

Article

# Automatic Detection of Objects in 3D Point Clouds Based on Exclusively Semantic Guided Processes

Jean-Jacques Ponciano <sup>1,\*†</sup>, Alain Trémeau <sup>2,†</sup> and Frank Boochs <sup>1,†</sup>

<sup>1</sup> i3mainz, Institute for Spatial Information and Surveying Technology, Mainz University of Applied Sciences, D-55128 Mainz, Germany; frank.boochs@hs-mainz.de

<sup>2</sup> Hubert Curien Laboratory, University Jean Monnet, 18 Rue Professeur Benoît Lauras, 42100 Saint-Etienne, France; alain.tremeau@univ-st-etienne.fr

\* Correspondence: jean-jacques.ponciano@hs-mainz.de; Tel.: +49-1517-1943626

† These authors contributed equally to this work.

Received: 17 August 2019; Accepted: 29 September 2019; Published: 8 October 2019



**Abstract:** In the domain of computer vision, object recognition aims at detecting and classifying objects in data sets. Model-driven approaches are typically constrained through their focus on either a specific type of data, a context (indoor, outdoor) or a set of objects. Machine learning-based approaches are more flexible but also constrained as they need annotated data sets to train the learning process. That leads to problems when this data is not available through the specialty of the application field, like archaeology, for example. In order to overcome such constraints, we present a fully semantic-guided approach. The role of semantics is to express all relevant knowledge of the representation of the objects inside the data sets and of the algorithms which address this representation. In addition, the approach contains a learning stage since it adapts the processing according to the diversity of the objects and data characteristics. The semantic is expressed via an ontological model and uses standard web technology like SPARQL queries, providing great flexibility. The ontological model describes the object, the data and the algorithms. It allows the selection and execution of algorithms adapted to the data and objects dynamically. Similarly, processing results are dynamically classified and allow for enriching the ontological model using SPARQL construct queries. The semantic formulated through SPARQL also acts as a bridge between the knowledge contained within the ontological model and the processing branch, which executes algorithms. It provides the capability to adapt the sequence of algorithms to an individual state of the processing chain and makes the solution robust and flexible. The comparison of this approach with others on the same use case shows the efficiency and improvement this approach brings.

**Keywords:** object detection; semantic; OWL; SPARQL; ontology; point cloud processing

## 1. Introduction

The real world is increasingly being digitized by various and continuously evolving sensor systems that produce vast amounts of data (e.g., images, point clouds). Several factors characterize this data. These factors depend on the sensor used, its characteristics and the way it is used on the one hand and the real-world objects and their geometrical and physical structure on the other hand. In addition, other external factors (such as the illumination of the scene) depend on the acquisition context. All these factors have an impact on the representation of the objects inside the data sets. Moreover, these factors vary broadly for different acquisition cases. For example, a 3D point cloud acquired by laser scanning acquisition technology as used in Reference [1] is dense, no noise and composed of regular geometrical surfaces. In contrast, a 3D point cloud acquired by a smartphone as used in Reference [2] or by a Kinect as used in Reference [3], often has a lower quality. This lower

quality leads to low and sparse density, much noise and the apparition of broken and irregular surfaces. Therefore, the object representation inside the data varies greatly.

The variation of object representation inside the data increases the difficulty of automatic processing techniques to detect and identify objects, since they have to manage every representation of objects. That is why many solutions are adapted and restricted to specific situations to limit the degrees of complexity introduced by this variation. An ideal solution should be very flexible to manage many cases. In addition, the more cases there are, the more human effort should be made to detect objects through a supervised or semi-supervised approach. Therefore, an ideal approach should be automatic to adapt the detection process itself to the diversity of cases. Indeed, such flexible solutions would need to consider all factors that impact the acquisition process. In order to detect an object, an automatic approach has to identify data characteristics that impact the object representation inside the data. Thus, detection processes contain three steps:

**Preprocessing:** improves the representation of the object inside the data (e.g., noise cleaning, transformations)

**Segmentation:** uses a strategy (model-driven/data-driven/learning-driven), based on certain assumptions helping to detect characteristics in the data sets which lead to the objects

**Identification:** evaluates the elements found (based on the characteristics provided by the detection process) and links it to the object categories

### 1.1. Related Work

The review in Reference [4] presents the most common approaches for preprocessing 3D point clouds. These approaches mainly perform filtering and denoising to improve the representation of objects in the data.

Then, for the segmentation step, some model-driven approaches, such as References [5,6], use a Hough transform adapted to 3D [7] to detect planes and identify the floor and buildings in order to reconstruct them in 3D. Other approaches such as that in Reference [8] use RANDOM SAMple Consensus (RANSAC) [9] to detect shapes (such as planes, spheres and cylinders).

The approach in Reference [10] projects and transforms the data into a 2D graph to segment elements according to local convexity criterion. The approach in Reference [11] uses a 3D voxel grid method to rasterize the point cloud after a first segmentation of the floor. However, these segmentation approaches do not use the information on the shape and surface of the objects to be segmented. They lose accuracy and therefore cannot be used to segment all types of object in different data contexts.

On the contrary, another approach [12] segments the data according to their smoothness. This approach provides accurate results but it is not suitable for a vast data set because it is quite time-consuming. Other more generic and traditional approaches such as region growing [13] segment all types of objects according to standard criteria but also segment the objects into several portions. Other approaches ([14,15]) segment point clouds in the domain of building information modeling (BIM). These approaches, similar to the majority of approaches in the BIM domain as presented by Reference [16], are mainly based on data characteristics for segmentation. Then these approaches mainly use object characteristics to identify the generated segments. The relevance of these approaches depends entirely on the data characteristics and the objects sought.

The data-driven methodology is composed of machine learning approaches and heuristic approaches. With the increase of 3D data and efficiency provided by machine learning, the machine learning approaches are being used more and more. The machine learning approach is comprised of a large variety of techniques. The machine learning approaches based on convolutional neural networks (CNN) like in Reference [17] and region-based convolutional neural networks (R-CNN) like in Reference [18] provide accurate object detection. In addition, the approach in Reference [19] based on ResNets [20] produces fast and efficient object detection in images. However, approaches using an R-CNN like in Reference [21] are only usable on dense data sets and organized into a tensor structure. Therefore they are mainly not applicable to 3D point clouds. The VoxelNet approach

in Reference [22] proposes an adaptation to 3D by using a voxel grid to obtain proper object detection. Similarly, the approach in Reference [23] adapts the convolutional network to the 3D. The approach in Reference [24] based on PointNet [25] uses the location of objects to improve their detection. Other approaches in References [26,27] combine 2D and 3D data to increase detection quality. Among the machine learning approaches, the approaches of References [28,29] obtain excellent results in the application case of room detection in a 3D modern building. That is why Section 4 uses the result of their works to compare with the result of the approach presented in this paper. However, all machine learning approaches require training on annotated data.

