *degree of ignorance* can be quantified. The process of transforming a fuzzy set to a fuzzy measure is described in Section 2.7 and involves applying Equation 2.18.

3.2.2 Adjustments to fuzzy reasoning

Evidence for and against a certain information is represented by a fuzzy set. This means that the two elements of said fuzzy set are by definition *the opposite of each other*. This has consequences for the logical operations which are used in the context of inference, since the fuzzy reasoning

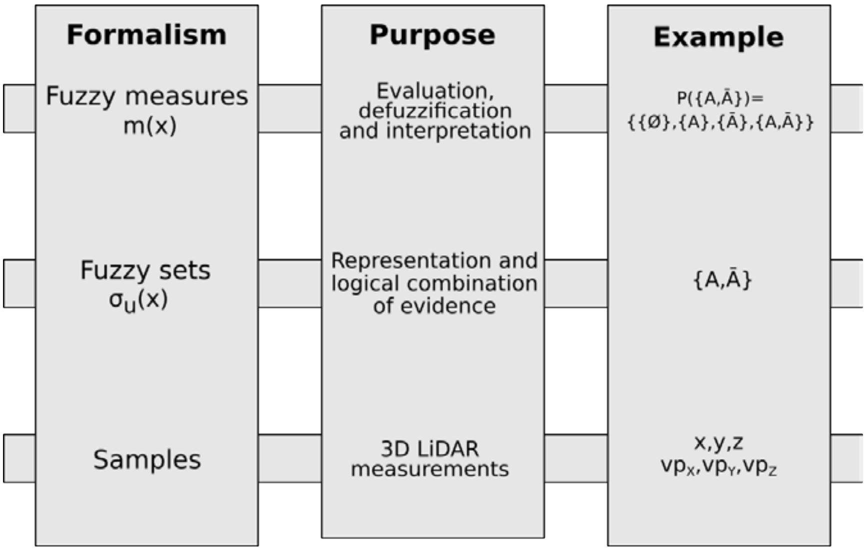
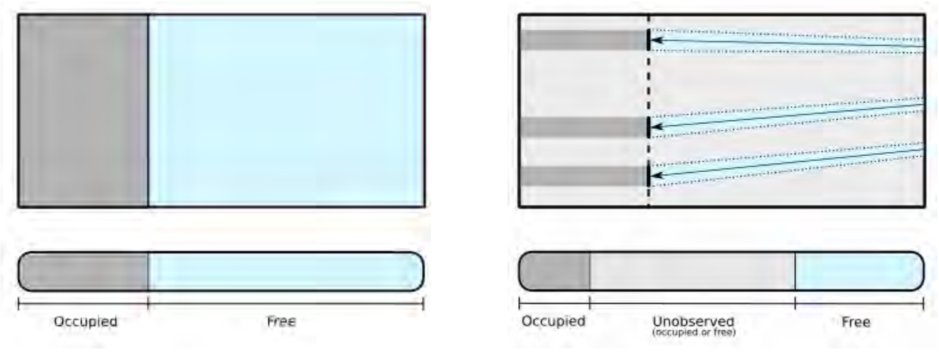


Figure 3.4: Overview over fuzzy spatial reasoning. Raw data (bottom layer), here in form of 3D measurements, is transformed into fuzzy sets used for logical operations (middle layer). The fuzzy sets in turn are translated into fuzzy measures (top layer) for the purpose of evaluation and interpretation.

described in Chapter 2 has to be adapted to the case at hand. This section explains the underlying problem and the solution found for it.

Consequences of an imperfect measurement function

A perfect measurement function would capture all information relevant in the context of an application not only with perfect precision and accuracy, but also with *completeness*. The latter is best illustrated by an example involving the occupancy of a volume of space. If the volume is measured to be 30 % occupied, then a perfect measurement function would allow the conclusion that the volume is also 70 % free. The entire space would be observed and unobserved space would not be present (cf. Figure 3.5a). Therefore, to represent occupied and free space, only a set with a single element representing the occupancy level is required. The degree of free space can then be derived using the standard complement, since the observations are generated by a physical process due to which the space can be either *free or occupied*, but *not both at the same time.*



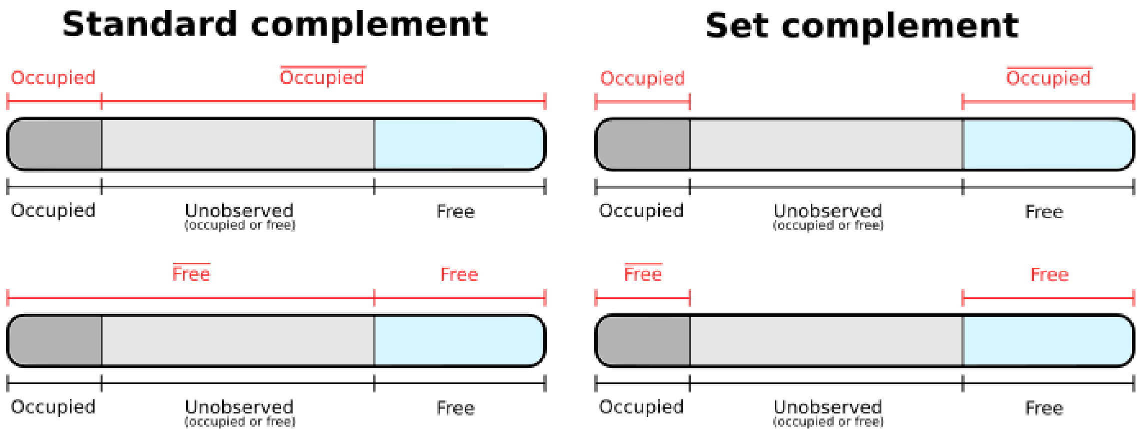
(a) (b)

Figure 3.5: Consequences of an imperfect measurement function. a) An exemplary scene with free (blue), occupied (dark gray) and unobserved (light gray) space, completely observed due to a perfect measuring function, b) partial observation due to an imperfect measurement function, as is the case with mobile mapping.

In reality, however, the environment to be measured is rarely fully recorded. In other words, there is always some part of the environment which is not observed. Therefore, sensor measurements are only an approximation of the environment. In the above example, the measurement with a laser scanner corresponds to the random sampling of free and occupied space (cf. Figure 3.5b). This results in both the degree of free space and the degree of occupied space being *underestimated*, but *never overestimated*. Consequently, there is a (not necessarily contiguous) area that has not been observed. It can be both *partly free and partly occupied*. As a consequence of the imperfect measurement function, it is no longer possible to conclude the degree of free space from the degree of occupied space. Whereas a perfect measurement model would have required only a set with a single element, in this case two elements are required. By definition, the *second element is always the inverse of the first element*, since the former contains all the evidence for the information to be represented and the latter all the evidence against it. Hence these two elements are called A and $\bar{A}$, each collects the evidence for one of the two cases mentioned. For the given example, the first element represents the degree of occupancy of a volume and the second element the degree of unoccupied space, i.e. free space.

The role of the complement

Given the representation proposed in the last section, negation based on the standard complement now no longer leads to the desired result. This can be shown using the boundary conditions of classical Boolean logic as a benchmark. The reason for this is the unobserved space, which plays the role of a third case, or rather a *pseudo-case*, since it is not directly measurable. Applying the standard complement to an element causes the element's complement to contain not only the evidence mass for the other element, but also the evidence mass associated with unobserved space (cf. Figure 3.6a). In case of occupied space, the complement corresponds to the evidence mass for free space plus the evidence mass for unobserved space. For free space it is similar, here the complement is equal to the evidence mass for occupied space plus that for unobserved space.



(a) (b)

Figure 3.6: Comparison of standard complement and set complement. a) The standard complement includes the evidence mass for unobserved space, b) the set complement includes only the evidence of the opposed case.

The use of logical operations that contain a negation leads to a violation of the *boundary conditions* requirement mentioned above. In other words, the results for edge cases are no longer comparable to those of classical Boolean logic. Another way, however, is to interchange the value of the element to be negated with the combined evidence mass of the other element(s) that have no intersection with said element. By doing this, the evidence mass of the unobserved space is explicitly excluded (cf. Figure 3.6b).

Operation	**Element $\boldsymbol{A}$**	**Element $\bar{\boldsymbol{A}}$**
And	$\min(\mu_{A_1}, \mu_{A_2})$	$\max(\mu_{\bar{A}_1}, \mu_{\bar{A}_2})$
Or	$\max(\mu_{A_1}, \mu_{A_2})$	$\min(\mu_{\bar{A}_1}, \mu_{\bar{A}_2})$
Xor	$\max(\min(\mu_{\bar{A}_1}, \mu_{A_2}), \min(\mu_{A_1}, \mu_{\bar{A}_2}))$	$\min(\max(\mu_{A_1}, \mu_{\bar{A}_2}), \max(\mu_{\bar{A}_1}, \mu_{A_2}))$
Inhibition of A	$\min(\mu_{A_1}, \mu_{\bar{A}_2})$	$\max(\mu_{\bar{A}_1}, \mu_{A_2})$
Inhibition of $\bar{A}$	$\min(\mu_{\bar{A}_1}, \mu_{A_2})$	$\max(\mu_{A_1}, \mu_{\bar{A}_2})$

Table 3.1: Operations required for using the set complement.

$$\mu_{A^c}(u) = \begin{cases} \mu(\bar{A}), & \text{if } u = A \\ \mu(A), & \text{if } u = \bar{A} \end{cases} \tag{3.3}$$

In this work, this is referred to as the *set complement*. For the two-element case, it is defined as shown in Equation 3.3. The application of this complement instead of the standard complement avoids the violation of the boundary conditions mentioned above and the results are compatible with those of classical logic.

Consequences for logical operations

Using the set complement has consequences for performing *logical operations*. Whenever an operation is applied in fuzzy reasoning, every element of the set is affected, i.e. the same operation is applied to every element of the set. As described above, the second element $\bar{A}$ is by definition the complement of the first element A. Thus, although the applied operation is the same for element $\bar{A}$, the former must also be negated before it is applied to the latter. This is done by applying De Morgan's laws. Starting from Equation 2.6, the *and*-operation is negated and rewritten using De Morgan's laws as follows:

$$\begin{aligned} \overline{s(\mu_{A_1}, \mu_{A_2})} &= \overline{\min(\mu_{A_1}, \mu_{A_2})} \\ &= \max(\mu_{\bar{A}_1}, \mu_{\bar{A}_2}) \end{aligned}$$

A similar procedure is carried out in the case of the *or*-operation. Every compound operation is treated analogously, i.e. it is first composed and then negated using De Morgan's laws. A list of common operations is shown in Table 3.1.

3.2.3 Definition of fuzzy evidence grids

The theoretical framework for fuzzy reasoning is described in detail in Chapter 2 and its modification for the case at hand is discussed in the previous section. However, the former is designed for handling abstract semantic information in the context of human language. Therefore, one of the main points of this work is the extension of said framework to spatial information. This involves addressing two major topics. The first one deals with creating a form of representation suitable for handling fuzzy information in a spatial context. The nature of the representation, which is denoted as a *fuzzy evidence grid*, is discussed in this section, as well as the reasons behind the design decisions made. Based on this, the second topic deals with the inference on fuzzy evidence grids, which is discussed in Section 3.2.4.

Representation of spatial information

Evidence grids, as described in Section 1.2.2, are a suitable starting point for linking information to space. These offer a number of properties that are advantageous for the representation of spatial data. One of them is the discretization of space, where certain areas are grouped together and described by a single set of parameters. The means for space partition is the so-called *voxel*. It is a unit of three-dimensional spatial information that describes the properties of the area enclosed within its bounding box. If a hierarchical grid in form of an *octree* is used instead of a grid, the resolution can be adapted locally to the spatial distribution of the information. Homogeneous areas can be represented by few large voxels, heterogeneous areas by several small ones. It should be noted that, from a theoretical point of view, the representation is still a grid, but with a variable resolution. Hence the term *evidence grid* is still appropriate.

The use of an octree has the additional advantage that several levels of resolution are available for the data. This can be useful, for example, in order to make preliminary calculations at a crude resolution, before applying a computationally expensive operation to a higher resolution. Furthermore, utilizing an octree allows the definition of a fixed upper memory limit. The maximum resolution and thus the maximum octree depth d_{max} is fixed, therefore the memory requirement is equal to a grid with the same resolution. Summing the memory requirements with those of the grids of the other resolution levels gives the maximum memory usage for the octree.

From a formal point of view, the voxels are not independent of one another. For example, if an occupancy grid representing the environment contains a car, all octree cells that encompass the space occupied by the vehicle depend on said vehicle and are therefore interlaced with each other. Modeling all interrelationships requires detailed knowledge of the environment and addressing all of them makes the problem very expensive from a computational point of view. This is the main reason why the *assumption of independence* is often promoted in the literature, knowing that said assumption is only an approximation of the actual events [Thrun et al., 2005]. It states that the information in each cell is independent of the information in all neighboring cells. The assumption reduces the computational requirements drastically and is therefore a necessity, even if it is not entirely correct.

Combining octrees and fuzzy information

Fuzzy information in a spatial context can be represented by choosing an *evidence distribution* as the voxel payload. At this point, it is worth remembering that this distribution is essentially a fuzzy set. The universe that defines the fuzzy set contains only two elements and is similar to the one shown in Figure 2.5. The elements represent the collected evidence mass *for and against the presence of a certain information*, represented by the elements A and $\bar{A}$. Note that the nature of this information does not have to be explicitly defined and the distribution is only given its meaning by interpretation. The lack of an explicit description of the nature of the information and the resulting need for interpretation offers the flexibility required to represent arbitrary information. It also allows the combination of different pieces of information that have different meanings.

The chosen universe resembles that of approaches such as the one published by Hornung et al. [2013]. However, the representation of the actual information is different. Since the two opposing cases are not represented by a single variable as it is the case with *log odds*, but are instead managed separately, *contradictions* can be modeled. A contradiction exists if there is both evidence for and against the represented information. Or in other words, that means a high membership for both A and $\bar{A}$. In the case of mobile laser scanning, for example, this can indicate an area of space that has been crossed by a moving object.

Differentiation from own previous work

The procedure motivated above for combining an octree with an evidence distribution for fuzzy information handling results from previous work. It is rooted in an approach utilizing so-called *density functions* [Gehrung et al., 2019b]. This form of representation describes the spatial distribution of the intensity of an information. The techniques related to the evidence grid underlying the approach are the same as presented here. Instead of an evidence distribution, the accuracy of the information was used as voxel payload. The combination of information was carried out using an arithmetic specially designed for this purpose. The particular approach had two disadvantages. First, it is not able to represent uncertainties, which means that the reliability of the information represented cannot be assessed. Second, contradictions cannot be dealt with. Both problems are resolved with the fuzzy reasoning based approach proposed in this work. *Fuzzy evidence grids* allow to derive the degree of uncertainty encoded in the fuzzy information. This allows to address the reliability of information without having to resort to a separate theoretical construct. More details are discussed in Section 3.2.5.

3.2.4 Logical inference

The topic of logical reasoning consists of three parts. The first part deals with the formalization of the inference process, the other two with the structural combination of the underlying data structure and with the methodological combination of evidence in terms of fuzzy logic.

Formalization

The logical inference describes the combination of two fuzzy evidence grids via an operation. This creates a new fuzzy evidence grid that contains the result of the operation. Since a fuzzy evidence grid is essentially a spatial grid, it can also be interpreted as a *set of voxels*. An operation on a fuzzy evidence grid can therefore be seen as a *set operation*. Equation 3.4 defines the combination of two fuzzy evidence grids $\boldsymbol{\sigma_x}$ and $\boldsymbol{\sigma_y}$ with the operation $\bullet$ as:

$$f_{\bullet}(\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y) := \{\sigma_x \bullet \sigma_y | \forall(\sigma_x, \sigma_y) \in \boldsymbol{\sigma}_x \times \boldsymbol{\sigma}_y, l(\sigma_x) = l(\sigma_y)\}. \tag{3.4}$$

The above definition is to be interpreted as follows. The function $f_{\bullet}(\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y)$ combines the fuzzy evidence grids $\boldsymbol{\sigma}_x$ and $\boldsymbol{\sigma}_y$. The return value of said function is an evidence grid that contains the result of the combination. All elements of both sets $\sigma_x \in \boldsymbol{\sigma}_x$ and $\sigma_y \in \boldsymbol{\sigma}_y$ are combined using the previously mentioned operation $\bullet$, but only if the location $l(.)$ of both elements is equal, which is expressed by $l(\sigma_x) = l(\sigma_y)$. In layman's terms, this means that only the grid cells at the same location are combined with one another.

So far, the combination of entire fuzzy evidence grids has been described from a formal point of view. However, the question arises as to how the evidence distributions stored in the cells are combined with one another. From a formal point of view, the combination of evidence distributions σ_x and σ_y is defined as shown in Equation 3.5. σ_z is the evidence distribution in the cell of the fuzzy evidence grid that stores the result of the operation.

$$\sigma_z = \sigma_{x \bullet y} \circ \sigma_{(x \bullet y) \Rightarrow z} \tag{3.5}$$

In the above equation, the result of operation $\sigma_{x \bullet y}$ is linked with the implication function. As explained in Section 2.5.4, this is done using the *compositional rule of inference.* Taking into account said rule from Equation 2.14 and the *Mamdani-relation* from Equation 2.16, the following equation for the combination of evidence results

$$\sigma_{x\bullet y\Rightarrow z}(v) := \max_{u\in\mathcal{U}}\Big\{\min\Big\{\sigma_{x\bullet y}(u), \min\left\{\sigma_x(u), \sigma_x(v)\right\}\Big\}\Big\}, \tag{3.6}$$

where $\sigma_{x\bullet y\Rightarrow z}$ is the resulting evidence distribution. The above equation is used to determine the said evidence distribution for each element $v \in \mathcal{U}$.

Structural combination

The inference in fuzzy evidence grids consists of two parts. Part one deals with the wrappers of the evidence distributions, i.e. the octrees, which are responsible for the efficient data storage of spatial data. That part is discussed here. The second part deals with the inference of the spatial information itself and is discussed in the next section.

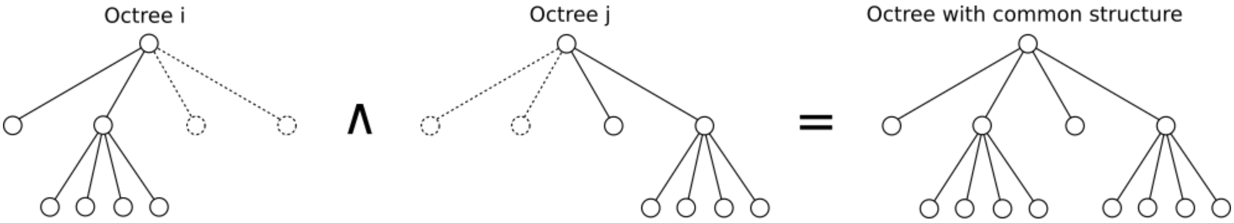


Figure 3.7: Combination of two octrees into an octree with a common structure. The latter stores the fuzzy evidence representation obtained by combining two fuzzy evidence grids.

When combining two fuzzy evidence grids, it is necessary that the result can be stored just as efficiently as it is the case with the input. If both octrees to be combined have an identical structure, then the structure of the common octree does not differ either. The result is thus represented efficiently. In a real application, however, it is unrealistic that this case will occur. Rather, it is more likely that both octrees resolve different areas with different levels of detail. Therefore, the common octree that does justice to both input octrees is resolved fine enough so that no information is lost, but at the same time resolved coarsely enough so that no non-existent information is added. This is illustrated in Figure 3.7. The octree with such a structure is also the most efficient representation of the information contained in both octrees. Needless to say, both octrees need to occupy the same space and therefore have the same bounding box. This is taken for granted, since each octree is located in a 3D tile with fixed boundaries.

The basic idea explained above leads to the algorithm for constructing the common octree in a recursive manner, as described in Algorithm 1. The algorithm is a modified version of the one published in own previous work [Gehrung et al., 2019a]. There are two major steps which are repeated recursively. The first step is executing the logical operation based on the evidence mass of both nodes. The exact procedure depends on the operation. The second step is selecting the nodes in both octrees that are used for further recursion. There are up to eight child nodes and recursion is done separately for each child node index. For a given index, the following case distinction is made:

- If both nodes have a child node for the given index, recursion is continued with said child nodes.

- If one of both nodes has no child node (or has not had a child node in the past, which can be determined based on its depth in the octree), a node without any evidence is generated. Recursion is continued with said node and the available child of the other node.

```
Data: Octrees $node_i$ and $node_j$.
Result: Octree $node_c$ with common structure.

Function construct($node_i$, $node_j$, $node_c$):
    // Merge payloads and store result in $node_c$.
    updatePayload($node_i$, $node_j$, $node_c$)

    // Construct the octree with common structure.
    for $child \leftarrow 0$ to 7 do
        // Ensure that at least one node has a child at index $child$.
        if $getDepth(node_i) > getDepth(node_j)$ then
            $hasChildNode \leftarrow$ hasChild($node_i$, $child$)
        else if $getDepth(node_i) < getDepth(node_j)$ then
            $hasChildNode \leftarrow$ hasChild($node_j$, $child$)
        else
            $hasChildNode \leftarrow$ hasChild($node_i$, $child$) $\vee$ hasChild($node_j$, $child$)

        // Select the nodes for recursion.
        if $hasChildNode$ then
            if $getDepth(node_i) < getDepth(node_j)$ then
                $next_i \leftarrow$ createEmptyNode()
            else
                if $hasChild(node_i, child)$ then
                    $next_i \leftarrow$ getChild($node_i$, $child$)
                else
                    $next_i \leftarrow$ createEmptyNode()
            if $getDepth(node_j) < getDepth(node_i)$ then
                $next_j \leftarrow$ createEmptyNode()
            else
                if $hasChild(node_j, child)$ then
                    $next_j \leftarrow$ getChild($node_j$, $child$)
                else
                    $next_j \leftarrow$ createEmptyNode()

        // Recurse.
        construct($next_i$, $next_j$, createAndGetChild($node_c$, $child$))
```

Algorithm 1: Recursive generation of the octree $node_c$ with common structure from two octrees $node_i$ and $node_j$.

The latter case is necessary whenever an area has not been observed in one of the two epochs. Hence it is simulated by a node without any evidence mass. It is possible to use the information stored in a node with a higher resolution, but this would lead to the formation of artifacts and thus to erroneous results. In order to terminate at the leaf nodes, the recursion is only continued if at least one node has a child node for the given index.

Operations for combining evidence

The formal and technical side of combining fuzzy evidence grids have been shown in the last sections. This section explains how to combine the evidence distributions contained in each grid cell. As mentioned above, two fuzzy evidence grids are combined using logical inference in order to derive a new representation with a different semantic meaning. Since evidence distributions are defined as fuzzy sets, the theoretical background described in Section 2.3 can be applied to logically link said distributions. The basic operations such as the single-valued *complement* as well as the two-valued *intersection* and *union* are defined in Equation 3.3, 2.5 and 2.6. They are applied as described in Algorithm 1. It is also possible to use compound two-valued operations derived from the basic operations, such as the *exclusive or* shown in Table 3.1. The fuzzy evidence grid obtained after applying a logical operation still contains the same information as the fuzzy evidence grid(s) involved, however, the semantic meaning is different. For example, by linking information about navigable surfaces and the footprint of vegetation using a logical *intersection*, knowledge about possible shady parking spaces can be derived. Another *intersection* with the sphere of influence of the local cinema highlights all areas where, most likely, one may not find a parking space on a sunny Saturday afternoon[*].

3.2.5 Uncertainty handling

As described in Chapter 3, the uncertainty is quantified using fuzzy measures. The relation between the fuzzy measures and the evidence distributions used to represent information is discussed extensively in Section 2.7. It is also stated there that a fuzzy measure is derived directly from a fuzzy set (and thus also from an evidence distribution). Because the combination of evidence distributions using logical operations again leads to an evidence distribution, the fuzzy measure can be derived at any time. No separate handling of the uncertainty is therefore necessary.

3.2.6 Membership function

A membership function establishes the relationship between measured evidence and the membership to the element of the evidence distribution that represents said evidence. Since the focus of the work is on 3D measurements, said evidence is available in the form of samples of the environment geometry. Any algorithm that acts as a membership function must therefore derive the membership for each voxel based on the number of samples contained therein. For each voxel there are i samples for element A and j samples for element $\bar{A}$. By definition, the membership to an element of the evidence distribution must be in the interval $[0, 1]$. A simple way to map the number of samples to a membership would be to divide them by the sum of all samples. However, this approach is misleading, as it results in $\sigma_A + \sigma_{\bar{A}} = 1$. This rules out the case of maximally contradicting evidence, expressed by $\sigma_A = 1$ and $\sigma_{\bar{A}} = 1$, which would lead to $\sigma_A + \sigma_{\bar{A}} = 2$. It also limits the ignorance class represented by $m(A, \bar{A})$ to a maximum value of 0.5. As a result of the normalization method described above, the overall system is restricted in such a way that its informative value is greatly restricted.

[*] This rule of thumb may not apply to the year 2020/2021, when large parts of this work were written.

In order to prevent this, both classes are handled *independently* of each other. A membership function is required for mapping from samples to the interval $[0, 1]$. The function should be continuous and allow as many intermediate stages as possible. Two functions are considered. The *linear function* is the most simple function that meets the criteria mentioned above. The *logistic function* is a more complex function that suppresses intermediate values and mostly maps towards extreme values at both ends of the spectrum. In addition, a third, task-specific membership function is examined to generate the occupancy representation, which, contrary to the above, generates a certain degree of dependency. This is required to compensate for the effect described in Gehrung et al. [2017].

Linear membership function

The linear membership function is defined as shown in Equation 3.7. The number of samples for a class u is represented by x.

$$\sigma_u(x) := m \cdot x + n \tag{3.7}$$

The parameters m and n are determined in a way that the line intersects with two points (cf. Figure 3.8a). The first point is the origin of the coordinate axes. To do this, n must be set to zero. The slope m is chosen so that the line intersects with the second point (x_1, y_1), where x_1 corresponds to the median number of samples per voxel and $y_1 = 1$. Therefore, the slope is determined as:

$$m = \frac{y_1}{x_1} = \frac{1}{x_1} \tag{3.8}$$

Since x_1 corresponds to the median number of samples, a normalization to the interval $[0, 1]$ is required.

Logistic membership function

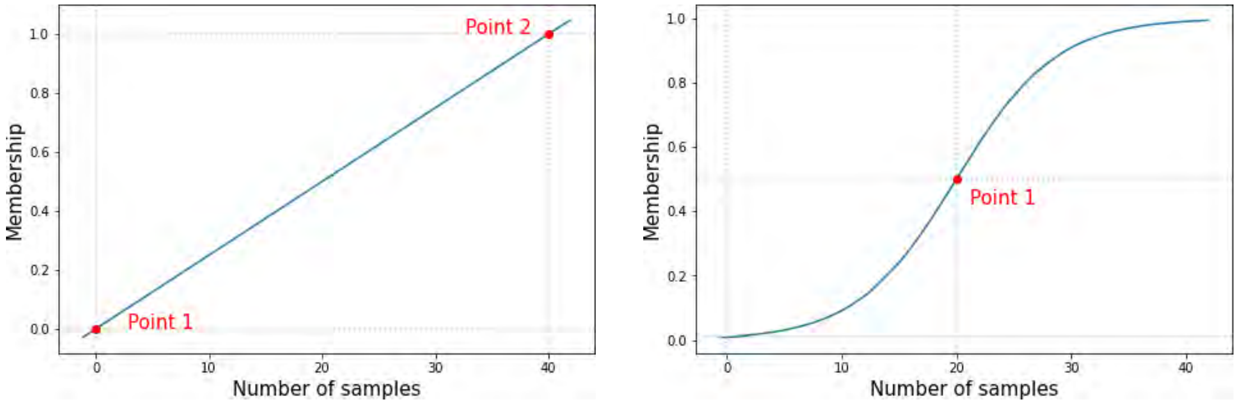
Another candidate for the mapping from samples to a membership is the *logistic function* from the class of *sigmoid functions*. It is computationally more expensive to calculate due to the exponential function, but has different properties than the simpler linear function. It is defined as:

$$\sigma_u(x) := \frac{1}{1 + e^{-k_u \cdot t_u(x)}} \tag{3.9}$$

The number of samples for a class u is represented by x. The design parameter k allows to adjust the steepness of the curve. The function $t_u(x)$ is a transfer function that is responsible for shifting the turning point of the logistic function. It is defined as

$$t_u(x) := x - s_u. \tag{3.10}$$

The number of samples s_u that describes the location of the turning point of the logistic function can be derived from the data. For this purpose, the number of samples contained for a class u is recorded for all voxels in the data set and a quantile such as the median is calculated. The resulting number of samples is used as s_u.



(a) (b)

Figure 3.8: Transfer function of linear and sigmoid membership function. a) The linear function directly maps samples to membership, b) the logistic function maps to extreme values.

$$\sigma_{u_{norm}}(x) := \max\left(0, \min\left(\frac{\sigma_u(x) - \sigma_u(0)}{\sigma_u(2 \cdot s_u) - \sigma_u(0)}, 1\right)\right), \tag{3.11}$$

The result of the membership function for each element of the fuzzy set is normalized linearly to map into the interval $[0, 1]$ using Equation 3.11, where the starting point is chosen at $x = 0$ and the end point at $x = 2 \cdot s_u$.

Occupancy membership function

This membership function is a variation of the logistic function. During the creating of an occupancy grid, an effect occurs which negatively influences the evidence distribution within. This has been described in detail in Gehrung et al. [2017] and arises when measuring rays cross an occupied voxel during ray casting. One method to compensate for this is to decrease the weight of the evidence for free space whenever there is high evidence for occupied space. This approach makes it possible to reduce the effect without having negative effects on the evidences within areas that contain only free space. The transfer function for occupied space is identical to that in Equation 3.10:

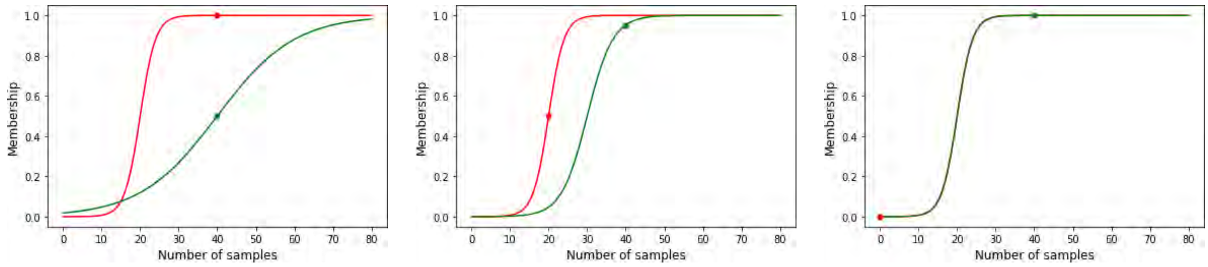
$$t_{occ}(x) := x - s_{occ} \tag{3.12}$$

The amount of evidence for free space depends on the amount of evidence for occupied space. The more pronounced the latter, the further the function value of the transfer function for free space is shifted in the direction of smaller values. Colloquially it can be said that this shifts the turning point of the logistic function to the right:

$$t_{free}(x) := x - \left(s_{free} + s_{free} \cdot \sigma_{occ}(x)\right) \tag{3.13}$$

In addition, the slope is reduced, graded linearly between the slope for occupied space k_{occ} and a minimum value k_{min}:

$$k_{free} := k_{occ} - \sigma_{occ}(x) \cdot (k_{occ} - k_{min}) \tag{3.14}$$



(a) (b) (c)

Figure 3.9: Visualization of the free space function of the occupancy membership function. a) High evidence for occupied space, b) medium evidence for occupied space, c) low evidence for occupied space. The evidence for free space (green point) is always the same, but the evidence for occupied space (red point) is varied.

The previously introduced rule regarding keeping membership functions independent is bend, but not broken. It is still possible to reach a degree of maximum ignorance, but this is now bound to certain conditions. Examples of the above-mentioned membership function for different evidences can be found in Figure 3.9. As before with the logistic function, the membership values for both free and occupied space need to be normalized as described in Equation 3.11.

Computation mode for parameter estimation

A membership function has at least one free parameter which, for the sake of simplicity, is estimated from the data. In the present case it is the median over the number of samples per voxel. Several approaches are conceivable as to how the parameters can be estimated. As part of a *global* approach, the parameter is estimated across all voxels of the representation. With the *tile-based* approach, only the voxels within the same tile are used. In another approach, a *sliding window* is used to take into account all voxels in the close vicinity for the parameter estimation. It works based on a grid with the same resolution as the voxel under consideration. The parameters for each cell of said grid are estimated. The choice of mode is decided during the evaluation.

3.2.7 Defuzzification

The process of sharpening a fuzzy information is called *defuzzification*. This is required for the purpose of interpretation, visualization and other tasks. The approach to defuzzification can be freely chosen. In the context of this work, however, three types of defuzzification have prevailed.

Threshold filter

For *threshold filtering*, a threshold is applied to the collected evidences in order to sharpen the fuzzy representation. Whenever the certainty $m(A)$ for the information exceeds the threshold $t_{certainty}$, the corresponding voxel is considered to belong to case A, that is to support the represented information:

$$\boldsymbol{\sigma}_\alpha := \{\sigma_\alpha | m(A) \geq t_{certainty}\} \tag{3.15}$$

The disadvantage of this method is that the quantity of the evidence is considered, not the ratio in between. This procedure will therefore not produce reliable results in location for which only little evidence has been collected.

Pro/contra filter

The *pro/contra filter* selects all those voxels for which there is more supporting than contradicting evidence. This is defined as follows:

$$\boldsymbol{\sigma}_\alpha \coloneqq \{\sigma_\alpha | m(A) > m(\bar{A})\} \tag{3.16}$$

In contrast to the threshold-based approach, this has the advantage that a more differentiated decision is made. The decision is also made based on the ratio between the supporting and contradicting evidence mass, not just based on the amount of collected evidence mass.

