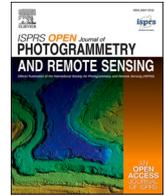


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

# ISPRS Open Journal of Photogrammetry and Remote Sensing

journal homepage: [www.journals.elsevier.com/isprs-open-journal-of-photogrammetry-and-remote-sensing](http://www.journals.elsevier.com/isprs-open-journal-of-photogrammetry-and-remote-sensing)

## Change detection in street environments based on mobile laser scanning: A fuzzy spatial reasoning approach

Joachim Gehrung<sup>a,b,\*</sup>, Marcus Hebel<sup>a</sup>, Michael Arens<sup>a</sup>, Uwe Stilla<sup>b</sup>

<sup>a</sup> Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, 76275, Ettlingen, Germany

<sup>b</sup> Photogrammetry and Remote Sensing, TUM School of Engineering and Design, Technical University of Munich, 80333, Muenchen, Germany

### ARTICLE INFO

#### Keywords:

Change detection  
Spatial information representation  
Evidence grids  
Fuzzy logic

### ABSTRACT

Automated change detection based on urban mobile laser scanning data is the foundation for a whole range of applications such as building model updates, map generation for autonomous driving and natural disaster assessment. The challenge with mobile LiDAR data is that various sources of error, such as localization errors, lead to uncertainties and contradictions in the derived information. This paper presents an approach to automatic change detection using a new category of generic evidence grids that addresses the above problems. Said technique, referred to as *fuzzy spatial reasoning*, solves common problems of state-of-the-art evidence grids and also provides a method of inference utilizing fuzzy Boolean reasoning. Based on this, logical operations are used to determine changes and combine them with semantic information. A quantitative evaluation based on a hand-annotated version of the TUM-MLS data set shows that the proposed method is able to identify confirmed and changed elements of the environment with F1-scores of 0.93 and 0.89.

## 1. Introduction

### 1.1. Change detection in urban environments

The goal of automatic change detection in urban areas is to analyze multiple epochs of spatial data, thereby identifying and quantifying changes without human interaction. This opens up many possibilities, such as assisting in updating city models either by automatic revision or by pointing out to a human operator where an existing model needs to be changed. Change detection can also be used for automatic city surveys or to determine damage from disasters such as storms and floods.

Mobile mapping platforms that use laser scanning enable the detailed, fast and cost-efficient recording of large urban areas along roads. The environment is captured at a high frequency at close range, which leads to a large amount of high-resolution 3D data in a short period of time. This allows changes to be detected with a high level of detail, even in areas that are not visible with other scanning techniques such as airborne laser scanning.

However, change detection in mobile LiDAR data also poses a number of challenges. One of the biggest issues here is dealing with the sources of error that a mobile mapping system is subject to. The two

major sources are boresight calibration errors and localization errors respectively the resulting registration residual errors. Both sources of error result in a degree of uncertainty remaining in the data. Environmental conditions such as moving objects and vegetation or process-related sources of error such as discretization artifacts lead to contradictions in the information derived from the measurement data. Furthermore, meaningful conclusions can only be drawn if sufficient measurement data is available. Explicitly addressing these factors is key to a solid description of the environment and to procedures that are robust to the above sources of error.

A variety of applications arise when such a robust form of representation is capable of representing arbitrary spatial data. This allows the definition of a form of inference that can then be used to solve any problem within a spatial context. This allows, for example, the results of change detection to be combined with other sources of information for further processing and evaluation.

### 1.2. Characteristics of conventional evidence grids

Starting from simple data structures for storing binary information, evidence grids have become complex representations with a

\* Corresponding author. Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, 76275, Ettlingen, Germany.

E-mail addresses: [joachim.gehrung@iosb.fraunhofer.de](mailto:joachim.gehrung@iosb.fraunhofer.de) (J. Gehrung), [marcus.hebel@iosb.fraunhofer.de](mailto:marcus.hebel@iosb.fraunhofer.de) (M. Hebel), [michael.arenis@iosb.fraunhofer.de](mailto:michael.arenis@iosb.fraunhofer.de) (M. Arens), [stilla@tum.de](mailto:stilla@tum.de) (U. Stilla).

<https://doi.org/10.1016/j.ophoto.2022.100019>

Received 16 February 2022; Received in revised form 26 July 2022; Accepted 26 July 2022

Available online 17 August 2022

2667-3932/© 2022 The Authors. Published by Elsevier B.V. on behalf of International Society of Photogrammetry and Remote Sensing (isprs). This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

probabilistic description of evidence (Hornung et al., 2013). The most popular form of evidence grid is the *occupancy grid*, which represents free and occupied space (Moravec, 1988). Despite the sophisticated state of evidence grids in general, this is a technology that still offers opportunities for further development and thus for new fields of application.

Using a probabilistic payload can lead to complications, for example when the *log-odd notation* introduced by Moravec (1988) and utilized by OctoMap (Hornung et al., 2013) is used to represent evidence. This representation doesn't keep track of evidences for and against the information, in this case occupancy, but instead stores the information using a ratio of both. Unfortunately, this doesn't allow to distinguish between the case of *no evidence at all* and the case of *contradicting evidence*, since information that needs to be stored in two buckets is compressed into a single bucket. This is a source of ambiguity that can lead to the effects discussed in Gehrung et al. (2017). Here it was observed that contradictions occur due to discretization errors, resulting in an apparent lack of observations for the affected areas as described above.

Since log-odds can again be converted into probabilities, a limited statement can be made about the degree of uncertainty associated with the evidence. For the reason mentioned above, however, it is not possible to make a statement about the *degree of ignorance*. In other words, no conclusion can be drawn about whether the representation contains enough information to make an informed statement about the state of the world.

Most evidence grids are designed exclusively for a specific purpose, such as the representation of spatial occupancy. Due to this lack of generalization, it is not possible to represent arbitrary information, only a specific one. A generic representation would, however, allow several pieces of information to be linked with one another.

### 1.3. Contribution of this work and differences to other works

This article addresses the above-mentioned points of criticism and introduces a novel form of evidence grid that is able to represent generic information, even if it is *contradictory*. Furthermore, it allows for a statement to be made about the associated *degree of uncertainty* and *ignorance*. The chosen approach also enables the combination of evidence grids by means of logical operations. To name an example of the resulting possibilities, this allows to detect changes only in structures that have been stable over a long period of time, by considering only structures that were confirmed in two epochs and then show a change in a third epoch. Another example is that this enables to identify all changes to vehicles within sight by simply linking changes with object classes and a field of view. The theoretical foundation for this is provided by *fuzzy logic*. We call our form of spatial representation a **fuzzy evidence grid** and the process of inference **fuzzy spatial reasoning**.

This work differs from other work in that it addresses all problems listed in Section 1.2, which arise from state-of-the-art procedures based on a log-odd representation in the sense of Moravec (1988). The approach presented here is the only one of its kind that can address the *degree of uncertainty, contradiction and ignorance*. Another difference is that we have generalized occupancy grids towards generic evidence grids. These allow the combination of arbitrary spatial information using Boolean logic. While previous own works such as Gehrung et al. (2019) are able to represent and combine arbitrary spatial information with each other, these were not able to do so taking into account the degree of uncertainty, contradiction and ignorance. The change detection method based on the above approach differs from the state-of-the-art in that it can deal with the mentioned factors. It also represents one of the few generic change detection methods on mobile LiDAR data that do not focus on a specific application.

### 1.4. Structure of this paper

After briefly discussing the state-of-the-art in Chapter 2, an overview about the aforementioned approach is given in Chapter 3. Its application

for change detection in urban environments is described in Chapter 4. Experiments to verify the validity of the proposed approach, as well as the results and their discussion are given in Chapter 5 and Chapter 6.

## 2. Related work

### 2.1. Spatial data representation

#### 2.1.1. Occupancy grids

*Occupancy grids* are a specific form of evidence grids that represent the degree of occupancy of a specific region of space. An early form was proposed by Elfes (1989) for the purpose of indoor mapping. The approach utilizes a two-dimensional grid, where the state of each cell is represented by a probability that implies either free, occupied or unseen space. Despite the low precision of the ultrasonic sensors applied to parametrize the grid, the approach allows for an accurate representation of the environment.

A technique by the name of *histogram in-motion mapping* for real-time robot navigation has been proposed by Borenstein and Koren (1991). By using rapid in-motion sampling based on ultrasonic range finders, a pseudo-probability distribution describing the environment is created. The content of each cell corresponds to the level of evidence that an obstacle is present.

A 3D grid extension was presented by Roth-Tabak and Jain (1989), but without taking any uncertainties into account. The disadvantage of all the methods mentioned here is that the boundaries of the spatial area have to be known in advance. Furthermore, the lattice structure consumes more memory than would be the case with more efficient data structures such as octrees, since it represents both homogeneous and heterogeneous areas with the same resolution.

#### 2.1.2. Elevation maps

*Elevation maps* or *2.5D maps* denotes a category of approaches that utilizes two-dimensional grid structures, where each cell encodes some form of height value (Herbert et al., 1989). In this way, overhanging structures, such as bridges, cannot be described. Therefore, the approach is only applicable in cases where a surface representation of the environment is sufficient. An example for this is the contribution to the scene understanding tasks of driver assistance and autonomous systems by Pfeiffer and Franke (2010). They present an approach based on so-called *Stixels*, rectangular sticks of a certain width that limit the free space in front of the vehicle. It is 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 are intended to increase the applicability of elevation maps. The technique proposed by Triebel et al. (2006) and Pfaff et al. (2007) utilizes several surfaces per cell. Gutmann et al. (2008) suggested to use multiple classes of cells in order to describe different structures. In order to allow a cell to represent more than a mere discretization of height, Ryde and Hu (2010) suggested to store a voxel list in each grid cell. Although this representation is volumetric, the authors make no distinction between free and unobserved space. Dryanovski et al. (2010) suggested to maintain a list of both occupied and free voxels per cell.

A hybrid approach has been presented by Douillard et al. (2010). It utilizes elevation maps to describe the background, while foreground objects are represented by high-resolution voxel structures.

#### 2.1.3. Octree-based environment representations

Octrees have the advantage over fixed-size grids that they enable dynamic resolution on one hand and multi-scale resolution on the other. The use of octrees to represent spatial data has been suggested in publications such as the one by Meagher (1982). Early techniques dealt with the representation of binary occupancy information (Wilhelms and Van Gelder, 1992). Octree-based approaches are still used today whenever a representation for arbitrarily shaped environments is required, as is the

case in robot navigation (Surmann et al., 2003).

Payeur et al. (1997) utilize octrees to expand the above-mentioned two-dimensional occupancy grid into three dimensions, thus presenting a probabilistic approach to modeling free and occupied space. Another technique with the name *Deferred Reference Counting Octree* was introduced by Fairfield et al. (2007). It is intended for use in the context of particle filter SLAM and for this reason shares subtrees between several octrees, which allows octrees to be copied quickly. The publication also includes a sophisticated update mechanism and a maximum likelihood compression mechanism for pruning child nodes with the same status.

An approach popular in the field of robotics is known by the name *OctoMap* (Hornung et al., 2013). One of the main extensions in relation to the work of Fairfield et al. (2007) is the use of probability clamping in order to achieve an almost loss-less compression. Multi-resolution queries and the problem regarding representations with exaggerated confidence are also addressed.

Based on the theoretical foundation provided by the state-of-the-art, we proposed a concept for a scalable occupancy representation (Gehrung et al., 2016). In subsequent work, we have shown that evidence grids using log-odds are susceptible to artifacts caused by *contradictions in evidence* (Gehrung et al., 2017). An approach based on a combination of plane-filtered raycasting and iterative refinement was presented as a heuristic to solve the problem during the construction of the evidence grid (Gehrung et al., 2018).