If annotated data are not available or not sufficient, heuristic approaches are superior. The detection of shapes and geometric characteristics of objects are the main basis of heuristic approaches. For example, the approach in Reference [30] combines several preprocessing, segmentation and classification algorithms to detect switches in a railway point cloud. These approaches produce accurate object detection. These approaches use a single sequence of algorithms to detect all objects in the data. Therefore these approaches adapt neither to characteristic and representation variations of the object type, nor to variations of data characteristics.

Through the emergence of the Semantic Web and the development of semantic technologies that formalize and use knowledge representations, the ontology-driven approaches have increased in different domains of computer vision. Among the semantic technologies, the Web Ontology Language (OWL) [31] allows a logical and explicit representation of knowledge in any particular domain of interest [32] by defining an ontology composed of classes and individuals. The use of semantic technologies in computer vision allows for introducing knowledge and use of this knowledge in the object detection process to improve its flexibility. Designing an object detection process based on explicit knowledge allows for adapting the process according to the knowledge content. Thus, an addition to or a change in knowledge representation implies a process adaptation facilitating a behavior adapted to new situations. The integration of explicit knowledge in current research work of computer vision appears to mainly support the classification process through the description of objects but also to support the segmentation process through knowledge about data, object and algorithm.

The semantic-based methodology of classification first segments a point cloud. Secondly, it estimates geometric properties of segments to generate ontologies expressed in OWL. Thirdly, it applies reasoning to classify and gather segments based on their features' description inside the knowledge base. This methodology is used in the cultural heritage domain ([33,34]) but also other domains as a straightforward extension of a previous approach in Reference [35]. Some variations of reasoning techniques and tools can be observed among the different approaches using this methodology. Contrary to Reference [35], the approach in Reference [36] uses SWRL rules to reason and semantically annotate railway objects (e.g., tracks, signal) in a 3D point cloud data set instead of OWL constraints [37]. This variation of semantic technologies in the semantic methodology mainly has an impact on the reasoning performance of the classification. However, each of them benefits from the advantage of a classification adapted to the defined knowledge that requires only a knowledge representation adaptation to classify new objects or process new data that would contain other object representations. This semantic-based methodology of classification is based on a classical segmentation processing that limits the performance of the semantic classification if the segmentation process is not adapted to the object to detect and its representation in the knowledge base. Therefore some approaches, such as References [38,39], combine the segmentation and classification process simultaneously through ontology-driven approaches. The use of explicit knowledge about data, object and algorithm in the segmentation process aims at providing adaptability in the selection of the algorithm to segment the data and thus allows an automatic adaptation of the segmentation process.

Nevertheless, data can contain diverse representations of the same object. The variation of object representation is due to the characteristics of the data or more precisely to its acquisition that can impact the object representation in the data. For example, an occlusion or a variation of density

inside the data lead to different representations of the same object. Thus, a complete and efficient semantic process of detection requires modeling every possible representation of an object inside the ontology to obtain an accurate object detection. The existing approaches are limited to a small set of representation descriptions due to the complexity and the variety of representation cases. The research work presented in this paper aims at addressing this limit by enriching the knowledge base through automatic reasoning and learning to improve object detection.

### 1.2. Contributions

The contribution of this paper is to solve the adaptation issues through the use of semantic technologies to provide an automatic detection process according to objects and data characteristics. The presented approach adapts algorithms for every combination of specific objects and data. This approach improves the previous approach [39] firstly by adding knowledge about data acquisition to deduce information about data characteristics and object representation inside the data. This knowledge aims at supporting the adaptation of the detection process to the variations of data characteristic and object representation inside the same data. Secondly, it improves the detection process's flexibility through a step-by-step process that considers the algorithms' results executed in the previous steps, instead of creating a sequence of algorithms. These two added values provide better flexibility and adaptability of the detection process according to data and object characteristics. However, the flexibility and the adaptability of the process cause its efficiency to be related to the quality of the knowledge base content, which cannot contain all possible cases and variations. That is why this approach also makes a contribution to a self-learning step based on knowledge. This learning uses experience provided by the detection process to enrich the knowledge according to the specificity of the use case. This enrichment aims at improving the detection process by adapting it to the use case. The improvement of detection process provided by these contributions is shown in Section 3. Moreover, the results of this approach are compared to the results of other approaches, on a data set representing a modern building, in Section 4.

## 2. Materials and Methods

The semantic method proposed is first presented through a system overview in Section 2.1 that explains it and shows the components of the system. Then Section 2.2 presents the knowledge representations that drive the detection process. This knowledge describes the three main domains, which are the algorithm, the object and the data domains. Finally, Section 2.3 explains the overall process of knowledge-driven detection. The knowledge-driven detection process is composed of three steps: the data processing through algorithm management, the classification and the self-learning.

### 2.1. System Overview

The proposed method is a semantic approach that uses knowledge domains of data, algorithm, objects and acquisition processing to drive the detection process. A knowledge analysis engine that allows for manipulating this knowledge is connected to a processing engine that executes algorithms chosen by the knowledge analysis engine. The reasoning that understands the requirements of each algorithm parameterizes the selected algorithms automatically. This processing works step by step to provide adapted processing at each step. This specific processing depends on the objects sought, the characteristics of the data (e.g., its content, its specific context, its acquisition process) and the results of the executed algorithms. It takes into account algorithms inputs, algorithms prerequisites, algorithms outputs, algorithms parameters and conditional-restrictions of algorithms interdependencies, in order to select algorithms adapted to each step. This strategy makes a detection process very flexible thanks to the full treatment under the regime of semantics. Such processing that selects and parameterizes the adapted algorithms step by step results in the execution of a sequence of algorithms specific to the processed use case. This capability of adaptation finally leads to a robust detection process. Thanks to the defined knowledge that guides the overall

process, the detection process can manage unexpected situations through a self-knowledge-based learning step, which proposes a possible continuation.

Several aspects must be taken into account to design such a detection process. First, a global process must drive the entire detection process by using explicit knowledge. Therefore, the whole process requires techniques to formulate and use this knowledge. Next, the global process needs algorithms to perform the processing. It must automatically assess the relevance of each algorithm to the specific detection context to select the most appropriate algorithm to run. The use of algorithms with knowledge requires an understanding of each algorithm's parameter to relate a parameter value to the value of an object or data characteristic. Then the overall process must use the relationship between the algorithm parameters and the characteristics of the objects or data to configure the algorithms. Moreover, the dynamic adaptation of the detection process requires the automatic execution of algorithms and the transformation of algorithmic results into a field of knowledge. To enrich the knowledge base, the automatic execution of algorithms and the updating of knowledge by interpreting algorithmic results require a technical and conceptual connection between the semantic paradigm and the algorithmic paradigm. Finally, conclusions must be drawn from the results obtained. These conclusions should not only serve to identify elements but should also guide the detection process to increase its effectiveness. Indeed, an analysis of these conclusions must be carried out to draw lessons from the experience from which these conclusions were drawn.