Ignorance filter

The *ignorance filter* is used to sort out those voxels for which the representation, to put it in layman's terms, has too little knowledge about. The resulting representation is referred to here as $\boldsymbol{\sigma}_\alpha$. This notation is motivated by the α-cut, which is similar to the operation executed here. The filter is defined as:

$$\boldsymbol{\sigma}_\alpha \coloneqq \{\sigma_\alpha | m(A, \bar{A}) \leq t_{ignorance}\} \tag{3.17}$$

The threshold $t_{ignorance}$ indicates the maximum allowed level of ignorance. A low threshold value means that there is a low degree of ignorance, whereas a high threshold value also allows a high degree of ignorance. Again in layman's terms, the former corresponds to a low level of missing knowledge, the latter to a high one.

3.3 Spatial predicates

In modern predicate logic, a *predicate* is the part of an atomic statement that is truth-functional. It is basically a function that maps one or more arguments to a Boolean value. As described in Section 2.5, a predicate x is $\tilde{A}$ maps an argument onto a membership function. The term *predicate* is preferred to the term *linguistic variable*, since it does not imply any connection with language. One of the key points of this work is to expand the fuzzy logic for reasoning in spatial information. For this reason, a *spatial predicate* denotes a function which maps spatial information, i.e. a three-dimensional argument, to a fuzzy truth value that is associated with some point in space. Spatial predicates are the building blocks that are combined using fuzzy logic in order to solve a problem within a spatial context. In the following, all spatial predicates that play a significant role in this work are explained in detail.

3.3.1 Fuzzy occupancy grids

A fuzzy evidence grid can represent arbitrary spatial information. However, much of the spatial representations important for this work can be derived from occupancy information. Changes can be derived, for example, from differences in the occupancy of two epochs. When referring to an occupancy representation, this is understood to be a fuzzy evidence grid, which describes the occupancy of space by means of a two-element evidence distribution. One element represents the evidence collected for *free space*, the other for *occupied space*. Contradictions are possible, i.e. that a position in space contains evidence of both the free and occupied case. The generation of an occupancy grid is a time-consuming process, since the way of each measurement ray has to be traced. This is particularly resource-intensive for high-resolution representations. In the following two methods for constructing a fuzzy occupancy grid are presented and their advantages and disadvantages are discussed.

Generation by means of raycasting

A frequently used approach for creating occupancy grids is *raycasting*. This means that in addition to the surface points measured by the sensor, the free space traversed by each measurement ray is also determined. This is usually done as proposed by Amanatides & Woo [1987]. Here, the ray, that is the path between sensor position and the corresponding surface point, is mapped onto a virtual grid. For this, first the address of the start cell is derived from the sensor position. Then, based on the direction of the ray, the addresses of all cells intersected by the ray are determined incrementally. The final cell is designated to contain the measured surface point, all other cells are assumed to correspond to free space, since they have been traversed by the measurement.

This process is repeated for each measurement. The result consists of two lists, one including all occupied cells, the other all free cells. Each cell can appear several times in the list, therefore each entry contains the cell address and a counter. Based on these lists, the occupancy grid is updated. Updating the occupancy statistics within a cell can be done in different ways, depending on the approach. Procedures that work with dynamic environments, such as the one proposed by Hornung et al. [2013], require the exact time sequence of the measurements. For procedures that compare static environments, as it is the case in this work, the number of traversing and ending rays per cells is sufficient information. The affiliation of a cell to the classes *occupied* and *free*, represented by the fuzzy elements u_{occ} and u_{free}, is defined as explained in Equation 3.9.

The lower resolution levels are built up from the highest resolution level. This has the advantage that the computationally expensive raycasting operation only has to be carried out once. The resolution pyramid is built up by manipulating the address for each cell in the list using a bit shift operation so that the address of the higher-level voxel can be read from it. By evaluating the two lists manipulated in this way, the number of samples for free and occupied space can now be derived for each affected voxel. This is repeated for each level of resolution until the root node of the octree is reached. The runtime for this procedure is negligible in comparison to performing another raycasting-step.

Since the process of generation is, as mentioned, very consuming in terms of computational resources, parallelization is required. Two approaches were examined as part of this work. The first is the parallelization at tile level, where each tile is assigned to a task that processes the entire measurements of said tile. The second option involves the distribution of all measurements within a single 3D tile to multiple tasks. The results of all tasks are combined by concatenating the partial lists created by each task. In practice, the second approach has proven to be more efficient despite the additional overhead, not least because it enables individual tiles to be processed quickly for test purposes.

Generation by means of iterative refinement

The disadvantage of raycasting is that all areas are resolved with the same resolution. However, this is not always necessary, since free space in particular represents a homogeneous area that can easily be summarized by several large voxels without a loss of information. Only the area on the border between free space and occupied space needs to be represented with a high resolution, as it is very heterogeneous. The most suitable data structure for the representation is an octree.

The method of *iterative refinement* was developed to generate an occupancy representation with high resolution in heterogeneous areas and low resolution in homogeneous areas [Gehrung et al., 2018]. As part of the method, a voxel is refined until the evidence it contains is either homogeneous or a maximum depth has been reached. An overview of the method is given in Algorithm 2. The procedure starts with an octree, of which only the root voxel exists. Said voxel is refined in an iterative manner. First, all rays assigned to a voxel are used to update its evidence

```
Data: Set of rays $r$, maximum octree depth $depth_{max}$.
Result: Octree $node_o$ with occupancy information.

Function refine($node_o$, $r$):
    updateOccupancyStatistics($node_o$, $r$)
    if hasContradictions($node_o$) and getDepth($node_o$) $<$ $depth_{max}$ then
        for $child \leftarrow 0$ to 7 do
            $node_{child} \leftarrow$ createChild($node_o$, $child$)

            // Find all rays intersecting the child node.
            $r_{child} \leftarrow$ determineRaySubsetByIntersection($node_{child}$, $r$)

            // Update the child node.
            refine($node_{child}$, $r_{child}$)
```

Algorithm 2: An iterative refinement based approach to derive an occupancy octree $node_o$ from a set of measurement rays r.

distribution using Equation 3.9. If there is a contradiction, i.e. if there is evidence for both *free* and *occupied*, the node is further refined and the measurements are distributed to each of the eight possible child voxels. For this purpose, an intersection test between each measurement ray and the voxel under consideration is carried out using the method proposed by Kay & Kajiya [1986]. The process described here is then repeated for each of the child voxels. This process automatically leads to a multi-resolution pyramid, but can only be parallelized at tile level. A slightly modified variation of the procedure allows to introduce free space with a lower resolution than the one used for the occupied space. This reduces computational costs, however, it also reduces the quality of the representation.

Comparison of both approaches

The main advantage of raycasting is that it generates very accurate results. Iterative refinement, on the other hand, leads to representations that consume less memory and usually also require fewer computational resources, especially if the free space is entered with a lower resolution. The number of voxels required is related to the complexity of the scene, not to the resolution implied by the scene, as is the case with raycasting.

Nevertheless, raycasting is still preferable to iterative refinement. As a result of the latter one, a voxel may not be further refined, although there is not evidence for *all* child voxels. Actually unobserved space is marked as either free or occupied space, whereby the former usually applies. If two epochs created in this way are compared in the context of change detection, phantom changes occur in places that were actually unobserved in one of the two epochs. The extent of these false positives depends on the voxel size and is usually not negligible.

The bottom line is that iterative refinement should be avoided when the comparison of two epochs is necessary. It is therefore not used in the further course of this work. However, it is mentioned here for the sake of completeness and to answer possible questions regarding occupancy representations with varying voxel size.

3.3.2 Object classes

The object class labels of the TUM-MLS data set described in Section 6.3.4 are also considered to be spatial predicates, since they can be used to describe the affiliation of a spatial area to an object class. This is useful additional information in order to evaluate the results of the change detection in more detail. To do this, the labels are converted into a fuzzy representation using the following method.

Each point in the labeled point cloud of the TUM-MLS data set has a number assigned to it. By applying a division and modulo operation, both the object class identifier as well as the object instance identifier can be derived from this number. However, the latter information is not used in this work. Since the *building* class is divided into both the *facade* and *interior* sub classes, a case differentiation for both subclasses and all classes with higher identifiers is required.

For each point in the TUM-MLS point cloud, the tile and then voxel in the representation for the corresponding object class is determined. As with the creation of the fuzzy occupancy grid, there are sample counters from which the evidence distribution is derived. For each sample, the counter for A in the representation of the corresponding object class is incremented by one. In addition, the same is done in the representations of all other object classes, but here the counter for $\bar{A}$ is incremented. In this way, an effective distinction is made as to which areas do belong to the object class, which do not and which are not covered by the data set. After all points are processed, the logistic membership function explained in Section 3.2.6 is applied to each voxel of all object class representations.

3.3.3 Compensation of registration errors

Although the data sets used here have been registered with one another as described in Section 6.3.2, a residual error remains. In some places this is larger than the selected voxel resolution of 0.1 m. One way to counteract this error in the context of the occupancy representation is to blur the latter. In the following, two methods are described which exercise spatial blurring based on the local neighborhood.

Max-pooling

This technique is inspired by *max-pooling* in *convolutional neural networks* [Scherer et al., 2010]. In max-pooling, the maximum value within a local neighborhood is determined. This is realized by a three-dimensional filter kernel that is slided over each voxel in the representation. The kernel size is always an odd number, since it must include the voxel under consideration and its N next neighbors in each direction. For each voxel covered by the kernel, the maximum values for both A and $\bar{A}$ are selected and assigned to the voxel in the center.

Low pass filtering

In low pass filtering, the representation is convolved with a *symmetric Gaussian function*, reducing high frequency components. Said approach is often used in computer vision [Forsyth & Ponce, 2012]. In the present case, the application of the filter has the effect of spatially blurring the fuzzy information. The kernel used for convolution is calculated with the following function:

$$G(x, y, z) = \frac{1}{(2\pi)^{\frac{1}{8}}\sigma^3} \exp(-\frac{x^2 + y^2 + z^2}{2\sigma^2}) \tag{3.18}$$

An illustration can be seen in Figure 3.10. The value for σ is given in meters, the function is then sampled based on the selected resolution level to calculate the values for the respective voxels. It is necessary to normalize the filter kernel to one, otherwise the representation will be subject to an effect that is similar to an image being brightened or darkened. The kernel is applied to each voxel and sets the fuzzy information to the weighted average of the voxel's neighbors. The original voxel's value receives the highest weight, the neighboring voxels get a smaller weight with increasing distance. From a statistical point of view it is recommended to set the range of the filter to 3σ in each direction. However, this can be very computationally expensive, so applying it to the nearest N neighbors in each direction is a necessary compromise.

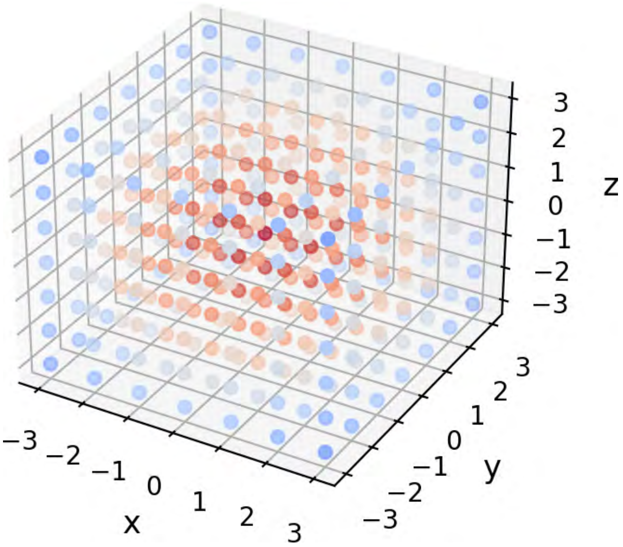


Figure 3.10: Visualization of a 3D Gaussian kernel. The Gaussian bell with $\sigma = 0.15$ is discretized in order to be used with a regular grid.

Comparison of both approaches

Both filters blur the representation, but with a different result. When the Gaussian filter is applied, the resulting representation only shows softened, vaguely defined edges and corners. The max-pooling on the other hand produces an inflated version of the occupancy grid, but with edges and corners as crisp as in the occupancy representation. While the latter operation only spatially blurs the representation, the former also seems to add noise due to the weight of the Gaussian function, therefore blurring the evidence distribution itself. This leads to a visible loss of quality in the changes derived from the blurred occupancy representation. Therefore, no further investigation of the Gaussian filter was performed. It is mentioned here only to answer the obvious question about using a Gaussian filter.

4 Change detection

This chapter presents a method to detect changes in urban environments utilizing *fuzzy spatial reasoning*. Figure 4.1 gives an overview over the process and its individual steps. The first step is the distribution of the 3D measurements to 3D tiles, which is described in detail in Section 3.1.3. This step comprises the distribution of the measurements to the individual tiles. The second process step involves the preparation of the fuzzy representations. This includes the construction of the fuzzy occupancy representation as described in Section 3.3.1. The max-pooling representations described in Section 3.3.3 are also generated here. The third process step involves the actual *change detection*. This includes the use of several fuzzy Boolean operations. These operations are formulated using the convention defined in Section 3.2.4 and are explained in detail below. The fourth step of the process includes the *defuzzification* of the results utilizing the means proposed in Section 3.2.7.

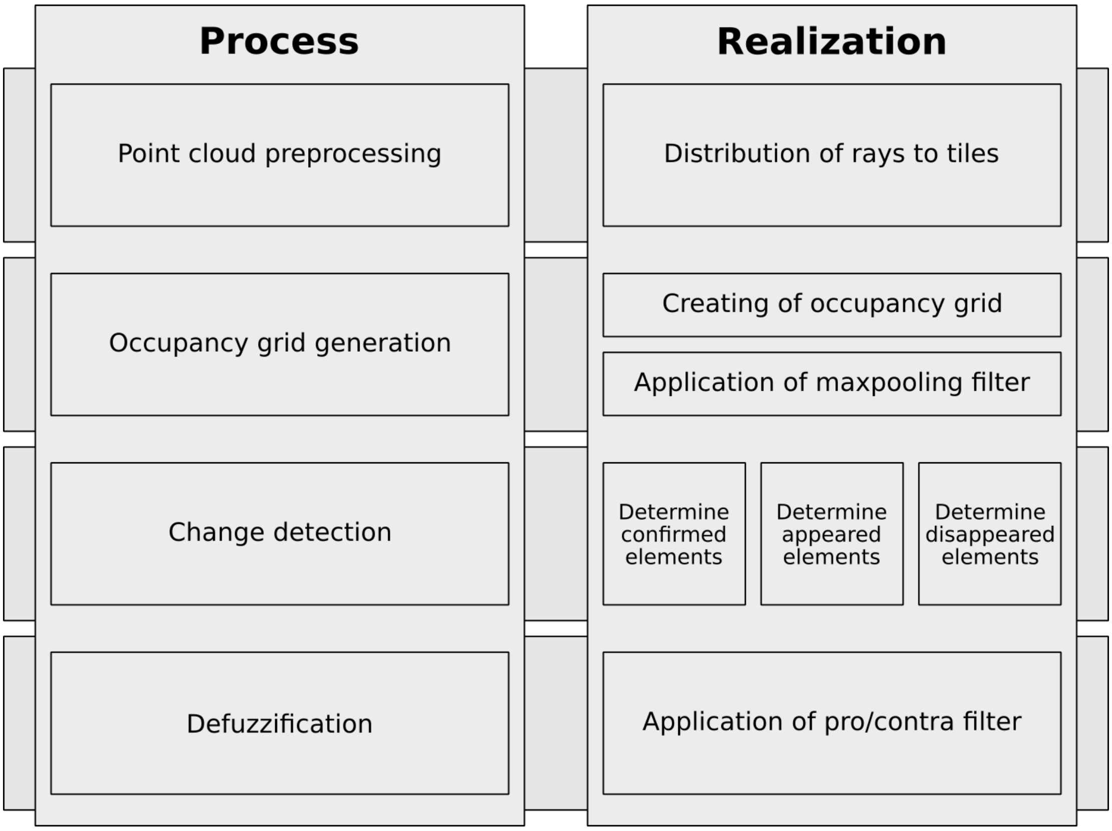

Process
Point cloud preprocessing
Occupancy grid generation
Change detection
Defuzzification
Realization
Distribution of rays to tiles
Creating of occupancy grid
Application of maxpooling filter
Determine confirmed elements
Determine appeared elements
Determine disappeared elements
Application of pro/contra filter

Figure 4.1: Overview over the change detection process. The individual process steps are shown on the left, their implementation on the right.

4.1 Representation of changes

There are five cases to be handled in change detection. The three main cases include the detection of *confirmed*, *appeared* and *disappeared* elements. These are the actual changes which every change detection algorithm should supply. In addition, the procedure must be able to identify areas that were *not observed in either of the two epochs* being compared. These are the two remaining cases. This should be done either explicitly or implicitly. If the method is not able to recognize these two cases, phantom changes occur which at best are actual changes that cannot be confirmed as such or, at worst, are false positives.

Changes can be represented in two ways. First, it is possible to use a single data structure to encode all five of the above cases. Such a structure has been used in a previous own work in which it was referred to as a *delta-octree* [Gehrung et al., 2019a]. Second, a separate data structure can be used for each of the above cases (e.g. one for disappeared elements, one for confirmed elements, etc.). In the context of this work, the latter was chosen because because this approach fits seamlessly into the theoretical framework described above.

4.2 Determining confirmed elements

An element is considered confirmed if it occurs in both of the compared epochs. From a logical point of view, this corresponds to applying an *and*-operation. However, executing this operation on the occupancy representation would lead to various false negatives. This is due to residual errors caused by poor registration of the MLS data or calibration errors of the MLS system. These can be compensated to a certain extent by utilizing a max-pooled version of the occupancy representation of each epoch. The resulting loss of detail is kept within limits if the neighborhood of the smoothing operation is kept low. The *and*-operation is applied between the epoch to be confirmed and the max-pooled version of the other epoch. The following equation demonstrates this for the first epoch:

$$\boldsymbol{\sigma}_{confirmed_1} \coloneqq f_\wedge(\boldsymbol{\sigma}_{occupancy_1}, \boldsymbol{\sigma}_{max-pooled_2}) \tag{4.1}$$

The second epoch is calculated using the following equation:

$$\boldsymbol{\sigma}_{confirmed_2} \coloneqq f_\wedge(\boldsymbol{\sigma}_{max-pooled_1}, \boldsymbol{\sigma}_{occupancy_2}) \tag{4.2}$$

This results into two result representations for the *confirmed* case. However, this is not a problem, but must be taken into account in the evaluation.

4.3 Determining appeared and disappeared elements

The Boolean operation used to determine all changes is the *xor*-operation. Here, however, no distinction is made between appeared and disappeared elements. For this reason, the two partial operations that make up the *xor* are used, namely the *inhibition*-operations. The case *appeared* is calculated using the Inhibition of $\bar{A}$. This corresponds to an *and*-operation with input negation:

$$\boldsymbol{\sigma}_{appeared} \coloneqq f_\wedge(\overline{\boldsymbol{\sigma}_{max-pooled_1}}, \boldsymbol{\sigma}_{occupancy_2}) \tag{4.3}$$

The procedure for the *disappeared* case is similar. It is computed by the Inhibition of A, which is also realized by an *and*-operation, but this time the other input is negated:

$$\boldsymbol{\sigma}_{disappeared} \coloneqq f_{\wedge}(\boldsymbol{\sigma}_{occupancy_1}, \overline{\boldsymbol{\sigma}_{max-pooled_2}}) \tag{4.4}$$

The use of max-pooled representations is necessary to avoid false positives. As in the case of the confirmed elements, incorrect conclusions would arise due to the aforementioned residual errors in the underlying data.

4.4 Determining elements unseen in the other epoch

As described above, an important point in change detection is dealing with areas that were only observed in one of the two epochs. This is a two-step process. The first step is to localize where the aforementioned case is present. As described in Section 3.2.4, this is done during the logical inference, more precisely when merging two octrees into one with a common structure. An area is not observed in the other epoch if a voxel exists in one octree, but not in the other.

The second step is to handle the aforementioned case. During the logical inference, whenever a voxel doesn't exist in the other epoch, a placeholder voxel without evidence is created. It is in the nature of the fuzzy reasoning that none of the logical conjunctions used for change detection will result in a false positive if one of the evidence distributions involved does not contain any evidence. Or in other words, a logical conjunction of evidence and complete ignorance does not lead to a result that suggests an information that is not present.

4.5 Filtering by object classes and instances

In order to make the changes for individual object classes and instances investigatable, these can be linked with one another. For example, all changes to vehicles can then be identified. In order to filter changes for objects of a certain class, these must be linked with an *and*-operation:

$$\boldsymbol{\sigma}_{confirmed_i} \coloneqq f_{\wedge}(\boldsymbol{\sigma}_{confirmed}, \boldsymbol{\sigma}_{class_i}) \tag{4.5}$$

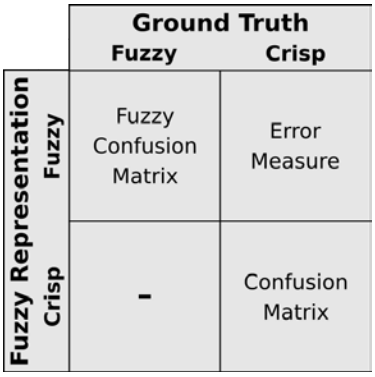
A comparable procedure is necessary to filter individual instances:

$$\boldsymbol{\sigma}_{confirmed_{ij}} \coloneqq f_{\wedge}(\boldsymbol{\sigma}_{confirmed}, \boldsymbol{\sigma}_{instance_{ij}}) \tag{4.6}$$

The equations shown above are exemplary. In principle, said information can also be included in another way, for example by first linking different object classes by applying an *or*-operation and then extracting the changes via an *and*-operation.

5 Fuzzy evaluation metrics

The *fuzzy spatial reasoning* approach presented in this thesis represents a novel form of spatial representation. Metrics are required to answer the question of the degree to which information can be reproduced correctly. For this purpose, two metrics for quantitative evaluation of fuzzy representations have been developed.



		Ground Truth	
		Fuzzy	Crisp
Fuzzy Representation	Fuzzy	Fuzzy Confusion Matrix	Error Measure
	Crisp	-	Confusion Matrix

Figure 5.1: Overview of the evaluation metrics. Comparing fuzzy representation and ground truth requires different metrics depending on whether the former has been defuzzified or the latter has been inserted into a fuzzy representation. The comparison of crisp representation and fuzzy ground truth does not play a role for the present case.

The *error measure* described in Section 5.1 is suited for comparing a fuzzy representation with a crisp ground truth such as an annotated point cloud. While the *confusion matrix* can be used to compare a defuzzified representation with a crisp ground truth, the *fuzzy confusion matrix* presented in Section 5.2 is used to compare a fuzzy representation with a fuzzified ground truth. The latter is obtained by importing the ground truth into a fuzzy representation. Figure 5.1 illustrates in which case which metric is to be used.

5.1 Error measure

The purpose of the error measure is to determine the difference between a fuzzy representation and a ground truth represented by an annotated point cloud. In other words, the error measure determines the deviation between a *set of labels* and the *evidences represented by the voxels.* The latter are represented by *certainties* so that contradictions in the evidence can be taken into account. A fuzzy representation summarizes the evidences for and against a single class. However, the labels of the ground truth usually contain more than two classes. With regard to the error measure, this is handled by applying a *binarization to the ground truth labels.* All labels that correspond to the class of the fuzzy representation are assigned to one class A, all other labels to another class $\bar{A}$. Since the error measure is based on the theoretical framework described in Chapter 4, the same formalization was chosen. What is referred to here as a *class* corresponds to an *element of the fuzzy set.* Since the ground truth was organized this way, the error measure only has to determine the difference between a binary representation and a binary ground truth. It is defined as:

$$e = \frac{1}{N} \sum_{i=1}^{N} |L_i - m(A_i)| + |1 - L_i - m(\bar{A}_i)| \tag{5.1}$$

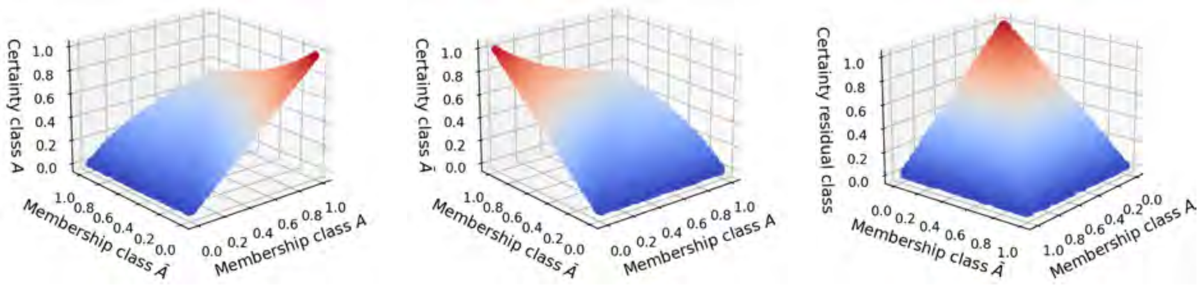
The more fuzzy representation and ground truth match, the smaller the error. N is the number of all labeled points in the ground truth, L_i is a numerical value that represents the affiliation of the label i to the class under consideration. The value is 1 if the label equals said class or 0 if it is a label of another class. $m(A_i)$ and $m(\bar{A}_i)$ represent the certainties associated with the voxel in which the labeled point is located. The former summarizes the certainty for the class under consideration, the latter against it. For the reasons explained below, the interval covered by the error measure is $[0, 2]$.

For a better understanding of the error measure and the associated possibilities, its individual components are explained here. The explanations are supported by 3D plots of the parameter space. These are generated by random sampling of the examined parts of the error measure. The basic idea of the error measure is that the numerical distance between the label and the evidence associated with the class is calculated. However, instead of using the evidence directly, the derived *certainties* are used, since this allows contradictions to be handled better. The error measure is structured as follows. For each labeled point, the corresponding voxel in the representation is determined and a distance function between the certainties stored within the voxel and the label is calculated. Since a voxel stores evidence for and against an information, the distance function consists of two parts. A numeric value of either 1 or 0 is assigned to the label, depending on whether it stands for or against the class represented by the fuzzy representation. Adding the individual errors for each labeled point results in the total error.

5.1.1 Relationship between membership and certainty

As explained in Section 2.7, the certainty is derived from the evidence distribution using Equation 2.18. One may assume that the relationship between both evidence and certainty is linear. However, this is not the case. Figure 5.2a shows that the relationship is non-linear. The surface plot can be divided into two areas, with the diagonal representing the dividing line. It represents the part of the function where there is exactly the same amount of evidence for both classes. If the greatest possible contradiction is present, i.e. both classes take a value of 1, then the maximum certainty for class A is 0.5. The area below the diagonal includes all points at which the evidence for class $\bar{A}$ is greater than that for class A. Accordingly, the certainty for class A grows very slowly and, as mentioned, is capped at a value of 0.5. Above the diagonal, the evidence for class A predominates, and the certainty increases rapidly, up to the maximum value of 0.1. The plots for class A and $\bar{A}$ are mirrored, but otherwise have the same shape (cf. Figure 5.2b).

The non-linear relationship between evidence and certainty can be explained by the residual class of the fuzzy measure. As a reminder, a fuzzy measure is used to describe the certainty associated with the elements of a fuzzy set, i.e. the elements of the evidence distribution. The evidence distribution is defined over the universe $\mathcal{U}$, while the fuzzy measure is defined over the power set of said universe. This means that instead of two elements of interest, there are three elements through which the evidence must be distributed. Any certainty mass that is not allocated to either class A or $\bar{A}$ is instead allocated to the third element, which reflects the state of ignorance. As illustrated in Figure 5.2c, the mass allocation for ignorance is highest when there is no evidence at all and lowest when there is full evidence for at least one of the two classes.



(a) (b) (c)

Figure 5.2: Relationship between membership and certainty. a) Relationship between the membership to both classes and the certainty for class A, b) the same for class $\bar{A}$, c) the certainty of the ignorance class.

5.1.2 Discussion of the error terms

Based on this information, the error measure and its individual elements are explained in more detail. The central element of the function is the distance between the label and the certainty of a class. Since the label either represents a certain class or its counter-class, it can only take the value 0 or 1. Figure 5.3 therefore illustrates the different terms of the error measure for both variants. The error terms for both classes A and $\bar{A}$ shows properties comparable to those described in the previous section, since certainty plays a dominant role in the error term. Compare Figures 5.3a, 5.3b and 5.3c, 5.3d for an illustration. The term is designed in a way that the error is at its maximum when the deviation between label and evidence is at its maximum.

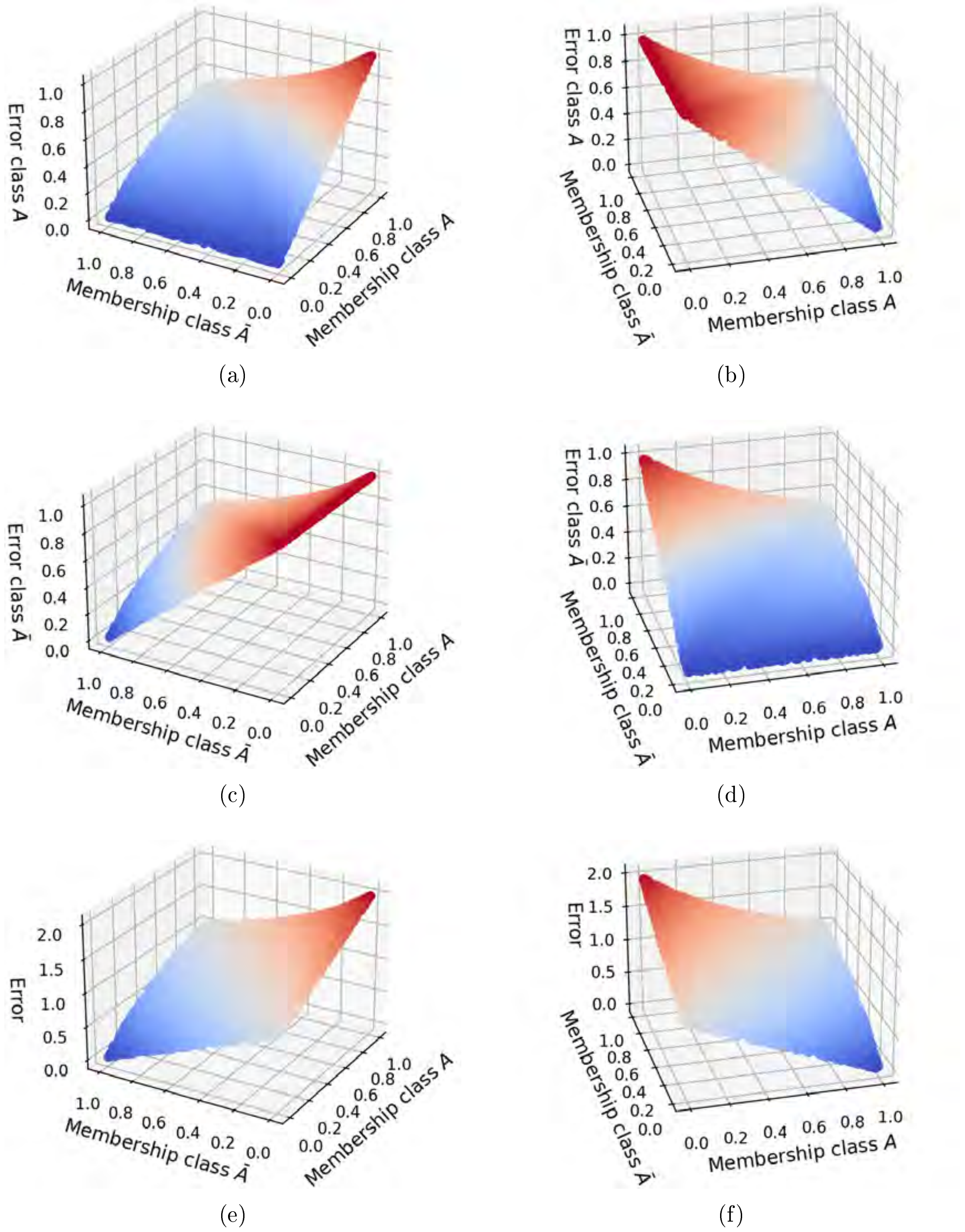
5.1.3 Discussion of the total error

The total error is the sum of the two error terms for class A and $\bar{A}$. Figures 5.3e and 5.3f show the total error for both label values of 0 and 1. As before, the diagram shows two different regions that need to be discussed separately. The lower part represents the case when label and class A are equal to a certain degree and class $\bar{A}$ supports this by having a value that is opposite to class A. If both classes are in such a way, the error is zero. At the transition between the two areas, the error is 1, because here either class A supports the label and class $\bar{A}$ is not properly opposed to class A or class A contradicts the label and class $\bar{A}$ is properly opposed to class A. Either way, either class A or class $\bar{A}$ is wrong, while the other is right. The error measure reaches its maximum in the remaining area. Here, both classes A and $\bar{A}$ contradict the label to a certain degree. At the most extreme point, the error measure reaches its maximum value of 2.

5.2 Fuzzy confusion matrix

The error measure is intended for the comparison between crisp labels and the fuzzy representation, as is required by the first research question. When comparing the results of the change detection with the ground truth, however, discretization errors should be avoided, especially in the case of coarser resolution levels. These errors occur when the resolution of the ground truth is higher than the resolution of the fuzzy representation to evaluate. For this reason, a fuzzy form of the classical confusion matrix was developed. For this purpose, the ground truth itself is converted into a fuzzy representation. The comparison then takes place as explained below.