## 2.2. Change detection

### 2.2.1. Point-based change detection

A point-to-point comparison, also known as the *surface difference*, is the most direct way of detecting changes between two point clouds. Basgall et al., 2014 utilized a subtraction method to calculate differences between LiDAR and stereo-photogrammetric point clouds. The Hausdorff distance was used by Kang et al. (2013) to calculate point-to-point distances in order to avoid issues related to local density variations.

Girardeau-Montaut et al. (2005) proposed a fast indicator for changes utilizing an octree for point cloud storage. A change is implied when the point distance within a voxel is beyond the noise. Empty voxels in one epoch and changes or shifts in the local normal of the points within a voxel are further indications of changes. The change is then verified by checking the neighboring voxels within the octree.

Xu et al. (2013) utilized a *3D surface separation map* that encodes the distance from a point to the nearest fitted plane from another epoch. A rule based classifier is used to classify changes identified by said map into different categories.

An automatic method for building change detection from LiDAR data and aerial images has been proposed by Du et al. (2016). Two data sets are co-registered using the ICP algorithm, then changes are derived from height differences and grey-scale similarities. A graph-cut technique is applied to further optimize detection results by utilizing contextual information.

A comprehensive discussion of change detection methods based on *digital elevation models* has been published by Matikainen et al. (2010). Murakami et al. (1999) compared digital elevation models and ALS data by projecting individual measurements onto a two-dimensional grid. Changes were derived by a subtraction method.

Change detection methods based on point-to-point distances are sensitive to variations in point density that occur in mobile laser scanning. In addition, point-based techniques do not consider free space and are therefore unable to handle occlusion.

### 2.2.2. Ray-based change detection

If the sensor position of each point is preserved, a point cloud can also be interpreted as a bundle of rays, with a ray representing both the laser pulse propagation path and the measured surface point. Zeibak and Filin (2008) proposed a so-called *visibility map*, a panorama depth map

that encodes the distance between the individual scene points and the laser scanner using spherical coordinates. Two scans can be compared by applying thresholds to said distance, dividing the scene points 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.

A method for change detection in ALS point clouds has been proposed by Hebel et al. (2013). A three-dimensional grid enables an efficient neighborhood search for rays. Changes are determined by combining the belief functions from all involved rays using the Dempster-Shafer theory. Meyer et al. (2022) utilized this approach to compare BIM models with measurements from a stationary terrestrial laser scanner. Xiao et al. (2015) further improved the method proposed by Hebel et al. (2013) with a point-to-triangle distance-based technique to conduct direct consistency evaluation on points. Both approaches produce good results and, unlike voxel-based methods, the resolution is not artificially limited by discretization.

### 2.2.3. Voxel-based change detection

In voxel-based change detection methods, point clouds are converted into a volumetric representation such as a grid, an octree or a set of voxels. Changes are found by comparing the occupancy information stored in said voxels.

A two-dimensional occupancy grid for representing the environment of an autonomous vehicle has been proposed by Pagac et al. (1998). Measurements are integrated into said grid using the Dempster-Shafer inference rule. A similar approach for SLAM-based robot navigation has been proposed by Wolf and Sukhatme (2004). It utilizes two occupancy grids to represent both the static and dynamic parts of a scene. A comparison of the occupancy states encoded in each voxel allows a conclusion to be drawn as to whether an element of the scene that was previously considered static has been moved. Azim and Aycard (2012) published a similar approach that utilizes conflict search on an occupancy grid based on the Octomap framework in order to determine dynamic elements. The latter ones are classified and tracked using a method based on finding the global nearest neighbor. Schachtschneider and Brenner (2020) utilize change detection based on an occupancy grid in order to extract feature maps for the localization of autonomous vehicles from point clouds.

Huang (2021) detects changes of construction sites using photogrammetric point clouds by considering both geometric and semantic changes. In a first step, semantic changes are detected using an occupancy-based change detection method inspired by Hebel et al. (2013). In a next step, geometric changes are considered to check consistency and detect conflicts. Wysocki et al. (2022) have demonstrated that voxel-based change detection can be utilized to refine semantic 3D building models with LiDAR measurements. Hirt et al. (2021) proposed an approach for detecting changes in trees in which a three-stage process consisting of trunk extraction, tree separation and crown expansion is used to extract tree instances from a point cloud. These instances are then linked to the geometric changes that are determined using an occupancy grid.

## 3. Fuzzy spatial reasoning

Section 1.2 lists a number of advantages and disadvantages of state-of-the-art evidence grids that are addressed and solved with the approach of **fuzzy spatial reasoning** presented here. At its core, a framework based on *fuzzy logic* is used. This allows to collect separate evidence for and against the information at hand. As a direct consequence, not only *certainties* but also *contradicting information* can be modeled. It is also possible to identify areas with insufficient evidence, which means that they cannot make any reliable statements about the information contained therein. An overview of the structural and procedural approaches of fuzzy spatial reasoning is given below.

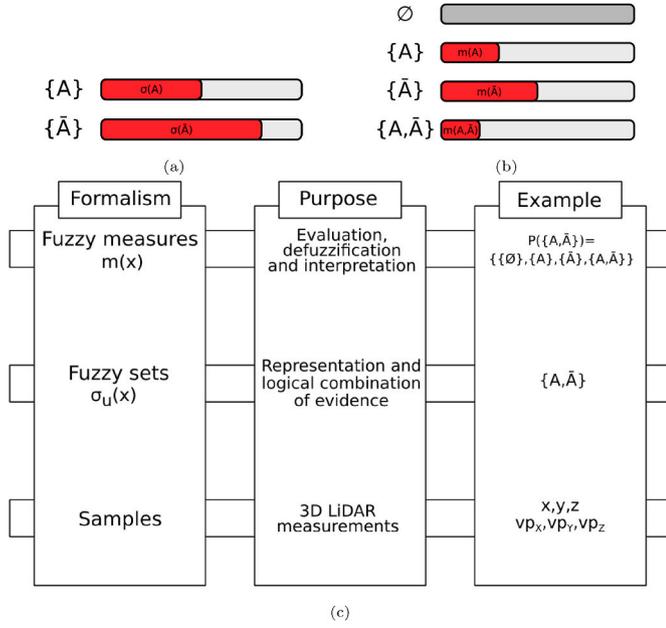


Fig. 1. Evidence representation in fuzzy evidence grids. (a) The evidence distribution in form of a fuzzy set is used to logically combine evidence. (b) The associated fuzzy measure is required to interpret the fuzzy set. (c) Process of deriving the aforementioned fuzzy set and measure from raw data using the example of occupancy.

### 3.1. Fuzzy evidence grids

#### 3.1.1. Large scale data handling

This approach leverages organizational data management practices from a previous work to manage and process the large volumes of measurements generated when processing mobile laser scanner data (Gehring et al., 2016). This involves the partitioning of all data (e.g. point clouds, fuzzy representations, etc.) into chunks using a virtual grid, so-called *3D tiles*, which allows for efficient memory management, parallelization and scalability. A downsampling/filtering step was deliberately omitted as this would negatively affect the functionality of the evidence grids presented in the next section.

#### 3.1.2. Definition of fuzzy evidence grids

A **fuzzy evidence grid** consists of two main components. The first component is for the handling of structural information. It is a spatial data structure in form of an octree that allows for multiple resolutions and adaptability to the underlying information. The second component is the representation of the evidence, which is multi-layered. Both components work together to assign evidence to a spatial context.

The representation of evidence within a voxel is based on a combination of an **evidence distribution** for the logical combination of evidence and a **fuzzy measure** for its interpretation. The evidence distribution  $\sigma_x$  is represented by a *fuzzy set*  $\mu_x$ . A fuzzy set is a vector, each element of the vector contains a number within the interval  $[0, 1]$  that describes the membership to the class represented by the element. In case of a fuzzy evidence grid, there are two classes  $A$  and  $\bar{A}$  that

correspond to *the evidence for a given information and the evidence against it* (cf. Fig. 1a). In case of occupancy, one of the classes collects all evidence implying that the voxel is *occupied*, the other class collects evidence that it is *free*. As mentioned above, the evidence distribution is not only responsible for representing the degree of evidence collected for a specific information, but also to *logically combine* information stored in multiple fuzzy evidence grids.

The fuzzy measure is derived from the evidence distribution and allows for a better interpretation of the information represented by the latter one. The annotation  $m(A)$  reflects the confidence that a fuzzy variable  $x$  has the value of the subset  $A \in \mathcal{U}_x$ . The fuzzy measure is defined over the power set of the evidence distribution. In addition to the certainties  $m(A)$  and  $m(\bar{A})$  for and against the information, it is therefore also possible to make a statement about the reliability of the information. Or in other words, it describes how high the level of ignorance represented by  $m(A, \bar{A})$  actually is (cf. Fig. 1b).

The three different forms of the information to be represented and their relationship to each other are summarized in Fig. 1c. In the case of an occupancy grid, the bottom layer consists of 3D-measurements. In a process described in Section 4.1, these are converted into a set of evidence distributions. Whenever required, the conversion of a fuzzy set into a fuzzy measure is carried out using the following case differentiation (Weisbrod, 1996):

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

The height of evidence distribution  $H(\sigma_x)$  is defined as

$$H(\sigma_x) := \max_{u \in \mathcal{U}_x} \{\sigma_x(u)\}. \quad (2)$$

Equation (1) can be interpreted as follows. The fuzzy measure is defined by the power set of the fuzzy set. The certainties for  $m(A)$  and  $m(\bar{A})$  correspond to the memberships of the fuzzy set, but are normalized by the height of the evidence distribution, i.e. the largest value. The ignorance  $m(A, \bar{A})$  corresponds to the remaining evidence mass, the empty set is omitted.

### 3.2. Logical inference

Logical inference describes the combination of two fuzzy evidence grids via a *logical operation*, the result is again stored in a fuzzy evidence grid. The process of inference consists of two parts. The first part includes the *structural combination of the octrees* that contain the evidence distributions. The second part is considered with *merging the evidence* itself.

#### 3.2.1. Structural combination

As explained, octrees are the data structure that assigns an evidence distribution to space. Whenever two evidence grids, i.e. the corresponding octrees, are combined, a third octree is required to store the result. The structure of this octree must resemble the *combined structure of both source octrees*, a structure that is able to store the information

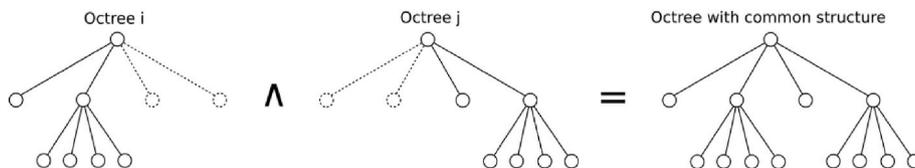


Fig. 2. An example demonstrating how two octrees  $i$  and  $j$  are combined into a single octree with a common structure. The latter one is used to store the fuzzy evidence representations obtained by combining two fuzzy evidence grids.

**Table 1**

A list of fuzzy logical operations tailored to the s-norm, t-norm and complement used in this work.

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

with a minimum of resources possible, but the maximum resolution required (see Fig. 2).

The algorithm used for this is a modified version of a similar one published in earlier works (Gehring et al., 2019) (cf. Algorithm 1). The procedure is largely unchanged, but there are differences in the handling and calculation of the payload. The main difference involves the insertion of placeholder voxels without evidence, as explained in Section 4.2. The approach is of recursive nature and equals a depth-first search as long as the structure of both octrees is similar. As soon as the structure

deviates, the recursion continues, but a node with an empty evidence distribution is replacing the missing sub-tree, thus emulating complete ignorance. It is not possible to use coarser resolutions as placeholders, as this leads to undesired artifacts.