Figure 1 shows the main components of the presented system that addresses these requirements.

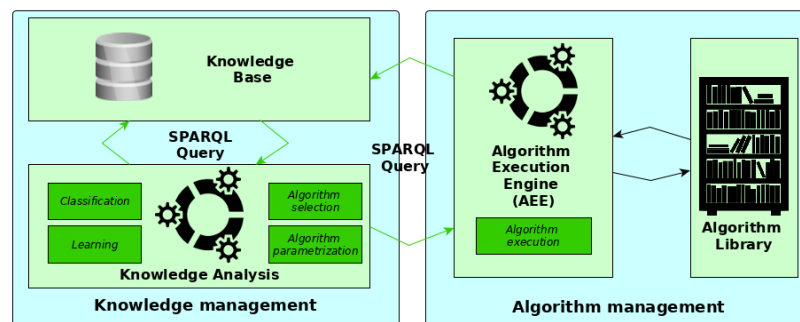


Figure 1. System Overview.

The proposed system is composed of algorithm libraries containing a multitude of algorithms that allow covering the processing of different 3D data. Each of these algorithms has a detailed description modeled in the knowledge base stored in a triplestore.

This knowledge is used and managed through a knowledge analysis engine (KAE), which aims at guiding the entire object detection process and adapting it to the data provided and the objects sought. This adaptation of the detection process begins with the selection of the most relevant algorithms, among all the algorithms available in the algorithm libraries. The estimation of this relevance depends on the characteristics of the objects sought and the data used. The selected algorithms are then automatically configured by KAE according to their execution context. In addition to the characteristics of the objects sought and the data used, the context of algorithm execution depends on the results obtained upstream from the execution of other algorithms. After the full parameterization of an algorithm, the KAE converts the algorithm information into SPARQL queries [40] using a SPARQL-function (similarly to the approach in Reference [41]) to call the execution of the algorithms.

An algorithm execution engine (AEE) uses information in the received SPARQL query to execute the algorithm related to this query. It then interprets the result to be returned through the SPARQL query to enrich the knowledge base. This enrichment then allows the KAE to reason on the new information to draw conclusions and identify objects through classification.

This step is followed by an analysis of the results to learn dynamically from the experience of achieved results. This learning then allows for specifically improving knowledge about objects and data, depending on the specific application context.

## 2.2. Knowledge Modeling

Knowledge is structured and modeled to guide the process of data understanding. This knowledge is defined as a knowledge base through the semantic standard OWL2 [42] which is an improvement on the semantic standard OWL [43]. Figure 2 shows the organization of knowledge.

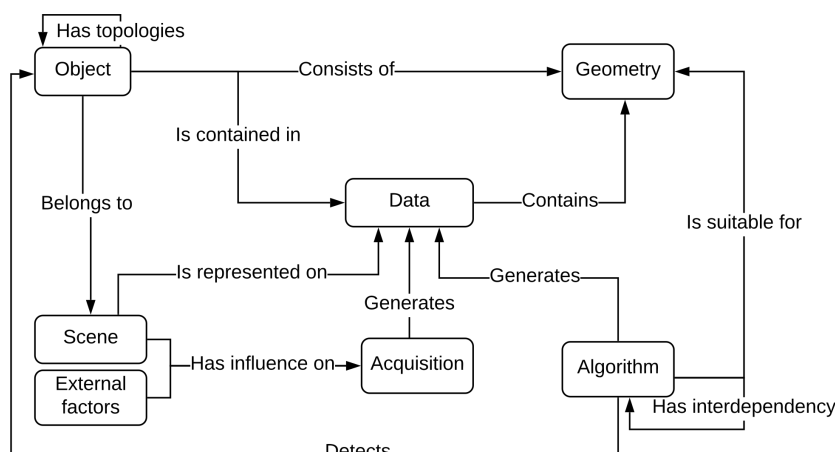


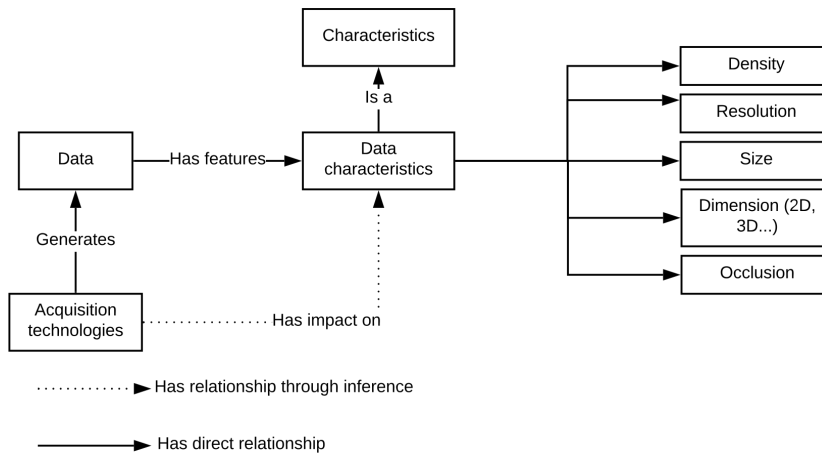
Figure 2. Overview of the knowledge structure.

Reference [44] presents the requirement of the knowledge modeling for the design of the object detection process. It is necessary to detect the geometric characteristics of the objects (e.g., shape, surface) and their topologic relationships (e.g., distance between objects, connection, perpendicular) in order to identify the objects that constitute the data. The detection of these characteristics requires the use of algorithms. Algorithms are designed to identify geometries (e.g., plane, line, sphere, segments, orientation). They provide new data (e.g., sampled data, filtered data) or new data characteristics (e.g., segments, density).

Let us consider an algorithm designed to detect planes (such as RANSAC or the Hough transformation) as an example. This algorithm generates segments (data parts) from data where each segment represents a plane. Thus it allows the detection of different objects with a planar geometry (e.g., wall, ceiling, floor, table, door). The behavior of algorithms depends mainly on the characteristics of the data (e.g., size, density), which depend on the acquisition process. However, the acquisition process is influenced on the one hand by the scene containing the objects to digitize (for example, any object may occlude another) and on the other hand, by various external factors (for example, light intensity and color, calibration of the measuring instrument, indoor or outdoor environment). Moreover, knowledge is organized hierarchically (i.e., one element can be a subset of another). For example, a vertical wall is a kind of wall that is a kind of object.