The basic idea is that the individual cases *true positive, true negative, false positive, and false negative* are assigned proportionally for each voxel. This is best illustrated by Figure 5.4. True positives and true negatives correspond to the overlap of the estimate P with the ground truth G. For the former, it is the overlap between A_P and A_G, for the latter, it is the overlap between



(a) (b)

(c) (d)

(e) (f)

Figure 5.3: Components of the error measure. The left column shows label values for 0, the right column label values for 1. a-b) Deviation between label and certainty for class A, c-d) the same for class $\bar{A}$, e-f) the overall error for both classes.

$\bar{A}_P$ and $\bar{A}_G$. A *false positive* occurs whenever the A_P of the representation overestimates the A_G of the ground truth. A *false negative* occurs whenever the $\bar{A}_P$ of the representation overestimates the $\bar{A}_G$ of the ground truth. The following formulas reflect this:

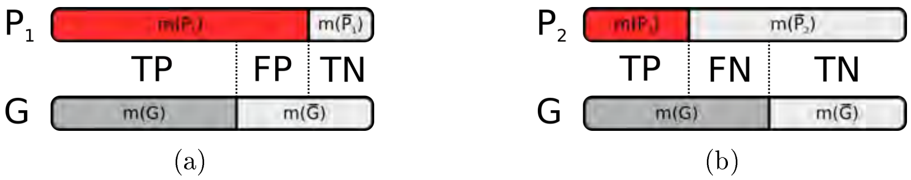


Figure 5.4: Stacked bar charts illustrating the concept of the fuzzy confusion matrix. (a) Example of a false positive caused by an overestimation of A. (b) Example of a false negative caused by an underestimation of A. True positives and true negatives correspond to the minimum overlap of the estimates with the ground truth.

$$\begin{aligned} TP &= \min\big(m(A_P), m(A_G)\big) \\ FP &= \max\big(0, m(\bar{A}_G) - (1 - m(A_P)\big) \\ FN &= \max\big(0, m(A_G) - (1 - m(\bar{A}_P)\big) \\ TN &= \min\big(m(\bar{A}_P), m(\bar{A}_G)\big) \end{aligned} \tag{5.2}$$

As already mentioned, A_P and $\bar{A}_P$ denote the evidences of the fuzzy representation, A_G and $\bar{A}_G$ those of the fuzzified ground truth. For each voxel, all four cases are computed and then normalized so that their sum equals one. When comparing two distributions of evidence that represent complete ignorance, this restriction does not apply since all four variables are zero.

6 Experiments

In this chapter, the experiment designs as well as the experiments are presented. Additionally, the data set acquired for the purpose of evaluation is presented, including the measurement system used, pre-processing steps such as intrinsic and extrinsic sensor calibration, intra-epoch and inter-epoch registration, and fine registration. Furthermore, ground truths for change detection and semantic classes built on the data set are presented.

6.1 Experiment design

In order to determine the performance of the methods presented in Chapters 3 and 4, a series of experiments was conducted. Among other things, the results of the experiments also allow conclusions to be drawn about the error measures presented in Chapter 5. The experiments can be divided into two major groups.

The first group of experiments deals with the verification of the fuzzy spatial representation and can be divided into two subgroups. The first subgroup deals with determining open parameters, which includes the choice of the membership function. The second subgroup deals with the determination of the properties of the fuzzy representation before and after defuzzification. This is performed by importing a known structure in form of an annotated point cloud into a fuzzy representation and then comparing it to said point cloud. The ground truth for the semantic object classes from the first epoch of the TUM-MLS data set is used for that point cloud (cf. Section 6.3.4). The comparison between fuzzy representation and crisp ground truth is performed using the error measure presented in Section 5.1.

The second group of experiments tests the ability of the fuzzy spatial reasoning to detect changes on a real-world data set that includes an urban street space. For change detection, the process shown in Figure 4.1 is performed. The data set is organized as explained in Section 3.1 and a fuzzy occupancy grid is created for each of the two epochs by the means described in Section 3.3.1. Open parameters are determined, which includes the choice of membership function for fuzzy occupancy grids, its computation mode, the max-pooling neighborhood and the defuzzification method. Subsequently, the properties of the change detection process are determined. Both epochs of the TUM-MLS data set are used for testing. The results are evaluated using the ground truth presented in Section 6.3.4. The comparison between the defuzzified change detection results and the crisp ground truth is done using a *classical confusion matrix*. The comparison between the fuzzy results and the fuzzified ground truth is done using the *fuzzy confusion matrix* presented in Section 5.2.

The experiments and their evaluation also allow conclusions to be drawn about the properties of the metrics used. For this reason, no additional experiments are necessary. Both the experiments and the methodology under investigation were implemented using C++. The system that the experiments were performed on had an Intel i9 CPU with 36 cores a 2.60 GHz and 64 GB of

RAM. The implementation of some components was based on the point cloud library PCL [Rusu & Cousins, 2011].

6.2 Summary of experiments

Two groups of experiments were developed. The first group deals with the verification of the fuzzy representation. The second group investigates the suitability of fuzzy spatial reasoning for change detection in urban street spaces.

6.2.1 Assessment of the spatial representation

The purpose of the series of experiments described in this section is to determine the properties of the fuzzy representation presented in Chapter 3 and to verify its ability to reliably reproduce information. First, a counting of all voxels in both epochs was performed, by means of which the average memory consumption for the investigated urban environment is estimated. The memory is calculated by multiplying the number of voxels with the bits required per voxel, where the result is then averaged over all tiles. Per voxel, the required memory equals 73 bit. That is one byte to indicate the presence of child nodes, a single bit for the presence of a payload and two times four byte to encode the evidences in form of two floating-point numbers.

Subsequently, the ground truth of the semantic object classes from the first epoch of the TUM-MLS data set was imported into a fuzzy representation (cf. Section 6.3.4). A separate representation for each object class was generated. The error measure explained in Section 5.1 was calculated for each resolution level of each representation. The average error was calculated by averaging the error measure for each resolution level over all object classes. This was then subsequently used to compare the linear and logistic membership function. The steepness parameter k of the logistic function was set to 1.0, all other free parameters were estimated from the data set as explained in Section 3.2.6. In addition, histograms over the number of samples as well as the evidence after application of the linear and logistic membership function were generated. The data for this was averaged over all voxels.

Based on the error measures of the object classes, it was then tested how well the corresponding information can be reproduced in a fuzzy way. In addition, following the same procedure as above, histograms were generated over the evidence for A and $\bar{A}$ and the error measures. Only voxels that contained one of the ground truth samples were considered. The representations generated from the object classes were then defuzzified using pro/contra filtering (cf. Section 3.2.7) in order to test how well the information is represented when defuzzification is involved. A confusion matrix was used for evaluation, as well as the derived measures precision, recall, and F1-score. In addition, the influence of discretization errors was investigated by examining the *false positives* and *false negatives*. This was also done based on the confusion matrix.

6.2.2 Assessment of fuzzy reasoning and change detection

The experiments described in this section are designed to validate the spatial fuzzy reasoning and test its suitability for change detection in urban urban street spaces. The process shown in Figure 4.1 is executed to generate the required fuzzy representations. Therefore, the data set is organized as explained in Section 3.1 and a fuzzy occupancy grid is created for each of the two epochs by the means described in Section 3.3.1. The steepness parameter k of the occupancy membership function was set to 1.0. Values of 5.0 and 1.0 were selected for the steepness parameters k_{occ} and k_{min} of the logistic membership function. The steepness parameters k of the logistic function was set to 1.0. All other free parameters were estimated from the data set as explained in Section 3.2.6.

Based on this, a max-pooled representation is generated (cf. Section 3.3.3). The determination of confirmed, changed, and unseen elements is performed as described in Section 4.2, 4.3 and 4.4 based on the max-pooled occupancy grids.

The experiments consist of a two-stage process. In the first stage, the still open parameters are clarified. Multiple occupancy grids were generated to determine the best fitting membership function and its computation mode. A *tile-based* computation mode was used for the former. The results are interpreted visually. Based on this, in the second stage multiple instances of the change detection process are executed to determine the size of the max-pooling neighborhood and the defuzzification method. For this, different max-pooling neighborhoods of 0-2 voxels are tested once with, once without pro/contra filtering. These are then evaluated using fuzzy confusion matrices as described in Section 5.2. The *Alte Pinakothek* subset of the two TUM-MLS data sets is used for all tests mentioned above. This was done because the fine structures of the scaffolding are very sensitive to any problems that may arise and the subset also contains many of the elements that are otherwise found in a street space. The ground truth described in Section 6.3.4 is imported into a fuzzy representation and compared to the change detection results using the *fuzzy confusion matrix* proposed in Section 5.2. The metrics *precision, recall and F1-score* are used for the comparison. The *F1-score* is preferred over the *accuracy*, since the former is more robust compared to said asymmetric distribution than the latter, because the comparatively large number of *true negatives* is not taken into account. For the same reason, *precision/recall curves* are to be preferred over *receiver operating characteristic (ROC)* plots, see Saito & Rehmsmeier [2015].

In the second stage, the properties of the change detection process are determined. This is carried out with the parameters that were determined in stage one. The change detection process is applied to the full data set and the result is compared to the fuzzified ground truth using a *fuzzy confusion matrix.* Since no meaningful ground truth could be created for changes in vegetation, it was removed from both the representations and the ground truth using a logical *and-not* operation. Since most of the changes are in the *Alte Pinakothek* subset, this is examined separately. Again, the vegetation was removed from representations and ground truth. For a more in-depth examination, the change detection results obtained on the subset are subdivided by object classes using fuzzy spatial reasoning and examined in more detail. For this, an *and* operation is performed with the ground truth of the semantic object classes.

6.3 Data sets and ground truth

For a better understanding of the data sets created for the evaluation, the mobile mapping system used for the acquisition is presented. Furthermore, it is explained how the quality of the recorded data was ensured. This includes measures for sensor calibration as well as for the registration of LiDAR data. In conclusion, the data sets recorded for the evaluation and the ground truth for change detection created from them are presented.

6.3.1 Measuring system

The data sets used for testing, calibration and evaluation have been recorded using the measurement vehicle *MODISSA* of the *Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB).* The designation *MODISSA* is an acronym and stands for *Mobile Distributed Situation Awareness.* The vehicle is used for hardware evaluation as well as research and software development in the contexts of automotive safety and security applications. An illustration of the vehicle and the build-in sensor and navigation system can be seen in Figure 6.1. The vehicle is also used to demonstrate the developed technologies in-house and how they interact. In the further course, an overview of the system architecture and the installed sensors is given. A comprehensive description of the overall system, the sensors as well as the hardware and software system architecture can be found in Borgmann et al. [2021].

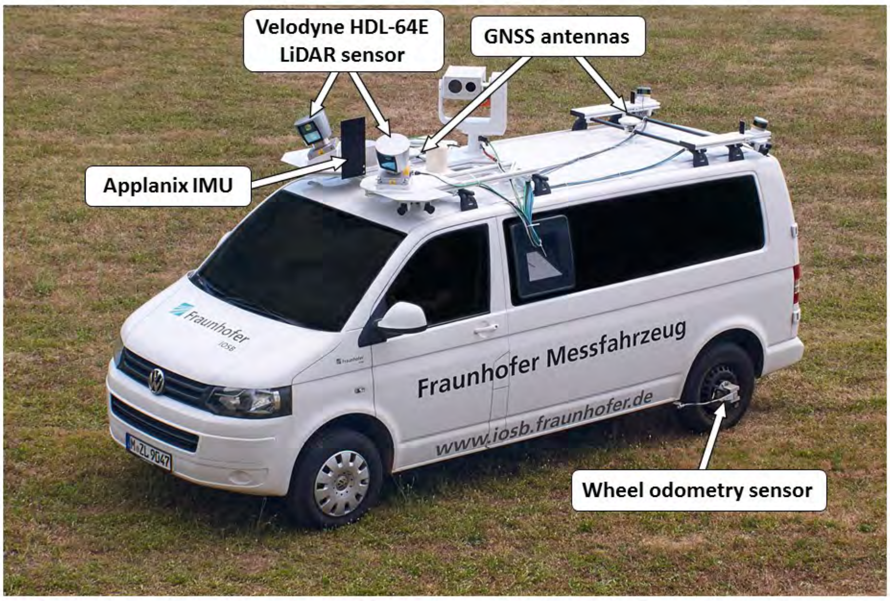


Figure 6.1: The mobile mapping system MODISSA.

System architecture

A transporter from a well-known German car manufacturer serves as the platform for the mobile mapping system. The vehicle has a long wheelbase and an all-wheel drive. Originally equipped with three rows of seats, the middle one was removed to make room for data processing equipment. The vehicle can accommodate up to five people, including a driver and a navigator in the front row as well as technical personnel in the back row. A server rack and monitor are mounted in the passenger area of the vehicle, replacing the middle row of seats. The rack is equipped with several computers and other electronic components used for sensor interaction, control tasks, data recording and on-board data processing. A 10 GBit high-bandwidth Ethernet network connects sensors, computers and other equipment. A WiFi access point allows to integrate mobile devices to view sensor data or to be able to control the measuring system from outside the vehicle. From a software point of view, a large part of the measuring system is based on the robot operating system ROS*. The power supply of the measuring system is completely independent of the electrical system of the vehicle. Behind the rear bench is a box with four large lithium-ion batteries, connected in a way that they provide 24 V/600 Ah. They are housed in a box in the trunk of the vehicle. An inverter provides up to 2000 W at 230 V AC for more than five hours before recharging is required.

Sensor systems

The sensors are mounted on specially designed platforms on the roof of the vehicle. Only the LiDAR sensors are relevant for the present work. However, in order to give a comprehensive overview of the vehicle, the other sensors are also briefly discussed.

*http://wiki.ros.org

At the time the first TUM-MLS data set was recorded in April 2016, two *Velodyne HDL-64E* LiDAR sensors were installed on the roof of the vehicle. Both sensors were tilted by a 25 ° angle to the horizontal by aluminum wedges installed under the base plate of each sensor. Also, each sensor was rotated outwards at a 45 ° angle. This unique sensor setup allowed to record both the area in the near vicinity of the vehicle as well as building facades and also parts of the roofs. The data collected in this way can be used for use cases such as pedestrian detection, mobile mapping and change detection. The field of view of both LiDAR sensors can be seen in Figure 6.2. Each LiDAR sensor is composed of 64 laser rangefinders with a maximum detection range of 120 m. In the present setup, it rotates with a frequency of 10 Hz and acquires about 130,000 range measurements per second. Although the data is acquired continuously, the stream of 3D measurements is split for easier processing to a sequence of scans of 1/10 second duration. This corresponds to a single 360 ° rotation of the scanner head.

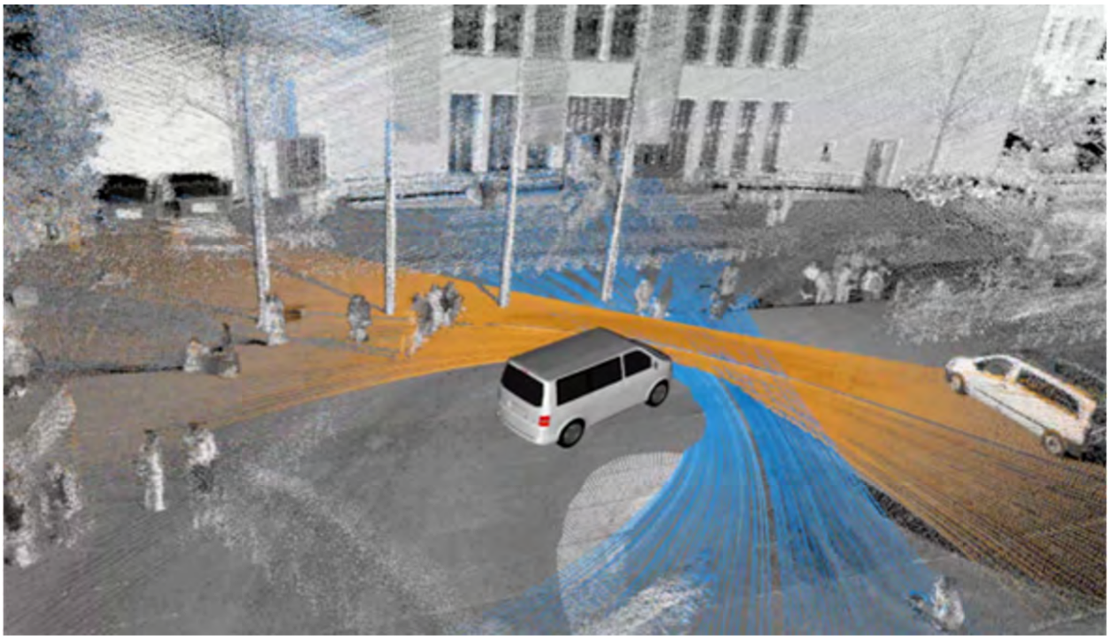


Figure 6.2: The field of view of the LiDAR sensors. Two Velodyne HDL-64E LiDAR sensors are installed on the roof of the measurement vehicle MODISSA.

In addition to LiDAR sensors, the measuring vehicle also has an array of digital cameras. These can for example be used for person detection or for texturing the point clouds generated by the LiDAR sensors. There are eight cameras in total mounted to the roof structure of the vehicle. Two cameras are attached to each corner, pointing away from each other at roughly a right angle. This setup is used to avoid obscurations by the vehicle roof and other components of the measuring system. All eight cameras create a panoramic setup that covers the entire area around the vehicle. The main advantage over an integrated panoramic camera is that there is no obstruction by the vehicle. A disadvantage of such a system, however, is that the cameras do not have a common projection center and therefore stitching the images into a 360 ° panorama is more complicated.

On top of the vehicle, a gyro-stabilized pan-tilt-unit with additional directional sensors is installed. These are a thermal infrared camera, a gray scale camera for visible light and a laser rangefinder. With the thermal infrared camera, for example, the heat radiation of building facades can be examined. In this case, the images from the gray-scale camera can be used for comparison of the infrared images with the visible parts of the facade. The laser rangefinder is used in the context of visual SLAM research.

In order to enable sensor fusion, i.e. the combination of the measurements from several sensors with one another, the timestamp of each recording must be known with high accuracy. For this reason, a system clock is available to synchronize all sensor systems. The system time is made available by the navigation system, which in turn gets it from the GNS system. The position computer of the navigation system provides four event inputs for timestamp signal pulses and four serial interfaces that can transmit National Marine Electronics Association (NMEA) messages and associated pulses. Both are used for different sensors. The LiDAR sensors have an internal clock that is set by receiving an NMEA message and a pulse-per-second (PPS) signal. Interface logic is required in order to mediate between the messages sent by the navigation system and the sensors. This logic is implemented in a small Field-Programmable Gate Array (FPGA). All cameras are triggered by an external signal. The signals required for this are generated by an additional FPGA. The synchronization system described here ensures that the timestamp of every measurement from every sensor is available with a high degree of accuracy. This is essential to combine sensor data with one another, but also to compute a consistent point cloud from the continuous LiDAR and INS measurements.

Navigation system and post-processing

The measuring vehicle is equipped with an inertial navigation system (INS) so that it can determine its 6D pose with a high temporal and geometrical precision. As mentioned above, this is required to generate consistent point clouds even if the vehicle is moving. The INS used in the measuring system is an *Applanix POS LV V5 520*. It includes the following components. An inertial measurement unit (IMU) is utilized to measure the vehicles specific forces and angular rates by a combination of accelerometers and gyroscopes. Two GNSS antennas allow to determine the position using global positioning systems such as GPS, Galileo and GLONASS. The second GNSS-Antenna is required to determine the heading of the vehicle using a technology called *GNSS Azimuth Measurement Subsystem (GAMS)*. A distance measuring indicator installed at one of the rear wheels is used to determine the distance covered by the vehicle. A build-in position computer combines all the measurements of the aforementioned navigation system components into a consistent pose measurement.

The georeferencing of all sensor measurements is based on the navigational data provided by the Applanix navigation system. Various interferences, such as GNSS signal reflections or atmospheric refractions, can occur during a measurement. Any orbit correction data of the GNSS satellites are also only known afterwards. Therefore the navigation data requires post-processing to increase the overall accuracy of the position measurements. This is done using the *POSPac Mobile Mapping Suite*. The correction is carried out by augmenting the recorded navigation data with correction data of the German SAPOS network. The sensor measurements are then georeferenced based on the revised navigation data. Based on the timestamps embedded in the LiDAR data and the corrected navigation data, it is possible to determine the position for each distance measurement. Since the navigation data is only delivered with a frequency of 200 Hz, linear interpolation is applied to determine the individual position of each measurement.

6.3.2 Data quality assurance measures

This section describes the measures used to ensure the quality of the data. The methods used for intrinsic and extrinsic calibration of the LiDAR sensors are described. Furthermore, it is described how the intra-epoch registration, inter-epoch registration as well as the fine registration of the recorded data sets was performed.

Intrinsic LiDAR sensor calibration

The intrinsic parameters are sensor-specific parameters such as the orientation of the laser range finders. Although the Velodyne HDL-64E are pre-calibrated at the manufacturer, the accuracy has been further improved by applying the calibration technique proposed by Gordon & Meidow [2013]. For this purpose, a point cloud generated from the measurements of the Velydyne HDL-64E was compared with the more precise measurement of a terrestrial laser scanner. The calibration parameters were finally derived from both data sets, resulting in a nearly halving of the planar surface misclosure, which has been reduced from 3.2 cm to 1.7 cm.

Extrinsic LiDAR sensor calibration

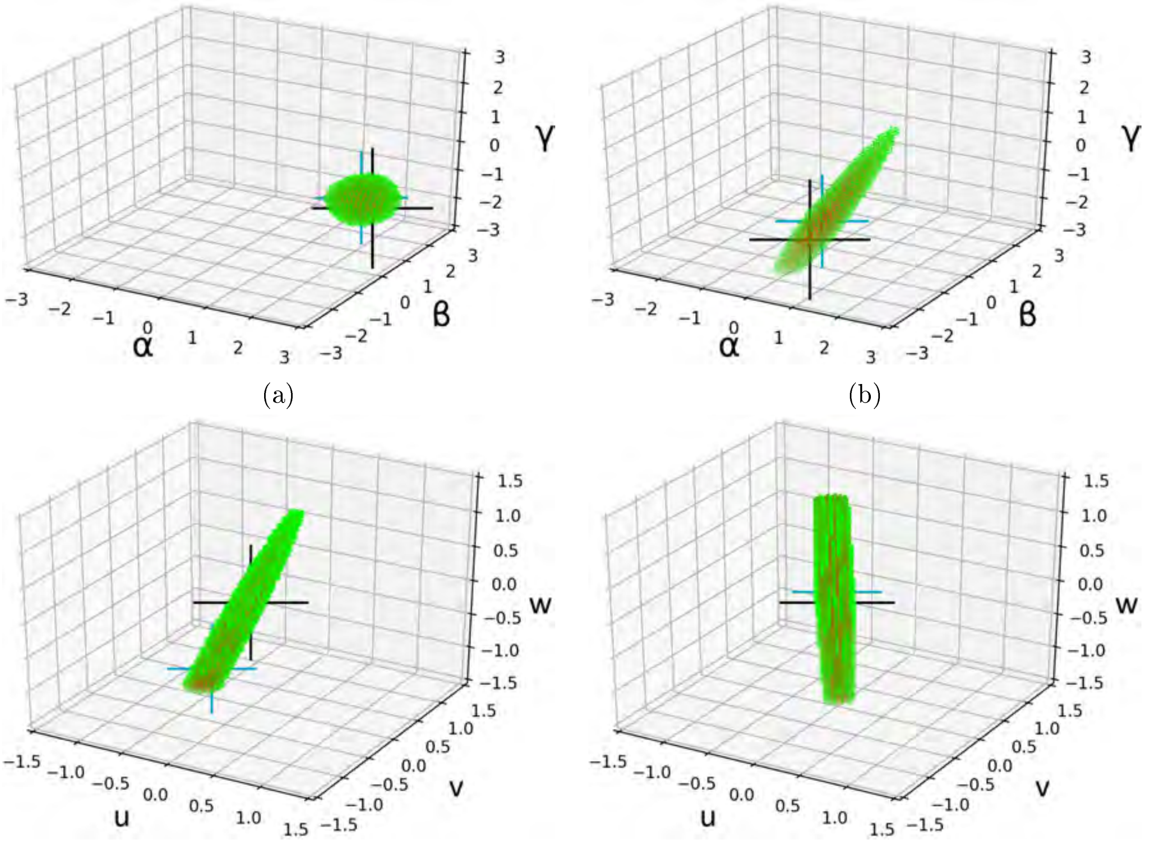
Extrinsic sensor parameters describe the rotation and translation between the sensor coordinate frame and the vehicle frame. The effect of inadequately precise extrinsic parameters leads to the resulting point cloud being distorted. This leads to problems that cannot be solved with the subsequent co-registration of the individual scans and therefore influences the change detection negatively. As part of this work, a contribution was made to the research of a new extrinsic calibration method for LiDAR sensors by participating in the research of Diehm et al. [2020].

Determining the transformation between the sensor and the vehicle frame is challenging because the latter is a coordinate system whose point of origin is not defined by a physical marker or object, and therefore the distance cannot be measured. The simplest solution is to derive the transformation from the vehicle's construction plans. Then, with sufficiently low manufacturing tolerances, any form of extrinsic calibration is completely unnecessary. However, the data in question is usually not available. Unfortunately, this also means that no ground truth is available to verify the functionality of a calibration method.

Although sensors can be calibrated extrinsically with one another relatively easily through data registration, sooner or later at least one sensor must be calibrated extrinsically to the vehicle frame. However, deriving this transformation in a data-driven way is not straightforward. One way of doing this is to set both the sensor and the vehicle in motion, determine the trajectory from both navigation system and sensor and then derive the extrinsic calibration parameters by comparing both trajectories. This would be possible using a flash LiDAR sensor, since it executes all measurements at the same time. In order to get consistent images of the environment from a rotating LiDAR sensor, however, the same extrinsic calibration is required to generate a consistent set of 3D scans.

The procedure used here to calibrate the vehicle is, from a logical point of view, the next best option to solve the problem at hand. The basic idea is to observe the average scatter of the measurement points along the normal of locally planar surfaces. This should tend towards zero if the extrinsic calibration is correct, even if the noise of the measurements or the intrinsic sensor parameters prevent the scatter from being completely zero. The higher the misalignment of the sensor, the greater the scatter along the normal to a planar surface. The solution is to roughly estimate the approximate values and then use a data-driven process to refine them, by minimizing an error measure based on the mentioned average scatter along planar surfaces. This lays down some requirements on the data set used for calibration. On the one hand, the measured terrain must contain smooth surfaces from which the scatter can be determined. On the other hand, appropriate driving maneuvers must have been carried out so that each of the parameters to be determined can also be reliably derived.

A total of six parameters are to be determined, three of which are translational (lever-arm) and three are rotational (boresight). The effect of incorrect translational parameters is far less serious than that of the rotational parameters. Therefore, an incorrect estimate of the translation only leads to relatively small errors. In the case of rotation, however, this looks different, since a misalignment of 0.6 ° results in a displacement of 1.0 m for points at a distance of 100 m. Since the translational aspect of the extrinsic parameters can easily be determined in another way, this is by measuring with a tape measure, and because an error does not play a major role, the main focus was placed on the *boresight calibration*, i.e. the estimation of the rotational component.



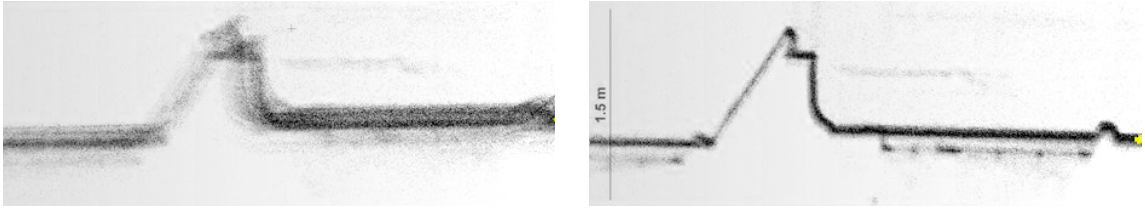
(a) (b)

(c) (d)

Figure 6.3: Parameter space resulting from the LiDAR calibration. a) Rotational parameters for an urban area data set and curvy driving style, b) rotational parameters for a rural area data set, c-d) translational parameters for the aforementioned urban data set for the left and right sensor. All plots show the 0.05 quantile of the parameter space.

To solve this problem, an error measure is first set up with which the quality of the point cloud generated from the measurements can be assessed. As mentioned, this assumes that the scene is characterized by smooth surfaces and that these have also been measured to an adequate extent. The occurrence of inhomogeneous structures such as vegetation leads to an almost constant background noise, since the detection of these structures does not produce smooth surfaces under any assumed sensor position. The scatter of the points is determined by means of the principal component analysis (PCA). For this purpose, the N nearest neighbors for each measuring point are determined using a k-d tree. A fixed number of neighbors was chosen instead of a radius so that the method can better adapt to variable local point densities. Using this neighborhood, the centroid and covariance matrix are computed and the PCA is carried out. Each eigenvalue, divided by the number of neighbors, can thus be viewed as a normalized measure for the scatter of the measuring points in the direction of the respective eigenvector. The smallest normalized eigenvalue belongs to the eigenvector, which can also be viewed as the normal of the planar surface. The error measure is now formed by determining the mean value of all the smallest eigenvalues, the latter being normalized with respect to the number of points in each local point neighborhood.

The optimal calibration parameters are determined by minimizing the error measure. This is done through an exhaustive search. Assuming a grid for sampling the rotational parameters with values between -3 ° and +3 ° in steps of 0.1 ° and another for the translational parameters with values between -1 m and +1 m in steps of 5 cm, then this would result in more than 10^{10} combinations. This is another reason in order to determine rotational and translational parameters independently of one another, as this reduces the number of samples to be examined to approximately 230,000 elements. An alternative is the recurrent dimensional search, in which only one parameter is varied at a time while the others are kept fixed. The recurrent dimensional search can be supplemented with a refined step size, i.e. by reducing the sampling distance of the parameter space in each iteration. However, this can mean that no global optimum is found.



(a) (b)

Figure 6.4: Example of the influence of the rotation LiDAR calibration parameters. a) Point cloud before applying said parameters, b) the same point cloud, after application of the parameters.

The procedure was evaluated, a detailed summary of the results can be found in Diehm et al. [2020]. Five different scenes were examined, which contain urban and rural areas, as well as fields and forests. Each scene also contained different driving maneuvers. The evaluation was carried out by intentionally applying manipulated extrinsic parameters on the measurement data, which then had to be compensated by the method. In the absence of a ground truth, this is the next best approach.

Figures 6.3a and 6.3b show the 0.05 percentile of the parameter set evaluated by the error measure for two scenes. The first one shows the result for an urban environment, the second one for a rural one. The black cross corresponds to the rotational parameters before intentionally adding the error and can therefore be viewed as a ground truth. The blue cross indicates the parameter set selected by the procedure. Examination of the plots shows that a clear global minimum emerges when the environment and driving maneuvers are appropriate. It is also demonstrated that a curvy driving style in an urban environment creates the better conditions for the calibration than other environments. From the small difference between the ground truth and the selected parameter set, it has been concluded that the method is successfully able to determine the rotational parameters with sufficient accuracy. The shape of the parameter space suggests that the parameter set is on the only global minimum. This can easily be verified by looking at the point cloud generated with the determined parameter set (cf. Figure 6.4). However, the results given in the publication also suggest that even under the best of conditions, the final result still varies on the order of 0.1 °, which is the resolution of the grid search. This corresponds to a point positioning accuracy of 17 cm at a distance of 100 m. Furthermore, it can be seen from Figure 6.3c and Figure 6.3d that the translational parameters can only be insufficiently derived, since the necessary driving maneuvers were not carried out. This is driving over hills or bumps in order to gain movement along the z-axis.

Intra-epoch registration

Due to various sources of error in connection with the GNSS and IMU components of the navigation system, the trajectory of a mobile mapping system cannot be determined with absolute accuracy.

This is particularly evident when the vehicle travels long distances that start and end at the same location. Both ends of the trajectory are then usually shifted several meters to one another, and that along all three translational axes. As described above, the quality of the trajectory can be improved by INS data post-processing. However, there still remain disruptive factors, such as GNSS signal interruptions or reflections caused by urban canyons. These and other sources of error mean that the measurement vehicle cannot be located absolutely in retrospect on the basis of its navigation data alone. How this affects the quality of the measurement data can be seen in Figure 6.5a. It shows the measurement data after the trajectory generated by the navigation system has been post-processed. It can be clearly seen that despite the relatively short driving distance between the scans, driving past the building three times already led to clearly visible deviations in the accumulated point cloud.