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

```

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

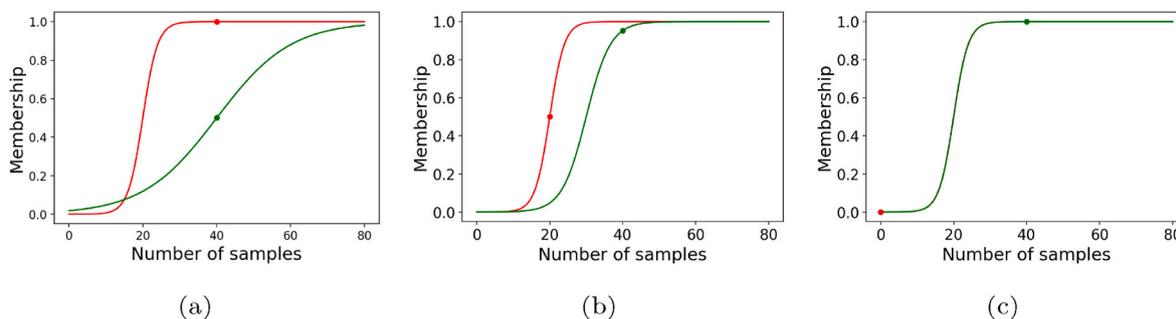
Function construct( $node_i, node_j, node_c$ ):
    // Merge payloads and store result in  $node_c$ .
    updatePayload( $node_i, node_j, node_c$ )

    // 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 $_i$   $\leftarrow$  createPlaceholderNode()
            else
                | if hasChild( $node_i, child$ ) then
                    | | next $_i$   $\leftarrow$  getChild( $node_i, child$ )
                | else
                    | | next $_i$   $\leftarrow$  createPlaceholderNode()
            if getDepth( $node_j$ ) < getDepth( $node_i$ ) then
                | next $_j$   $\leftarrow$  createPlaceholderNode()
            else
                | if hasChild( $node_j, child$ ) then
                    | | next $_j$   $\leftarrow$  getChild( $node_j, child$ )
                | else
                    | | next $_j$   $\leftarrow$  createPlaceholderNode()

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

```



**Fig. 3.** 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. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

### 3.2.2. Combination of evidence

Logical inference on fuzzy sets requires a DeMorgan-Triplet, i.e. an **s-and t-Norm** representing the operations for union and intersection, as well as a **complement** operation. Any other logical function can be derived using these three components. In this work, the s-norm is

$$\sigma_{A \cup B}(u) := \max\{\sigma_A(u), \sigma_B(u)\}. \quad (3)$$

and the t-Norm is defined as

$$\sigma_{A \cap B}(u) := \min\{\sigma_A(u), \sigma_B(u)\}, \quad (4)$$

The **set complement** was chosen as the complement operation. The negation is carried out by choosing the evidence of the other class:

$$\sigma_{A^c}(u) := \begin{cases} \sigma(\bar{A}), & \text{if } u = A \\ \sigma(A), & \text{if } u = \bar{A} \end{cases} \quad (5)$$

The two norms do not imply prior knowledge and the complement takes into account the fact that one class of evidence is the opposite of the other. This also has implications for the logical operations. Since the second class  $\bar{A}$  is by definition the complement of the first class  $A$ , each operation must be negated using De Morgan's law before being applied to  $\bar{A}$ . If this does not happen, the behavior of an operation no longer corresponds to that of classic Boolean logic. This leads to the operations described in Table 1.

### 3.3. Handling of certainty and ignorance

Both *certainty* and *ignorance* are determined based on the fuzzy measure. As mentioned above, the latter is derived from the evidence distribution using Equation (1). The *certainty* is quantified by using the elements  $\{A\}$  and  $\{\bar{A}\}$  of said measure. They are similar to the classes of the evidence distribution, but the *degree of ignorance* is factored out. This is encoded in the third element  $\{A, \bar{A}\}$  and equals the residual evidence mass that has not been assigned to any of the other classes. If there is only few evidence, then the degree of ignorance is high and vice versa. The degree of ignorance can be used to assess the reliability of the information, i.e. the amount of evidence in support.

### 3.4. 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. One simple way is **threshold filtering**. Whenever the evidence for the information exceeds the threshold  $t_{certainty}$ , it is considered to belong to class  $A$ :

$$\sigma_a := \{\sigma_a | m(A) \geq t_{certainty}\} \quad (6)$$

Another method of defuzzification is **pro/contra filtering**. A voxel is considered as belonging to class  $A$  if the evidence for that class is larger

than the evidence for class  $\bar{A}$ :

$$\sigma_a := \{\sigma_a | m(A) > m(\bar{A})\} \quad (7)$$

In the context of this work, the second defuzzification method has been found useful. The decision is made based on the ratio between the supporting and contradicting evidence mass, not just based on the amount of collected evidence mass. This allows for more reliable conclusions in areas where there is little evidence.

### 3.5. Differences to the Dempster-Shafer theory of evidence

At this point, the difference between the Dempster-Shafer theory of evidence and the method presented here is addressed. In fact, there is a connection between the fuzzy measure used in this paper, Dempster-Shafer and also probability measures. However, the major difference with Dempster-Shafer is that the method presented in this paper uses fuzzy measures in combination with fuzzy sets. This has two major advantages over Dempster-Shafer.

The first advantage is that a theoretically sound and computationally efficient inference mechanism is possible. This allows to combine arbitrary spatial information using Boolean operations. A similar approach can be performed using Dempster-Shafer, since said theory as well as the Dempster rule of combination can handle any number of classes. Such an approach would require that, for each information domain to be considered, two classes of evidence (one for and one against said information) were defined. However, since all classes of a logical term must exist in the same *universe*, the evidence mass is distributed among all elements of the power set. For  $n$  classes,  $2^n$  elements are required, which is unfeasible for cases with many classes. The inference approach presented in this paper is considered to be more efficient because it always suffices with 2 classes, i.e. one class each for and against the available evidence for an information, no matter the length of the logical term.

The second advantage is that dealing with contradictions leads to more intuitive results than is the case with Dempster's rule of combination. Regarding the latter, contradictory probability mass assignments lead to conflicts being resolved by distributing all of the probability mass to the other elements of the set. This could lead to non-plausible results. In the approach presented here, inference based on fuzzy sets allows to deal with contradictions in such a way that they are preserved and the effect mentioned above does not appear. However, due to the fuzzy measure based on the fuzzy sets, it is possible to calculate measures such as Belief and Plausibility as in Dempster-Shafer.

## 4. Change detection

The process of change detection applied in this work requires the construction of a **fuzzy occupancy representation** for each epoch. Logical operations are then used to determine the confirmed and

**Table 2**

Runtime complexity of both the individual stages as well as the overall change detection.

Step	Classical octree (guaranteed)	Linear octree (guaranteed)	Hash map (best case/worst case)
Preprocessing	$O(4N)$	$O(4N)$	$O(4N)$
Insertion of measurements	$O(N \log N)$	$O(N)$	$O(N)/O(N^2)$
Generation of representation	$O(N \log N)$	$O(N)$	$O(N)/O(N^2)$
Comparison of both epochs	$O(2N)$	$O(2N)$	$O(2N)$
Change detection (total)	$O(10N + 2N \log N)$ $\cong O(N \log N)$	$O(12N)$ $\cong O(N)$	$O(12N)/O(10N + 2N^2)$ $\cong O(N)/O(N^2)$

changed elements of the environment and also to combine the results with other sources of information in order to determine the changes per object class.

In change detection, there are *four possible cases* to consider. The first case concerns the issue of *observability*. If a location was observed only in one epoch, but not in the other, it is not possible to distinguish between a missing observation and an actual change. The second case is about determining all elements of the environment that are unchanged, i.e. *confirmed*. The two remaining cases are considered with the actual changes, that is with *appeared* and *disappeared* elements of the environment. Each of the points mentioned here will be addressed below.

#### 4.1. Generation of fuzzy evidence representations

Change detection is based on the comparison of fuzzy evidence representations. The latter ones are generated on a per tile base. The rays are truncated to the tile boundaries using a technique published by Kay and Kajiya (1986). The algorithm proposed by Amanatides and Woo (1987) is used to determine the voxels traversed by each ray. Per voxel the number of rays ending inside is stored, also is the number of rays traversing. These numbers represent the evidences for both occupied and free space, as it is shown in the bottom part of Fig. 1c.

A membership function determines the affiliation of each voxel to the classes *occupied* and *not occupied*. For the sake of simplicity, the latter will also be referred to as *free*. Determining the affiliation is done based on the samples collected in the previous step.

A suitable function must allow a gradation between both extreme values. Furthermore, it must be able to weight evidence for free space less if evidence for occupied space is available. This is necessary to compensate for the effect discussed in Gehrung et al. (2017). A whole range of functions come into question, a logistic function was selected, as this can vary between an approximately linear function and a step function. The equation for the logistic function is as follows:

$$\sigma_u(x) := \frac{1}{1 + e^{-k_u \cdot u(x)}} \quad (8)$$

A transfer function is utilized to shift the turning point of the logistic function. For occupied space it is defined as

$$t_{occ}(x) := x - s_{occ}, \quad (9)$$

where  $s_{occ}$  is the median of samples for occupied space per voxel. The transfer function for free space depends on the aforementioned membership  $\sigma_{occ}(x)$ :

$$t_{free}(x) := x - (s_{free} + s_{free} \cdot \sigma_{occ}(x)) \quad (10)$$

The median of samples for free space is represented by  $s_{free}$ . By utilizing  $\sigma_{occ}(x)$ , the input to the logistic function is shifted in favor of a lower function value, the higher the available evidence for *occupancy* is. In addition, the slope  $k_{free}$  of the logistic function is reduced:

$$k_{free} := k_{occ} - \sigma_{occ}(x) \cdot (k_{occ} - k_{min}) \quad (11)$$

As a result, the function value is further reduced if there is high evidence for occupied space.  $k_{occ}$  is the steepness used for occupied space,  $k_{min}$  is the minimum steepness. The membership for both the free and occupied case need to be normalized linearly to map into the interval [0,

1]. The starting point for this normalization is chosen at  $x = 0$  and the end point at twice the median. Examples of the above-mentioned membership function for different evidences can be found in Fig. 3.

Moving objects are identified based on the single scans using the method proposed by Underwood et al. (2013). A fuzzy representation is generated and used to subtract the moving objects from the occupancy grid by linking both representations via an *and not*-operation.

#### 4.2. Consideration of unseen areas

Unseen areas are expressed by the fact that a voxel exists in one of the two octrees to be linked, but not in the other. As described in Section 3.2.1, this is handled by creating a placeholder voxel without evidence during the creation of the common octree. It is in the nature of the fuzzy representation presented here that none of the logical conjunctions used for change detection will result in a false positive. Or in other words, a logical conjunction of information with ignorance does not lead to a result that implies an information that is not present.

#### 4.3. Determination of confirmed elements

An element of the environment is considered confirmed if it occurs in both epochs. From a logical point of view, this corresponds to applying an *and*-operation. Applying this operation would lead to various false negatives, because despite all efforts, discrepancies remain due to poor registration of the MLS data or calibration errors of the MLS system. These can be compensated for by smoothing the fuzzy representation of each epoch with a *maxpool*-operation. The resulting loss of detail is limited if the neighborhood of the smoothing operation is kept small. The *and*-operation is then applied between the epoch to be confirmed and the smoothed version of the other epoch. The following equation demonstrates this for the first epoch:

$$\sigma_{confirmed_1} := f_{\wedge}(\sigma_{occupancy_1}, \sigma_{smoothed_2}) \quad (12)$$

The second epoch is calculated using the following equation:

$$\sigma_{confirmed_2} := f_{\wedge}(\sigma_{smoothed_1}, \sigma_{occupancy_2}) \quad (13)$$

The *and*-operation is symbolized by  $f_{\wedge}(\cdot)$ . The representations  $\sigma_{occupancy_1}$  and  $\sigma_{occupancy_2}$  are the fuzzy occupancy representations of the first and second epoch. The same applies to  $\sigma_{smoothed_1}$  and  $\sigma_{smoothed_2}$ , which are the smoothed fuzzy occupancy representations. As a result of this approach, there are two representations that contain the confirmed elements for both epochs.