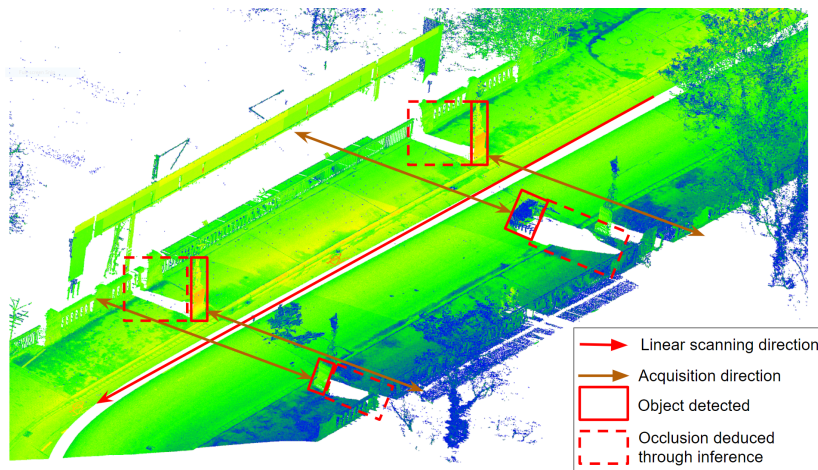
### 2.2.1. Data Knowledge

As shown in Figure 2, data is a representation of a scene, obtained through acquisition. This acquisition process, more precisely the instruments and the methods used for the acquisition, influences the data characteristics. In addition, elements influencing the acquisition process itself, such as external factors and the acquired scene, transitively impact the data characteristics. The data characteristics such as density, noise, resolution, dimension, size and occlusion comprise the specificity of each data. Figure 3 shows the generic semantic description and the relation between any data and its acquisition process.



**Figure 3.** Semantic description of the data and of the acquisition process.

The specificity of the data has an impact on the process of understanding the data content. That is why the characteristics of data must be considered to guide the process of understanding. The semantic representation of the relationship between data and its acquisition process allows for inferring data characteristics from the information of the acquisition process. Thus, information about the acquisition process indirectly guides the process of detection through the data characteristics. Let us take a striking example in an outdoor scene acquired by the Lidar laser scanner. This example is illustrated in Figure 4.



**Figure 4.** Illustration of occlusion inference.

This example shows an acquisition process by horizontal scanning (see brown arrows in Figure 4) along a linear path (illustrated by the red arrow in Figure 4). Such acquisition process using a Lidar laser scanner as an acquisition instrument systematically produces a local loss of information caused by objects that occlude parts of the scene. The loss of information often translates into an over-segmentation (the same object segmented into several parts) and detection failure during the process of data understanding.

However, a reasoning mechanism can anticipate and compensate for this loss of local information. In this example, the reasoning allows for inferring the location of possibly occluded areas (highlighted by the red dotted rectangle in Figure 4) from the position of the measuring instrument and the detection of an object (illustrated by the red rectangle in Figure 4) in the alignment of the horizontal scan. Thus, reasoning on information about the acquisition process (acquisition method, measuring instrument, the content of the acquired scene) allows for anticipating and locating the occluded areas. The identification of occluded areas provides information that improves the detection

process. For example, it allows for gathering each part of an object which was separated due to some occlusions into the same object. It also provides essential information that influences the selection of algorithms through relations between algorithm requirement and the characteristics of the data.

### 2.2.2. Object Knowledge

The understanding of data requires a description of the objects contained in the acquired scene to facilitate the object searching and thus the data content understanding. This object description is semantically represented inside the knowledge base. The description of object characteristics aims at guiding the strategy of object detection. Indeed, the simpler the object shape is, the simpler its detection strategy can be. On the contrary, objects with complex shapes require more elaborate detection strategies.

Three main parts compose the semantic description of objects: object characteristics, its geometry and the scene that it belongs to. They aim at facilitating the adaptation of detection strategies. Figure 5 illustrates the generic semantic description of an object.

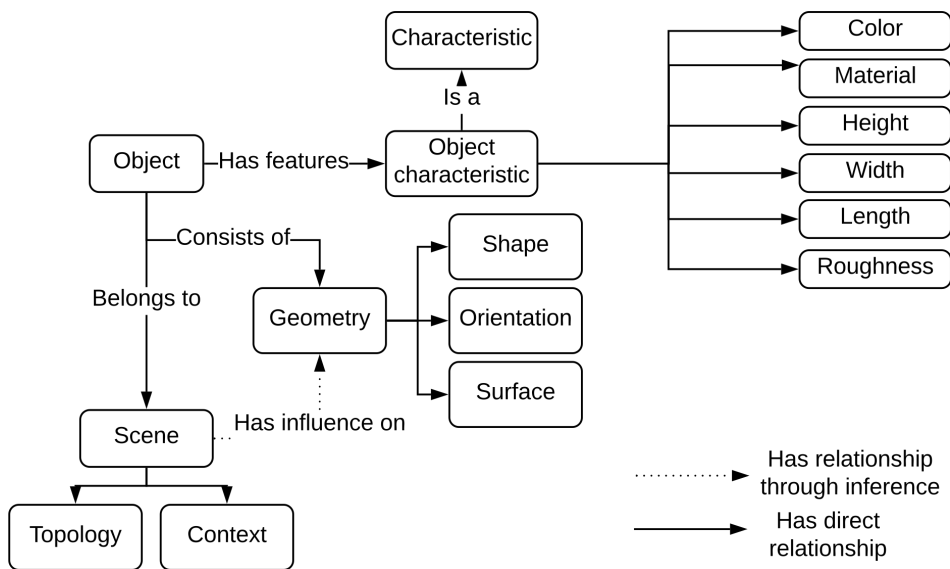


Figure 5. Semantic description of any object.

Characteristics are the first part of the object description. These characteristics are specific to each object and influence the process of acquisition (e.g., size, color, material). Let us take as an example the acquisition of an indoor scene with a transparent glass table. Figure 6 shows such scene acquired with laser scanner technologies by the company NavVis (NavVis: <https://www.navvis.com/>).

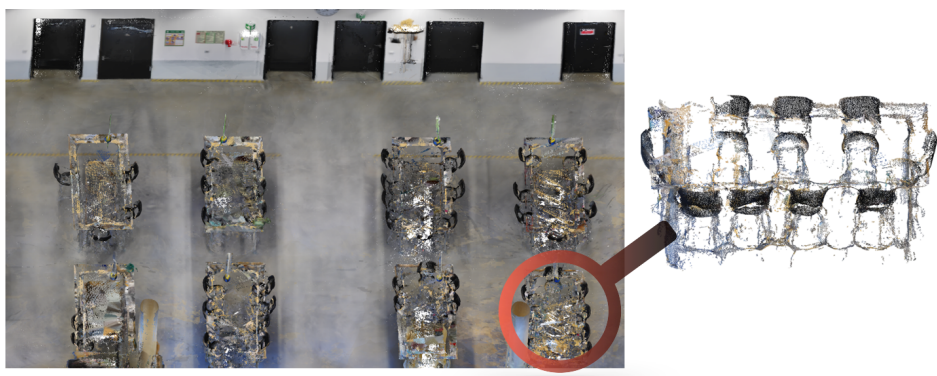


Figure 6. Point cloud from the company NavVis representing a room composed of glass tables with chairs.