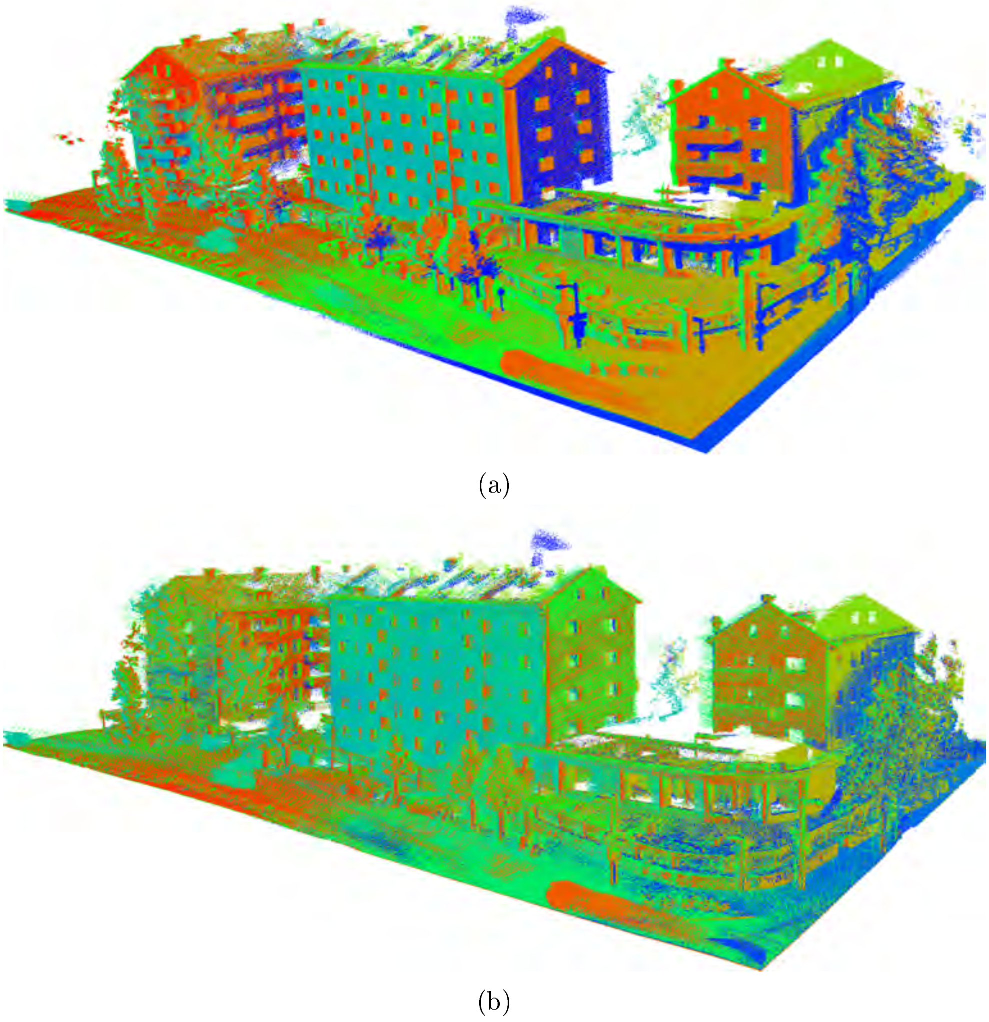


Figure 6.5: Examples of intra-epoch registration. a) Point cloud after INS data post-processing, b) the same point cloud after SLAM-based registration. Different pass-by recordings of the same area are shown in red, green and blue.

If absolute position accuracy cannot be achieved, the next best solution is to modify the trajectory so that the measurements are inherently consistent. This means that they fit together without contradiction, even if the same location was recorded at different points in time. The event that occurs when the measuring vehicle crosses its previous path or comes within visual range is also known as a *loop closure*. In order to close a loop, a correction vector needs to be determined. Said vector superimposes the measurements associated with both points in such a way that they result in a consistent point cloud. It also indicates how the trajectory needs to be shifted and rotated since the last pass in order for the measurements to align.

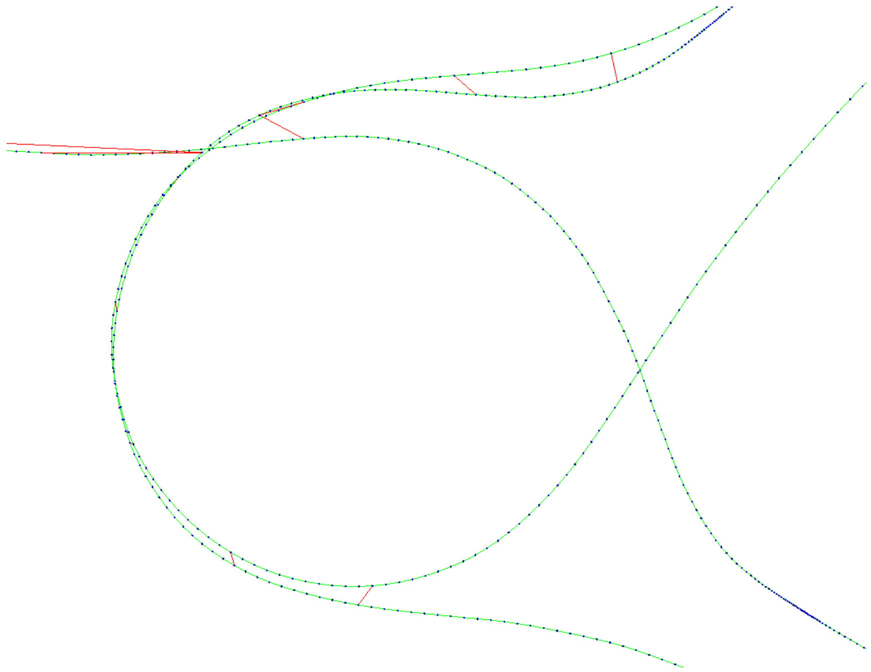


Figure 6.6: Example graph as used in graph based SLAM. Positions measured by the navigation system are shown as blue dots, odometry constraints and loop closures as green or red lines.

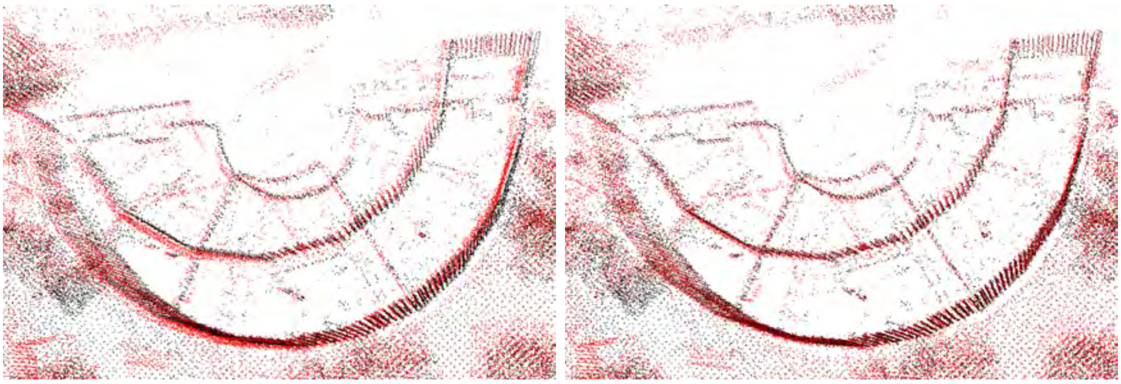
Since many loop closures can occur during a measurement run, the correction vectors must be distributed over the entire trajectory. For this purpose, a process referred to as *graph-based SLAM* was utilized [Grisetti et al., 2010]. A *pose graph* is created, in which the nodes correspond to the measured vehicle poses, i.e. the translation and rotation of the vehicle at the respective point in time. The nodes are connected in chronological order via so-called *odometry constraints*. In the present case, these are taken from the post-processed trajectory. Whenever a loop is found, an additional *loop closure constraint* is inserted between the current node and the matched node. A part of the pose graph for the scene shown in Figure 6.5 can be found in Figure 6.6. The trajectory is optimized by interpreting the graph as an optimization problem, which is then solved using a standard optimization method such as Levenberg-Marquardt. The first vehicle pose is fixed, otherwise the optimization problem cannot be solved. In addition, the loop closure constraints are weighted higher than the odometry constraints, because they are considered more accurate. In a final step, the correction vectors for rotation and translation are derived from the corrected graph and added to the individual measurements.

To determine loop closures, references to all poses are stored in a two-dimensional grid. The vehicle poses are processed in sequence. Before inserting a pose into a cell, a neighborhood search on said cell and its neighbors is performed. If a reference is found, the corresponding measurements are compared and the correction vector between the two scans is determined using an automatic registration algorithm such as ICP. Whether or not the ICP procedure was successful, i.e. a reasonable match between the two measurements was found, is determined by applying a threshold to the ICP error measure. If a match has been found, a new loop closure constraint is added to the pose graph. The final result can be seen in Figure 6.5b. As can be seen, most of the registration errors have been successfully corrected.

Inter-epoch registration

Inter-epoch registration is required whenever an epoch of new measurements is to be registered to an existing data set. As already described in the last section, the new measurements are subject to the same sources of error, which is why a location with absolute accuracy is not feasible. As before, the data must therefore be registered in such a way that it is consistent. Consistent here means that the measurements from the new measurement run agree both with themselves and with the previously recorded data set of the same area.

The procedure is similar to that of the intra-epoch registration. Again, a grid for storing the references to the vehicle poses is utilized. However, before the vehicle poses of the new measurement run are added to the grid, the ones from the trajectory of the first measurement run are inserted. From here on, the procedure is analogous to that in the last section, with the exception that loop closures are now not only possible within the current measurement run, but also with the first one. From a logical point of view, the pose graph constructed in this way is therefore divided into two parts. First, as described, a reference pose graph is constructed. It contains only nodes, which are set to be fixed in the context of the optimization. There are no odometry or loop closure constraints. Then the pose graph of the measurement ride to be registered is constructed by adding the vehicle poses step by step, this time with odometry contraints. Loop closure constraints are possible between nodes of the second pose graph, but at the same time also with nodes of the reference graph. As in the previous section, an optimization problem is then created from this combined pose graph and solved using standard methods.



(a) (b)

Figure 6.7: Effect of chunk-based fine-registration. a) Point cloud after SLAM-based inter-epoch registration, b) the same point cloud after chunk-based detailed registration.

The quality of the results is comparable to that of the inter-epoch registration. However, as Figure 6.7a is intended to show, despite the subsequent registration, some smaller registration errors may still remain. Although the ends of the trajectory subsection between two loop closures are properly registered, this does not automatically mean that the sections in between also are. The longer said subsection, the higher the probability of a deviation, since a navigation system can only cover a route for a certain period of time without any noteworthy tracking errors. However, since the registration of such a subsection is already pretty good, a simple technique for fine-registration can be used to improve the results. Said technique is described in the next section.

Fine-registration

An accumulated point cloud based on the first measurement run is used as a reference for the fine-registration. The basic idea is to register every scan of the new measurement run against the reference. A scan is a full rotation of the sensor head, i.e. a 360 ° point cloud. This procedure is not a substitute for inter-epoch registration, as there can be large shifts between the data sets or changes in the environment that can only be dealt with by correcting the trajectory. After inter-epoch registration, however, the shifts are usually small enough that registration techniques such as ICP deliver satisfactory results. Combining multiple scans into a so-called *chunk* enables the registration of scans that are poor in detail. It also significantly reduces the likelihood of incorrect registration. In the procedure implemented in this work, 5 scans are combined into a chunk, whereby the scans are spaced 10 scans apart. This distance corresponds to exactly one second. A chunk corresponds to 5 seconds of driving for which a stable trajectory can be assumed. All scans of the chunk are then combined into a single point cloud and registered to the aforementioned reference point cloud using ICP. The reference point cloud is downsampled to a resolution of 10 cm, which increases the runtime drastically. The lack of detail is compensated for by the spatial extent of each chunk. Thus, the registration is essentially done using large or distinct structures such as the ground plane, walls, vehicles, or tree trunks. The benefit of the chunk-based fine-registration can be seen in the upper right corner of Figure 6.7b.

6.3.3 Data sets

This section presents the data sets that were used for extrinsic sensor calibration as well as for evaluation of the methods presented here.

Ettlingen calibration data set

The data set was recorded in May 2019 using the mobile mapping system MODISSA and includes a measurement run over the compound of the Fraunhofer IOSB in Ettlingen. Only a few minutes of data were recorded. Several short sections were selected and used to determine the extrinsic calibration parameters between the coordinate frame of the LiDAR sensors and the vehicle coordinate frame. The quality of the urban features contained in the data set only plays a subordinate role. Instead, the focus is on the geometry as this is used by the calibration process. The data set is particularly suitable for calibration, since the majority of the surfaces recorded by the sensor vehicle are planes. These planes are at different distances from the vehicle, so it is likely that there will always be planes where the calibration errors are sufficiently evident.

It was ensured that the vehicle was moved along all rotation and translation axes during the recording of the data set. This was achieved by placing cable bridges on the ground, so that the vehicle was raised by a few centimeters and rotated by a few degrees along the roll and pitch axis when driving over it. A rotation around the yaw axis was achieved by driving serpentine lines. The trajectory can be seen in Figure 6.8. The data set is short enough that the trajectory supplied by the navigation system can be viewed as ground truth of the vehicle location over time. Dead reckoning, based on the IMU, works reliably for short periods of just a few seconds, so GNSS localization errors are irrelevant. The scans from both LiDAR sensors were used to perform the extrinsic sensor calibration described in Section 6.3.1.

TUM-MLS data sets

The TUM-MLS data set was recorded as part of a cooperation between the Chair for Photogrammetry and Remote Sensing at the TU Munich and the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation in Ettlingen. The data set consists of a total of two measurement runs, the trajectories of which can be found in Figure 6.9. Two recordings were made at a large time interval in order to allow the environment to change in a way that makes the search for non-trivial changes possible. The first epoch was recorded in April 2016 and includes the TUM city campus as well as the adjoining street areas. The record of the second epoch took place in

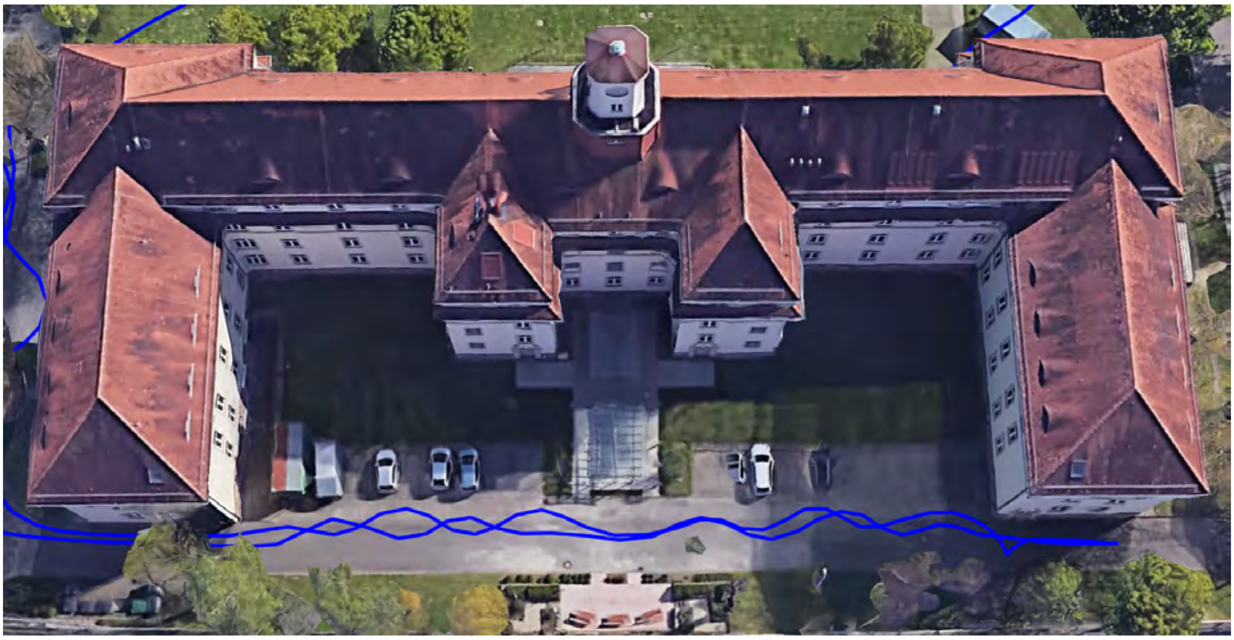
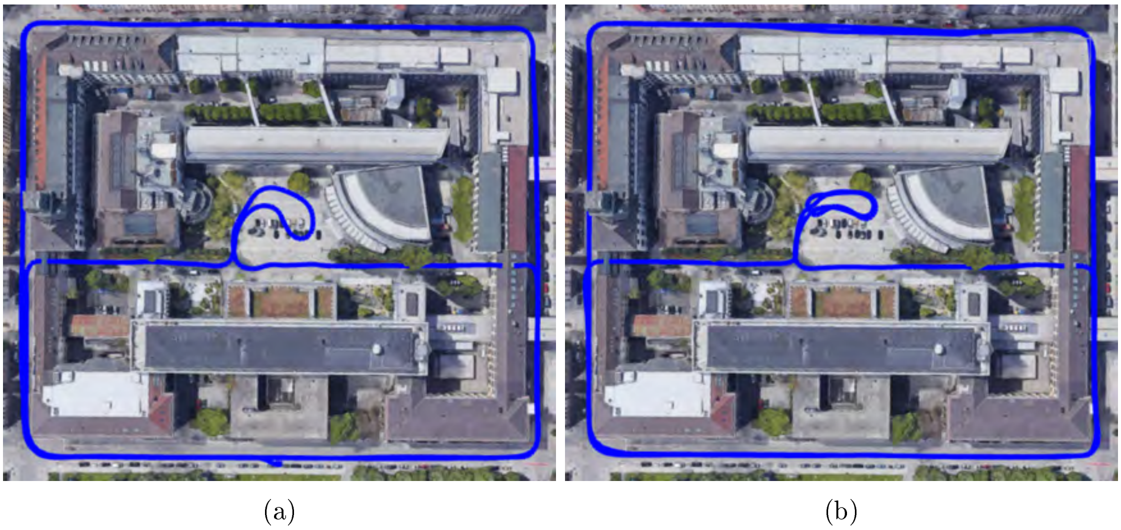


Figure 6.8: Overview of the trajectory used for the extrinsic sensor calibration. The source for background image and 3D model is Google Earth.

December 2018. Both measurement runs were made with the MODISSA measuring vehicle as part of this work.



(a)

(b)

Figure 6.9: Overview of the trajectories for the TUM-MLS-2016 and TUM-MLS-2018 data set. a) Trajectory for the first data set recorded in April 2016, b) trajectory for the second data set recorded in December 2018. The source for background image and 3D model is Google Earth.

A variety of sensors was used during both measurement runs. As described in Section 6.3.1, this includes two LiDAR sensors for three-dimensional recording of the environment, several cameras working in the visible spectrum and an uncooled microbolometer for recording longwave infrared (LWIR). Georeferencing was supported by SAPOS and a Leica reference station, which was installed on the roof of one of the TUM buildings for the duration of each measurement drive. The 3D recordings of both data sets are available in form of georeferenced single 360 ° scans. Said scans and the images of the bolometer were made available to the public under a Creative Commons License[†].

Both data sets cover many scenarios and occurrences that can be found in an urban environment. Large numbers of people were recorded on the campus, some of whom were seated, standing or walking. Various traffic situations such as cars turning at intersections and road crossings were recorded in the street space. Cars, vans, pedestrians and two-wheelers were involved. Buildings with complete facades, vegetation at different times of the year and various street furniture were also recorded. An overview of the data set can be found in Figure 6.10.

A subset of the data sets was selected for the purpose of parameter estimation. The area around the *Alte Pinakothek* was chosen for this purpose. This section contains the multitude of all objects that can be found in an urban street space. Furthermore, many changes are present here, as well as small fine structures that can be used to investigate quantization errors. The area of the test data set in marked in Figure 6.10.

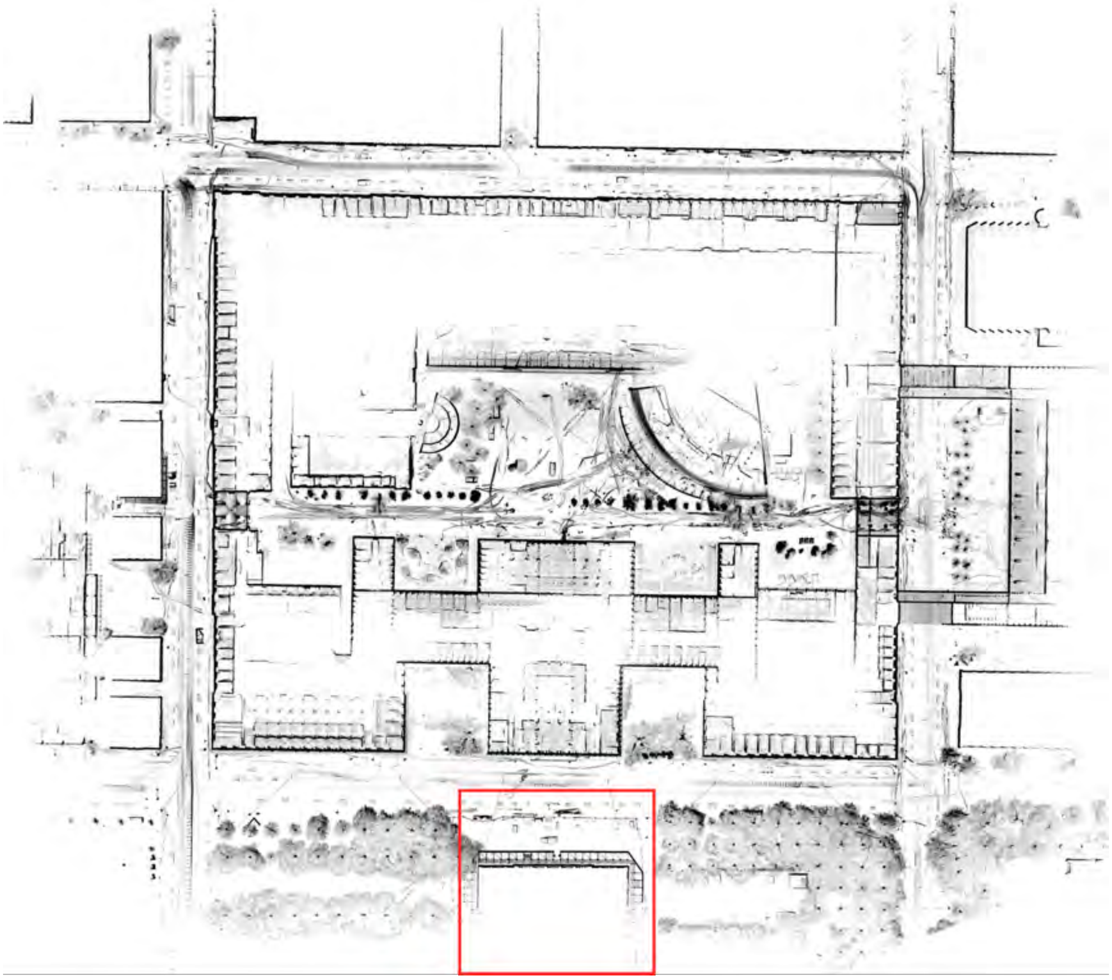


Figure 6.10: Accumulated 3D point cloud of the first epoch of the TUM-MLS data set. The point cloud is shown in birds-eye view. The area of the subset used for parameter estimation is indicated by a red box.

The approach of Underwood et al. [2013] was used to remove the moving objects from both epochs. Said approach is designed to compare two point clouds with each other. However, the measurements of the mobile mapping system consist of a sequence of individual 360 ° scans, where one scan represents a geolocated point cloud with about 100,000 individual measurements. Un-

[†] The TUM-MLS 2016 and TUM-MLS 2018 data sets can be downloaded at http://s.fhg.de/mls1

derwood's process has been adapted to efficiently compare the above mentioned individual 360 ° scans. It was ensured that the scans depict the same location as far as possible, but that the time interval between them is as large as possible. This was solved by utilizing a two-dimensional grid in which each scan is stored based on its sensor position.

First, all scans were distributed on said grid. Then all scans are processed in sequence, the corresponding grid cells and their neighbors are determined, and another scan is selected that is as far in the future or past as possible. Both scans are then compared and the moving objects, i.e. the changes, are determined. A fuzzy representation is then created from the information about the locations of all moving objects. This can then be subtracted from the occupancy grid using an *and not* operation. This approach has the advantage that said representation can be swapped against another one generated from a ground truth, if additional sources of error are to be excluded during the evaluation.

6.3.4 Ground truth

This section presents the ground truth that was created for the quantitative evaluation of the change detection method presented in this work. Furthermore, the ground truth of semantic object classes, also used in the evaluation, is discussed.

Ground truth for change detection

In order to be able to quantitatively evaluate the results of the change detection methods presented in this work, a *ground truth* is required. The ground truth describes the elements of the environment that changed between epochs. Since there is no ground truth for the real-world data available, there is no choice but to create it manually. Here, however, it must be considered that it can only be an approximation to a real ground truth that is at most as good as the person that makes the annotation. Especially when free space information plays a major role, there is a potential for error, because its clear visualization is not easily possible. A subset of the TUM-MLS data set containing the *Arcisstrasse* was selected and manually annotated. Encoding changes requires two point clouds for each pair of epochs, one to encode the changes between epoch A and B and vice versa. Each point has one of the following labels:

- unchanged
- moving object
- appeared in epoch B
- disappeared from epoch A
- not observed in epoch A
- not observed in epoch B

The *unchanged* and *moving object* labels are applied to points in both point clouds in order to mark unchanged surface points and points that are part of objects that moved during recording. In addition, the point cloud that describes the transition from epoch A to B encodes both surface points that *disappeared* and those that are in an area that was *not observed in epoch B*. The other point cloud representing the transition from epoch B to A describes *appeared* surface points as well as all points that were *not observed in epoch A*. The use of two point clouds also simplifies the process of manual labeling, as both epochs can be visualized side by side. The point clouds on which the manually created approximated ground truth is based are generated by accumulating and downsampling all individual measurements in the above-mentioned area. The downsampling was carried out using a grid with a resolution of 10 cm, with the first observed point per cell being selected as a representative for all other points in said area.

In order to remove points that are the result of reflections and sensor noise, all points that have less than 5 neighbors within a half meter radius were removed. Remaining clutter was then removed in a manual post-processing step. Subsequently, an initial version of the annotations was bootstrapped by transferring automatically generated suggestions to both point clouds. This is necessary to identify areas that were not seen in the other epoch. This requires information about the space that was traversed by the measurement rays, therefore this can only be done automatically, since this information cannot be presented in a meaningful way during manual post-processing. Furthermore, the automatic suggestions simplify the issuing of manual labels, since existing labels only have to be corrected and not completely reassigned. In a final step, labels are corrected manually. This task took approximately four weeks to complete. The results of the automatic labeling as well as the final labels can be seen in Figure 6.11 and Figure 6.12.

As already mentioned, a preliminary version of the labels was created automatically. There are two main reasons why suggestions in form of automatically generated labels are recommended. Annotating the complete data set is a very cumbersome and time-consuming process. On the other hand, it is straightforward to automatically generate labels by a simple comparison of occupancy information. In some cases, this approach is even superior to the manual approach, for example when multi-layered structures such as scaffoldings are involved. Furthermore, it is easier for an observer to check existing labels for plausibility and correct erroneous ones, instead of having to assign labels to a completely unlabeled point cloud. The second reason concerns the abilities to visualize the information that is required to make informed decisions. It is not straightforward to visualize areas that have been observed as free, i.e. that have been traversed by measuring rays. In order to be able to differentiate correctly between a changed object and an object not observed again, this information is essential. Of course the required information can be visualized by a bundle of lines, but not to the extent that the observer can still interpret the result. The main reason for this is the lack in depth perception caused by the many visual elements that obscure each other and the points to be labeled.

Similar to the method presented in this thesis, the annotations were created on the basis of a comparison of occupancy information. At this point it should be mentioned that all labels were corrected manually until they appeared plausible to a human observer. The author of this work would like to emphasize that to the best of his knowledge and belief he has minimized any bias in the ground truth. Occupancy grids with a resolution of 10 cm were generated, one for each epoch. Each grid cell contains statistics derived from the measurement data, namely two counters that store the number of measurement rays that pass through the cell and that represent surface points. Said information is obtained by means of raycasting [Amanatides & Woo, 1987]. If a cell is traversed by more than 10 rays, then it is considered to be free. It it contains more than 10 surface points, than it is considered to be occupied. If both cases are present, then the one with the higher number of samples is preferred. In order to automatically derive the labels, the occupancy grids of both epochs are compared with each other cell by cell. Depending on the result of the comparison, all points that fall into each cell c are annotated accordingly. If two compared cells are occupied, the points are labeled as *unchanged*. In case one of the cells is occupied and the other one is free, then the points are annotated as changed. That means that if a cell in epoch A is compared to the same cell in epoch B, the points are marked as *disappeared*, otherwise as *appeared*. If the counters in one of both cells are zero, than the area hasn't been observed and the points are marked as either *not observed in epoch A* or *not observed in epoch B*, depending on the

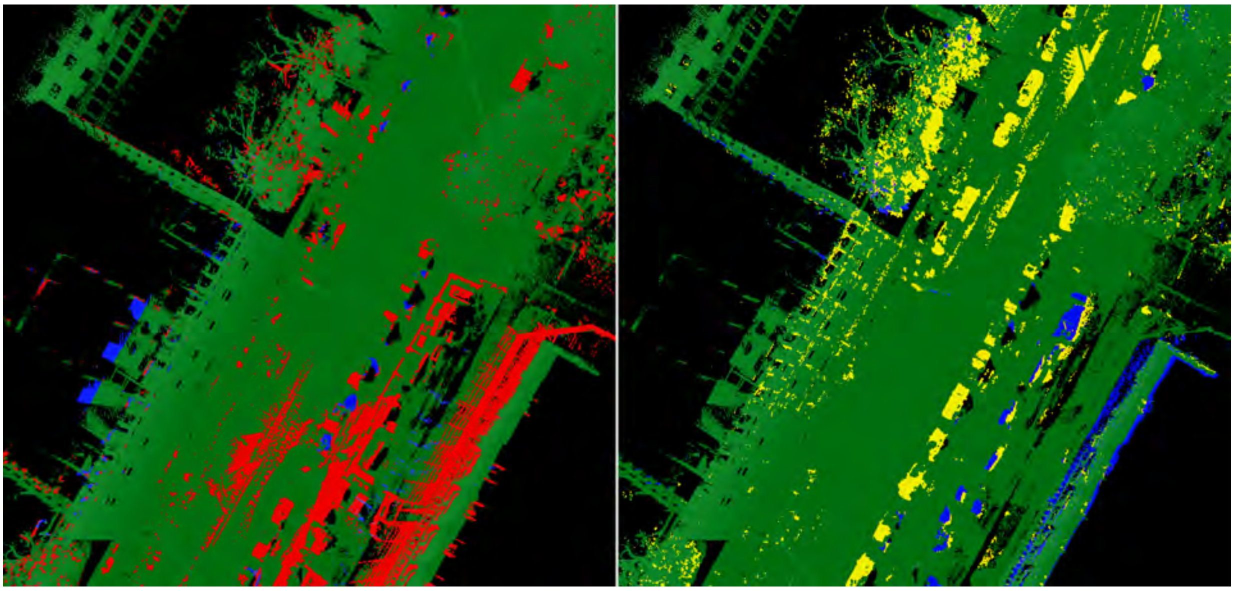


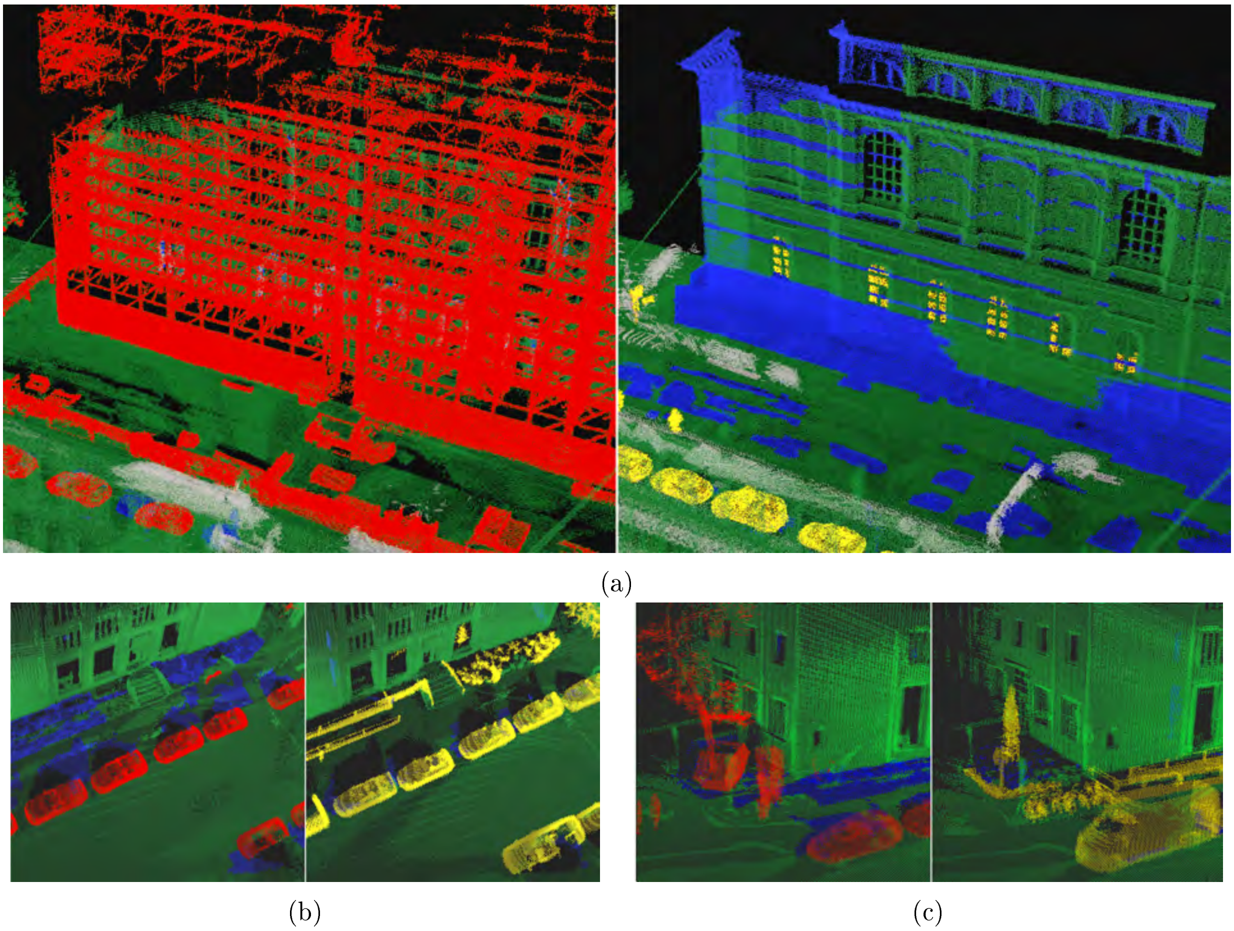
Figure 6.11: Overview of automatically generated change detection labels. Unchanged points are green, appeared and disappeared points are yellow and red. Points unseen in the other epoch are marked in blue. Moving objects are not part of the automatically generated labels.

direction of comparison. In order to annotate both point clouds, the process has to be repeated symmetrically. That means that epoch A has to be compared to epoch B and vice versa. The resulting point clouds can be seen in Figure 6.11. The labeling errors resulting from the process and the measures for manual correction are explained in the next section.