#### 4.4. Determination of appeared and disappeared elements

Changes in general are determined using the *xor*-operation. However, since a distinction is to be made between *appeared* and *disappeared*, the two halves of said operation are used, i.e. the *inhibition*-operations. The case *appeared* is calculated using the inhibition of  $\bar{A}$ . This corresponds to an *and*-operation with input negation:

$$\sigma_{appeared} := f_{\wedge}(\overline{\sigma_{smoothed_1}}, \sigma_{occupancy_2}) \quad (14)$$

The procedure for the *disappeared* case is similar. It is computed by the inhibition of  $A$ , which is also achieved by an *and*-operation, but this

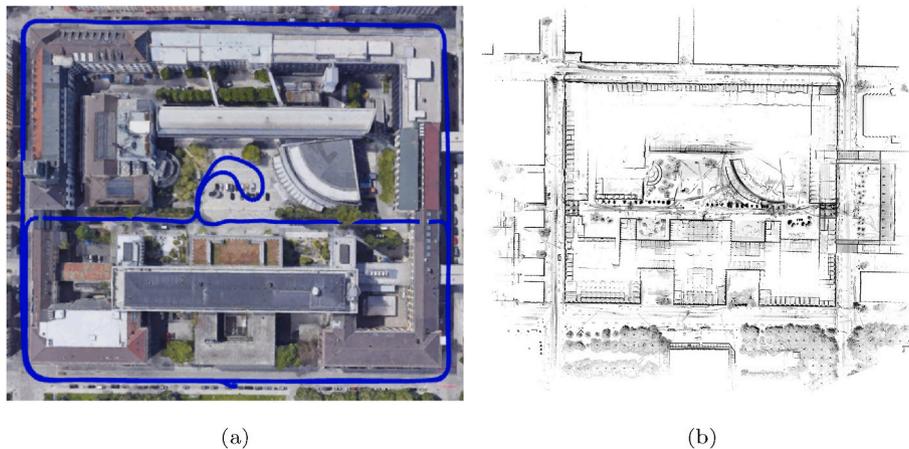


Fig. 4. Overview over the TUM-MLS data set. (a) Trajectory of the epoch of the TUM-MLS-2016 data set recorded in April 2016, visualized in Google Earth. (b) A section through the corresponding accumulated point cloud, without ground plane and roofs.

time the other input is negated:

$$\sigma_{disappeared} := f_{\wedge}(\sigma_{occupancy_1}, \overline{\sigma_{smoothed_2}}) \quad (15)$$

The use of smoothed representations is necessary to avoid false positives. As in the case of the confirmed elements, incorrect conclusions would arise due to registration errors in the underlying data.

#### 4.5. Combination with object classes

One of the great strengths of this approach is its ability to combine arbitrary spatial information. With respect to change detection, this is necessary because changes in different object classes are expressed in different ways. If a tree or a building is removed, the change can be clearly identified. If, on the other hand, a vehicle is moved, changes are only visible at both ends of the vehicle. Similar effects appear when a tree has grown or a building has been reconstructed. In order to be able to better assess and handle such effects during further processing and evaluation, it is helpful to consider the changes based on the different object classes. The present changes can be restricted to a given object class, such as *trees*, by linking both with an *and*-operation:

$$\sigma_{disappeared-trees} := f_{\wedge}(\sigma_{disappeared}, \sigma_{trees}) \quad (16)$$

It should be emphasized that distributivity does not play a role here. Isolating the trees in the changes found is identical to isolating the trees first and then determining the changes. The usefulness of the approach presented here becomes clear for a more complex example. Based on several epochs, it is possible to identify renovations of buildings that only took place between the second and the third epoch. This can be formulated as follows. For the sake of simplicity, an abbreviated notation was used, in which  $\wedge$  stands for  $f_{\wedge}(\cdot)$ .

$$\sigma_{renovations} := \left( \begin{array}{c} \sigma_{confirmed_{1,2}} \wedge \\ \left( (\sigma_{appeared_{2,3}} \wedge \sigma_{buildings_3}) \vee \right. \\ \left. (\sigma_{disappeared_{2,3}} \wedge \sigma_{buildings_2}) \right) \end{array} \right) \quad (17)$$

The result can further be linked to other sources of information, such as the field of view from a given position in order to identify all renovated buildings visible from a given point of view.

#### 4.6. Differentiation from other methods

In this section, the difference to other methods is elaborated. First, the runtime complexity of the change detection approach presented here is determined so that it can be compared with other approaches. Then the difference to ray-based methods is explained, which are another approach to change detection that takes free space into account.

##### 4.6.1. Runtime complexity

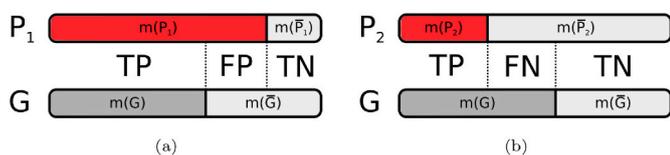
Since the prototype implementation used here uses parallelization extensively, runtime complexity is considered instead of runtime. The steps performed in the process have the complexities shown in Table 2, where  $N$  is the number of measurements in the point cloud. For reasons of completeness, factors and terms with lower importance are retained. The creation of an epoch first requires a preprocessing step that consists of distributing measurements over tiles ( $O(N)$ ), determining the number of traversed voxels using raycasting ( $\sim O(N)$ ) and calculating the medians for  $A$  and  $\bar{A}$  ( $\sim O(2N)$ ). For the sake of clarity, the average number of traversed voxels was not included in the terms.

An octree can be implemented in several ways, that is as a classical octree, a linear octree (Schrack, 1991) or as a hash map (Nießner et al., 2013). Linear octrees are faster than classical octrees, but have a very high memory consumption. Hash maps are memory efficient and fast, but the performance is not guaranteed. As can be seen in Table 2, the steps for inserting the measurements and generating the representation vary between the types of octrees. The overall complexity of the change detection approach is composed of the effort required for creating the representation of both epochs and an effort of  $O(2N)$  for performing the comparison using Algorithm 1. In total, this results in a runtime complexity that varies between linear and quadratic depending on the realization of the octree.

##### 4.6.2. Differences and similarities with ray-based methods

The comparison of the voxel-based method presented here with ray-based methods helps to better contextualize the former within the state of the art. In order for change detection to handle occlusions, free space must be considered. In approaches such as the one presented here, a raycasting operation is usually required to determine the latter. This operation is computationally expensive and can be considered as one of the major bottlenecks in the process chain. Approaches such as that of Xiao et al. (2015) and that of Underwood et al. (2013) do not require raycasting by employing a cylindrical and spherical reference frame, respectively. In both approaches, surface measurements can thus be identified which are in conflict with the free space information of the measurement rays. It is not excluded that change information can be integrated into the presented framework in the way outlined above. Currently, however, the authors do not see a way that allows to determine the memberships to the evidence sets in a way that allows for an intuitive and explainable *degree of ignorance*. Furthermore, omitting the explicit free space information also means that visibility calculations can no longer be reliably performed, although this is of limited relevance to the present work.

In addition to runtime complexity, another criterion for the comparison of voxel- and ray-based methods is the resolution. For voxel-



**Fig. 5.** 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.

based methods, the maximum resolution is usually limited by the voxel size. This results in a blurring of information that has little to great impact depending on the voxel resolution. This can also be used deliberately to construct resolution pyramids that in turn can be used to indicate areas of interest with low computational effort. The advantages of ray-based methods are that the maximum resolution corresponds to that of the sensor system and therefore there is no artificial limitation of the resolution as in voxel-based methods.

## 5. Experiments and results

### 5.1. TUM-MLS data set for change detection

The TUM-MLS data set was recorded in a cooperation between the Chair for Photogrammetry and Remote Sensing at the Technical University of Munich (TUM) and the Fraunhofer Institute of Optics, System Technologies and Image Exploitation in Ettlingen. It includes two epochs of LiDAR measurements of the TUM city campus and the adjoining street areas (cf. Fig. 4). The epochs were recorded in April 2016 and December 2018 and contain point clouds with 1.7 and 2.2 billion georeferenced mobile LiDAR measurements covering an area of approximately 29000 m<sup>2</sup>.

Both epochs were recorded with the MODISSA measuring vehicle (Borgmann et al., 2021). At the time of recording, it was equipped with two Velodyne HDL-64E LiDAR sensors mounted at an angle of 25° on the vehicle front roof. This allowed to record both the area at street level and building facades. The data sets consist of a sequence of individual 360° scans, with each scan including approximately 0.1 s of data acquisition. Each measurement has individually been georeferenced using the on-board Applanix POS LV navigation system that combines navigation data from two GNSS antennas, an inertial measuring unit and a distance measurement indicator (Diehm et al., 2020). Methods from graph-based SLAM have been applied in order to provide a reasonable good intra- and inter-epoch registration.

Two sets of labels were created manually. The first one is the *TUM-MLS Semantic Segmentation Benchmark* which comprises two subsets, one with semantic object classes and another with instance labels. A detailed description of the benchmark can be found in Zhu et al. (2020).

The second set provides annotations for both epochs describing the moving objects and changes in the scene. Each epoch is represented by a point cloud, which was created by accumulating all the individual measurements of the epoch and then annotated accordingly. Each point in each epoch is either labeled as *confirmed*, *appeared*, *disappeared*, as *unseen* in the other epoch or as part of a *moving object*. Said data set is published in the context of this work.<sup>1</sup>

### 5.2. Fuzzy confusion matrix

To compare the change detection results with the ground truth, two options are available. Either the fuzzy representation is sampled with

<sup>1</sup> The annotations and TUM-MLS data sets can be downloaded at <http://s.fhg.de/mls1>.

the ground truth and thus a *confusion matrix* is used to determine the difference between the two. The alternative is that the ground truth is transferred into the fuzzy domain and then compared to the results using an adapted version of the confusion matrix, which was developed specifically for this case and called a *fuzzy confusion matrix*. Before comparison, the representation with the ground truth should be sharpened, to eliminate deviations from the extreme values caused by the import procedure. While there is not much difference between the two approaches for the resolution at hand, especially for coarser resolution levels the influence of discretization errors cannot be neglected. Here, the method based on the fuzzy confusion matrix is more suitable, because in this case the ground truth has the same systematic errors as the results. The individual elements of the fuzzy confusion matrix are calculated based on the fuzzy measure:

$$\begin{aligned} TP &= \min(m(A_P), m(A_G)) \\ FP &= \max(0, m(\bar{A}_G) - (1 - m(A_P))) \\ FN &= \max(0, m(A_G) - (1 - m(\bar{A}_P))) \\ TN &= \min(m(\bar{A}_P), m(\bar{A}_G)) \end{aligned} \quad (18)$$

$A_P$  and  $\bar{A}_P$  denote the evidences of the fuzzy representation,  $A_G$  and  $\bar{A}_G$  those of the ground truth that has been converted into a fuzzy representation. For each pair of voxels of the fuzzy representations, all four variables are computed and then normalized so that their sum equals one. If two evidence distributions representing complete ignorance are compared, then this restriction does not apply because all four variables are zero.