The glass tray of the table presented in this figure has an insufficient density of points to allow the detection of the table. On the contrary, the acquisition of a matt object such as chairs in Figure 6 provides a proper density of points that facilitates its detection.

The Equation (1) shows a formal logical rule that compares the characteristics (?c) of the object (?o) with the acquisition technologies (?t) to automatically infer the density variation in a data (?d). Moreover, an object represented with a low density means that several segments represent it after the data segmentation. The knowledge base contains a class called SegmentsSet, defined as composed of several similar segments (having similar characteristics). Therefore, if an object is represented with a low density, then it is a set of segments.

$$Data(?d) \wedge Object(?o) \wedge LaserScanner(?t) \wedge GlassMaterial(?c) \wedge isContainedIn(?o, ?d) \wedge hasCharacteristics(?o, ?c) \wedge generates(?t, ?d) \Rightarrow hasLowDensityFor(?d, ?o) \Rightarrow SegmentsSet(?o) \quad (1)$$

The geometries representing the objects constitute the second part of the object description. A shape (e.g., rectangular, triangular, cubic, cylindrical, spherical, free), an orientation (e.g., vertical, horizontal, oblique) and a surface (e.g., regular, irregular, planar, linear) are the components of the geometry definition. Moreover, an object can be defined as a compound of several objects.

Let us continue with the example of the table to illustrate the geometry part of its semantic description. A table is composed of one tray. A tray is a horizontal plane. This description guides the selection of algorithms to identify objects. The object identification comes from the detection of geometries that belongs to its description. However, the geometries of an object may not be sufficient to differentiate objects (cf. the same geometrical definition of ceiling and floor presented in Table 1 of the Section 3.2) or to detect objects whose points' density does not allow the detection of the geometry (e.g., the glass table in Figure 6). That is why the object description requires a third description part, which is the scene description.

The scene description is composed of two aspects: the topological links between objects (e.g., parallel, perpendicular, in contact, on, inside, next to, above, below, surrounding) and the context in which they are found (e.g., outside, inside, street, modern building, archaeological excavation).

Topological links facilitate the object detection by providing further information to classify it or by deducing the position of an object through the detection of another. Let us take as an example the topological relationship between a table and chairs in the use case presented in Figure 6. A table is described as surrounded by chairs. The property surround of an object by a set of objects is further defined through a relationship between their shapes. As shown in Equation (2), the definition of the surround property applied to this case of tables say that "a table (?t) is surrounded by a set of chairs (?cs)" is equivalent to says that "the shape (?s<sub>t</sub>) of the table (?t) overlaps the shape (?s<sub>cs</sub>) built from the set of chairs (?cs)".

$$Table(?t) \wedge ChairsSet(?cs) \wedge surround(?cs, ?t) \Leftrightarrow Shape(?s_t) \wedge shapeOf(?s_t, ?t) \wedge Shape(?s_{cs}) \wedge shapeOf(?s_{cs}, ?cs) \wedge overlaps(?s_t, ?s_{cs}) \quad (2)$$

Thus in the context illustrated in Figure 6, this topological information can be used to infer the position of the table and facilitate its detection. The context of the scene can influence the geometry and other characteristics of an object. That is why linking the semantic object description to a scene context supports the detection process to search the geometry adapted to the context.

Let us take as an example two different geometries of tables. The first geometry of a table defined as a working table is associated with a scene context of working rooms. It corresponds to the context shows in Figure 6. The working table is described as a table composed of a specific tray that has a rectangular shape. The second geometry of a table defined as round table is associated with a context of lounge rooms. The round table is described as a table composed of a specific tray that has a circular shape. Both these specific descriptions of table respect the general semantic description of a table presented in Section 2.2.2. Figure 7 illustrates this difference of geometry linked to a specific context.

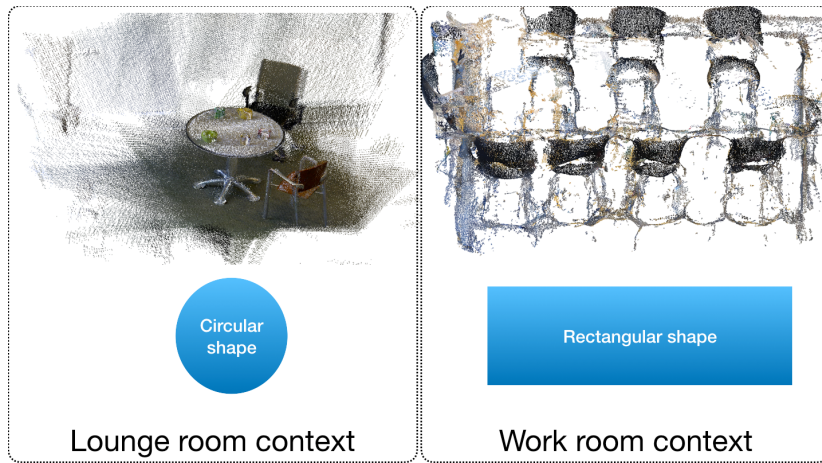


Figure 7. Illustration of two tables having a different shape of tray in two different contexts.

Thanks to such definitions of these two specific tables, the system can adapt the detection process according to the table definition corresponding to the scene represented in the data processed.

### 2.2.3. Algorithm

Algorithms are the components that perform the process of object detection. They aim at providing data or data characteristics that give information contributing to detecting geometries and objects contained in the data. Therefore the semantic description of an algorithm defines an algorithm as generating data, detecting objects and being adapted to geometries. This generic semantic description of any algorithm is presented in Figure 8.

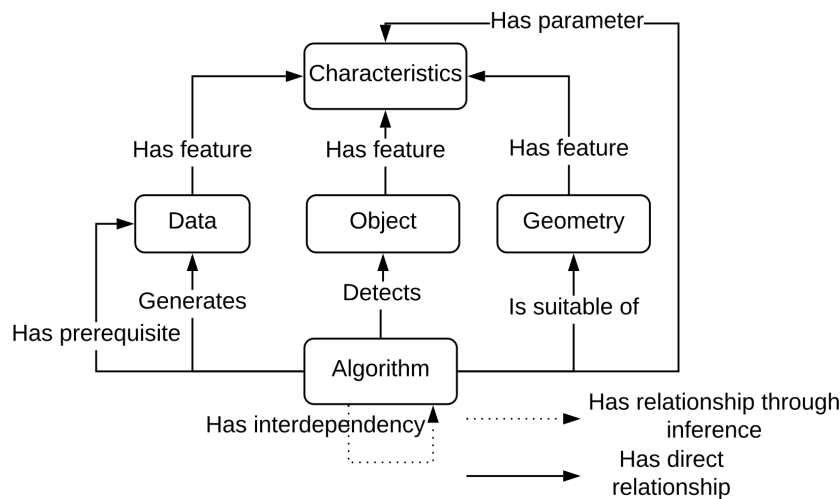


Figure 8. Semantic description of the algorithm.