The automatically annotated data is too noisy to be used as ground truth, which is why manual post-processing is required. For this purpose, a special editor was developed that visualizes both epochs side by side. Both views are synchronized, so changing the viewing angle in one view automatically changes the other view. This ensures that both views always show the same area from the same perspective. A grid can be displayed to enable comparison of partial areas. The labels of one or more points can be changed by dragging a frame around them with the mouse. For all selected points, the label is then replaced by one with the previously selected class. All points within the frame are affected, including those at a great distance from the camera. For this reason the annotation was simplified by dividing the data set into several tiles, with each tile being annotated independently of the others. Since the subdivision into tiles takes place at runtime, the complete data set is available at all times. Depending on the object class, different challenges have arisen during annotation. The following subsections provide a detailed explanation. The manually revised labels can be seen in Figure 6.12.

Buildings Significant changes in the building structure such as destroyed or newly constructed buildings are not part of the data set, only minor changes such as extensions and bricked up windows. Since these are significant changes in flat structures, these were also detected reliably by the bootstrapping method. Smaller deviations at the building edges and windows were manually corrected.

Narrow structures Despite the relatively high resolution of mobile laser scanning, the resolution of narrow structures such as people and bicycles is relatively low. This is due to the fact that only few measuring points are recorded on their surface. Due to the discretization of the bootstrapping method, this often leads to incorrect labeling results, especially at the edges. Affected labels were adjusted manually.



(a)

(b)

(c)

Figure 6.12: Sample scenes of manually revised labels. a) Alte Pinakothek, b) street scene, c) construction site. Unchanged points are green, appeared and disappeared points are yellow and red. Points unseen in the other epoch are marked in blue. Moving objects are marked in gray, but are only shown in a) for reasons of clarity.

Partially displaced objects The problem of partially displaced objects is particularly evident in case of vehicles, especially for those which are parked at the roadside. If a vehicle is slightly displaced between two epochs, this usually results in two symmetric occupancy changes. Surface points seem to have appeared at one end of the vehicle and disappeared at the other end. The slight displacement of cars between two epochs can be explained by marked parking bays at the roadside. A similar effect occurs whenever a vehicle is replaced by one with a different shape. All vehicles were checked manually and the entire vehicle was marked as changed if it was moved or replaced.

This introduces a potential for error, as it mixes semantic annotations and an otherwise purely geometric annotation. However, from the author's point of view, there was no justification here for annotating only the geometric changes from the perspective of a potential application.

Moving objects Marking moving objects in an accumulated point cloud is easier and less time-consuming than in single scans, and also leads to similarly good results. Moving objects can usually be recognized by the tubular structures they form in accumulated point clouds. Their positions are also limited to certain areas within the data set, such as sidewalks and crosswalks. The automatic annotation procedure marked most parts of said tubular structures as unchanged, because there were too few points per cell to accept a change.

Noise and artifacts The automatic annotation is susceptible to noise and other error sources. The reason for both can be explained by the discretization of space. In case of noise, spatial areas are combined, which can lead to points or point clusters being given an inappropriate label. Some artifacts can be explained by measurements that pass by flat surfaces at a small angle [Gehrung et al., 2017]. The automatic annotation procedure then recognizes these areas as changed based on differences in the perceived occupancy. Both effects were corrected by manual post-processing.

Vegetation Labeling vegetation is challenging, both for the automatic annotation procedure and a human observer. Foliage has many fine structures, too fine to be accurately reproduced by mobile LiDAR scans. Due to shadowing from leaves, fine branches cannot be recognized properly. The degree of foliage is approximately the same in most cases, only slightly more in the first epoch, since the recording took place in another season. The suggestions of the bootstrapping method were partially adopted, but since they are difficult to confirm or refute manually, the vegetation class is only marginally considered in this work.

Occluded areas An important task with regard to change detection is to identify areas that are visible only in one epoch, but not in the other. This is because a change can only be detected if the area under consideration has been observed in both epochs. In the data set, such events usually occur in two places, that is within buildings and behind vehicles. Unobserved areas in building interiors are caused by shutters or doors that are open in one epoch, but closed in the other. For example, if a shutter has been partially lifted, as was the case in one of the campus buildings, more space is visible in one epoch than the other. A human observer can recognize such a case on large areas such as walls, but only with difficulty on an apparently appeared independent object such as a ceiling lamp. Such occurrences can usually only be deduced from the automatically generated labels, since it is difficult to intuitively visualize the space traversed by measuring rays.

The other type of area that is usually subject to occlusion can be found behind vehicles. Since the sensors cannot detect this area, a kind of *observation shadow* seems to form here. The areas of unobserved space annotated by the automatic process are usually too small and are surrounded by a halo of points that are marked as changed. However, since these areas are located on planar surface such as a streets or sidewalks, it is straightforward to correct the labels manually. Two tools were used to simplify this process. The first one is a lock for points marked as *changed* that prevents them from being edited. This is necessary in order to be able to label points under the vehicle without changing the labels of the vehicle points. The second tool is the above mentioned grid that is projected onto the screen. This is inspired by techniques used to copy paintings. Without both tools it would not be possible to reproduce complex occlusion patterns accurately and correctly localized.

Ground truth for object classes and instances

The change detection method examined in this work allows additional information to be included. These are not required for finding changes, but can be used for a more in-depth analysis of the results. In the present case, the benefit of semantic object classes is that they enable a class-specific examination of the results of the change detection. Data sets containing semantic object classes are usually used as benchmark data sets, i.e. to determine the performance of automatic semantic annotation processes. The results generated by a procedure are compared to a ground truth and the correspondence between the two is expressed numerically. As discussed in detail by Matrone et al. [2020], the creators of the ArCH benchmark data set, it is important that the classes contained in a benchmark data set fit thematically to the examined environment. A wide range of benchmark data sets for semantic labeling exists. For those based on 3D data, the Oakland outdoor MLS data set [Munoz et al., 2009], Semantic3D.net TLS data set (Semantic3D) [Hackel et al., 2017], Paris-Lille-3D MLS data set [Roynard et al., 2018], Toronto-3D MLS data set [Tan et al., 2020], Daimler urban segmentation data set [Scharwächter et al., 2013] and A2D2 data set [Geyer et al., 2020] can be given as examples.

In addition to the thematic suitability, the conditions that prevailed during the recording must also match. The platform used for the measurements plays a significant role because accuracy, density, and quality of the resulting point clouds vary significantly between recording techniques. Therefore, if one only limits to semantically annotated data sets that are based on mobile laser scanning data, then this list contains the Oakland 3D data set [Munoz et al., 2009], the Sydney Urban Objects data set [DeDeuge et al., 2013], iQmulus [Vallet et al., 2015], Paris-Lille-3D [Roynard et al., 2018], SemanticKITTI [Behley et al., 2019], Toronto-3D [Tan et al., 2020], the Daimler urban segmentation data set [Scharwächter et al., 2013], the A2D2 data set [Geyer et al., 2020] and both the TUM-MLS-2016 and TUM-MLS-2018 data set [Zhu et al., 2020].

The latter two are based on the data sets that were recorded for this work. The manual annotation has been done by Zhu et al. [2020]. Two different types of labels are provided, the first are semantic annotations with eight classes, the second are annotations for instance segmentation. Both are available as voxel grids with a resolution of 20 cm. For ease of use, the annotations have been transferred to accumulated point clouds with the same resolution. In addition, the object class labels were transferred back to the 360 ° laser scans. The advantage of the TUM-MLS data set is that, in addition to the semantic labels, the sensor position is also preserved for each measuring point, which is required for the change detection within the scope of this work

Annotations of the data set were made directly in 3D, which is a labor-intensive but reliable operation. The underlying point cloud equals the one used for annotating changes. Statistical outlier removal as well as downsampling has been applied to the point cloud. The latter has the advantage of removing duplicate points that were generated by waiting periods of the measuring vehicle, for example at a traffic light. The annotation itself was carried out using the software CloudCompare. More than 40 million points were labeled with eight semantic classes following the ETH standard, namely the Semantic3D.net benchmark [Hackel et al., 2016].

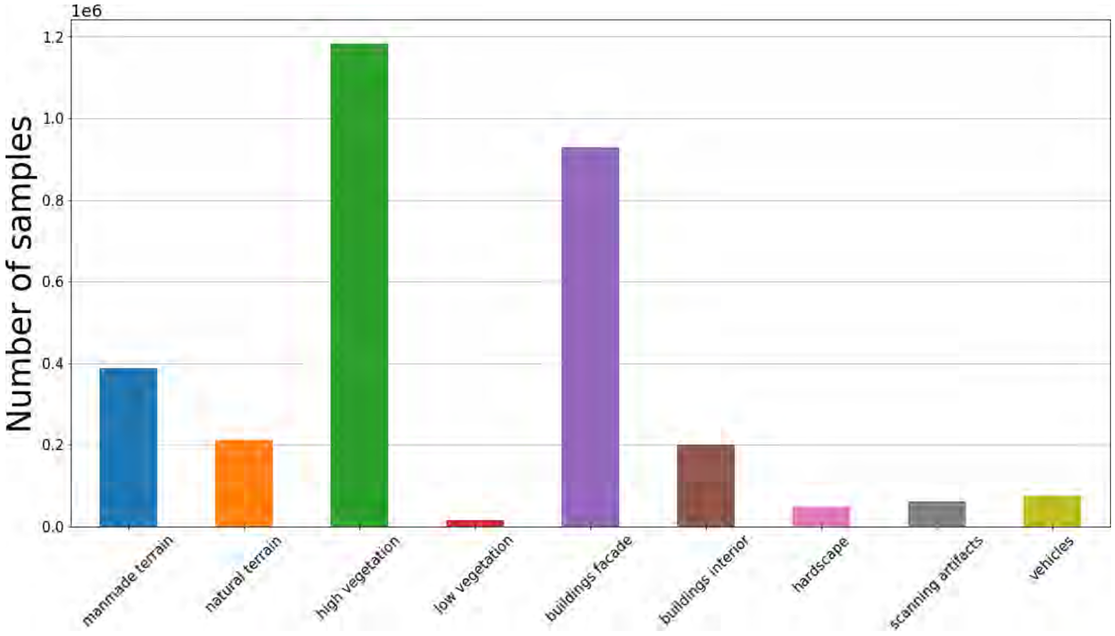
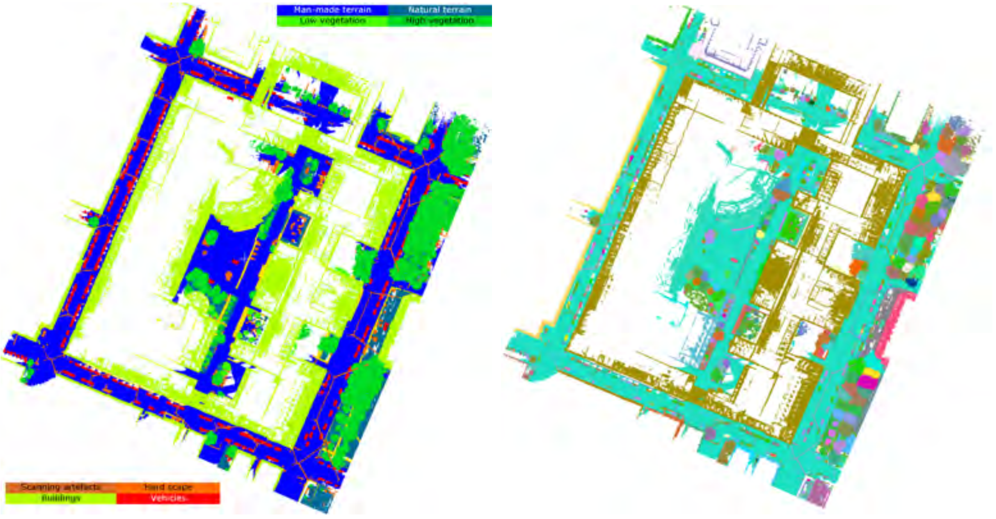


Figure 6.13: Number of samples per object class. Some classes are several orders of magnitude more represented than others.



(a) (b)

Figure 6.14: Overview over the TUM-MLS semantic segmentation benchmark data set. a) Overview of object class labels, b) overview of object instance labels.

Classes	Label Index	Content
Man-made terrain	1	Roads and impervious ground.
Natural terrain	2	Grass and bare land.
High vegetation	3	Trees.
Low vegetation	4	Bushes and flower beds.
Buildings	5	Building facades, roofs and interior.
Hardscape	6	Walls, fences, light poles.
Scanning artifacts	7	Power cables and artificial objects.
Vehicles	8	Parked cars and buses.
Unclassified	0	Noise, outliers, moving vehicles, pedestrians, and unidentified objects.

Table 6.1: Annotated object classes of the TUM-MLS semantic segmentation benchmark data set.

An overview of all classes can be found in Table 6.1, an illustration of the annotated data set in Figure 6.14a. The classes *natural terrain*, *low vegetation* and *high vegetation* describe natural structures usually found in an urban environment. Artificial structures are represented by the classes *man-made terrain*, *buildings* and *hardscape*. The *scanning artifacts* class largely includes hose-like structures that are caused by moving objects. All points that have not been assigned a label are summarized by the *unclassified* class. Since the data set was published, the *buildings* class has been expanded to include interior points. The samples are unequally distributed across all classes, as can be seen in Figure 6.13.

As with the annotation of the change detection data, the instance labels provided by the benchmark data set were first automatically annotated and then manually corrected. Therefore, both automatically and manually generated labels were created in a two-step approach. The first step comprises the automatic segmentation using an unsupervised clustering [Xu et al., 2018]. The second step is a manual error correction and object boundary refinement. The manually corrected labels can be seen in Figure 6.14b. At the time the benchmark was published, there have been 1002 objects of the eight classes mentioned above.

7 Results and Analysis

This chapter presents the qualitative and quantitative results of the experiments described in Chapter 6, which were used to evaluate the methods described in Chapters 3-5. The results presented here provide the foundation for the discussion of the method under investigation.

7.1 Spatial representation results

This section summarizes the results of all experiments used to evaluate the generic evidence grid for spatial representation.

7.1.1 Memory consumption

Figure 7.1 shows the average memory requirements per tile. The average memory requirement has a maximum value of about two MiB* per tile for the highest resolution. The memory consumption grows exponentially to the depth of the octree, which is to be expected due to its structure.

7.1.2 Membership function

Figure 7.2 shows the plots of the average error measures of the linear and logistic membership function. The error is plotted against spatial resolution, not octree depth, since only this properly reflects the nonlinear relationship. The various octree resolution levels are marked by dotted vertical lines, the average error measures are specified by points.

The error of both functions is close to zero for the maximum resolution of 20 cm. This is as expected since the resolution is exactly the maximum resolution of the ground truth incorporated into the representation (cf. Section 6.3.4). The average errors for the coarser resolution levels show clearly evident differences. The linear function's error increases abruptly and then settles at a relatively high level, showing a logistic growth. The logistic membership function's error increases less abruptly. A peak forms at a spatial resolution of 0.4 m, but quickly flattens out for higher resolution levels. Apart from said peak, an approximately linear growth can be seen here.

The histograms shown in Figure 7.3 illustrate the distributions of the number of samples as well as the memberships for the evidence obtained by applying the linear and logistic membership function. They were generated exemplarily on the basis of the class *vehicles* for a resolution of 40 cm, the data was averaged over all voxels. As can be seen in Figure 7.3a and Figure 7.3b, the distribution of the samples strongly resembles the distribution resulting from the linear membership. All values along the x-axis are finely graduated and use the full range of values specified by the samples. The distribution of membership values generated by the logistic function shown in Figure 7.3c shows a different behavior. A large part of the evidence mass is allocated to the

[*] The notation used here for memory requirements corresponds to the international standard ISO 80000-13. This means that statements referring to memory are made on the basis of powers of 2, not on the basis of powers of 10.

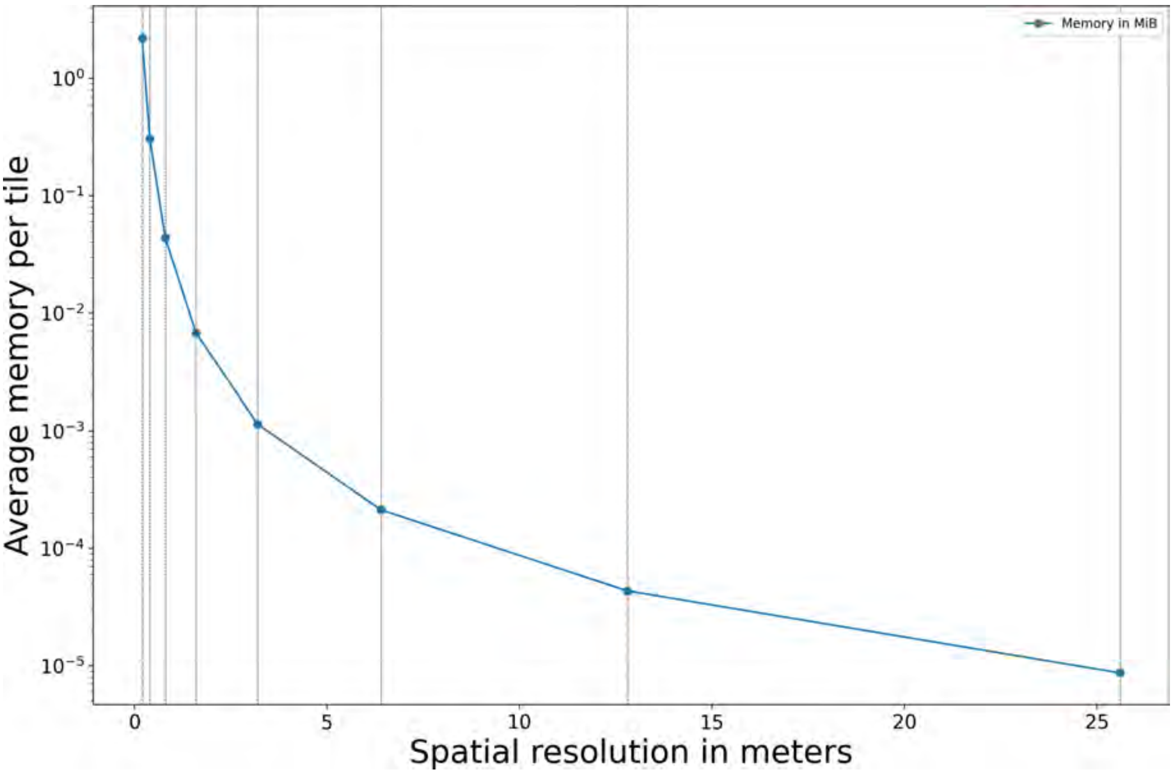


Figure 7.1: Memory consumption in relation to spatial resolution.

extreme values 0 and 1 at each end of the interval. Negligible to no amounts of evidence are distributed on the values in between. Instead of soft intermediates, hard values are enforced.

7.1.3 Reproducibility before defuzzification

In order to determine the cause of the peak in the error measure visible in Figure 7.2, both the first local maximum at 0.4 m as well as the following local minimum at 1.6 m were examined in more detail. The overlap with other classes was examined, since a lower overlap can be a possible explanation for a lower error. For the spatial resolution of 0.4 m, 12.99 % of all voxels containing a ground truth point have an overlap with another class. In contrast, at the spatial resolution of 1.6 m, 56.23 % do. The coarser the resolution, the more voxels have a collision with another class. This implies that the reason for said peak is not a higher overlap, but that the average deviation between representation and ground truth is much lower in total at the coarser resolution of 1.6 m.

Figure 7.4 shows the histograms of the certainty for class A and $\bar{A}$ and the error measure for the aforementioned resolutions. It can be seen that the certainty of class A for the 0.4 m resolution comprised the entire spectrum (cf. Figure 7.4a). In other words, the certainty distribution is not particularly clearly pronounced, as there are many intermediate levels. On the other hand, at the 1.6 m resolution, the distribution has fewer intermediate levels and is therefore more clearly pronounced (cf. Figure 7.4d). The certainties for class $\bar{A}$ of both resolutions are essentially comparable and therefore do not play a major role in terms of the effect under discussion (cf. Figure 7.4b and Figure 7.4e). The errors behave as shown in Figure 7.4c and Figure 7.4f. For the 0.4 m resolution, there are many errors of small and intermediate size. These errors can be thought of as noise superimposed on the signal. At 1.6 m resolution, the signal-to-noise ratio is better, with fewer small and medium errors but more large errors.

Also to be noted in Figure 7.2 is the visible change in the slope of the error measure that occurs between 3.2 m and 6.4 m. This is a clear indication that the ground truth can still be reproduced

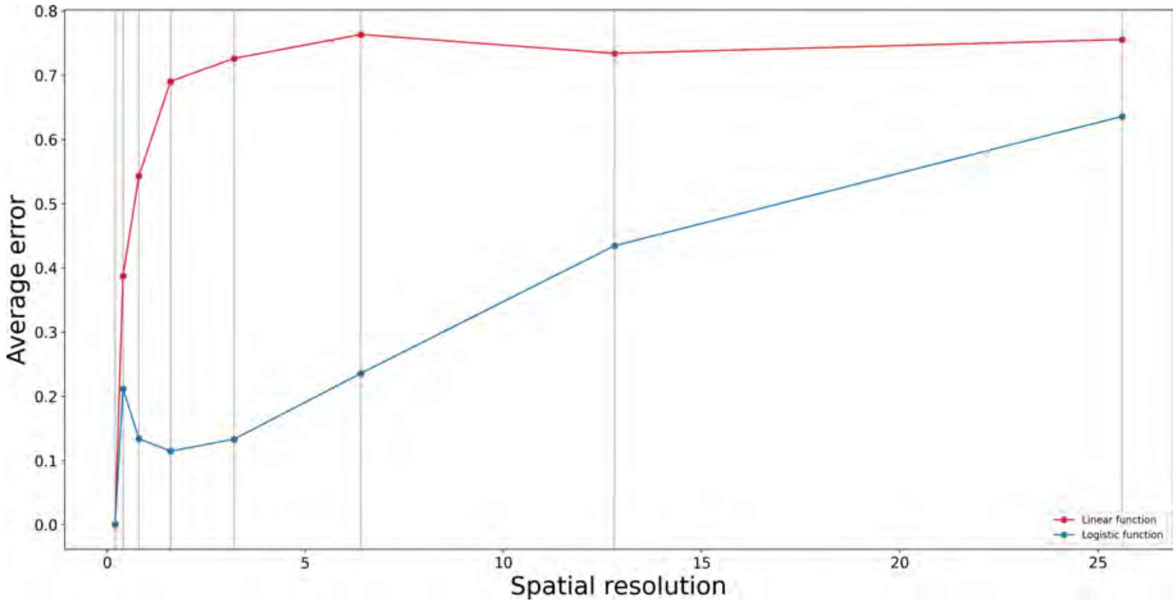
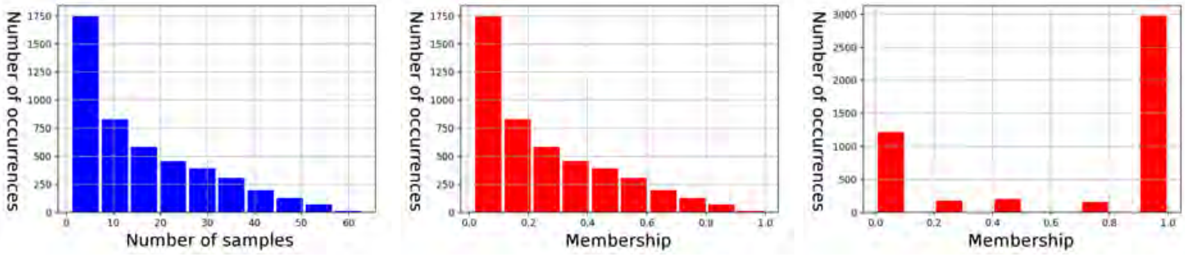


Figure 7.2: Average error resulting from applying the linear (red) and logistic (blue) membership functions. The average error of both functions is different, because the logistic function is biased towards the extreme values, while the linear function is a direct, unbiased mapping.

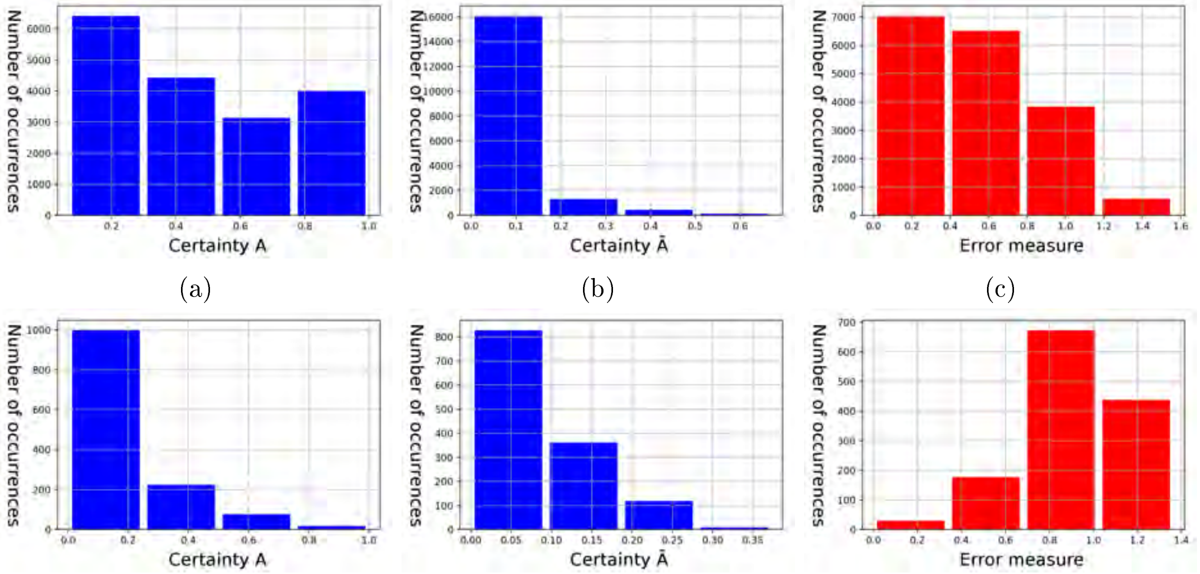


(a) (b) (c)

Figure 7.3: Histograms of samples and membership for the linear and logistic membership function. a) The distribution of samples for all voxels for the vehicle class at a 0.4 m resolution, b) the membership computed by the linear membership function, c) the membership computed by the logistic membership function.

sufficiently well at a resolution of 3.2 m, but no longer at the next higher resolution. This may have to do with the fact that, except for the few areas where class instances are located close to each other, there is a large free space area between them. When the resolution is reduced, at some point several class instances previously separated by free space overlap. This point seems to be located somewhere between the spatial resolution levels of 3.2 m and 6.4 m.

Figure 7.5 illustrates the average error per object class as a function of the spatial resolution. It shows that although each class has its own characteristics, the general behavior is largely identical. The error is close to zero at the 0.2 m resolution for all classes, because this corresponds to the resolution of the input data. The peak for the classes *high vegetation*, *building facades* and *manmade terrain* is only slightly pronounced. All three classes were prominently located within the field of view of the LiDAR sensors during the recording. The class *low vegetation* and *building interior* have a very low error across all resolutions. Both occur in a very isolated fashion and have few points of contact with other classes. Two-thirds of the *low vegetation* class do not have any ground recorded nearby due to the recording angle of the LiDAR sensors and *building interior* has no contacts to other classes except for the *building facade* class. The error in the *scanning*



(a) (b) (c)

(d) (e) (f)

Figure 7.4: Histograms for examining the peak in the error measure. a-c) Histograms for a resolution of 0.4 m, d-f) histograms for a resolution of 1.6 m. From left to right, histograms of the certainty for class A, the certainty for class $\bar{A}$ and the error measure are shown. Note the difference of an order of magnitude in the y-axis.

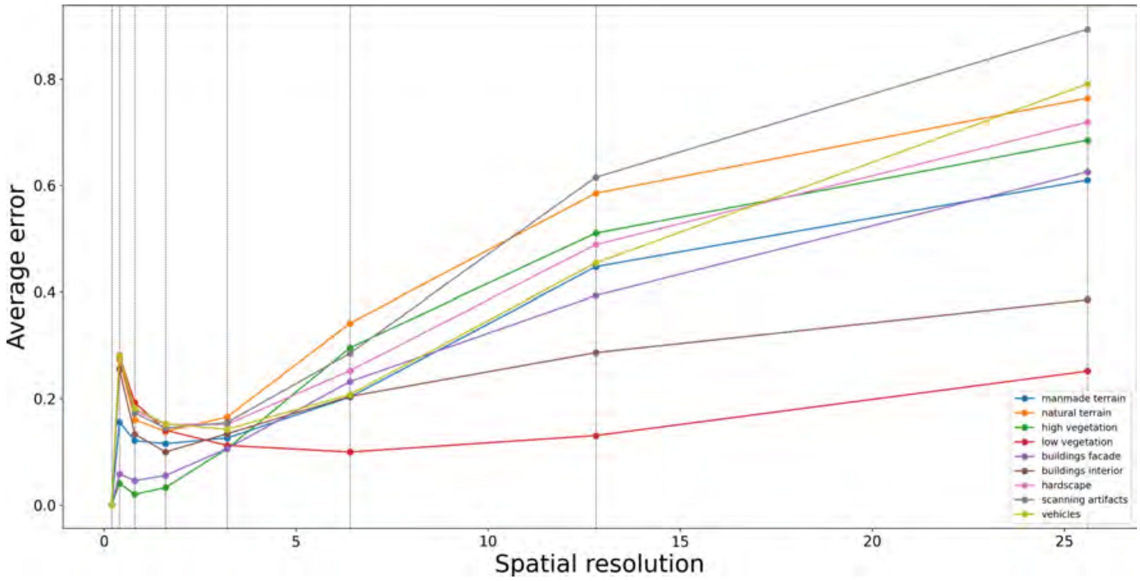


Figure 7.5: Average error per object class.

artifacts class is particularly pronounced, as this class contains small, mostly elongated structures with many points of contact with other classes.

7.1.4 Reproducibility after defuzzification

Figure 7.6 illustrates the F1-scores for all object classes. As with the error measure in the last section, the F1-score is optimal for the finest resolution, i.e. it has a value close to one, since the resolution of ground truth and octree are identical. However, the plot is missing the peak-like anomaly discussed in the previous section. Instead, a steadily decline of all scores can be observed for decreasing resolution, albeit with one minor exception.