The basic idea is that the individual cases *true positive*, *true negative*, *false positive*, and *false negative* are proportionally assigned for each pair of voxels. This is best illustrated by Fig. 5. True positives and true negatives correspond to the minimum overlap between estimate  $P$  and ground truth  $G$ . For the former, it is the overlap between  $A_P$  and  $A_G$ , for the latter, it is the overlap between  $\bar{A}_P$  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 positive* occurs whenever the  $A_P$  of the representation overestimates the  $A_G$  of the ground truth.

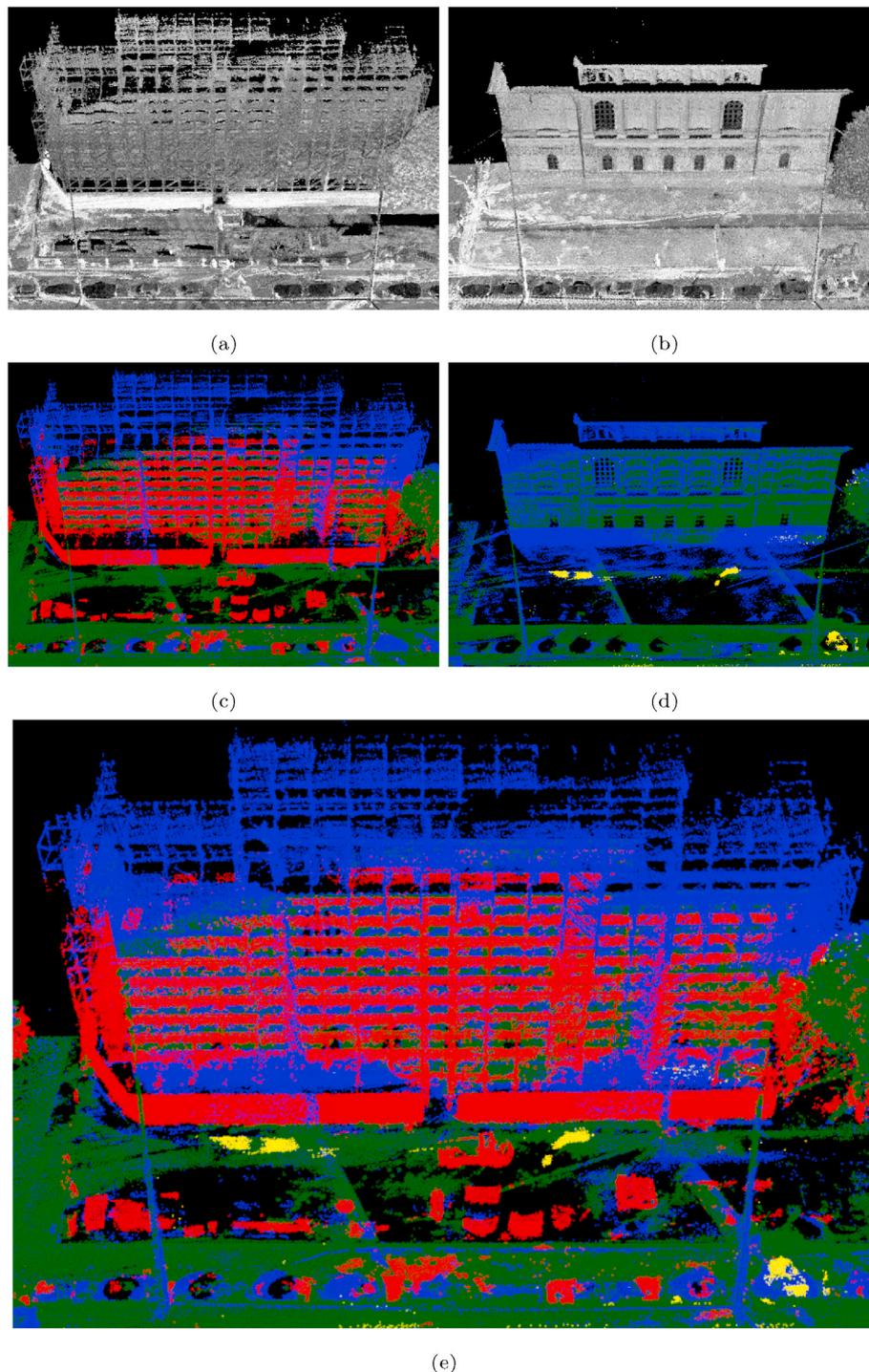
The results are then evaluated using the following metrics. Due to the imbalance between positive and negative samples, it was decided to use *Precision* and *Recall* as well as the *F1-Score*, which is based on both. A deeper explanation can be found in Saito and Rehmsmeier (2015).

### 5.3. Qualitative change detection

Due to renovation work carried out, the *Alte Pinakothek* shown in Fig. 6 is a good example to illustrate the capabilities of the approach presented in this work. The wall behind the site fence was not visible during the first epoch, so in the second epoch it is correctly marked as not seen, and not as changed. It is similar for areas along the facade, as they were covered by the walkways of the scaffolding. Both areas are marked in blue in the figure. Another noticeably large area is marked as not visible, that is the upper part of the scaffolding. However, since this section of scaffolding was recorded in the first epoch and the measuring vehicle took the same route in both epochs, a change is expected here. A review of the underlying representation revealed that there is no evidence for said area in the second epoch, although the area was in focus of the sensors.

The Arcisstrasse in Fig. 7 shows examples of parking behavior. The detected changes show vehicles that have used one of the parking spaces that are also visible in the image. The figure also illustrates changes in foliage, caused by different seasons. Even minor changes in the foliage can be seen. The construction site in the Gabelsbergerstrasse in Fig. 8 shows a situations that has an effect on the street environment. In the first epoch, not only a house is wrapped in scaffolding, one lane of the road in front is also occupied by a large concrete silo, which was surrounded by a site fence.

Fig. 9 shows a number of changes that were selected for different

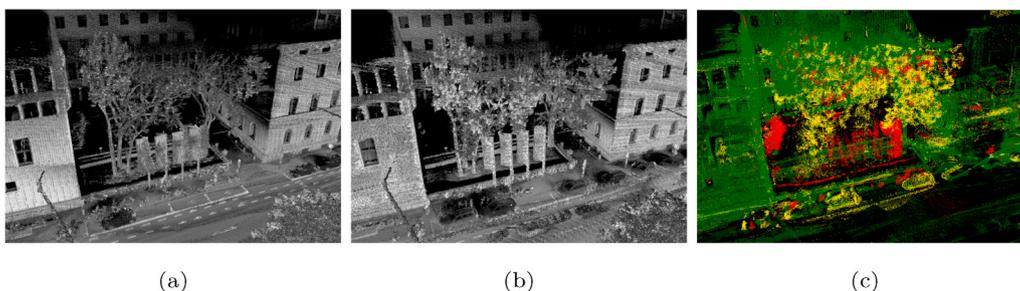


**Fig. 6.** Changes due to renovation work on the *Alte Pinakothek* in Munich. (a,b) Point clouds of both epochs. (c,d) Disappeared and appeared elements. (e) Overview of changes and unseen areas of both epochs (red = disappeared, yellow = appeared, blue = unseen in other epoch). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

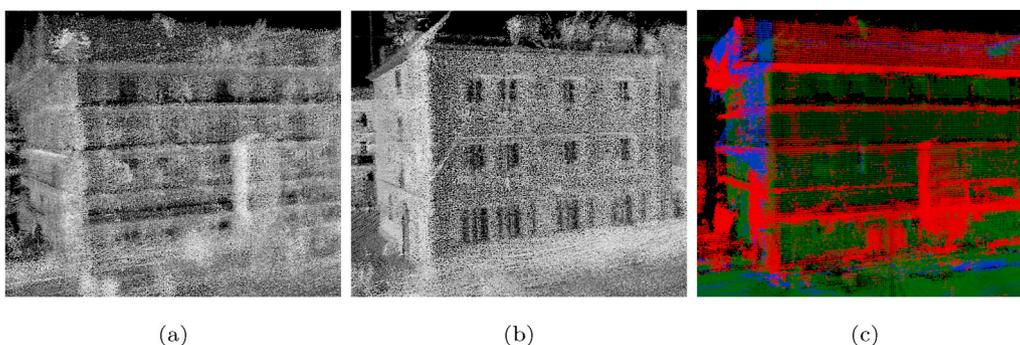
reasons. Row 9A shows flags on the TUM campus that appeared in the second epoch. In row 9B, an appeared Christmas tree in a window was detected. Row 9C shows a tree and a container that disappeared and were replaced by a new, freshly planted tree. It also shows the appeared person who is leaning against the wall in the house entrance at the center right side of the image.

Fig. 10a and b show the effects of uncertainty in the information derived from the measurement data. The degree of uncertainty is passed through to the change detection results. This behavior is desirable because the Boolean operations should not cause any distortion of the

underlying information. Figs. 10c and d illustrate the impact of contradictions on change detection results. Despite a high level of contradiction in the evidence of the occupancy grid visible in Fig. 10c, the changes are clearly pronounced. In a log-odd based procedure such as the one proposed by Moravec (1988), the evidences for and against the information would cancel each other almost completely out. It would not be possible to derive change from such occupancy grids.



**Fig. 7.** Changes in a street environment in the *Arcisstrasse* in Munich. (a,b) Point clouds of both epochs. (c) Overview of all changes and unseen areas (red = disappeared, yellow = appeared, blue = unseen in other epoch). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)



**Fig. 8.** Changes caused by a construction site in the street space in the *Gabelsbergerstrasse* in Munich. (a,b) Point clouds of both epochs. (c) Overview of all changes and unseen areas (red = disappeared, yellow = appeared, blue = unseen in other epoch). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

#### 5.4. Quantitative change detection

A section from both TUM-MLS data sets showing the *Alte Pinakothek* was selected as a test data set for the evaluation of our approach (cf. Fig. 6). This particular subset was chosen because large-scale construction work was carried out at one epoch of the dataset, for the purpose of which scaffolding and site fences were erected in large numbers. The section can therefore be regarded as statistically meaningful because of the large amount of both *changed* and *confirmed* geometry. Also, the changes are distributed across different object classes. It should be noted that most of the changes fall into the *disappeared* geometry class. However, since the operations used to identify *disappeared* and *appeared* objects are symmetric to each other, this is not a disadvantage. In all tests, the vegetation class was removed by applying logical operations such as  $\sigma_{\text{epoch}_{\text{occupancy without vegetation}}} := f_{\wedge}(\sigma_{\text{occupancy}}, \bar{\sigma}_{\text{vegetation}})$ . The treatment of this object class is a topic in itself and is handled in works such as that by Hirt et al. (2021). However, vegetation is still part of the qualitative evaluation and can for example be seen in Fig. 7c.

The fuzzy evidence grids representing the changes were derived as stated in Section 4. Values of 5.0 and 1.0 were selected for the steepness parameters  $k_{\text{occ}}$  and  $k_{\text{min}}$  of the logistic membership function. The comparison of the change detection results with the ground truth requires that the latter one is converted to a fuzzy representation for the reasons stated in Section 5.2. The comparison is carried out using the *fuzzy confusion matrix* and visualized with *precision-recall diagrams*.