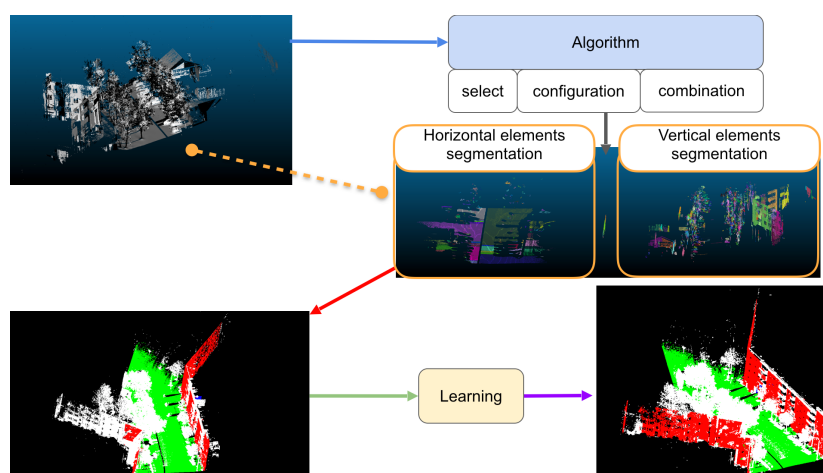
In the process of algorithm selection, the relevance of an algorithm to detect objects is estimated according to the semantic description of each algorithm. This relevance is defined through a link “is suitable” between algorithms and the geometries of the objects.

Algorithms can also have data prerequisites and can produce data characteristics. Therefore, some algorithms satisfy the prerequisites of other algorithms by producing data characteristics required by others. The inference process automatically creates such relationship (shown by the dotted arrow in Figure 8) that makes these algorithms interdependent. Let us take the example of the segmentation algorithm by normal; this algorithm requires a point cloud with a normal estimated for each point. The algorithm of normal estimation estimates the normal of each point and

thus produces a point cloud with estimated normals. Therefore the normal estimation algorithm satisfies the prerequisite of the normal segmentation algorithm. That is why the inference process deduces the interdependency relationship between normal estimation and normal segmentation algorithms that allows them to be combined during the detection process. Some algorithms also require parameters, whose value influences the algorithm result. The role of a parameter is generally to configure mathematical functions or to define a threshold. That is why parameters are mainly primitive values like integers or doubles. The choice of these values by experts depends on the characteristics of data and objects. That is why the choice of parameter values in this methodology is defined through an equation, whose computation depends on elements from object and data descriptions. These elements can be characteristics or geometries of an object but also data characteristics, including factors related to the acquisition process that impact the data. For example, the radius parameter of the normal region growing is computed through an equation that depends on the data resolution.

### 2.3. Knowledge-Driven Object Detection

The detection of objects and geometries in the data is entirely knowledge-driven to provide knowledge that is then analyzed in a learning phase to generate new and more accurate knowledge about the application case, thus improving the effectiveness of detection. Figure 9 shows an overview of a facade detection process in an urban point cloud.



**Figure 9.** Overview of an example of facade detection process.

Data processing is performed using algorithms (illustrated by the blue arrow in Figure 9). These algorithms must be selected, configured and combined (shown under the blue box in Figure 9) according to the application case (e.g., acquisition context, objects sought) and the prerequisites of each algorithm (e.g., low noise, high density, estimated normal).

Let us take the example of a facade defined as a planar surface perpendicular to the ground with a height of at least 12 m. The detection of this facade requires algorithms for plane detection, size estimation and topological link assessment (e.g., parallel, perpendicular). These algorithms may have prerequisites such as normal estimation or whether the data is small in size or not noisy. Thus other algorithms such as normal estimation, denoising and, sampling algorithms must be combined in a sequence to satisfy the needs of some selected algorithms in some cases.

The results provided by each executed algorithm are analyzed and correlated with each other (shown by orange frames in Figure 9) to identify objects and geometries in the data (shown by the red arrow in Figure 9). In this case, the analysis and correlation of algorithm results allow identifying ground (in green) and some facades (in red). The management of algorithms and the interpretation of results are entirely driven by reasoning. This reasoning is based on the knowledge presented in Section 2.2.

However, it is not possible to formulate the knowledge in such a way as to describe all possible cases. In the case illustrated in Figure 9, parts of the facades are not detected because the information on these parts (height less than 12 m, not connected to the ground) differs from the knowledge of the facades (height greater than 12 m and perpendicular to the ground). The discrepancy between the information collected and the knowledge at the preliminary stage may be due to multiple factors (e.g., sensitivity of the acquisition instrument, acquisition condition, insufficient light, objects too far away). Consequently, the knowledge needs to be adjusted according to the application case and the information computed.

Knowledge adaptation requires understanding how objects are represented in the data to anticipate possible variations and to compensate for the discrepancy between the information obtained and the knowledge. Understanding the representation of objects and geometries (“learning” frame in Figure 9), requires analyzing the objects and geometry detected to formulate hypotheses on the characteristics, allowing them to be better identified.

In the case presented by Figure 9, the analysis of the detected facades allows the inference that facades can have a height between 10 and 13 m and that they may not be connected to the ground, which has discontinuities (areas without ground). This new knowledge is integrated into prior knowledge and allows the behavior of the detection process to be changed. The detection strategy becomes specialized for the application case, which leads to better results (shown by the purple arrow in Figure 9).

### 2.3.1. Algorithm Management

The management of algorithms is carried out through reasoning on global knowledge as explained in Reference [45].

This reasoning aims firstly at automatically selecting the algorithms the most adapted to the processed use case. Next, it configures the selected algorithms according to the description of data and objects corresponding to the use case. Finally, it executes configured algorithms and retrieves their result to integrate them into the knowledge base.

The selection of the most adapted algorithms is applied through three steps. The first step consists of selecting algorithms that allow the detection of either objects or characteristics of objects present in the processed data and modeled into the knowledge base. The second step is to add algorithms which capable of working on the processed data to the previous selection. In the third step, algorithms producing data or characteristics that satisfy prerequisites of the previously selected algorithm are added to the set of selected algorithms,. The set of selected algorithms resulting from these three steps is analyzed to filter only algorithms whose prerequisites are satisfied. This set of selected algorithms with prerequisites satisfied is then configured and executed. The configuration of each algorithm is done by first setting its inputs, then adjusting its parameters.