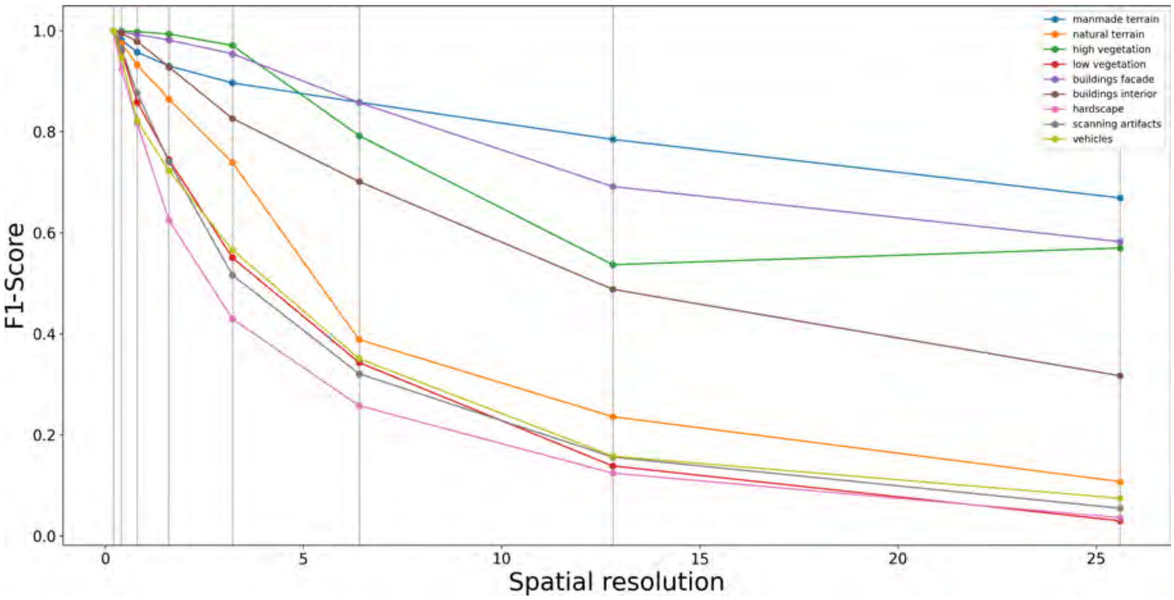


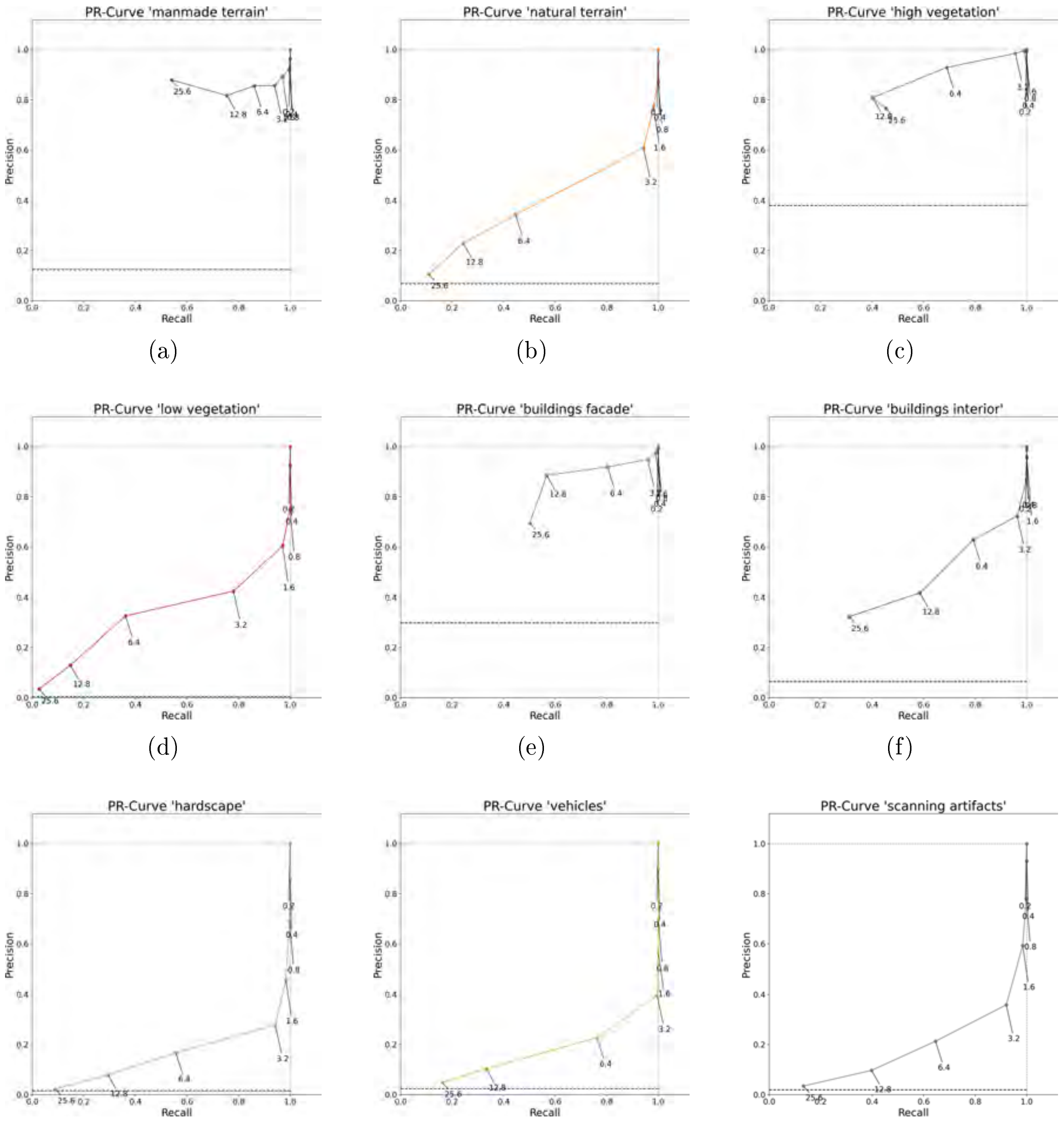
Figure 7.6: F1-score per object class calculated after applying defuzzification.

Examining the F1-score shows that some classes can be better represented than others. Based on Figure 7.6, two groups of classes can be identified. The first group, with a rather high average F1-score, includes the classes *manmade terrain*, *building facade*, *high vegetation*, and *building interior*. These classes were already mentioned during the investigation of the average error. They have in common that they are prominently observed, are large in scale and have a small overlap with other classes. The second group with the rather low average F1-score includes the remaining classes *natural terrain*, *vehicles*, *scanning artifacts*, *hardscape* and *low vegetation*. These classes have a large overlap with other classes, contain relatively few samples, especially the *low vegetation* class, or are unfavorably shaped, as it is the case with the *scanning artefacts* class. It is also noticeable that the first group is represented with far more samples than the second group, up to two orders of magnitude (cf. Figure 6.13). The class *building interior* is an exception, since it has approximately as many samples as the class *natural terrain*. However, the former has a higher F1-score, since it is very isolated.

The precision/recall diagrams created from the confusion matrix are shown in Figure 7.7. The data points of all object classes for each resolution are above the respective *no-skill line*. This implies that even at the most coarse resolution, every representation is better than a classifier that is not able to discriminate between the classes, e.g. a classifier that makes random decisions. In addition, the two groups described in context of the F1-score can be clearly distinguished from one another. For *manmade terrain*, *building facade* and *high vegetation* a consistently high precision and recall can be seen. This means that, on one hand, the majority of all predictions is correct and, on the other hand, most of the ground truth is correctly predicted. The class *building interior* is an exception, although not quite as extreme as the classes in the second group. The object classes of the second group, i.e. the classes *natural terrain*, *vehicles*, *scanning artifacts*, *hardscape* and *low vegetation* consistently show a high recall, but a strongly varying precision. This means that a large part of the ground truth is correctly classified, but in addition, many labels of other classes have also falsely been classified as belonging to the class.

7.1.5 Discretization errors

A false positive occurs whenever a voxel has strong evidence that it belongs to one class, but then a sample of a different class falls within said voxel. False positives are examined based on



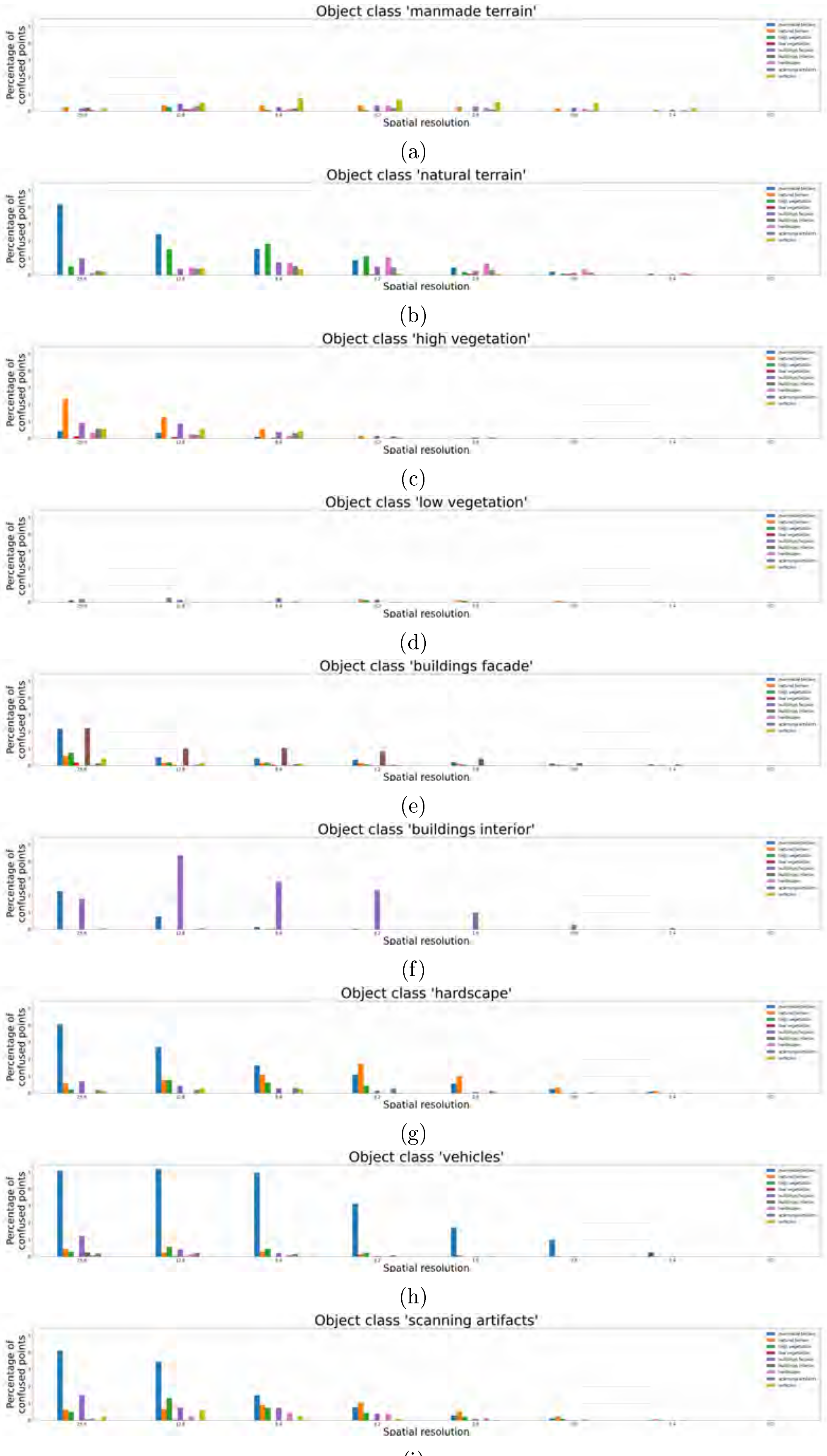
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7.7: Precision/recall diagrams for all object classes. The plots are generated from the confusion matrix resulting from the comparison of object class labels and the corresponding fuzzy representations. The parameter under consideration is the spatial resolution.

confusions between object classes, which have been recorded during the generation of the confusion matrix. Figure 7.8 illustrates these in form of a bar chart. Looking at the relationship between *false positives* and *resolution*, it can be seen that the former are either consistently low, if there are hardly any points of contact with other classes, or decrease with increasing fineness of the resolution. Occasionally a local maximum for one of the medium resolutions can be seen, as it is the case with *building facade* in the class *building interior*. However, this only implies that the evidence for said classes is particularly pronounced for the respective resolution levels. The proportion of false positives to the amount of ground truth varies between classes. For some of them, it is about 4-5 % for the coarse resolution levels, and below about 0.5 % for others. For fine resolutions, the percentage of false positives tends towards zero. Thus, it can be highlighted that even at the most coarse resolution of 25.6 m, only a very small percentage of false positives occur.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 7.8: Bar chart of the confusion between object classes for different resolution levels. The y-axis corresponds to the percentage of all points that were interpreted incorrectly.

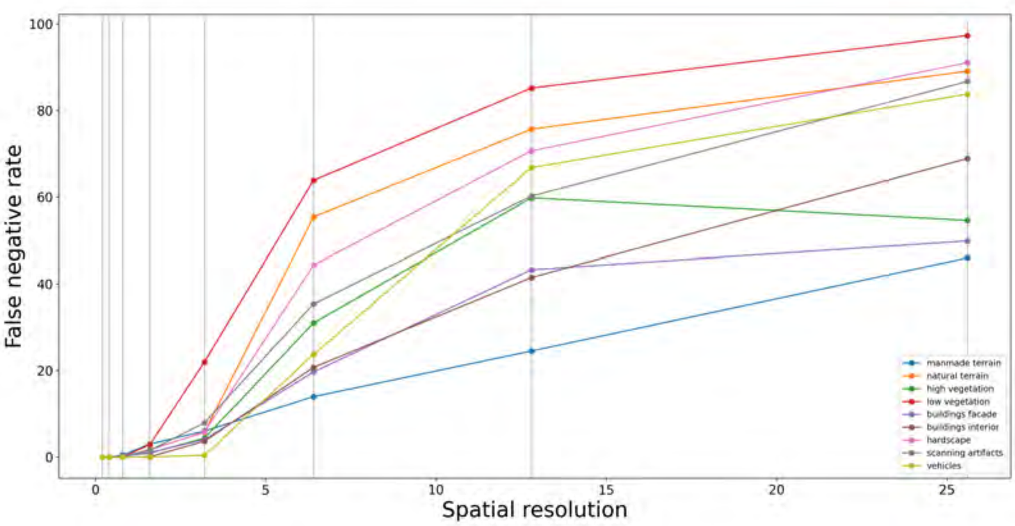


Figure 7.9: The percentage of false negatives, broken down by object class. The percentage refers to the number of all samples.

A *false negative* occurs whenever a class is not recognized as such, i.e. when the evidences in a voxel are too weak to counter the preponderance of evidences of other classes. This occurs especially in areas where many classes interface. In other words, if a sample of a class is classified as not belonging to said class, this leads to a false negative. False negatives are examined by visualizing their percentage of the number of all samples, broken down by object class. The percentage of false negatives out of the number of all samples can be seen in Figure 7.9. As with the false positives, the two groups *manmade terrain*, *building facade*, *high vegetation* and *building interior* as well as *natural terrain*, *vehicles*, *scanning artifacts*, *hardscape* and *low vegetation* can be distinguished.

7.2 Change detection results

This section summarizes the results of all experiments used to evaluate the change detection method. The results of the first stage are presented, which were used to decide on the free parameters of the procedure. Subsequently, the results of the change detection are presented, which are based on said parameters.

7.2.1 Membership function and computation mode

In this section, the results of the experiments to identify the best membership function required to create occupancy grids are presented. This is part of the first stage of experiments to determine the free process parameters. The results of the comparison of the linear, logistic and occupancy membership function can be seen in Figure 7.10. All three functions produce satisfactory results. This means that the underlying geometry can be accurately recognized in the occupancy grid and the associated certainties can be considered sufficiently high. As the color coding in Figure 7.10a shows, the linear function produces the results with the least certainty. This can be seen from the fact that the fewest red voxels are present here, which imply a certainty close to the value 1. Instead, green voxels predominate, which mean a certainty in the range of around 0.5. The certainties generated by the logistic function are more pronounced, especially in the area containing fine structures. This can be recognized by the higher number of red voxels. The occupancy membership function produces the results with the highest certainty.

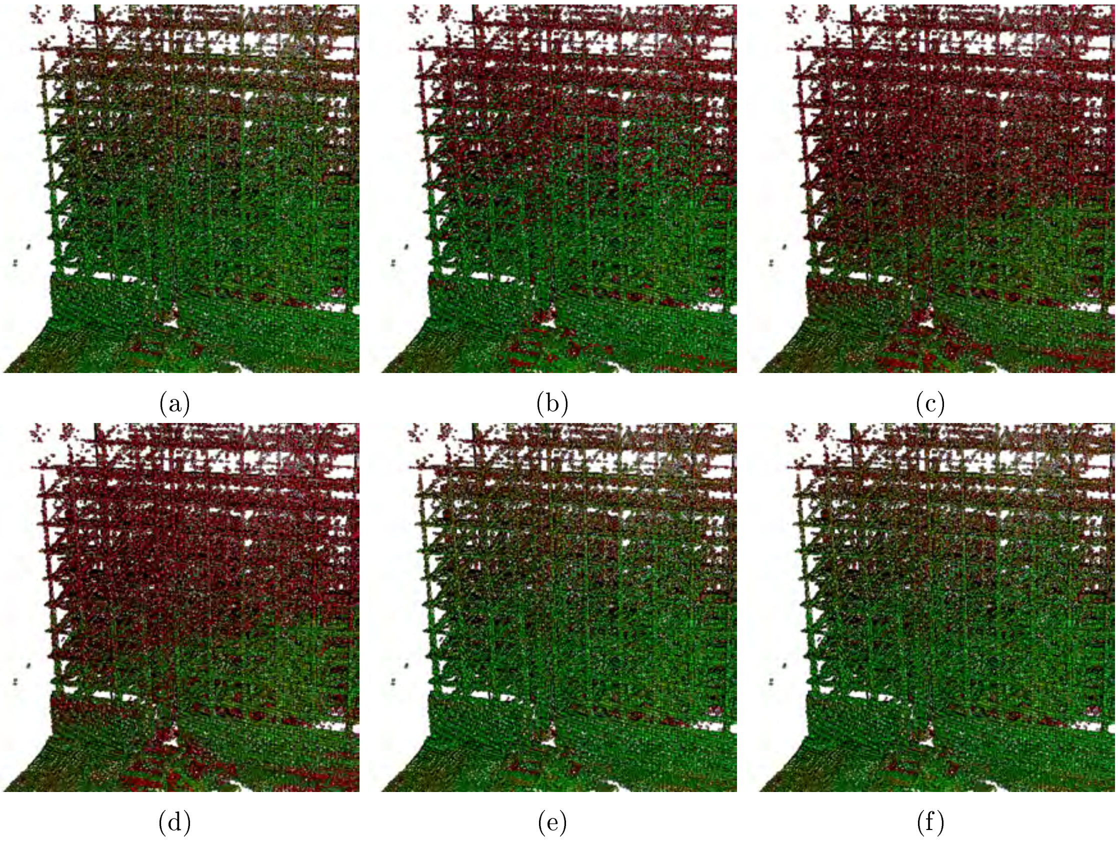


Figure 7.10: Occupancy representations generated with different membership functions and computational modes. a) Linear membership function, b) sigmoid membership function, c) occupancy membership function, d) tile-based computation mode, e) sliding-window computation mode, f) fixed-window computation mode. The occupancy membership function with the tile-based computation mode leads to the best result. The color encodes the degree of certainty: blue=low, green=medium, red=high.

The comparison of the *computation modes* was carried out based on the membership function selected beforehand. The *global* mode was not examined because it proved to be too computationally intensive. The results of the application of the *tile-based*, *fixed-window* and *sliding-window* can be seen in Figures 7.10d, 7.10e and 7.10f. The *tile-based* computation mode produces the results with the highest certainty. The other two modes result in a significantly lower certainty in the area containing fine structures, but otherwise produce comparable results.

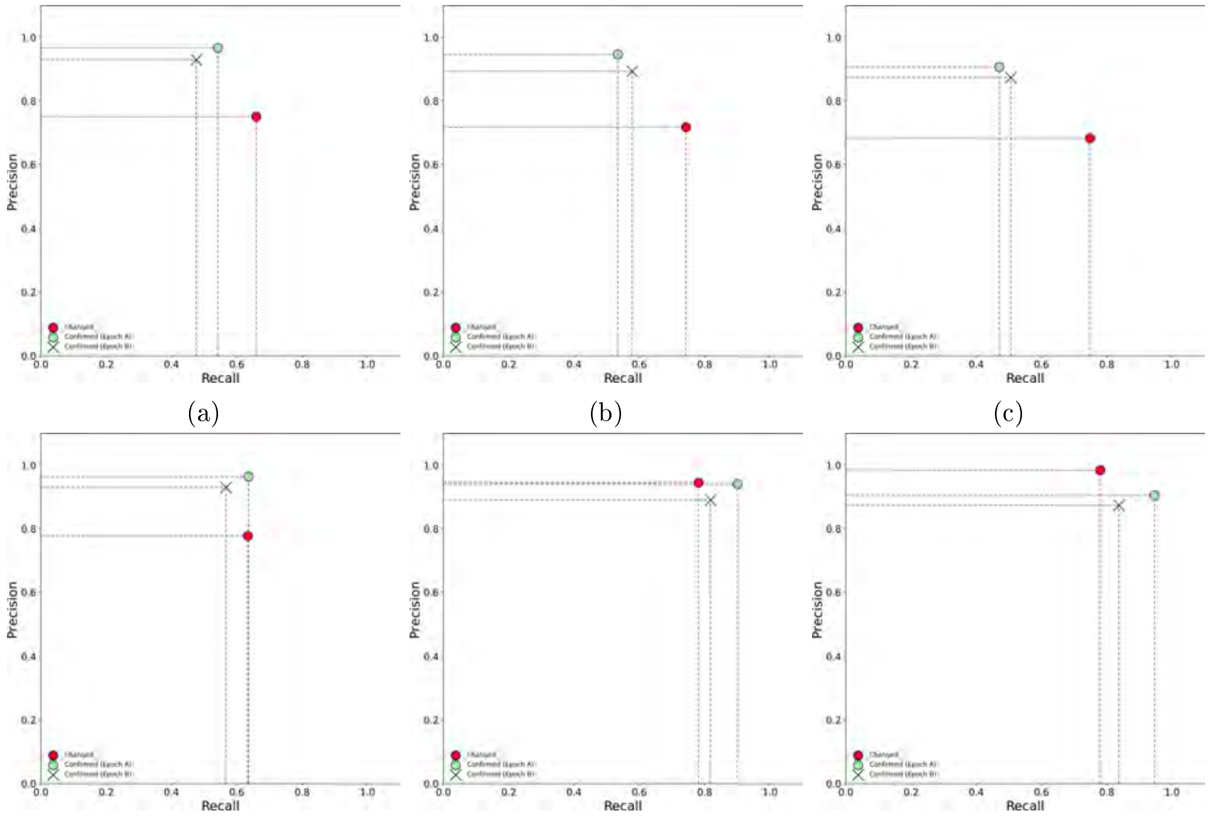
7.2.2 Max-pooling neighborhood and defuzzification

This section presents the results of the experiments conducted to determine the max-pooling neighborhood and to decide whether to use defuzzification. This is part of the first stage of experiments to determine the free process parameters. In total, six experiments were performed to test the max-pooling neighborhoods of $0 - 2$ voxels once with, once without defuzzification. The precision/recall diagrams constructed from the corresponding fuzzy confusion matrix of each experiment can be seen in Figure 7.11. The plots show that defuzzification leads to a better result, which, however, is only visible when max-pooling is applied. The application of max-pooling leads to an increase in recall for both *confirmed* and *changed* geometry. In addition, the precision for *changed* geometry is also increased. The numerical results for the experiments with defuzzification are shown in Table 7.1. They coincide with the bottom row in Figure 7.11. The table shows that

	No max-pooling			max-pooling (n=1)			max-pooling (n=2)		
Case	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Confirmed (A)	0.96	0.65	0.78	**0.94**	**0.92**	**0.93**	0.91	0.97	0.94
Confirmed (B)	0.93	0.57	0.71	**0.89**	**0.82**	**0.85**	0.87	0.84	0.86
Disappeared	0.78	0.66	0.71	0.95	0.81	0.88	**0.98**	**0.81**	**0.89**

Table 7.1: Results of the change detection on the *Alte Pinakothek* subset. The results achieved with the final configuration of the method are marked in bold.

not applying max-pooling leads to high precision but only mediocre recall. The recall improves significantly as soon as max-pooling with a neighborhood of one voxel is applied. While the precision of *confirmed* geometry remains largely unchanged, a significantly increase can be seen with *changed* geometry. Increasing the max-pooling neighborhood to two voxels slightly improves the F1-score, which leads to an improvement in recall and in some cases to a slight reduction in precision.

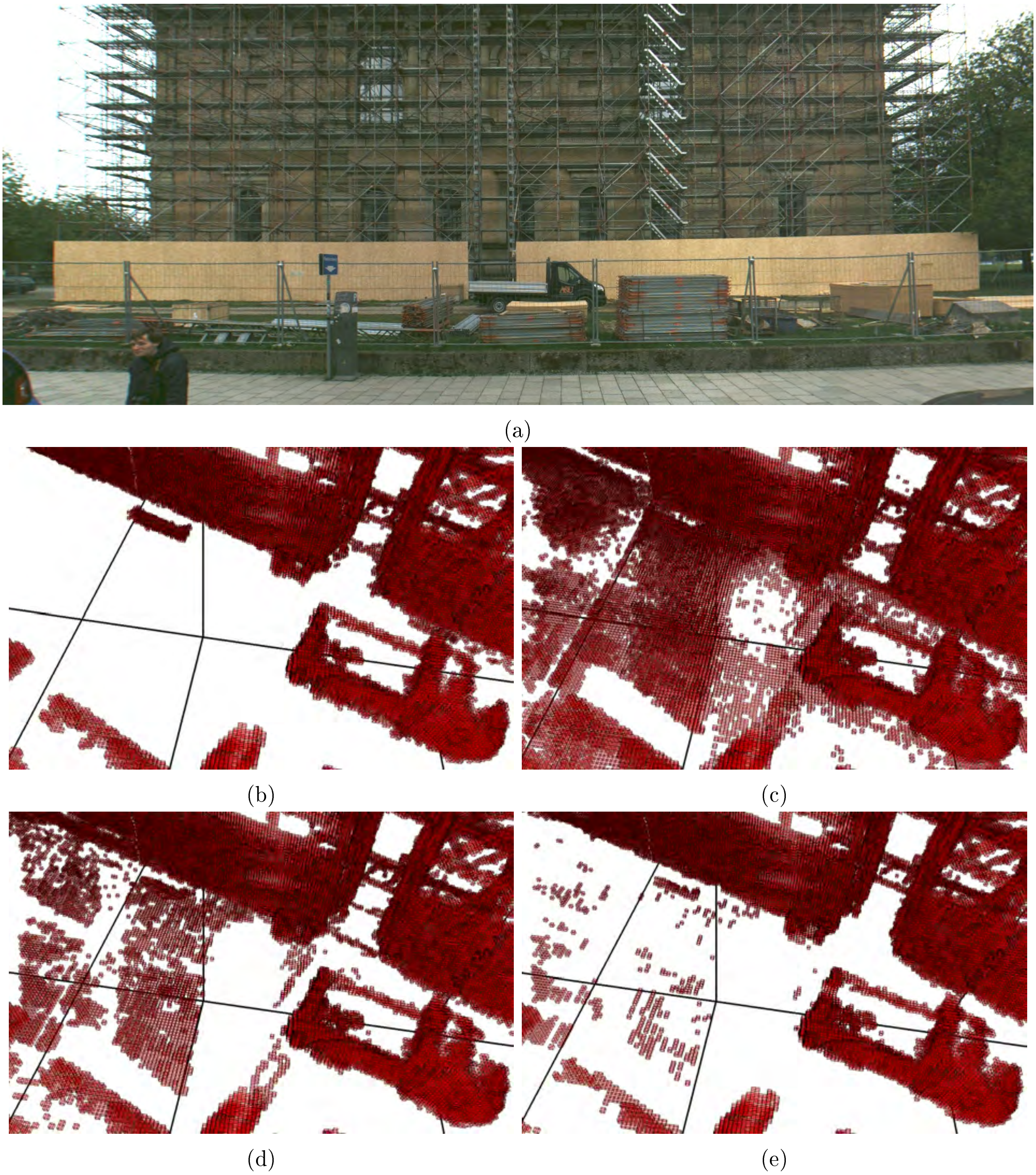


(a) (b) (c)

(d) (e) (f)

Figure 7.11: Change detection results for different max-pooling neighborhoods and defuzzification methods. a,d) Results without max-pooling, b,e) results with a max-pooling neighborhood of one voxel, c,f) results with a max-pooling neighborhood of two voxels. No defuzzification has been applied in the top row, the bottom row has been defuzzified with pro/contra filtering.

Figure 7.12 shows the evidence grid containing the disappeared geometry for all max-pooling neighborhoods surveyed. Although the increase in F1-score is minimal, it can be seen from the figures that max-pooling has a noticeable positive impact. It can also be observed that, contrary to the implications of the F1-score, the size of the neighborhood has a significant impact on the quality of the result. This can be justified by the fact that with a neighborhood of only one voxel, there is a significant amount of clutter in the representation. With a neighborhood of two voxels, on the other hand, this clutter is almost non-existent and the representation is very similar to the



(a)

(b)

(c)

(d)

(e)

Figure 7.12: Influence of max-pooling neighborhoods on the results of the change detection. a) Camera image of the scene, b) fuzzy representation of the ground truth, c) change detection results without max-pooling, d) results with a max-pooling neighborhood of one voxel, e) results with a max-pooling neighborhood of two voxels. The color encodes the degree of certainty: blue=low, green=medium, red=high.

ground truth. The evidence grids for confirmed geometry do not show any clutter, so the effect observed here seems to be present only in case of changes.

7.2.3 Qualitative change detection results

This section presents the qualitative results of the change detection method, which are based on the free parameters determined in the first stage. The change detection results were transferred to

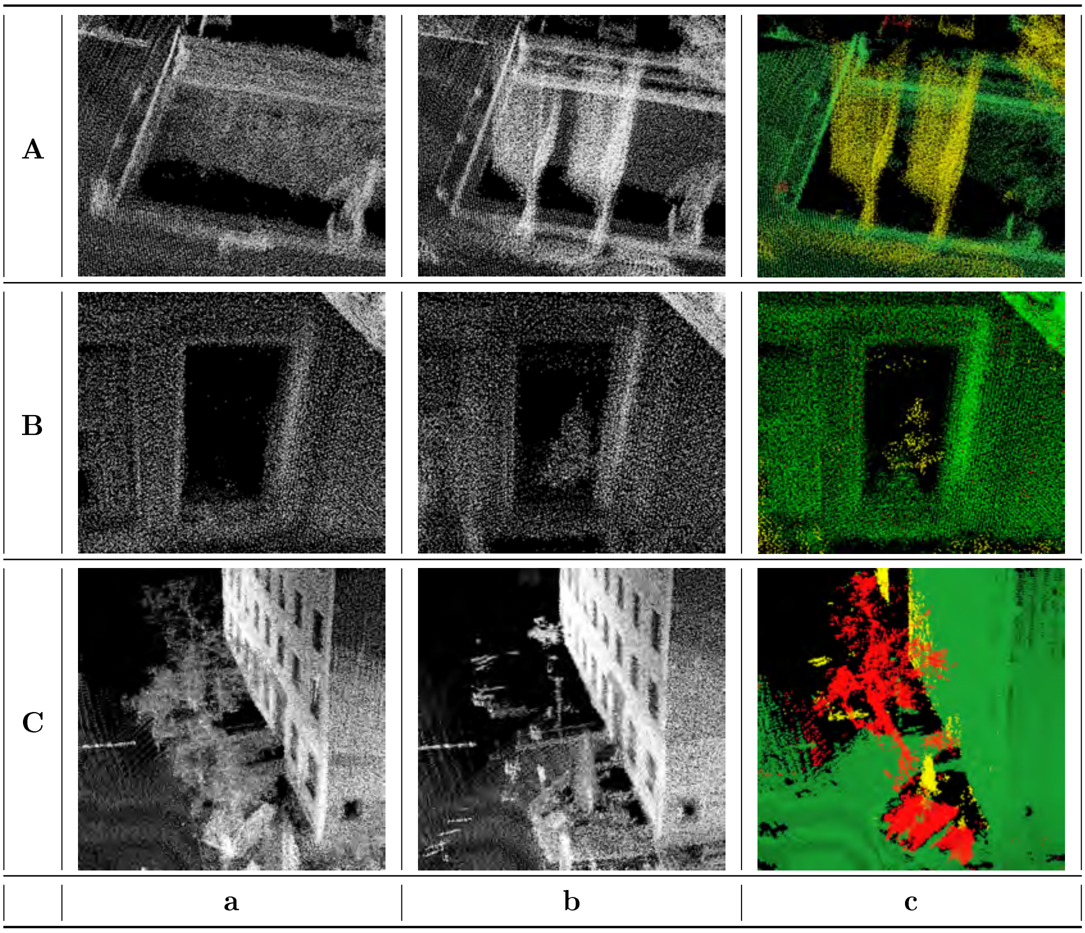


Figure 7.13: Selected changes in the TUM-MLS data set. A) Flagpoles erected on the TUM campus, B) Christmas tree in the window of a residential building, C) a removed tree and container. a) Point cloud of the first epoch, b) point clouds of the second epoch, c) overview of all changes and unseen areas (green=confirmed, red=disappeared, yellow=appeared).

an accumulated point cloud for closer examination. A few selected examples are explained in more detail. The *Alte Pinakothek* shown in Figure 7.16 shows a number of changes concerning various objects related to the street space. This is mainly due to the construction site, which was located there in the first epoch due to renovation work on the building. When the construction site was closed, all related objects such as scaffolding and fences were removed. The facade of the building, which was covered by a construction fence in the first epoch, was correctly marked as unseen in the second epoch, not as changed. The same effect can be seen wherever the wall was covered by the walkways of the scaffolding. Both areas are marked in blue. The chosen example also shows that even inconspicuous changes can be recognized. For example, the curtains in the windows on the lowest floor of the Pinakothek were closed in the second epoch, which is why they are correctly marked as *appeared*. Furthermore, right in front of the site fence, there are two areas in which the ground was leveled, i.e. material was added to mend holes in the ground that probably originated from the construction work. Small and large changes can be seen in the vegetation in the right-hand area of the figure. Here it can be seen that the tree has lost its foliage due to the different seasons. Entire branches have been removed in the lower area, probably due to tree maintenance work. Changes also include larger objects like construction fences, containers, material stacks and a flatbed truck. Changes related to the *vehicles* object class can be found in the lower area of the figure. In roadside parking lots, there are some vehicles that have either changed type,

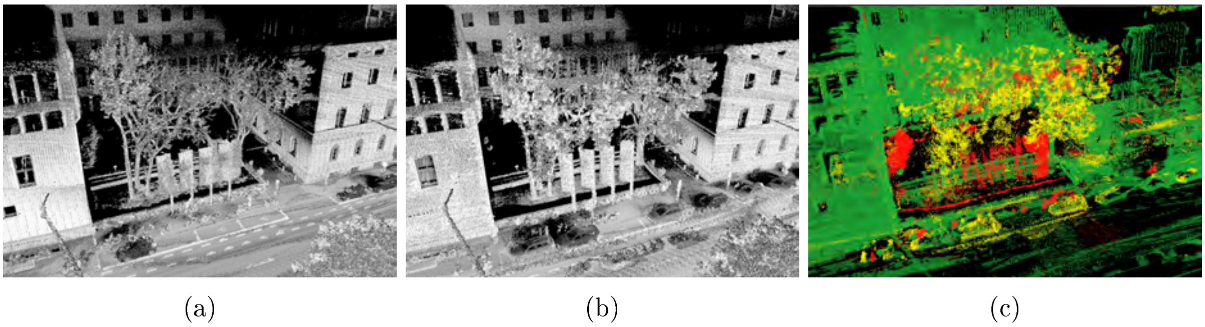


Figure 7.14: Changes in a street environment in the *Arcisstrasse* in Munich. a) Point cloud of the first epoch, b) point clouds of the second epoch, c) overview of all changes and unseen areas (green=confirmed, red=disappeared, yellow=appeared).