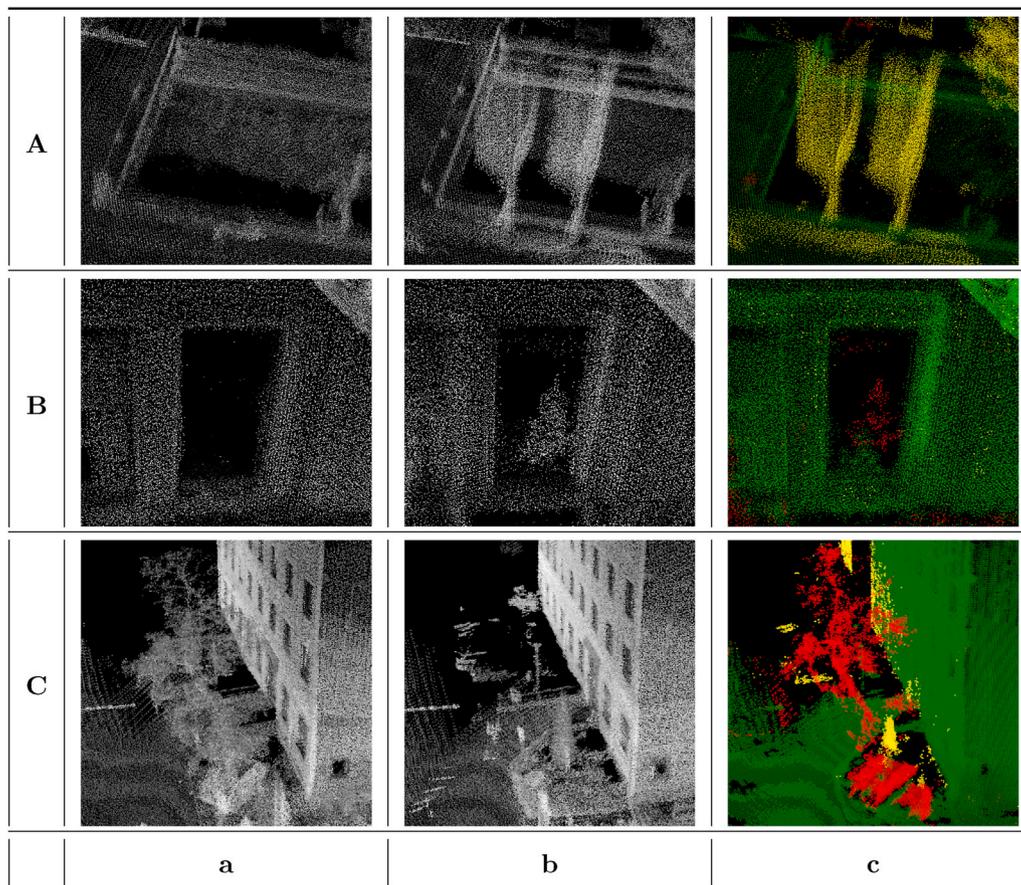
A series of experiments was performed to determine the maxpooling neighborhood. One experiment each was performed without maxpooling, with a maxpooling neighborhood of one voxel, and one with a neighborhood of two voxels. To investigate the effects of defuzzification, the same set of experiments was performed after applying *pro/contra filtering* to the change detection results. The precision-recall diagrams illustrating the results of both experiments can be found in Fig. 11. A clear difference in the results can be seen when comparing them before

and after defuzzification. While the difference is still relatively small when maxpooling is not used, there are clear differences when maxpooling is used. Specifically, there is an increase in recall for both *confirmed* and *changed* geometry as well as an additional increase in precision for *changed* geometry.

The values for the experiment with defuzzification, i.e. the bottom row of the precision-recall diagrams, are listed in Table 3. An examination of precision and recall shows that not using maxpooling leads to high precision, but only to a mediocre recall. Applying maxpooling with a neighborhood of one voxel is already sufficient to improve the recall significantly. The precision for *confirmed* geometry remains largely unchanged, but increases significantly for *changed* geometry. Increasing the maxpooling neighborhood from one voxel to two voxels improves the F1-score, if only a little. In most cases, this goes hand in hand with an improvement in recall and a slight reduction in precision. In order to examine this further, the evidence grid containing the disappeared geometry was visualized. An illustration of this can be found in Fig. 12.

It can be clearly seen that the results are significantly better when using maxpooling. Furthermore, it can 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. While a neighborhood of one voxel still shows significant clutter, a neighborhood of two voxels shows almost none. The neighborhood of the maxpooling does not seem to have any effect on the actual changes. A review of the evidence grids for confirmed geometry shows that there is no clutter.

Table 4 shows the change detection results for the *Alte Pinakothek* subset, subdivided by object classes. For the full subset, the *disappeared* case has a significantly higher F1-score of 0.89 than the *appeared* case, which has a F1-score of 0.30. A comparable effect is seen when the changes are broken down by object class. The reason seems to be a low recall, most likely because discretization errors on small objects turn a large part of all points into false negatives. This reflects the situation with the data set, here are many large *disappeared* objects and few small



**Fig. 9.** 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,b) Point clouds of both epochs. (c) Overview of all changes and unseen areas (red = disappeared, yellow = appeared, blue = unseen in other epoch). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

*appeared* included. Since *vegetation* has been removed and there are no changes available for *building interior* and *manmade terrain*, these cases are not listed in the table. In addition, for some classes there is no *confirmed*, *appeared* or *disappeared* geometry. It seems that if an object class has a large area and is accordingly represented with many samples, it is reflected positively in the F1-score. The high F1-score of 0.85 for *disappeared* geometry for the *building facade* reflects this. Similar behavior can be observed in case of *hardscape*, where a large construction fence has disappeared and the F1-score is 0.92. For smaller changes, as it is the case with *appeared* geometry in *natural terrain* and *scanning artifacts*, 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 the vehicles were annotated as modified in their entirety. *Confirmed* geometry does not occur, because it is not annotated in this data subset.

### 5.5. Comparison with other works

The comparison of the presented method with the state-of-the-art is not readily possible. To the best of the authors' knowledge, there is a lack of a change detection benchmark data set that includes the sensor position for each measurement point. The latter is necessary so that free space can be taken into account and thus changes can be distinguished from occlusions. Even if one would exist, the approaches mentioned in the state-of-the-art did not use it. To solve this problem, the authors of this paper would like to propose the hand-annotated ground truth used in this work as a benchmark data set for generic change detection methods. Said ground truth was published to complement the MLS point clouds of the TUM-MLS data set from Gehring et al. (2017) as well as the

semantic object classes from Zhu et al. (2020). The use of a common data set, as well as a corresponding evaluation approach such as the F1-score, allows the comparison of future works with the present method.

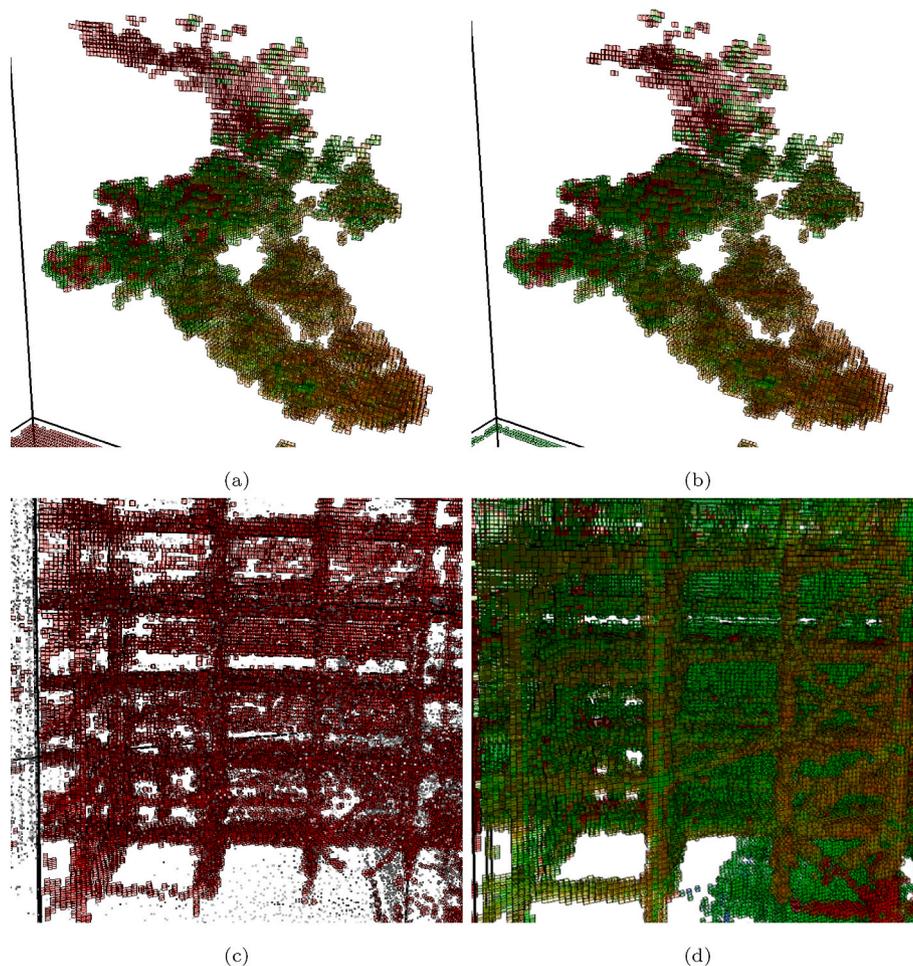
## 6. Discussion

### 6.1. Qualitative results

The Arcisstrasse in Fig. 7 serves as an example of how traffic situations, such as parking behavior, can be assessed using change detection. The changes show that there is a very high need for parking spaces in the second epoch compared to the first. It can be clearly seen that significantly more vehicles are parked along the street. Even some disabled parking spaces are used and a driveway is blocked. The figure also serves as an example to demonstrate the suitability of the change detection approach for assessing changes in vegetation, although, as mentioned above, these changes are not examined quantitatively. It turns out that changes in the foliage, which are due to different seasons, are clearly recognizable.

The construction site in the Gabelsbergerstrasse in Fig. 8 serves to demonstrate the suitability of our approach for the identification of situations that have an effect on the street environment. Based on the change detection, a construction site could be identified in the street area. Since this is an obstacle directly within the street space, the traffic was forced to evade and coordinate with the oncoming vehicles. Such information can be used in further applications, e.g. to evaluate the trafficability of the road for heavy transport.

The changes in Fig. 9 illustrate, among other things, specific properties of our approach. The Christmas tree was chosen because this



**Fig. 10.** Demonstration of the handling of uncertainty and contradictions. (a) Occupancy grid with uncertainties. (b) Derived changes, the uncertainties remain the same. (c) Contradictions in an occupancy grid. (d) Despite the contradictions derived changes.

change is small and inconspicuous and can therefore easily be overlooked. Even if this change is recognized by a human observer, then there is still a chance that it would be discarded subconsciously. The last row in the above mentioned figure was chosen for similar reasons. It shows an appeared person who is leaning against the wall in the house entrance at the center right side of the image. This change was initially overlooked when the annotations were created and is a good example that LiDAR data cannot always be reliably interpreted. Both cases are a clear argument in favor of using automated change detection.

### 6.2. Influence of defuzzification

The comparison of the experiments with and without defuzzification using the precision-recall diagrams in Fig. 11 suggests that defuzzification can rapidly improve the results. A closer look at the evidence grids shows that the fuzzy change detection results use a much wider range of the evidence interval than the largely crisp ground truth, which tends towards extreme values. The main reason for the discrepancy between results and ground truth can therefore be explained by the fuzziness of the data. After defuzzification, the evidence intervals of the results are in their extreme values, just like the ground truth, so all remaining errors are of a geometric nature. Or in other words, the errors no longer result from differences in fuzziness, but from all areas where result and ground truth do not match. This is a much more meaningful result in terms of change detection, because it highlights actual errors instead of

peculiarities of the representation.

### 6.3. Determination of the maxpooling neighborhood size

Looking at the results in Table 3, it is clear that the use of maxpooling leads to an increase in the quality of the results. This can also be clearly seen in the visualizations of the associated evidence grids in Fig. 12. If one only considers the F1-score as evaluation criteria, one comes to the conclusion that the actual size of the maxpooling neighborhood does not seem to play a role, since the results do not improve significantly. This would lead to the conclusion that a neighborhood of one voxel is to be preferred, since the computational effort involved is much smaller than it is the case for a neighborhood of two voxels.