Data is assigned as an input of an algorithm if and only if it has all the characteristics defined as a prerequisite by the algorithm. For example, a color segmentation algorithm requires that the data be colored, so only data with the characteristic of being “colored” will be assigned as input to the algorithm.

The parameters of the algorithms determine their behavior. The parameter values of each algorithm are calculated using equations defined in the semantic description of the algorithm. These equations match characteristics of the objects or geometries for which algorithms are preferable with the characteristics of the data that the algorithms take as input. Thus the parameter values of each algorithm are adapted to the data and the objects or geometries sought.

Then the configured algorithms are executed through the SPARQL function call. Semantic values (e.g., xsd:string, xsd:double, xsd:int) are converted to algorithmic values (e.g., string, double, int) and the algorithm is instantiated, configured and executed dynamically. The results of the algorithms are then converted into the semantic paradigm and incorporated into existing knowledge.

For example, an estimated normals segmentation algorithm produces as knowledge segments whose orientation is homogeneous. The homogeneity threshold of the segments is defined according to the parameters of the algorithm. It is thus adapted to the characteristics of the data (e.g., density, resolution) and the characteristics of the objects (e.g., volume, size, roughness).

### 2.3.2. Classification

The result of executed algorithms provides further information about the data and its content that support the detection of objects and geometries. Their adding to the knowledge base enriches it, allowing further reasoning and information deduction to pursue the detection process.

The semantic descriptions of objects allow for identifying the class of a segment. The reasoning compares the segment characteristics to each object description (characteristics, geometry and scene) to identify the object descriptions that correspond to the segment characteristics.

The classification process uses the characteristics and geometries of the object description first. The description of characteristics and geometries of an object is translated into a logical rule, used to classify segments.

Let us pursue the example of the class called “Table.” This table class is composed of a horizontal plane. It also has characteristics, which are an area greater than 0.5 m<sup>2</sup> and a height between 30 m and 70 m. The following description in Listing 1 corresponds to the “Table” definition in the Manchester syntax:

Listing 1: Example of Table modeling in the Manchester syntax.

- 1 Table :
- 2 (Object or SegmentsSet)
- 3 and (isComposedOf exactly 1
- 4 (Plane and (hasOrientation only Horizontal))
- 5 and (hasArea exactly 1 xsd:double [> 0.5]))
- 6 and (hasHeight exactly 1 xsd:double [> 30, < 70]))

The following logical rule is the translation result of the above “Table” description:

$$\begin{aligned}
 & (Segment(?s) \vee SegmentsSet(?s)) \wedge isComposedOf(?s, ?c) \wedge Plane(?c) \wedge hasOrientation(?c, ?o) \wedge \\
 & \quad Horizontal(?o) \wedge hasHeight(?c, ?h) \wedge between(?h, 30, 70) \wedge hasArea(?c, ?a) \wedge \\
 & \quad GreaterThan(?a, 0.5) \Rightarrow Table(?c)
 \end{aligned} \tag{3}$$

All segments satisfying the rule above are identified as a table. However, although the geometries and characteristics of object description can be adapted to different contexts (as explained in Section 2.2.2), they are not always sufficient to detect objects as shown by the use case of Figure 6.

Indeed, in this use case, the glass material of the table produces a representation with a low density of points inside the data. During the detection process, this low density produces an over-segmentation of tables. It means a table is represented by several segments, rather than a single segment. Thus, during the step of classification by geometry and characteristics, such over-segmentation of tables leads to only two segments representing a part of a table satisfying the Equation (3). Therefore, only two segments are classified as a table (on a total of eight tables) and these segments do not adequately represent a table. This use case shows that the detection of an object represented by a low density of points in the data and whose detection process only takes into account its geometry and characteristics obtains a low and inaccurate result of the detection. That is why this methodology also uses topological relationships to improve the classification.

The topology-based classification uses the topological relationship between segments to classify them. Similar to the previous classification, the topological descriptions of objects are translated into a logical rule. Thus a segment is classified into an object class if and only if the segment satisfies all the topological relationships of a given object class. Let us pursue the example of the topological

relationship of a “Table.” The definition of the “surround” property results in the creation of the logic rule shown by the Equation (4).

$$\begin{aligned} &Segment(?s) \wedge ChairsSet(?cs) \wedge Shape(?s_{cs}) \wedge shapeOf(?s_{cs}, ?cs) \\ &\wedge overlaps(?s, ?s_{cs}) \Rightarrow surround(?cs, ?s) \end{aligned} \quad (4)$$

Thanks to the identification of segments that are surrounded by chairs, the process can create segment sets composed of segments that have some characteristics of a table (e.g., horizontal plane, height between 30 cm and 70 cm, surrounded by the same chair set). Then sets that satisfy the Equation (3) are classified as a table.

Figure 10 shows steps to detect tables thanks to the topological relationship between a table and chairs. First the process detects chairs. The result of the chair detection is shown in Figure 10a. Secondly, an algorithm creates an energy minimalization graph gathering chairs into sets, as illustrated in Figure 10b. Finally, the detection process searches segments that overlap the chair graphs to classify these segments as tables. The result of this classification is presented in Figure 10c.

This application case shows the role of topological links in locating objects when the acquisition conditions are not optimal. Moreover, it also illustrates that the study of the topological relationships allows for identifying objects that are composed of several objects.

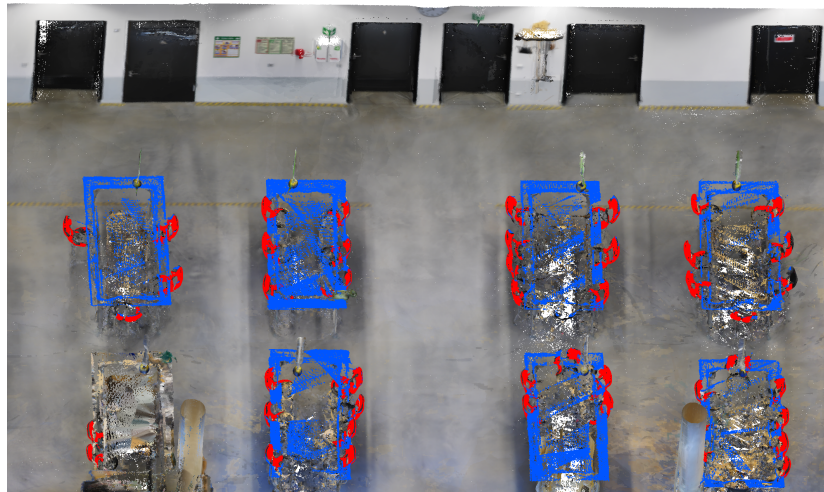


(a) Results of chairs detection in red.



(b) Graph of chairs sets in green.