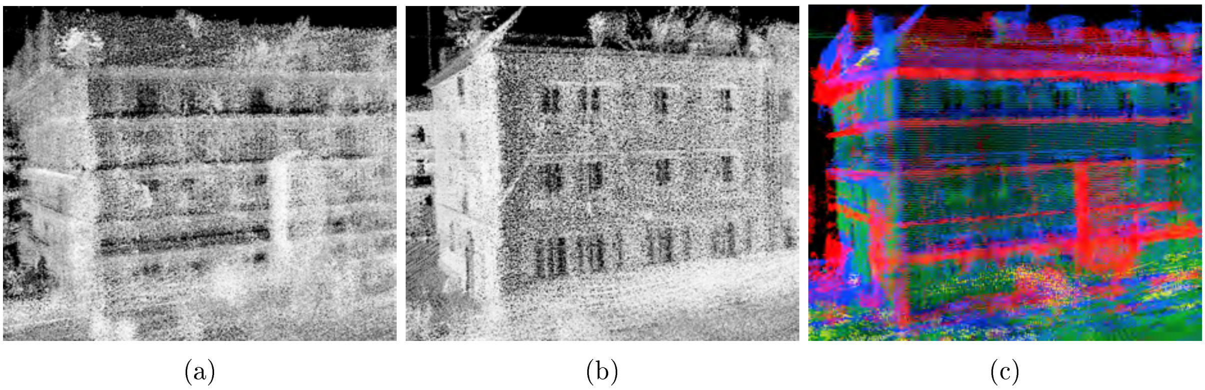


Figure 7.15: Changes due to construction work in the *Gabelsbergerstrasse* in Munich. a) Point cloud of the first epoch, b) point clouds of the second epoch, c) overview of all changes and unseen areas (green=confirmed, red=disappeared, yellow=appeared, blue=unseen in other epoch).

been moved, or completely replaced with another vehicle. Some changes are more pronounced than others because the shape or position of the associated vehicles changed significantly between epochs.

The Arcisstrasse in Figure 7.14 shows how the parking situation has changed between the two epochs. In the second epoch there were many more vehicles parked along the road. The parking situation is so tense that even disabled parking spaces are being used and one driveway is blocked. In addition, changes in the foliage can be recognized due to the different seasons, namely spring and winter. Foliage is much less pronounced in the second epoch, which is recognized as a change.

Figure 7.15 shows a construction site in the Gabelsbergerstrasse that has an effect on the street environment. Here, too, the change detection provides a number of relevant changes that are helpful for the interpretation of the scene. In the first epoch, scaffolding surrounds the house. In addition, one lane of the road is occupied by a large concrete silo and separated from the rest of the street by a construction fence. Since this is an obstacle directly within the street space, traffic was forced to evade and coordinate with oncoming vehicles.

Figure 7.13 shows a selection of changes that were discovered and are worth mentioning. Row 7.13A shows a number of flags on the TUM campus. These appeared in the second epoch and illustrate one of the many challenges of change detection. A flag is a deformable object, and even if the actual flag does not change, the wind can still cause a change to be detected here. Row 7.13B shows an appeared Christmas tree in a window. This change was chosen because it is easy to overlook. 7.13C shows a tree and a container that both disappeared, whereby the former was most likely transferred piece by piece to the latter. This can be deduced from the fact that in the

	Confirmed (A & B)			Appeared			Disappeared		
Subset	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
Full data set	0.98/0.97	0.28/0.34	0.43/0.50	0.65	0.27	0.38	0.74	0.50	0.60
Alte Pinakothek subset	0.94/0.89	0.92/0.82	0.93/0.85	0.85	0.19	0.30	0.98	0.81	0.89
Natural terrain	0.96/0.87	0.89/0.75	0.93/0.81	0.24	0.68	0.36	-	-	-
Building facade	0.96/0.90	0.93/0.82	0.94/0.86	-	-	-	0.98	0.75	0.85
Hardscape	0.25/0.98	0.31/0.87	0.28/0.92	0.31	0.41	0.35	0.97	0.88	0.92
Scanning artifacts	0.89/0.86	0.87/0.91	0.88/0.89	0.75	0.13	0.22	0.95	0.71	0.81
Vehicles	-	-	-	0.95	0.14	0.25	0.95	0.42	0.58

Table 7.2: Change detection results for the full data set and the *Alte Pinakothek* subset, subdivided by object classes.

second epoch the tree was replaced by a new one. It also shows a person leaning against the wall in the house entrance in the right center of the image, which was also initially overlooked when the annotation was created.

7.2.4 Quantitative change detection results

This section presents the quantitative results of the change detection method, which are based on the free parameters determined in the first stage. These include applying the *occupancy membership function* with *tile-based computation mode*. A max-pooling neighborhood of one voxel for confirmed geometry and two voxels for modified geometry was used. The results were defuzzified before evaluation.

Results for the full data set and subset

Table 7.2 shows the change detection results for both the full data set and the *Alte Pinakothek* subset. The results for the entire data set show that the two cases *confirmed* and *disappeared* are roughly comparable, with F1 values of 0.50 and 0.60, respectively. The case *appeared* has a significantly lower F1-score of 0.38. In all three cases, the results have a low recall, which in turn leads to a low F1-score. A point cloud of the results shows that there is a lot of *confirmed* and *disappeared* geometry in the data set, but hardly any *appeared* geometry. A correlation is therefore present between the number of samples per case and the F1-score.

The *Alte Pinakothek* data set was chosen to contain most of the changes in the data set. The same correlation between *disappeared*, *appeared* and *confirmed* geometry and the associated F1-scores is evident here. The F1-scores for *confirmed* are 0.93 and 0.85, the F1-score for *disappeared* is 0.89. Appeared has a low recall of 0.19, which is reflected in the also low F1-score of 0.30. As with the complete data set, there is much less *appeared* geometry than *confirmed* and *disappeared* geometry.

Results subdivided by object classes

Table 7.2 shows the results of the *Alte Pinakothek* subset, divided by object class. Since *vegetation* has been removed and there are no changes for *building interior* and *manmade terrain*, these are not listed in the table. Furthermore, for some classes there is no *confirmed*, *appeared* or *confirmed* geometry. The *disappeared* geometry in the subset includes:

❑ **Hardscape:** street signs, construction fence around the Pinakothek

❑ **Building facade:** scaffolding

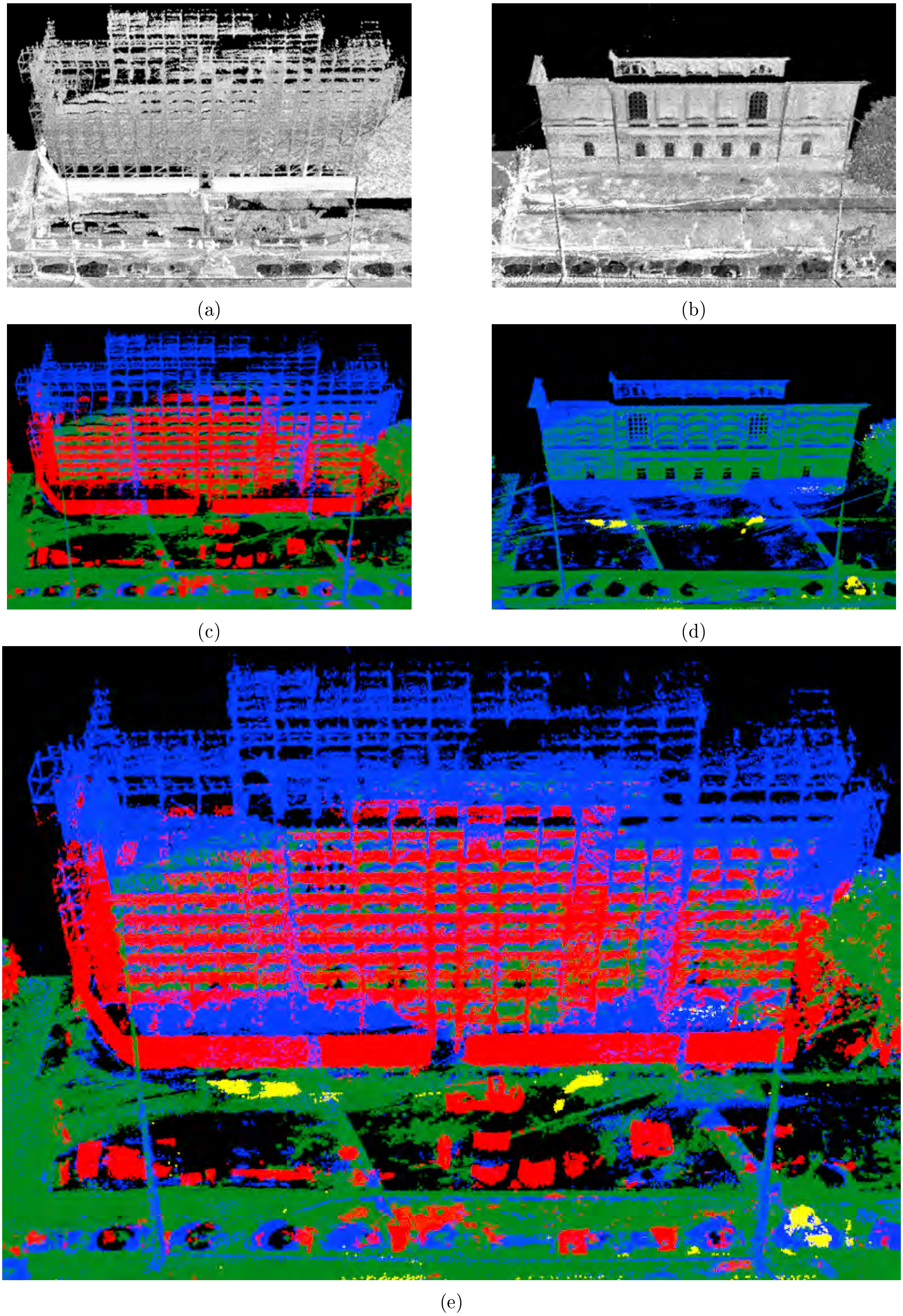
❑ **Vehicles:** construction vehicle, vehicles on the road

- **Scanning artefacts:** container, street sign

The *appeared* geometry includes:

- **Scanning artefacts:** street signs
- **Building facade:** curtains in the windows
- **Natural terrain:** filled ground in front of the Pinakothek

The above changes are also illustrated in Figure 7.16. Some of the changes are large in area and therefore represented by more samples than smaller changes. This is reflected in the corresponding F1-scores in Table 7.2. *Building facade* has a high F1-score of 0.85 for *disappeared* because a large scaffolding along the facade of the Pinakothek has disappeared. A similar behavior can be observed for *hardscape*, where a large construction fence disappeared and the F1-score is 0.92. In the case of smaller changes, as it is the case with *appeared* geometry in *natural terrain* and *scanning artefacts*, correspondingly lower F1-scores of 0.36 and 0.22 are observed. The values for precision and recall for the *vehicle* class can be explained by the fact that each vehicle that was moved or replaced by another was annotated in its entirety. *Confirmed* geometry does not occur for *vehicles* because every vehicle in the data set was subject to change in the two years between the recording of the epochs.



(a)

(b)

(c)

(d)

(e)

Figure 7.16: Changes caused by renovation work on the *Alte Pinakothek* in Munich. a,b) Point clouds of both epochs, c,d) disappeared and appeared elements, e) overview of changes and unseen areas of both epochs (green=confirmed, red=disappeared, yellow=appeared, blue=unseen in other epoch).

8 Discussion

In this chapter, the three research questions are addressed and discussed in light of the results. The main points are the capability of fuzzy representations to represent information, the suitability of fuzzy spatial reasoning for change detection and the assessment of the metrics used for evaluation.

8.1 Discussion of the fuzzy spatial representation

The results of the experiments in Section 7.1 demonstrate that the fuzzy spatial representation presented in this work can successfully be applied to represent an urban street space scene. The individual factors influencing representability are discussed in detail in the following sections.

8.1.1 Membership function

The results presented in Section 7.1.2 show that the linear membership function makes better use of the entire range of possible membership values. This implies that it introduces little to no additional prior knowledge, since it has no visible bias towards a particular range of values. The identical shape of the sample and membership histograms in Figure 7.3 illustrates this. The logistic function, on the other hand, does not use the complete bandwidth of the interval, almost only the part of the interval close to the two extreme values. Compared to the linear function, additional prior knowledge is introduced here. The approximately linear shape of the plot of the logistic membership function indicates that the error grows logistically with resolution and is compensated for by the use of said membership function. It can be argued that by favoring extreme values, noise is reduced without sacrificing possible intermediate values.

This focus on extreme values can also be seen as the reason for the peak that appeared in Figure 7.2, which is discussed in detail in Section 8.1.4. The comparison of the membership functions leads to the conclusion that *the logistic function is to be preferred over the linear one.* The main reason for this is that applying the logistic function leads to a lower average error per voxel. The extreme values preferred by the function reduce the work of defuzzfication, as this leads to a higher contrast between classes and thus reduces noise. The linear function, on the other hand, has a higher average error, which is due to the many intermediate steps. The logistic function introduces additional knowledge through which the information can be presented with greater contrast, but without having to give up the aspect of fuzziness.

8.1.2 Memory consumption and recommended resolution

The finer the resolution of the spatial representation, the higher the required memory consumption. Each resolution level that is omitted leads to a large reduction in the required memory. However, the total memory consumption is negligible, averaging around 2 MiB per tile. The main advantage of omitting one resolution level is the reduction of the overall computing time, since creating an occupancy grid is an extremely computationally intensive operation due to the traversion of free space.

The choice of resolution levels depends on whether a defuzzification is intended. If this is the case, any resolution level that can fulfill the task at hand can be selected. Sharpening the data removes noise and effects such as the aforementioned peak, of course blurring effects due to discretization errors remain. If no defuzzification is intended, then it is recommended to consider the error measure when choosing the resolution level. However, it is important to consider how representation and ground truth interact. For the present data set of an urban street space, the maximum resolution of 0.2 m is equal to the ground truth resolution. This creates the impression that said resolution level allows an almost flawless representation, because only one ground truth sample coincides with one voxel each. In fact, however, the error measure has only limited significance here due to the circumstances mentioned.

A resolution level cannot be recommended in general. A voxel must be large enough to enclose several measurement points and thus make the evidence statistically significant. At the same time, however, it should be small enough so that not too much geometry is combined. It depends on the structure of the environment, the application and, of course, the resolution of the sensor system. If memory or computational effort is to be saved, a rule of thumb is to select the appropriate resolution using the precision/recall curves. This includes analyzing all plots and then selecting the resolution for which all plots have an acceptable *precision*. For the plots presented here, good results are obtained for a resolution of 0.8 m. Results that are still acceptable are obtained for a resolution of 1.6 m. The highest resolution studied was 0.2 m, which is a good compromise between spatial resolution and runtime for an urban environment.

8.1.3 Properties favorable to representability

The quality of the representation of a spatial information depends on several factors. It is best if the corresponding class has been well observed, i.e. the recorded samples have a high spatial density. It is also advantageous if a class is isolated from other classes and has few if any interfaces with them. The quality of representation is worst if a class has been poorly observed, i.e. only a few samples have been recorded for it and they are widely distributed spatially. This occurs especially with elongated or narrow objects. Many interfaces to other classes are also unfavorable. All these factors lead to the conclusion that discretization errors have a high influence, which negatively affect the quality of representation.

The above mentioned conclusions have been confirmed experimentally, e.g. by the two groups identified in Figure 7.6. The histograms of the certainty and error distributions in Figure 7.4 and the average errors of the individual object class in Figure 7.5 also support that hypothesis. Despite many points of contact the object classes *high vegetation*, *building facades* and *manmade terrain* have only a small peak in the error measure, which supports the hypothesis that there is a connection between high point density and strong evidence. The object classes *low vegetation* and *building interior* show a very low error, although they are represented by only a few samples. However, they have little contact with other classes. This supports the hypothesis that spatial distance between classes is conducive to representation.

It can be summarized that the quality of the representation is best whenever there is a low overlap between the represented information as well as a high amount of samples. If this is not the case, a moderate or even high overlap can be tolerated *if the class is represented by enough samples.* On the other hand, if the overlap is high and the number of samples is insufficient, the information is only poorly represented.

8.1.4 Expressiveness of the fuzzy representation

The peak in the error measure in Figure 7.2 indicates a possible limitation in the ability of the representation to describe information. However, since Figure 7.6 no longer shows such an effect after defuzzification, it can be concluded that it does not represent a significant limitation of the representation's descriptiveness. The peak can be explained by the fact that large voxels are more likely to get a significant number of measurements. This results in enough measurements to completely fill the interval within the evidence distribution. This is shown by the histograms in Figure 7.4. For smaller voxels, the probability is correspondingly smaller and not enough measurements end up in a voxel, which means that the interval of the evidence distribution is less filled. The effect is therefore caused by the *law of large numbers*. This effect is counteracted by the effect of blurring, that is, the aggregation of information due to the decreasing voxel size. This explains the subsequent increase in the error curve. It can be assumed that the effects shown in Figure 7.2 are not specific to the data set, but also occurs in other scenarios, since it is essentially caused by the discretization of space.

The *precision/recall curves* show that even for the coarsest resolution level there is still a clear difference to random guessing. This can be attributed to the fact that all data points are above the *no-skill line*. This implies that the evidence distribution represents the signal clearly enough so that it can still be distinguished from noise even at the coarsest resolution level. This finding underscores the value of the fuzzy evidence representation. From an application perspective, this allows coarser resolution levels to be used as indicators. A coarse resolution level can be computed many times faster than the investigated maximum resolution of 0.2 m, but is still expressive enough to give indications of e.g. changes in the represented volume of space.

In Figure 7.8, a trend can be seen that the proportion of false positives decreases with increasing resolution. This shows that the data is approximated better with increasing resolution. This trend is not always clearly pronounced, but this only shows that the spatial resolution is not the only relevant factor for the representation of the data. This is further supported by the *false negatives* shown in Figure 7.9 and the fact that all precision/recall curves are above the no-skill line.

Given the above conclusions, it is deducted that the fuzzy representation can be used not only in the general case for the representation of spatial information, but also for the special case of an urban environment.

8.1.5 Effects and necessity of defuzzification

Fuzzy gradation of information is relevant, since it takes into account the vagueness of information. However, decision-making requires sharp information, which in turn is generated by applying a defuzzification method to the fuzzy information. As Figure 7.2 demonstrates, a direct evaluation of the fuzzy information for high resolution levels leads to a higher error than for some of the coarser resolution, which is expressed by a peak in the average error measure. Since the peak in the F1-score in Figure 7.6 is not present after defuzzification, this suggests that the defuzzification was able to compensate for the noise in the certainty responsible for the peak.

The hard distinction made by the pro/contra filter, i.e. $m(A) > m(\bar{A})$, ensures that the decision is no longer made based on the quantity of the evidence, as in the case with the error measure, but on the ratio instead. This allows for more reliable result in poorly observed areas. In general, a defuzzified representation has the advantage over the fuzzy one that only the geometric differences between it and the ground truth remain. Defuzzification thus suppresses the noise and brings out the actual result more clearly.

8.1.6 Effects of discretization errors

Discretization errors are negligible for sufficiently fine resolution levels. As mentioned above, classes with few measurements, classes with a small expansion or a coarse resolution are problematic. Regarding *false positives* it can be noted that their number decreases with increasing resolution. Also, their characteristics depend on the represented object class. If there are enough samples for a class, little to none false positives occur in the classes marginal areas. However, this means more errors with other classes. The class to be represented should therefore always be supported with many samples and therefore measured prominently. Both *false positives* and *false negatives* do not occur for correspondingly fine resolutions and are therefore negligible.

8.1.7 Inferability of scene knowledge from false positives

Although not directly relevant to the present evaluation, it should be mentioned that the confusion of classes illustrated in Figure 7.8 can be used for a data-driven semantic interpretation of the scene. The extent of the false positives can be considered as a measure of the overlap between classes and therefore reflects where such classes are spatially close to each other. These statistical correlations represent possible semantic relationships between classes. The ones discovered during the evaluation reflect the typical correlations for an urban environment. For example, the class *building facade* is in relation with the classes *building interior* and *manmade terrain*. Also, the classes *vehicles* and *scanning artifacts* are related to *manmade terrain*, which is not surprising since vehicles and road signs occur in the context of roads. As can be clearly seen, this accurately describes the semantic relationships between object classes that occur in an urban environment.

8.2 Discussion of fuzzy reasoning and change detection

The results of the experiments in Section 7.2 demonstrate that geometric changes can be detected, provided that a max-pooled variant of the fuzzy occupancy grids is used to compensate for any remaining residual errors of the measurements. The findings are discussed in more detail in the following sections.

8.2.1 Membership function and computation mode

As noted in Section 8.1.1, the *logistic membership function* is more suitable than the *linear membership function*. However, Figure 7.10 shows that the results of the *occupancy membership function* developed specifically for generating occupancy grids are better than those of the *logistic membership function*. This is due to the deliberate suppression of the evidence for free space in presence of evidence for occupied space.

Figure 7.10 leads to the conclusion that a tile-based computation mode creates the best results. Using the other two modes results in a higher uncertainty. The reason for the poor performance of said modes is that the median is determined only on a very small sample size and therefore each value is close to the median. As a result, the confidence of the resulting occupancy grid is located around the middle of the interval. Also, it is far less stable from a statistical point of view. As a result, the *occupancy membership function* and the *tile-based computation mode* are recommended to create the occupancy grids that are needed for change detection.

8.2.2 Influence of defuzzification

The results of the defuzzification experiment illustrated in Figure 7.11 imply that defuzzification improves the change detection results. This is consistent with the findings in Section 8.1.5. The fuzziness of the data can be seen as the main reason for the discrepancy between the results and ground truth. An inspection of the evidence grids reveals that the crisp ground truth uses a much narrower interval than the fuzzy change detection results. The defuzzification of the results leads to the fact that results and ground truth both are in identical sections of the interval. This means that all remaining errors are no longer due to a different expression of the evidences, but rather to geometric deviations, i.e. real differences between results and ground truth. In terms of change detection, the deviations in the evidence mass are irrelevant, only the actual differences from the ground truth are meaningful. From this point of view, defuzzification helps to limit the result to the relevant parts, i.e. to separate signal from noise.

8.2.3 Determination of the max-pooling neighborhood

The experimental results for determining the max-pooling neighborhood illustrated in Figure 7.11 and Table 7.1 demonstrate that applying max-pooling leads to an increase in the quality of the change detection results. The visualizations of the associated evidence grids in Figure 7.12 confirms this. Using only the F1-score as an evaluation criterion, one can conclude that the actual number of neighbors of the max-pooling operation does not seem to matter, since the results do not improve significantly with a larger neighborhood. Furthermore, one can conclude that a neighborhood of one voxel is preferable due to the lower computational effort. A neighborhood of two voxels is more expensive and only leads to comparatively similar results.

However, the above-mentioned visualization of the evidence grid in Figure 7.12 clearly shows that a higher number of neighbors results in less clutter. This means there are fewer geometric errors in the form of false positives. Two conclusions can be drawn from this. First, a max-pooling neighborhood of two voxels is preferable, since the changes found by the method are more likely to correspond to the actual changes. This does not apply to *confirmed* geometry, since a max-pooling proximity of one voxel seems to be sufficient here. The second conclusion is that the confusion matrix has limited usefulness as a tool for evaluation. This is discussed comprehensively in Section 8.3.2.

Note that the size of the max-pooling neighborhood is related to the residual registration errors. The reason why the neighborhoods were determined to the above extent is that the remaining residual errors are in the range of one to two voxels. Thus, the statement made here about the size of the neighborhood cannot be generalized to arbitrary other data sets.

8.2.4 Precision-recall trade-off

The main factor influencing the quality of the results is the neighborhood size of the max-pooling. It determines how much the representation is blurred. Figure 7.11 shows a trend related to the size of the neighborhood. If it increases, the representation becomes more blurred, which decreases the precision, but increases the recall. In the case of *precision*, this can be explained by the fact that the number of false positives increases with the increase of the occupied volumes. This prevents the negative effect of minor registration errors. In case of *recall*, the number of false negatives decreases. This also is an effect of the increase in occupied volume. This effect is here referred to as the *trade-off between precision and recall* and depends on the size of the max-pooling-neighborhood. It allows fine-tuning in terms of the sensitivity of the approach and the reliability of the changes detected.

8.2.5 Discussion of qualitative change detection results

A demonstration of how change detection can be used to detect traffic situations affecting the road space is shown in Figure 7.14. This shows an example of parking behavior in the Arcisstrasse.

Based on the changes, it is possible to see how the demand for parking spaces increased between the first and second epoch. In the second epoch, there are far more vehicles, which even block a driveway and a handicapped parking space. The figure also shows the ability of the approach to determine changes in vegetation. The changes in foliage, which can be explained by the difference between winter and summer, are clearly visible.

Another situation, which has a direct impact on the street space, can be found in the Gabelsbergerstrasse example in Figure 7.15. Based on the changes, a construction site present there in the first epoch can be identified. Since this is an obstacle directly in the street space, traffic is forced to evade and coordinate with oncoming vehicles. Information like this can form the foundation for other applications, such as evaluating the passability of a road for heavy transport.

Specific properties of the change detection approach presented here can be illustrated by the changes shown in Figure 7.13. The Christmas tree can be used to demonstrate that small and inconspicuous changes can easily be overlooked. Even if such changes are detected by a human observer, there is still the possibility that they will be subconsciously discarded as irrelevant. For similar reasons, the scenario in row 7.13C was chosen. In the right part of the picture a person is leaning against the wall in a house entrance. This is difficult to see in the point cloud and therefore was not detected during annotation. This is an informative example of how LiDAR data cannot always be reliably interpreted by a human annotator.

8.2.6 Comparison of the quantitative results for full data set and subset

The correlation between the number of samples and the F1-score for the cases *confirmed*, *appeared*, and *disappeared* is evident in both the full data set and the *Alte Pinakothek* subset. Since the changes found in both cases can be described as comprehensive and well-defined from a qualitative point of view, this suggests that the error measure only provides meaningful results on *statistically representative* data. The *Alte Pinakothek* subset contains much *confirmed* and *disappeared* geometry, but little *appeared*. Based on the above conclusion and the consideration of the qualitative results, it can be concluded that the F1-scores for *confirmed* and *disappeared* roughly correspond to the actual circumstances, but the F1-score for the case *appeared* underestimates the actual circumstances. The same is true for the full data set. The examination of the changes in relation to the individual object classes has shown that object classes represented by many samples have a higher F1-score than object classes represented by few samples. This confirms the conclusion that the data set must be statistically representative for the confusion matrix based metric to produce meaningful results.

8.2.7 Systematic effect of the sensor data evaluation concept

A close inspection of Figure 7.16e reveals an area that does not behave as expected. The upper part of the scaffold is marked as *unseen* instead of *changed*, although the corresponding area was in the focus of the sensors in both epochs. This can be said with certainty, because the *Alte Pinakothek* was recorded in both epochs driving by on the street directly in front of it. The explanation for this is that the occupancy grids on which change detection operates are based on *surface measurements*. This implies that the complete free space contained in the data set is located only in the area between a surface point and the associated sensor position. Furthermore, this means that the free space is only captured if there is a *surface in the background*.

Free space information is required to distinguish between changes and occlusions. A change can only be inferred with confidence if it is actually known whether there was free space in the other epoch. If this is not known, an automated procedure can only mark the corresponding area as *unseen*, otherwise there would be the possibility of a false positive. This is especially relevant

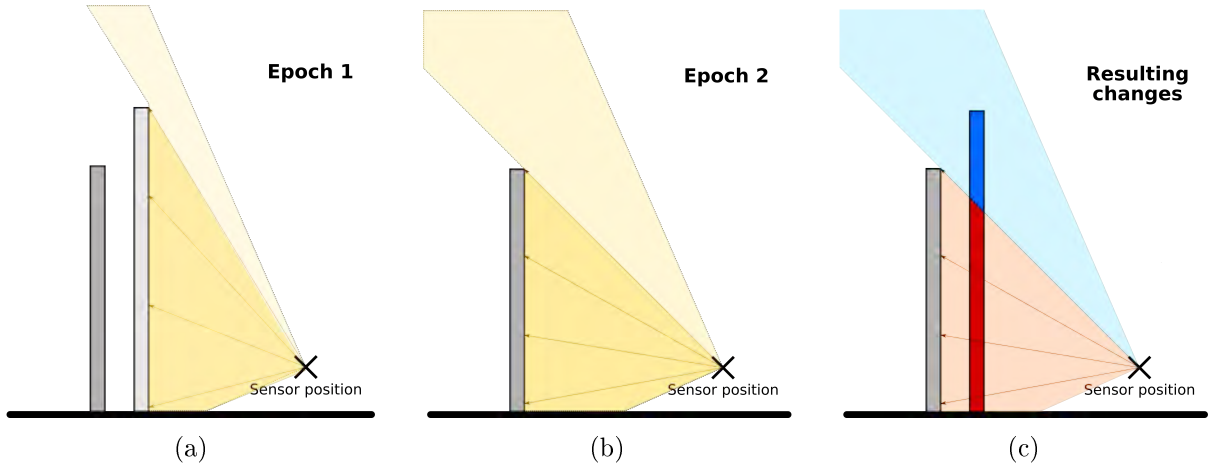


Figure 8.1: Example to illustrate the apparently missing observations. a) An object is measured in the first epoch, b) the object is gone in the second epoch, c) illustration of the resulting blind area (blue) when free space (red) is only derived from surface measurements. Only the red part of the completely disappeared object is recognized as a change, but not the blue part. Fields of view are marked in yellow.

in urban environments, because here the limited space is used as efficiently as possible, resulting in many occlusions. The effect caused by the fact that free space is only captured in the context of surfaces is illustrated in Figure 8.1.

The effect described here is of practical relevance, since it always occurs when *changed* geometry is measured *against the sky or a free surface*. It is possible to prevent the effect by storing the direction vector for each measurement. The free space can then be extrapolated from said vector and an assumed maximum sensor range. Unfortunately, it is not possible to subsequently apply the solutions discussed here to the available data. The direction information of each measurement was lost when the latter were transformed into a point cloud. For future generations of measuring systems it is recommended to keep the direction vector. Furthermore, it is recommended to consider the effect discussed here for all change detection methods that utilize free space, otherwise relevant changes cannot be detected.

8.3 Discussion of the evaluation metrics

The results in Sections 7.1 and 7.2 provide empirical evidence about the performance of the used metrics. This made it possible to work out their characteristics and limitations. The three metrics examined in this work are discussed in detail in the following sections.

8.3.1 Discussion of the error measure

The error measure described in Section 5.1 was designed to determine the degree of deviation between a non-defuzzified representation and a ground truth. As Section 7.1 demonstrates, the error measure can quantify the deviation from the ground truth with a degree of accuracy that enabled to detect the interaction of two effects, which were verified by the histograms in Figure 7.4. Said effects are explained in detail in Section 7.2.

An assessment of the error measure with the empirical data gathered during the experiments was performed. Geometric errors and errors due to vagueness are both intermixed by the error measure. The influence of defuzzification determined in the last two main sections suggests that errors due to vagueness are a notable influence. This leads to the conclusion that it should only be used when this effect is desired, i.e. for the comparison of a non-defuzzified representation and a ground truth, as was the case in Section 7.1.

The experiment underlying Section 7.2 has shown that the fuzziness of the information has a visible impact on the accuracy of the represented information. This influence would therefore have to be removed by means of defuzzification in order to be able to use the error measure for change detection, where only the geometric deviations are relevant. However, as argued in the next section, the informativeness of the confusion matrix is higher than that of the error measure.

The data also shows that the error measure has limited explanatory value as soon as ground truth and representation exist on a grid with the same resolution. This case is present in Figure 7.2 for the highest resolution level. As expected, the error here tends to zero because the representation exactly reflects the ground truth it is created from. However, small deviations, such as those caused by the membership function, are lost in the noise.

8.3.2 Discussion of classical and fuzzy confusion matrix

The use of the classical confusion matrix is motivated by the fact that change detection requires an error measure that evaluates only the crisp geometric deviations between representation and ground truth. From a theoretical point of view, this task can also be performed by the error measure discussed in the last section. However, as confirmed experimentally, the confusion matrix produces more meaningful results than the error measure. This can be inferred from Section 7.1.4. The precision/recall diagrams (derived from a confusion matrix) shown in Figure 7.7 demonstrate clearly that the data point for each resolution is above the *no-skill line*. As explained in Section 8.1.4, this suggests that even coarse resolution levels still have a good enough signal-to-noise ratio to enable conclusions to be drawn that are better than random guessing. Such a finding cannot be derived from the error measure alone.

When an information is inserted into a fuzzy representation, a systematic error is incorporated into it. This is the motivation for extending the *classical confusion matrix* to the *fuzzy confusion matrix*. When comparing a fine resolution level of the representation with the ground truth, this is not particularly noticeable, however, with a coarse resolution level, it is even more so. Since said systematic error cannot be removed, the only way to have a meaningful comparison is to apply the same systematic error to the ground truth. This is done by inserting the latter into a fuzzy representation. That the fuzzy confusion matrix meets the requirements can be inferred from the values in Table 7.1, since the metric results shown there match the circumstance shown in Figure 7.16.

Both the classical and fuzzy confusion matrices have the disadvantage that the measures derived from them are no longer meaningful if the underlying data is not statistically representative. This is shown by the fact that there is a clear difference between the numerical results in Table 7.1 and the visualization of the corresponding scene in Figure 7.12. The F1-scores for the max-pooling neighborhoods of one and two voxels barely differ, but the visualization of the former shows large patches of clutter, while the visualization of the latter doesn't. The reason for this is that qualitative aspects like the above mentioned clutter are lost in the large amount of data points. A purely quantitative error measure only identifies an issue if it is also statistically representative. This is a limitation, as it only permits a meaningful evaluation of changes that are represented by a large number of samples. This is evident in Section 7.2.4, as the results on the whole data set appear to be worse than their visualization, but on the statistically representative *Alte Pinakothek* data set they match the visualization.