However, the visualization of the evidence grids in Fig. 12 shows that a higher neighborhood leads to less clutter and thus to fewer geometric errors in the form of false positives. This leads to two conclusions. On the one hand, a neighborhood of two voxels is to be preferred for the chosen voxel size and data set, as this improves the results visibly. On the other hand, this leads to the conclusion that the confusion matrix has only limited use as an evaluation tool. The reason for this is probably that more qualitative aspects such as the clutter shown seem to be lost in the amount of data points.

Since no clutter occurs with confirmed geometry, a maxpooling neighborhood of one voxel is already sufficient here. Setting the parameters based on the insights gained in the last two sections, one gets

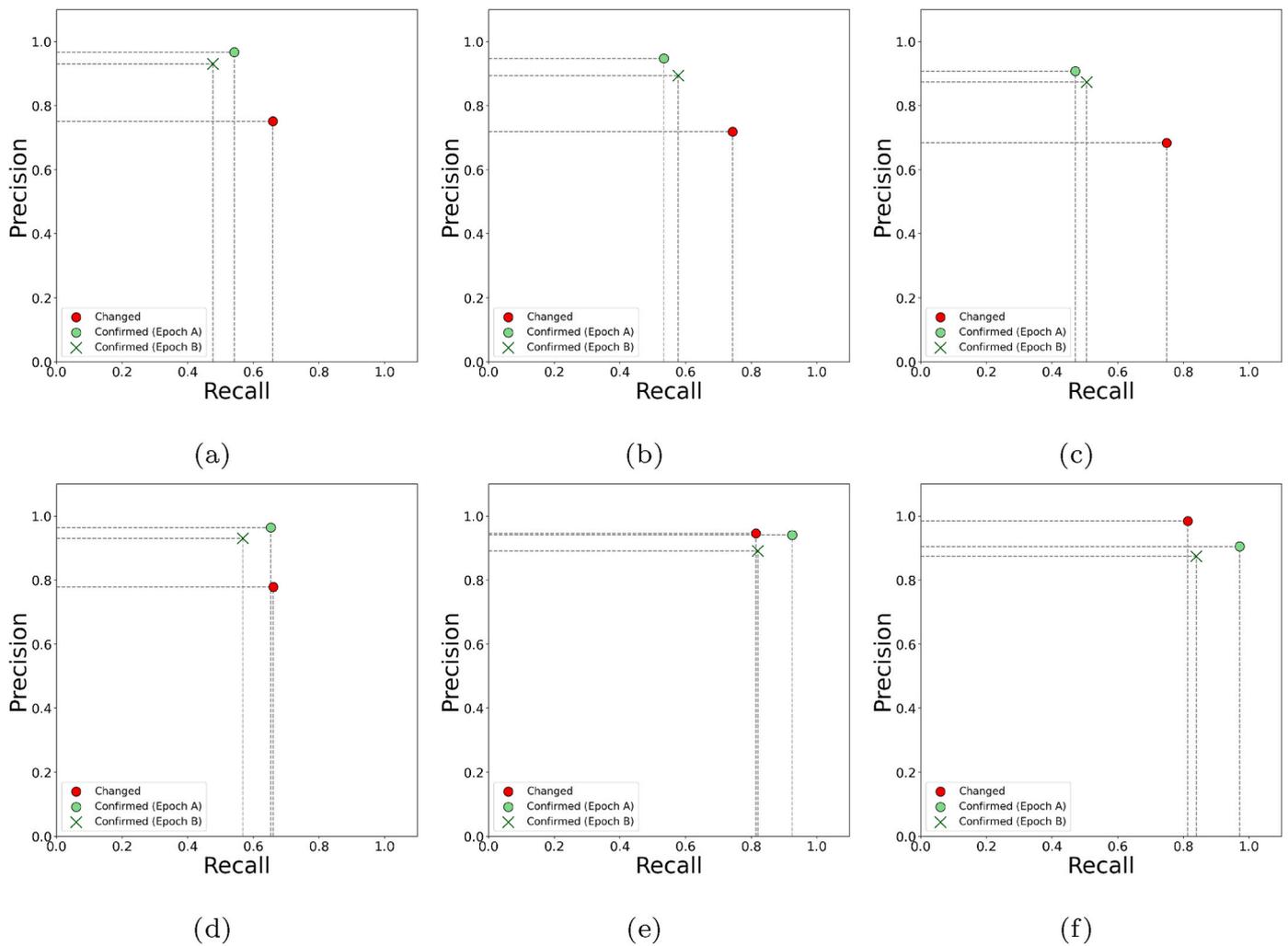


Fig. 11. Precision-recall diagrams for the change detection results with no defuzzification (top row) and defuzzification with pro/contra filtering (bottom row). (a,d) Results without maxpooling. (b,e) Results with a maxpooling neighborhood of one voxel and (c,f) two voxels.

Table 3

Results of the change detection on the *Alte Pinakothek* subset with no maxpooling and maxpooling with neighborhoods of 1 and 2. The results achieved with the final configuration of the approach are marked in bold.

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

the results marked in bold in Table 3. Please note that while the results were obtained for *disappeared* geometry, the conclusions can also be transferred to that of *appeared* geometry. As mentioned before, this can be explained by the fact that both cases are generated by operations that are symmetrical to each other.

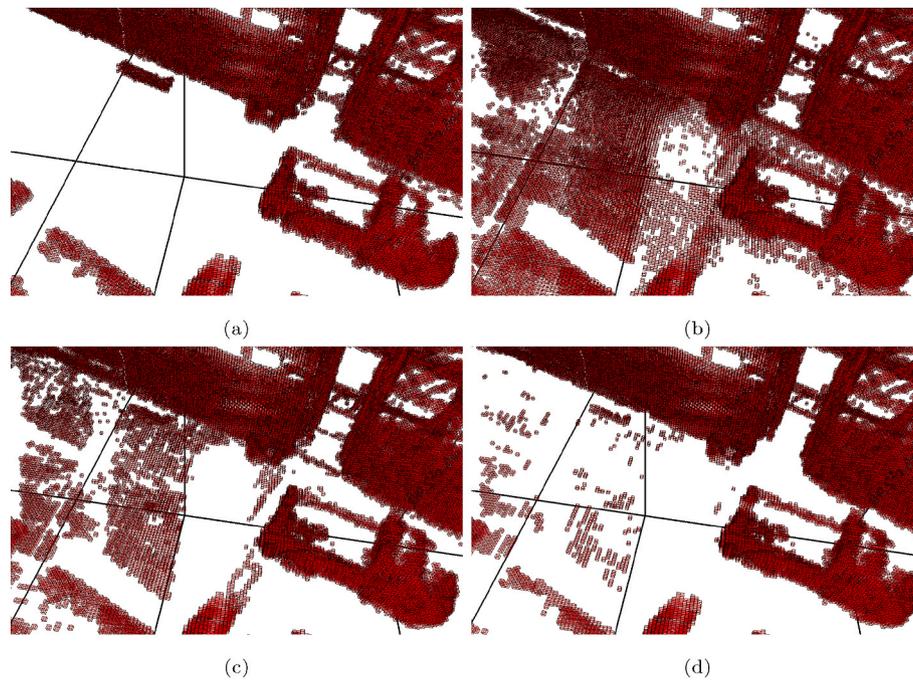
#### 6.4. Systematic effect of the sensor data evaluation concept

As can be seen from Fig. 6e, the upper part of the scaffolding was marked as unseen instead of changed, although the corresponding area was in the focus of the sensors in both epochs. This can be explained by the fact that the data used for change detection consists of *surface measurements*. This means that all free space in the data set is in the area between the respective sensor position and the measured surface point. Conversely, this means that free space is only recorded *whenever a*

*surface is in the background.*

Information about free space is required in order to distinguish between occlusions and changes. This is particularly relevant in an urban environment, since the efficient use of compact spaces leads to many occlusions. If there is no information about free space for the above reasons, then the distinction is not possible and as a result, the procedure marks corresponding areas accordingly, i.e. as *unseen* in one of both epochs. Fig. 13 illustrates this.

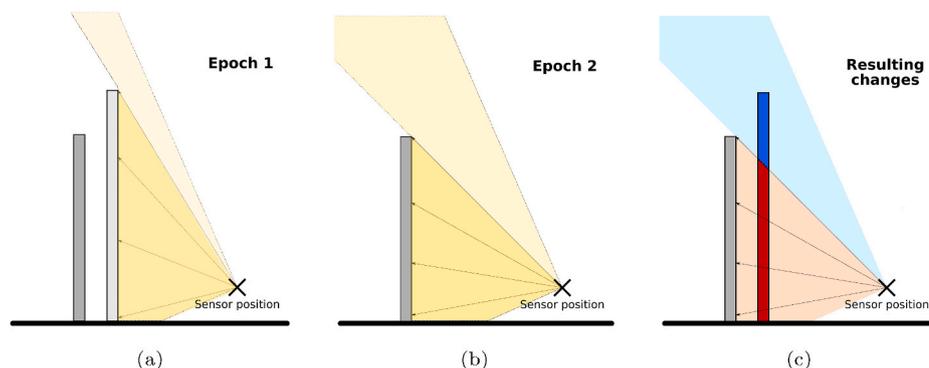
The implications of this effect are of practical relevance, since it occurs whenever a changed object is *measured against the sky or an open space*. The effect can be circumvented by storing a direction vector for all measurements, so that the free space can be extrapolated from this in the event of a missing sensor reading. Unfortunately, it was not feasible to subsequently apply the above solutions to the data at hand. Since the cause of the effect described here is due to missing sensor data and not to



**Fig. 12.** Influence of maxpooling neighborhoods on the results of the change detection. (a) Fuzzy representation of the ground truth. (b) Change detection results without maxpooling. (c) Results with a maxpooling neighborhood of one voxel and (d) two voxels.

**Table 4**  
Change detection results for the *Alte Pinakothek* subset, subdivided by object classes.

Subset	Confirmed (A & B)			Appeared			Disappeared		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
<i>Alte Pinakothek</i> 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



**Fig. 13.** Example to illustrate the apparently missing observations. (a) An object is measured in the first epoch. (b) The same location is measured in the second epoch, but the object is gone. (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. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

the performance of the overall system, the decision regarding the ground truth was made in favor of the method and the annotations were adjusted accordingly.

**7. Conclusions**

This work introduces the new technology of *fuzzy spatial reasoning* and demonstrates that it can be a powerful tool, not only in the study of urban street spaces but for spatial information processing in general. It has been shown that by employing simple logical operations, it is

possible to successfully locate changes in large buildings, provide the basis for the analysis of parking space requirements and determine areas such as road-side construction sites that have an effect on the street environment. Above all, the approach is able to detect subtle changes which can only be recognized with difficulty by a human observer, such as closed curtains, Christmas trees in windows, people standing in house entrances or filled holes on construction sites.

In our test data set, the method was able to identify *confirmed* geometry with an F1-score of 0.93 and *changed* geometry with an F1-score of 0.89. Experiments have shown that by using maxpooling, registration

errors and similar sources of error can be compensated to a certain extent. The authors came to the conclusion that the presented method is in some cases superior to humans in terms of its performance. This is not only because it can detect even small, inconspicuous changes, but primarily because it is able to access free space information, allowing it to distinguish between geometry observed only in one epoch and geometry that has actually changed. This is not easily possible for humans, since this three-dimensional information is difficult to visualize.

While the method used for evaluation fulfills its purpose, a detailed examination of the evaluation result suggests that it is only meaningful to a limited extent. Using excerpts from the data set, it was found that the confusion matrix often paints a much worse picture of the situation than is actually the case. An object-related evaluation concept would be more appropriate, as this corresponds more to human perception. However, since its implementation depends on the application that utilizes the change detection results, it is considered a subject of future work.