Thus, the explanatory power of the two metrics based on the confusion matrices is limited. In terms of their usage, this means that they should always be applied in conjunction with a visualization or other metrics. After all, this is necessary to identify situations in which the metric underestimates the actual quality of the results. One possible solution would be to include semantic clusters to emphasize the qualitative aspect to a greater extent. However, semantics is dependent on a specific application and therefore such an approach does not qualify as a generic metric.

9 Conclusions and outlook

The purpose of this final chapter is twofold - to present the most important conclusions resulting from the work carried out within this thesis and to outline possible directions for further research that address limitations of the presented methods. The conclusions are grouped according to their relationships to specific objectives pursued within the work, as well as to the three research questions posed in Section 1.3.

9.1 Conclusion

Research question I: To what degree can the influences of vagueness and contradiction in information derived from mobile LiDAR measurement data be reduced and what properties of method and data support this?

The first objective of this thesis was to develop a method for spatial representation that allows handling vague and contradiction information, that allows to make a statement about the degree of ignorance and provides an inference mechanism. The latter is required to perform the actual change detection, the former to handle the effects of the error sources and conflicting evidence contained within the measured data. For clarity, the answer to this question is divided into three parts.

The first part of the answer deals with the extent to which the vagueness of information can be resolved. The main sources of error in the generic evidence grid developed for change detection are discretization errors. Experiments show that this leads to two overlapping effects in the fuzzy representation. The first effect is negligible as soon as the information is defuzzified, the second effect is the smearing of information common for voxels. Using the latter, the influence of vagueness can be determined, that is, by looking at the different resolution levels, which artificially introduce vagueness into the system. The vagueness can be quantified via the F1-score by comparing the object classes inserted into fuzzy representations with the original data. For all object classes, the F1 score for a resolution of 0.4 m is greater than 0.95. For a resolution of 0.8 m, the F1-score is still greater than 0.80 and even for a coarse resolution of 1.6 m, it is still greater than 0.80 for object classes represented with many samples. Furthermore, it was experimentally determined that even for the coarsest resolution level of 25.6 m, the signal-to-noise ratio is still sufficiently good that the information contained in the representation is better than random guessing. This is particularly true for object classes that are represented with statistical significance. For less significant represented object classes, this effect is less pronounced, although still present. In summary, the fuzzy evidence grid can compensate for vagueness to the extent that it is not only sufficient for change detection, but also coarse resolution levels can be used as an indicator for changes.

The second part of the answer deals with the extent to which the contradictions in an information can be resolved and which characteristics of the method have an influence on the results. Regarding the resolution of contradictions, it must be emphasized that they can only be resolved where the evidence for the information actually present is predominant. By applying the logistic membership function, the evidence was shifted towards the extreme values, which simultaneously reduces the possibility for a contradiction. The occupancy membership function, specifically designed for the creation of occupancy grids, reduces the presence of contradictions by suppressing part of the evidence responsible for it. In case of change detection, this alone greatly reduces the possibility for contradictions. By applying defuzzification, all contradictions are resolved where the evidence for the information actually present prevails. The change detection experiments have shown that omitting the defuzzification of the data and thus resolving vagueness and inconsistencies has a clearly visible impact on the derived F1-score of the results. Hence, the F1-score for confirmed geometry decreases from 0.94 to 0.62 and for modified geometry from 0.89 to 0.71 when defuzzification is omitted. Defuzzification is considered the most important influencing factor for resolving both vagueness and contradictions. In summary, all contradictions are resolved when the appropriate information is available and defuzzification is applied. In practice, this means that almost all contradictions have been resolved.

The third part of the answer explains which properties of the data have an influence on their representability. The experiments conducted on the representability of individual object classes suggest that information are well represented whenever it is supported by many samples and when there is a clear spatial separation of areas with evidence for and against the information at hand. In terms of object classes, this means that spatial separation from other classes is advantageous. This is supported, among other things, by the fact that semantic scene knowledge regarding the context of object classes can be derived from false positives. The results further suggest that it is already sufficient if only one of the two mentioned factors is present for the information to be well represented. However, a high number of samples, i.e. statistical significance, seems to be generally advantageous and the dominating factor.

Research question II: With what accuracy is it possible to automatically detect changes in an urban street space based on vague, contradicting and spatially blurred mobile mapping point clouds and does it allow the use of the method in the context of applications such as city surveys and security applications, either supportive, or fully automatic?

The top-level goal of this thesis was to develop a change detection algorithm capable of handling the vagueness and contradictions that are part of the point clouds acquired by a mobile mapping system. The characteristics of the method in terms of compensating vague and contradictory information have already been answered in the last research question, which is why this question is limited to the capability of the method to detect changes. Again, the answer to this research question is divided into three parts.

The first part of the answer is focused on the accuracy of the change detection method. A quantitative assessment of the results was performed using the F1-score derived from a fuzzy confusion matrix. For *confirmed* geometry, the F1-score is 0.94, and for *changed* geometry, the F1-score is 0.89. The quantitative results were determined on a data subset, since the underlying confusion matrix only reliably reflects the situation if the changes are statistically significant. It should be noted that the *quantitative* evaluation outside of this test data set is limited, although this is primarily due to the metric used. The use of max-pooling to address the challenge of spatially blurred information increased the F1-score for *confirmed* geometry from 0.78 to 0.94 and for *changed* geometry from 0.71 to 0.89. In terms of *confirmed* geometry, this has led to a significant increase in recall; in terms of *changed* geometry, it has led to a significant increase in both precision and recall. Hence, the use of max-pooling has a significant impact on the improvement of the results. From this, it can also be deduced that the residual error on the data remaining after boresight calibration and SLAM-based intra- and inter-epoch registration is less than 0.1 m.

A qualitative review of the entire data set showed that, with few exceptions, the elements of the environment that were considered to have changed were reliably detected. False positives occurred in the form of noise at random locations in the data set and specifically at locations with significant registration mismatches. When there was a change, it was usually detected not only partially, but almost completely. False negatives usually occur at contact points between a changed object and a confirmed object such as the road plane. These usually occur due to discretization errors. Despite the occurrence of false positives, false negatives and an occasional noise, the results give the impression of consistency. Not least because even subtle changes like drawn curtains, people in building entrances, and backfilled holes in a construction site are detected.

The second part of the answer elaborates on the previous statement about the accuracy of the results with respect to the classes of objects found in an urban environment. Quantitative evaluation is not always possible here due to the aforementioned limitations of the metrics used for evaluation, which is also the subject of the next research question. However, in addition to qualitative statements, corresponding conclusions can also be derived from the experiments regarding spatial representability, which in turn are supported quantitatively. It was quantitatively confirmed that changes and confirmations of planar surfaces such as *manmade terrain*, *natural terrain*, *hardscape* and *building facade* can be reliably detected. Traffic signs are challenging because of their small size, which is why the number of samples per voxel and thus the accumulated evidence is rather weak. It has been experimentally confirmed that these factors have a negative impact on representability. However, there are not enough of such changes in the data set, which is why no deeper quantitative statement can be made. Vehicles present a challenge because they are movable objects. Additional semantic knowledge is required here, since a merely geometric approach can only identify the parts that were changed when the vehicle was moved, yet this information in itself has little value. Detected changes to vegetation appear plausible, but were determined only qualitatively and not quantitatively because no reliable ground truth could be generated.

The third part of the answer deals with the question of whether the quality of the process is suitable for a supporting or fully automated application. The method is definitely suitable for supporting a task, as it can reliably determine changed locations despite the prevailing uncertainties and a spatial blurring of the data and a visualization of the results can be easily interpreted by an observer. The question as to whether the results of the procedure are suitable for automatic interpretation can generally be answered in the affirmative, since the quality of the results permits this despite noise. In particular, however, the answer is that it depends on the application. Since the method was developed as a generic approach, it is not focused on a specific application. The generic evidence grid and the fuzzy spatial reasoning based thereon are suitable for answering complex questions, provided that the problem to be solved is based on spatial information and can be formulated by a Boolean expression.

The applicability of the method is, however, limited by a characteristic of the measurement principle used. When only the free space between the measuring point and the sensor position is taken into account, then changes measured against the sky or an open area cannot be detected. This limitation does not apply if the entire free space in its field of view is measured. For a LiDAR system, this means that measurements without return are preserved and interpreted as free space. This can be realized, for example, by keeping the direction vector of the measurement and assuming a fixed maximum range. This is a limitation of the method only insofar as it depends on the free space information to distinguish occlusions from actual changes.

Research question III: What metrics are appropriate for evaluating the spatial representation developed in this work and what are their limitations?

This final section addresses the metrics used for evaluation of the spatial representation. The introduction of the generic evidence grid made it necessary to develop metrics to compare the information represented by the evidence grid to a ground truth. This was required to quantitatively evaluate the results of the change detection method, the development of which was the top-level goal of this work. Three metrics were introduced, one for each (plausible) way to compare fuzzy representation and ground truth. It was found that while the investigated numerical error measure is suitable for evaluating the inner workings of the generic evidence grid, especially the degree of vagueness of the represented information, the ability of making statements beyond that is limited. This is because errors due to vagueness and geometric errors are intermixed. Applying a defuzzification solves this, but the resulting error measure is far less meaningful than the other metrics examined.

The evaluation scheme based on the confusion matrix represents a more meaningful tool than the error measure, since metrics such as precision and recall based on it allow a more detailed statement about different aspects of the method. The extension presented in this work, the *fuzzy confusion matrix*, allows to compensate for the systematic discretization errors caused by the generic evidence grid. The evaluation of the change detection results showed that from both metrics, insights could be drawn about the inner workings of the generic evidence grid and change detection. With respect to the generic evidence grid, properties that support the representability of objects could be determined. It was also found that for some object classes the coarsest resolution level is still better than random guessing. With respect to change detection, open questions could be answered regarding the max-pooling neighborhood and the usefulness of defuzzification. Such detailed statements would not have been possible on the basis of the error measure alone. As demonstrated, the metrics based on the confusion matrix can be considered as meaningful enough to answer complex questions.

However, the change detection experiments have shown that the metrics based on confusion matrices are only applicable if the results under consideration are statistically representative. Unlike the evaluation of a classifier, not every evaluated sample has the same significance, since a distance measurement identified as *changed* does not necessarily correspond to a full change, e.g., the change of a complete object. A changed object is usually represented by a large number of distance measurements. A more significant unit for the evaluation of changes would be the *semantic object*, since it represents a quantifiable entity that can be recognized as either changed or not changed. An error measure using semantic objects would be more meaningful for evaluating the change detection results than the present one. However, this metric would be application specific and could only be applied to change detection results, which contradicts the basic idea of a generic metric for a generic evidence grid.

9.2 Outlook

The generic nature of the evidence grid developed in this work, as well as the equally generic inference mechanism built on top of it, suggest that the theoretical framework developed here can be applied to other topics as well. This is supported by the encouraging change detection results. Any kind of problem which can be solved by combining different geo-related information can be considered. An example of this would be determining the area that would be shaded by a high-rise building after its construction. For this purpose, a representation of the shadow cast by the existing buildings at a given time of day needs to be computed. This can be done by means of a visibility calculation from the approximate location of the sun. Applying the logical inhibition operation on this representation of the shadow highlights the occluded area. This example can of course be extended to an arbitrarily complex question, such as the determination of the shadow cast in the field of view of all south-facing windows of a certain building in the late afternoon, but with a maximum distance of 100 meters from the window.

Since discretization of space is one of the main sources of error in change detection, increasing the maximum resolution would be one of the most important ways to improve the results. However, the central bottleneck here is the generation of the occupancy representations using raycasting. Even if the state-of-the-art in this field has been widely exploited in the context of this work, the area of highly parallel methods represents a further area of research. Another starting point to eliminate the bottleneck would be to calculate the free space ad hoc only where necessary. Using a spatial data structure to organize the measurement data as in Hebel et al. [2013], only those measurement rays that are required to distinguish a change from missing measurements can be chosen.

From the point of view of the generic evidence grid, another way to reduce discretization errors without simultaneously increasing resolution would be to develop some sort of *super resolution enhancement*. Currently, only one evidence distribution is stored per voxel. However, if several spatially localized evidence distributions were stored per voxel, a spatially accurate smooth transition between distributions can be computed using the N nearest neighbors and linear interpolation. With respect to the free space, which is expensive to compute, one could use existing or cheaply computable information such as the ray intersections with the voxel to roughly determine the affected areas where additional evidence distributions would need to be deposited within a voxel. Thus, the free space could be resolved more coarsely than the occupied space without causing false positives due to the different resolution levels of free and occupied space. The extent to which softer transitions would prevent the formation of artifacts would need to be investigated.

The work has shown that the quantitative error measures presented here are only meaningful if the underlying evidence is statistically significant. To compensate for this, the incorporation of semantic information is proposed. The number of vehicles marked as changed is much more meaningful than the number of measurement points recorded on the vehicle surface. This would also fix the disadvantage that small objects contribute less to the result than large ones. The specific nature of the semantics depends on the application, so this can only be done in the context of a specific application.

Bibliography

Aijazi AK, Checchin P, Trassoudaine L (2013) Detecting and updating changes in LiDAR point clouds for automatic 3D urban cartography. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-5/W2: 7–12.

Amanatides J, Woo A (1987) A fast voxel traversal algorithm for ray tracing. In: Eurographics '87: 3–10.

Asvadi A, Peixoto P, Nunes U (2016a) Two-stage static/dynamic environment modeling using voxel representation. In: Proc. Robot 2015, SecondIberian Robot. Conf.: 465–476.

Asvadi A, Premebida C, Peixoto P, Nunes U (2016b) 3D LiDAR-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes. Robotics and Autonomous Systems, 83: 299–311.

Azim A, Aycard O (2012) Detection, classification and tracking of moving objects in a 3D environment. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV): 802–807.

Basgall PL, Kruse FA, Olsen RC (2014) Comparison of LiDAR and stereo photogrammetric point clouds for change detection. In: Turner MD, Kamerman GW, Thomas LMW, Spillar EJ (eds) Laser Radar Technology and Applications XIX; and Atmospheric Propagation XI, 9080: 214 – 227.

Behley J, Garbade M, Milioto A, Quenzel J, Behnke S, Stachniss C, Gall J (2019) SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In: Proceedings of the IEEE International Conferenceon Computer Vision

Berrio JS, Ward J, Worrall S, Nebot E (2019) Identifying robust landmarks in feature-based maps. In: 2019 IEEE Intelligent Vehicles Symposium (IV): 1166–1172.

Black M (1937) Vagueness: An exercise in logical analysis. Philosophy of Science, 4: 425–455.

Borenstein J, Koren Y (1991) Real-time map building for fast mobile robot obstacle avoidance. In: Proc. SPIE 1388, Mobile Robots V

Borgmann B, Schatz V, Hammer M, Hebel M, Arens M, Stilla U (2021) MODISSA: A multipurpose platform for the prototypical realization of vehicle-related applications using optical sensors. Applied Optics, 60 (22): F50–F65.

Cavegn S, Haala N (2016) Image-based mobile mapping for 3D urban data capture. Photogrammetric Engineering and Remote Sensing, 82 (12): 925–933.

Chizhova M (2019) Virtual 3D reconstruction of destroyed Russian orthodox churches from incomplete point clouds. PhD thesis, Technische Universität München, München.

DeDeuge M, Quadros A, Hung C, Douillard B (2013) Unsupervised feature learning for classification of outdoor 3D scans. In: Australasian Conference on Robitics and Automation, 2: 1.

Dempster AP (1967) Upper and lower probabilities induced by a multivalued mapping. The Annals of Mathematical Statistics, 2 (38): 325–339.

Diehm AL, Gehrung J, Hebel M, Arens M (2020) Extrinsic self-calibration of an operational mobile LiDAR system. In: Turner MD, Kamerman GW (eds) Laser Radar Technology and Applications XXV, 11410: 46–61.

Dorninger P, Pfeifer N (2008) A comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. Sensors, 8 (11): 7323–7343.

Douillard B, Underwood JP, Melkumyan N, Singh SPN, Vasudevan S, Brunner CJ, Quadros AJ (2010) Hybrid elevation maps: 3D surface models for segmentation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): 1532–1538.

Dryanovski I, Morris W, Xiao J (2010) Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): 1553–1559.

Du S, Zhang Y, Qin R, Yang Z, Zou Z, Tang Y, Fan C (2016) Building change detection using old aerial images and new LiDAR data. Remote Sensing, 8 (12).

Dutta S (1991) Approximate spatial reasoning: Integrating qualitative and quantitative constraints. International Journal of Approximate Reasoning, 5 (3): 307–330.

Elfes A (1989) Occupancy grids: A probabilistic framework for robot perception and navigation. PhD thesis, Carnegie Mellon University.

Fairfield N, Kantor G, Wettergreen D (2007) Real-time SLAM with octree evidence grids for exploration in underwater tunnels. Journal of Field Robotics, 24 (1-2): 03–21.

Fine TL (1977) Review: Glenn Shafer, A mathematical theory of evidence. Bull. Amer. Math. Soc., 4 (83): 667–672.

Forsyth DA, Ponce J (2012) Computer Vision - A Modern Approach, Second Edition. Pitman.

Freeman J (1975) The modelling of spatial relations. Computer Graphics and Image Processing, 4 (2): 156–171.

Gallup D (2011) Efficient 3D reconstruction of large-scale urban environments from street-level video. PhD thesis, University of North Carolina.

Gehrung J, Hebel M, Arens M, Stilla U (2016) A framework for voxel-based global scale modeling of urban environments. In: ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W1: 45–51.

Gehrung J, Hebel M, Arens M, Stilla U (2017) An approach to extract moving objects from MLS data using a volumetric background representation. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1/W1: 107–114.

Gehrung J, Hebel M, Arens M, Stilla U (2018) A voxel-based metadata structure for change detection in point clouds of large-scale urban areas. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2: 97–104.

Gehrung J, Hebel M, Arens M, Stilla U (2019a) A fast voxel-based indicator for change detection using low resolution octrees. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W5: 357–364.

Gehrung J, Hebel M, Arens M, Stilla U (2019b) A representation of MLS data as a basis for terrain navigability analysis and sensor deployment planning. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2/W7: 39–46.

Geyer J, Kassahun Y, Mahmudi M, Ricou X, Durgesh R, Chung AS, Hauswald L, Pham VH, Mühlegg M, Dorn S, Fernandez T, Jänicke M, Mirashi S, Savani C, Sturm M, Vorobiov O, Oelker M, Garreis S, Schuberth P (2020) A2D2: Audi autonomous driving dataset.

Girardeau-Montaut D, Roux M, Marc R, Thibault G (2005) Change detection on point cloud data acquired with a ground laser scanner. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36-3/W19: 30–35.

Gordon M, Meidow J (2013) Calibration of a multi-beam laser system by using a TLS-generated reference. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-5/W2: 85–90.

Grisetti G, Kuemmerle R, Stachniss C, Burgard W (2010) A tutorial on graph-based SLAM. Intelligent Transportation Systems Magazine, IEEE, 2 (4): 31–43.

Gutmann JS, Fukuchi M, Fujita M (2008) 3D perception and environment map generation for humanoid robot navigation. International Journal of Robotics Research, 27 (10): 1117–1134.

Hackel T, Savinov N, Ladicky L, Wegner JD, Schindler K, Pollefeys M (2017) Semantic3D.net: A new large-scale point cloud classification benchmark. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1/W1: 91–98.

Hackel T, Wegner JD, Schindler K (2016) Fast semantic segmentation of 3D point clouds with strongly varying density. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, III-3: 177–184.

Hebel M (2012) Änderungsdetektion in urbanen Gebieten durch objektbasierte Analyse und schritthaltenden Vergleich von Multi-Aspekt ALS-Daten. PhD thesis, Technische Universität München.

Hebel M, Arens M, Stilla U (2013) Change detection in urban areas by object-based analysis and on-the-fly comparison of multi-view ALS data. ISPRS Journal of Photogrammetry and Remote Sensing, 86: 52–64.

Hellendoorn H (1973) Reasoning with fuzzy logic. PhD thesis, Delft.

Herbert M, Caillas C, Krotkov E, Kweon IS, Kanade T (1989) Terrain mapping for a roving planetary explorer. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2: 997–1002.

Hinz S (2008) Automatic object extraction for change detection and GIS update. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 37 (B4): 277–284.

Hirt PR, Xu Y, Hoegner L, Stilla U (2021) Change detection of urban trees in MLS point clouds using occupancy grids. PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science, 89 (4).

Hofmann A, Hoskova-Mayerova S, Talhofer V (2013) Usage of fuzzy spatial theory for modelling of terrain passability. Advances in Fuzzy Systems, 2013: 506406:1–506406:7.

Hornung A, Wurm KM, Bennewitz M, Stachniss C, Burgard W (2013) OctoMap: An efficient probabilistic 3D mapping framework based on octrees. Autonomous Robots, 34 (3): 189–206.

Huang R, Xu Y, Hoegner L, Stilla U (2022) Semantics-aided 3D change detection on construction sites using UAV-based photogrammetric point clouds. Automation in Construction, 134: 104057.

Kang Z, Zhang L, Yue H, Lindenbergh R (2013) Range image techniques for fast detection and quantification of changes in repeatedly scanned buildings. Photogrammetric Engineering and Remote Sensing, 79 (8): 695–707.

Kay TL, Kajiya JT (1986) Ray tracing complex scenes. SIGGRAPH Comput. Graph., 20 (4): 269–278.

Klir GJ, Folger TA (1987) Fuzzy sets, uncertainty, and information. USA: Prentice-Hall.

Krishnapuram R, Keller JM, Ma Y (1992) Quantitative analysis of properties and spatial relations of fuzzy image regions. In: Proceedings of the North American FuzzyInformation Processing Society '92

Kruse R, Gebhardt J, Klawonn F (1993) Fuzzy-Systeme. Stuttgart: Teubner.

Lee CC (1990) Fuzzy logic in control systems: fuzzy logic controller (Part I and II). IEEE Transactions on Systems, Man, and Cybernetics, 20 (2): 404–435.

Lee J, Yoo S, Hong S, Farkoushi MG, Bae J, Park I, Sohn HG (2020) Automated algorithm for removing clutter objects in MMS point cloud for 3D road mapping. Sensors, 20 (15).

Liang X, Kankare V, Hyyppä J, Wang Y, Kukko A, Haggrén H, Yu X, Kaartinen H, Jaakkola A, Guan F, Holopainen M, Vastaranta M (2016) Terrestrial laser scanning in forest inventories. ISPRS Journal of Photogrammetry and Remote Sensing, 115: 63–77.

Lindenbergh R (2010) Airborne and terrestrial laser scanning, chapter 7, (pp. 237–369). Whittles Publishing: Dunbeath.

Lindenbergh R, Pietrzyk P (2015) Change detection and deformation analysis using static and mobile laser scanning. Applied Geomatics, 7 (2): 65–74.

Matikainen L, Hyyppä J, Ahokas E, Markelin L, Kaartinen H (2010) Automatic detection of buildings and changes in buildings for updating of maps. Remote Sensing, 2 (5): 1217–1248.

Matrone F, Lingua A, Pierdicca R, Malinverni ES, Paolanti M, Grilli E, Remondino F, Murtiyoso A, Landes T (2020) A benchmark for large-scale heritage point cloud semantic segmentation. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B2-2020: 1419–1426.

Meagher D (1982) Geometric modeling using octree encoding. Computer Graphics and Image Processing, 19 (2): 129–147.

Menger K (1951) Ensembles flous et fonctions aléatoires. C. R. Acad. Sci., (232): 2001–2003.

Michel U, Thunig H, Ehlers M, Reinartz P (2012) Rapid change detection algorithm for disaster management. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, I-4: 107–111.

Moravec H, Elfes A (1985) High resolution maps from wide angle sonar. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA): 116–121.

Moravec HP (1988) Sensor fusion in certainty grids for mobile robots. AI Magazine, 9 (2): 61.

Munoz D, Bagnell JA, Vandapel N, Hebert M (2009) Contextual classification with functional Max-Margin Markov Networks. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition: 975–982.

Murakami H, Nakagawa K, Hasegawa H, Shibata T, Iwanami E (1999) Change detection of buildings using an airborne laser scanner. ISPRS Journal of Photogrammetry and Remote Sensing, 54 (2-3): 148–152.

Ning X, Shihuang S, Xizhou F (1993) A fuzzy approach to the weighted region problem for autonomous vehicles. In: Proceedings of 8th IEEE International Symposium on Intelligent Control: 400–404.

Pagac D, Nebot E, Durrant-Whyte H (1998) An evidential approach to map-building for autonomous vehicles. IEEE Transactions on Robotics and Automation, 14 (4): 623–629.

Payeur P, Hebert P, Laurendeau D, Gosselin CM (1997) Probabilistic octree modeling of a 3D dynamic environment. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2: 1289–1296.

Pfaff P, Triebel R, Stachniss C, Lamon P, Burgard W, Siegwart R (2007) Towards mapping of cities. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA): 4807–4813.

Pfeiffer D, Franke U (2010) Efficient representation of traffic scenes by means of dynamic stixels. In: Proceedings of the IEEE Intelligent Vehicles Symposium: 217–224.

Pollefeys M, Nistér D, Frahm JM, Akbarzadeh A, Mordohai P, Clipp B, Engels C, Gallup D, Kim SJ, Merrell P, Salmi C, Sinha S, Talton B, Wang L, Yang Q, Stewénius H, Yang R, Welch G, Towles H (2008) Detailed real-time urban 3D reconstruction from video. International Journal of Computer Vision, 78 (2): 143–167.

Poullis C (2013) A framework for automatic modeling from point cloud data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (11): 2563–2575.

Pu S, Vosselman G (2009) Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 64 (6): 575–584.

Robinson V (2003) A perspective on the fundamentals of fuzzy sets and their use in geographic information systems. Transactions in GIS, 7: 3–30.

Rosenfeld A (1998) Fuzzy geometry: An updated overview. Information Sciences, 110 (3): 127–133.

Roth-Tabak Y, Jain R (1989) Building an environment model using depth information. Computer, 22 (6): 85–90.

Roynard X, Deschaud JE, Goulette F (2018) Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. The International Journal of Robotics Research, 37 (6): 545–557.

Rusu RB, Cousins S (2011) 3D is here: Point Cloud Library (PCL). In: 2011 IEEE International Conference on Robotics and Automation: 1–4.

Ryde J, Hu H (2010) 3D mapping with multi-resolution occupied voxel lists. Autonomous Robots, 28 (2): 169–185.

Saito T, Rehmsmeier M (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10 (3): 1–21.

Schachtschneider J, Brenner C (2020) Creating multi-temporal maps of urban environments for improved localization of autonomous vehicles. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B2-2020: 317–323.

Scharwächter T, Enzweiler M, Uwe Franke SR (2013) Efficient multi-cue scene segmentation. In: German Conference on Pattern Recognition: 435–445.

Schauer J, Nüchter A (2018) The Peopleremover — Removing dynamic objects from 3-D point cloud data by traversing a voxel occupancy grid. IEEE Robotics and Automation Letters, 3 (3): 1679–1686.

Scherer D, Müller A, Behnke S (2010) Evaluation of pooling operations in convolutional architectures for object recognition. In: Proceedings of the 20th International Conference on Artificial Neural Networks: Part III: 92–101.

Shafer G (1976) A mathematical theory of evidence. London: Princeton University Press.

Singh A (1989) Review article digital change detection techniques using remotely-sensed data. International Journal of Remote Sensing, 10 (6): 989–1003.

Surmann H, Nüchter A, Hertzberg J (2003) An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. Robotics and Autonomous Systems, 45 (3): 181 – 198.

Tan W, Qin N, Ma L, Li Y, Du J, Cai G, Yang K, Li J (2020) Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW): 797–806.

Thrun S, Burgard W, Fox D (2005) Probabilistic Robotics. Intelligent Robotics and Autonomous Agents series. MIT Press.

Triebel R, Pfaff P, Burgard W (2006) Multi-level surface maps for outdoor terrain mapping and loop closing. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): 2276–2282.

Tunstel E (1995) Fuzzy spatial map representation for mobile robot navigation. In: Proceedings of the 1995 ACM Symposium on Applied Computing: 586–589.

Turksen I, Tian Y (1993) Combination of rules or their consequences in fuzzy expert systems. Fuzzy Sets and Systems, 58 (1): 3–40.

Tuttas SA (2017) Acquisition of building elements by photogrammetric point clouds and verification of a 4D building model for construction site monitoring. PhD thesis, Technische Universität München.

Underwood JP, Gillsjö D, Bailey T, Vlaskine V (2013) Explicit 3D change detection using ray-tracing in spherical coordinates. In: 2013 IEEE International Conference on Robotics and Automation: 4735–4741.

Vaaja M, Hyyppä J, Kukko A, Kaartinen H, Hyyppä H, Alho P (2011) Mapping topography changes and elevation accuracies using a mobile laser scanner. Remote Sensing, 3 (3): 587–600.

Vallet B, Brédif M, Serna A, Marcotegui B, Paparoditis N (2015) TerraMobilita/iQmulus urban point cloud analysis benchmark. Computers & Graphics, 49: 126–133.

Vilém Novák, Irina Perfilieva JM (1999) Mathematical principles of fuzzy logic. Kluwer Academic Publ.

Voelsen M, Schachtschneider J, Brenner C (2021) Classification and change detection in mobile mapping LiDAR point clouds. PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science.

Vosselman G, Gorte B, Sithole G (2004) Change detection for updating medium scale maps using laser altimetry. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 34 (B3): 207–212.

Wang Z, Klir GJ (1992) Fuzzy measure theory. New York: Plenum Press.

Weisbrod J (1996) Unscharfes Schließen. PhD thesis, Universität Karlsruhe.

Wilhelms J, Van Gelder A (1992) Octrees for faster isosurface generation. ACM Transactions on Graphics, 11 (3): 201–227.

Wolf D, Sukhatme GS (2004) Online simultaneous localization and mapping in dynamic environments. In: 2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04, 2: 1301–1307.

Xiao W, Vallet B, Brédif M, Paparoditis N (2015) Street environment change detection from mobile laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 38: 38–49.

Xu S, Vosselman G, Elberink SO (2013) Detection and classification of changes in buildings from airborne laser scanning data. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, II-5/W2: 343–348.

Xu Y, Hoegner L, Tuttas S, Stilla U (2018) A voxel- and graph-based strategy for segmenting man-made infrastructures using perceptual grouping laws: comparison and evaluation. Photogrammetric Engineering & Remote Sensing, 84 (6): 377–391.

Xu Y, Stilla U (2021) Toward Building and Civil Infrastructure Reconstruction From Point Clouds: A Review on Data and Key Techniques. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 14: 2857–2885.

Xu Y, Tong X, Stilla U (2021) Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry. Automation in Construction, 126: 103675.

Zadeh LA (1965) Fuzzy sets. Information and Control, 8: 338–353.

Zadeh LA (1973) Outline of a new approach to the analysis of complex systems and decision processes. IEEE Transactions on Systems, Man, and Cybernetics, 3 (1): 28–44.

Zeibak R, Filin S (2008) Change detection via terrestrial laser scanning. In: International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36 (Part 3): 430–435.

Zhang Z, Liu X, McDougall K, Wright W (2017) Fuzzy analysis of airborne LiDAR data for rainforest boundary determination. In: Proceedings of the 6th International Conference on Telecommunications and Remote Sensing: 48–53.

Zhu J, Gehrung J, Huang R, Borgmann B, Sun Z, Hoegner L, Hebel M, Xu Y, Stilla U (2020) TUM-MLS-2016: An annotated mobile LiDAR dataset of the TUM city campus for semantic point cloud interpretation in urban areas. Remote. Sens., 12 (11): 1875.

Acknowledgment

It is only at the end of a long journey that one realizes its true extent. After successfully reaching the end of the journey that obtaining my doctorate degree represents, I would like to express my deepest gratitude to all the professors, colleagues, friends and family members who have accompanied me along my journey and contributed in making my venture a success.

First and foremost, I would like to express my sincere gratitude to Prof. Dr.-Ing. Uwe Stilla, who supervised my doctoral work at the Technical University of Munich, for the opportunity to pursue my PhD in his research group. In particular, I would like to thank for his guidance in the academic community, his helpful professional advice for my academic studies as well as for teaching me the importance of networking. During my doctoral work, Prof. Stilla shared with me his many years of experience in structuring and publishing articles, dealing with reviewers and navigating the academic world. His guidance contributed significantly to the successful completion of my doctoral thesis. Furthermore, I would also like to thank apl. Prof. Dr.-Ing. Claus Brenner of Leibniz Universität Hannover for his work as a reviewer of my dissertation as well as for the time and effort he invested in preparing my review. I would also like to express my thanks to Prof. Dr.-Ing. Christoph Holst for his chairmanship of the examination committee.

This thesis was written alongside my work as member of the research staff at the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB in Ettlingen. I would like to express my special thanks to my department head Dr. rer. nat. Michael Arens and my team leader and mentor Dr. rer. nat. Marcus Hebel for giving me the freedom to write this thesis as well as for valuable academic discussions and suggestions. I would also like to thank my colleagues Dr. rer. nat. Marcus Hammer, Björn Borgmann, Axel Diehm, Dr. rer. nat. Volker Schatz, Ann-Kristin Grosselfinger and Dr.-Ing. Stefan Becker for their support.

The academic part of my work was situated at the Chair of Photogrammetry and Remote Sensing at the Technical University of Munich. I would like to express my special thanks to all the members of the chair that I had the pleasure of getting to know over the years. At this point, I would like to mention Prof. Dr.-Ing. Ludwig Högner in particular, who always had an open ear for the two external doctoral candidates from the Fraunhofer IOSB. I am grateful for being welcomed with open arms, even though being an external PhD candidate. I feel privileged to have been a part of the chair.

A house does not stand upright without a firm foundation; the same is true for a person. My most heartfelt thanks therefore goes to both my parents Klaus and Gabriele Gehrung and to my wife Natalija Gehrung. Thank you so much for your patience, your uplifting words, and your many years of support. It is only thanks to your unconditional love and selfless support that I was able to endure this demanding time and bring my doctoral thesis to a successful conclusion.