In terms of change detection, ray-based methods represent an alternative to the voxel-based approach used here, which has comparable capabilities but different characteristics. One of its advantages is that the maximum resolution is not artificially limited. From the authors' point of view, there is a great opportunity to apply the fuzzy reasoning developed within the scope of this work to ray-based methods as well. A hybrid process combining the advantages of both process classes is also conceivable.

It must also be mentioned that the effect described in this work, which stems from the processing of the sensor data, must be taken into account when constructing occupancy grids. If this is not the case, all changes recorded against the sky or open space will not be recognized as such. Due to its nature, the aforementioned effect plays a major role in urban street spaces in particular, since the conditions for it are more prevalent here.

## Funding

This work was funded by the Fraunhofer society and the German Ministry of Defence.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

The authors would like to thank Ann-Kristin Grossefinger and David Münch from Fraunhofer IOSB for their work related to the anonymization of MODISSA's sensor data. We also thank Yusheng Xu, Jingwei Zhu, Rong Huang and Zhenghao Sun from the Technical University of Munich for their efforts in annotating the TUM-MLS-2016 and TUM-MLS-2018 data sets. This work was carried out within the framework of the Leonhard Obermeyer Center (LOC) at Technical University of Munich (TUM).

## References

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

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)*, pp. 802–807. <https://doi.org/10.1109/IVS.2012.6232303>.

Basgall, P.L., Kruse, F.A., Olsen, R.C., 2014. Comparison of lidar and stereo photogrammetric point clouds for change detection. In: Turner, M.D., Kamerman, G. W., Thomas, L.M.W., Spillar, E.J. (Eds.), *Laser Radar Technology and Applications XIX; and Atmospheric Propagation XI*, vol. 9080. International Society for Optics and Photonics; SPIE, pp. 214–227. <https://doi.org/10.1117/12.2049856>. URL:

Borenstein, J., Koren, Y., 1991. Real-time map building for fast mobile robot obstacle avoidance. In: *Proc. SPIE 1388. Mobile Robots V*, 10.1117/12.25455.

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. *Appl. Opt.* 60 (22), F50–F65. <https://doi.org/10.1364/AO.423599>. URL: doi:10.1364/AO.423599.

Diehm, A.L., Gehrung, J., Hebel, M., Arens, M., 2020. Extrinsic self-calibration of an operational mobile LiDAR system. In: Turner, M.D., Kamerman, G.W. (Eds.), *Laser Radar Technology and Applications XXV*, 11410. International Society for Optics and Photonics; SPIE, pp. 46–61. <https://doi.org/10.1117/12.2556476>. URL: doi: 10.1117/12.2556476.

Douillard, B., Underwood, J.P., Melkumyan, N., Singh, S.P.N., Vasudevan, S., Brunner, C. J., Quadros, A.J., 2010. Hybrid elevation maps: 3D surface models for segmentation. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 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)*. IEEE, pp. 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. *Rem. Sens.* 8 (12) <https://doi.org/10.3390/rs8121030>. URL: <http://www.mdpi.com/2072-4292/8/12/1030>.

Elfes, A., 1989. *Occupancy Grids: a Probabilistic Framework for Robot Perception and Navigation*. Ph.D. thesis. Carnegie Mellon University.

Fairfield, N., Kantor, G., Wettergreen, D., 2007. Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *J. Field Robot.* 24 (1–2) <https://doi.org/10.1002/rob.20165>, 03–21. URL: doi:10.1002/rob.20165.

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, pp. 45–51. <https://doi.org/10.5194/isprs-archives-XLII-2-W1-45-2016>.

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, pp. 107–114. [10.5194/isprs-annals-IV-1-W1-107-2017](https://doi.org/10.5194/isprs-annals-IV-1-W1-107-2017).

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*, vols. IV-2, pp. 97–104. <https://doi.org/10.5194/isprs-annals-IV-2-97-2018>. URL: <https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/IV-2/97/2018/>.

Gehrung, J., Hebel, M., Arens, M., Stilla, U., 2019. A Representation of MLS Data as a Basis for Terrain Navigability Analysis and Sensor Deployment Planning, vols. IV-2. W7, pp. 39–46. <https://doi.org/10.5194/isprs-annals-IV-2-W7-39-2019>. URL: <https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/IV-2-W7/39/2019/>.

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*, umc 36-3/W19, pp. 30–35.

Gutmann, J.S., Fukuchi, M., Fujita, M., 2008. 3D perception and environment map generation for humanoid robot navigation. *Int. J. Robot Res.* 27 (10), 1117–1134.

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 J. Photogrammetry Remote Sens.* 86, 52–64. <https://doi.org/10.1016/j.isprsjprs.2013.09.005>.

Herbert, M., Caillias, C., Krotkov, E., Kweon, I.S., Kanade, T., 1989. Terrain mapping for a roving planetary explorer. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2. (ICRA), pp. 997–1002. <https://doi.org/10.1109/ROBOT.1989.100111>.

Hirt, P.R., 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). <https://doi.org/10.1007/s41064-021-00179-4>.

Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W., 2013. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Aut. Robots* 34 (3), 189–206. URL: <http://octomap.github.com>.

Huang, R., 2021. *Change Detection of Construction Sites Based on 3D Point Clouds*. Ph.D. thesis. Technische Universität München (TUM), Ingenieur fakultät Bau Geo Umwelt.

Kang, Z., Zhang, L., Yue, H., Lindenbergh, R., 2013. Range image techniques for fast detection and quantification of changes in repeatedly scanned buildings. *Photogramm. Eng. Rem. Sens.* 79 (8), 695–707.

Kay, T.L., Kajiya, J.T., 1986. Ray tracing complex scenes. *SIGGRAPH Comput Graph* 20 (4), 269–278. <https://doi.org/10.1145/15886.15916>. URL: doi:10.1145/15886.15916.

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

Meagher, D., 1982. Geometric modeling using octree encoding. *Comput. Graph. Image Process.* 19 (2), 129–147. [https://doi.org/10.1016/0146-664x\(82\)90104-6](https://doi.org/10.1016/0146-664x(82)90104-6). URL: doi:10.1016/0146-664x(82)90104-6.

Meyer, T., Brunn, A., Stilla, U., 2022. Change detection for indoor construction progress monitoring based on bim, point clouds and uncertainties. *Autom. Construct.* 141, 104442 <https://doi.org/10.1016/j.autcon.2022.104442>. URL: <https://www.sciencedirect.com/science/article/pii/S0926580522003156>.

- Moravec, H.P., 1988. Sensor fusion in certainty grids for mobile robots. *AI Mag.* 9 (2), 61. <https://doi.org/10.1609/aimag.v9i2.676>. URL: [https://ojs.aaai.org/index.php/ai\\_magazine/article/view/676](https://ojs.aaai.org/index.php/ai_magazine/article/view/676).
- Murakami, H., Nakagawa, K., Hasegawa, H., Shibata, T., Iwanami, E., 1999. Change detection of buildings using an airborne laser scanner. *ISPRS J. Photogrammetry Remote Sens.* 54 (2–3), 148–152. [https://doi.org/10.1016/S0924-2716\(99\)00006-4](https://doi.org/10.1016/S0924-2716(99)00006-4). URL: <http://www.sciencedirect.com/science/article/pii/S0924271699000064>.
- Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M., 2013. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph.*
- Pagac, D., Nebot, E., Durrant-Whyte, H., 1998. An evidential approach to map-building for autonomous vehicles. *IEEE Trans. Robot. Autom.* 14 (4), 623–629. <https://doi.org/10.1109/70.704234>.
- Payeur, P., Hebert, P., Laurendeau, D., Gosselin, C.M., 1997. Probabilistic octree modeling of a 3D dynamic environment. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2. (ICRA), pp. 1289–1296.
- Pfaff, P., Triebel, R., Stachniss, C., Lamon, P., Burgard, W., Siegart, R., 2007. Towards mapping of cities. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. (ICRA), pp. 4807–4813. <https://doi.org/10.1109/ROBOT.2007.364220>. URL: doi:10.1109/ROBOT.2007.364220.
- Pfeiffer, D., Franke, U., 2010. Efficient representation of traffic scenes by means of dynamic Stixels. In: *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 217–224. San Diego, CA.
- Roth-Tabak, Y., Jain, R., 1989. Building an environment model using depth information. *Computer* 22 (6), 85–90. <https://doi.org/10.1109/2.30724>.
- Ryde, J., Hu, H., 2010. 3D mapping with multi-resolution occupied voxel lists. *Aut. Robots* 28 (2), 169–185. <https://doi.org/10.1007/s10514-009-9158-3>. URL: doi:10.1007/s10514-009-9158-3.
- 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. <https://doi.org/10.1371/journal.pone.0118432>.
- Schachtschneider, J., Brenner, C., 2020. Creating multi-temporal maps of urban environments for improved localization of autonomous vehicles. *Int. Arch. Photogram. Rem. Sens. Spatial Inf. Sci.* <https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-317-2020>. XLIII-B2-2020:317–323. URL: <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLIII-B2-2020/317/2020/>.
- Schrack, G., 1991. Finding neighbors of equal size in linear quadrees and octrees in constant time. *CVGIP Image Underst.* 55 (3), 221–230. [https://doi.org/10.1016/1049-9660\(92\)90022-U](https://doi.org/10.1016/1049-9660(92)90022-U). URL: doi:10.1016/1049-9660(92)90022-U.
- 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. *Robot. Auton. Syst.* 45 (3), 181–198. <https://doi.org/10.1016/j.robot.2003.09.004>. URL: <http://www.sciencedirect.com/science/article/pii/S0921889003001556>.
- 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)*. IEEE, pp. 2276–2282. <https://doi.org/10.1109/IROS.2006.282632>. URL: doi:10.1109/IROS.2006.282632.
- Underwood, J.P., 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*, pp. 4735–4741. <https://doi.org/10.1109/ICRA.2013.6631251>.
- Weisbrod, J., 1996. *Unschärfes Schließen*. Ph.D. thesis. Universität Karlsruhe.
- Wilhelms, J., Van Gelder, A., 1992. Octrees for faster isosurface generation. *ACM Trans. Graph.* 11 (3), 201–227. <https://doi.org/10.1145/130881.130882>. URL: <http://doi.acm.org/10.1145/130881.130882>.
- Wolf, D., Sukhatme, G.S., 2004. Online simultaneous localization and mapping in dynamic environments. In: *2004 IEEE International Conference on Robotics and Automation 2004*, vol. 2. *Proceedings. ICRA '04*, pp. 1301–1307. <https://doi.org/10.1109/ROBOT.2004.1308004>.
- Wysocki, O., Hoegner, L., Stilla, U., 2022. Refinement of semantic 3d building models by reconstructing underpasses from mls point clouds. *Int. J. Appl. Earth Obs. Geoinf.* 111, 102841 <https://doi.org/10.1016/j.jag.2022.102841>. URL: <https://www.sciencedirect.com/science/article/pii/S1569843222000437>.
- Xiao, W., Vallet, B., Brédif, M., Paparoditis, N., 2015. Street environment change detection from mobile laser scanning point clouds. *ISPRS J. Photogrammetry Remote Sens.* 38, 38–49.
- Xu, S., Vosselman, G., Elberink, S.O., 2013. Detection and classification of changes in buildings from airborne laser scanning data. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, tome II-5/W2, pp. 343–348.
- Zeibak, R., Filin, S., 2008. Change detection via terrestrial laser scanning. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, pp. 430–435.
- 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. *Rem. Sens.* 12 (11), 1875. <https://doi.org/10.3390/rs12111875>.