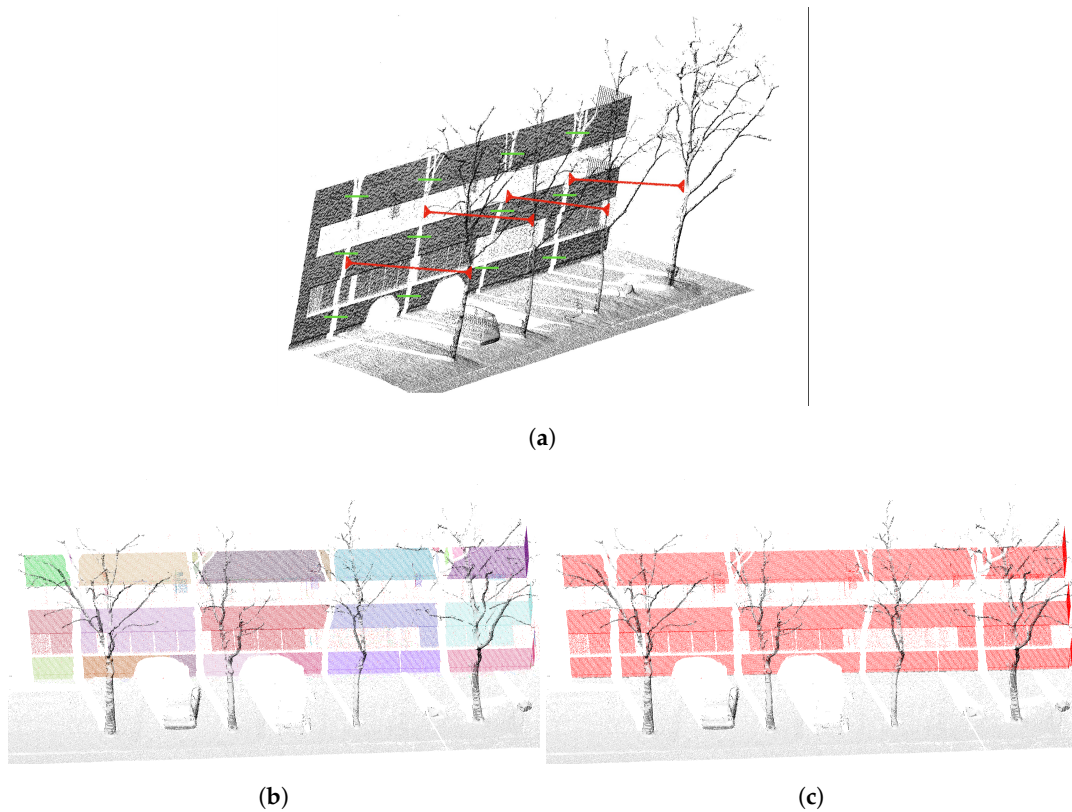
Figure 10. Cont.



(c) Results of table detection in blue.

**Figure 10.** Steps for tables detection using topologic relationship between tables and chairs.

Let us take another example in a completely different use case to illustrate the role of topological relationship in the management of occlusions due to the acquisition process. Figure 11a shows a use case of wall detection, where occlusions during the acquisition process lead to dividing the same wall into several segments, identified as independent walls by the first steps of the detection process (as shown in Figure 11b).



**Figure 11.** Detection of a wall from occluded areas (a single color is assigned to each detected wall). (a) Example of recomposition of a wall divided into several parts, using occlusion deduction (in red) and topological link of building (in green); (b) Wall detection before the inference on the occluded areas; (c) Wall detection after the inference that allows for unifying segments.

In this use case, the occlusions are generated by trees. Thanks to knowledge about the acquisition process and the detection of trees, the system is able to predict the presence of occlusions (illustrated in red in Figure 11a). The information about occlusions is inferred into the knowledge base, by creating a topological “occlusion” relationship (illustrated in green in Figure 11a) between elements separated by an occlusion. Thus by specifying in the knowledge base that two walls having the same geometry and having a relationship of connection or occlusion are a single wall, it allows the gathering of several walls into a single wall (as shown in Figure 11c).

Despite the definition and classification according to geometries, characteristics and topologies of objects, some objects still not detected due to, on the one hand, a difference between the representation inside the data and the semantic description of the object and on the other hand, insufficient description of topological relationships. That is why this methodology is also composed of a self-learning step based on knowledge.

### 2.3.3. Self-Knowledge-Based Learning

Self-knowledge-based learning aims at improving the detection process through new knowledge generation or compensation of any deviation between the semantic description of an object and its representation inside the data. This learning uses the experience gained from the first detection process to generate or adapt to global knowledge. This experience corresponds to new knowledge coming from the study of the results obtained in each application case. This new knowledge specific to each application case provides information to guide the detection process more precisely than the general knowledge previously used. This knowledge that is more adapted to the application case improves the detection process by selecting and configuring more adapted algorithms and by refining the classification through object definitions more specific to the processed use case.

The inference of new knowledge (e.g., new relationships between objects) or the updating of knowledge (e.g., modification of object geometries) requires the understanding of the structure of the data. The analysis of classification results and information extracted by algorithms must be smart in order to improve the knowledge base efficiently, and thus the detection process, and not regress them.

That is why this methodology uses a self-knowledge-based learning system that consists of three steps. The first step consists of enriching the knowledge base with new information about the detected objects. The second step analyzes information common to the same class to formulate new hypotheses on the data structure. The third step checks the consistency of the hypothesis on the processed use case, either to integrate the validated hypotheses in the form of knowledge or to reformulate the invalidated hypotheses.

#### 2.3.3.1. Step 1: Knowledge Enrichment

The enrichment of knowledge aims at identifying new geometries, characteristics of objects and topological relationships between objects to add them to the knowledge base as specific knowledge about the use case. This enrichment requires a diversity of algorithms to identify topological relationships, extract characteristics and geometries. Thus the more algorithms there are and the more diverse capacities algorithms provide, the more the knowledge base can be enriched. The enrichment comes through generation of queries that request algorithm execution to add segment characteristics, which are not yet present in their semantic description. Let us continue to the example of table detection (cf. Figure 6). The application of the methodology without learning step results in the detection of seven tables among eight. Figure 12 highlights the undetected table in an orange rectangle.

Tables have been identified (in blue in Figure 12) thanks to their topological relationship with chairs. The undetected table (encompassed in orange in Figure 12) is not surrounded by enough chairs to detect it through its relationship with chairs. Despite this topological relationship between table and chairs, there is no topological relationship described between tables. Therefore, the enrichment of knowledge searches to identify further information characterizing the relations between tables. Thus executed algorithms allow for computed distances between each table and their relationship



(e.g., aligned, parallel) from each other but also their dimension (e.g., width, height, length). Then this information of distance and position between tables is added to the knowledge base to be analyzed.

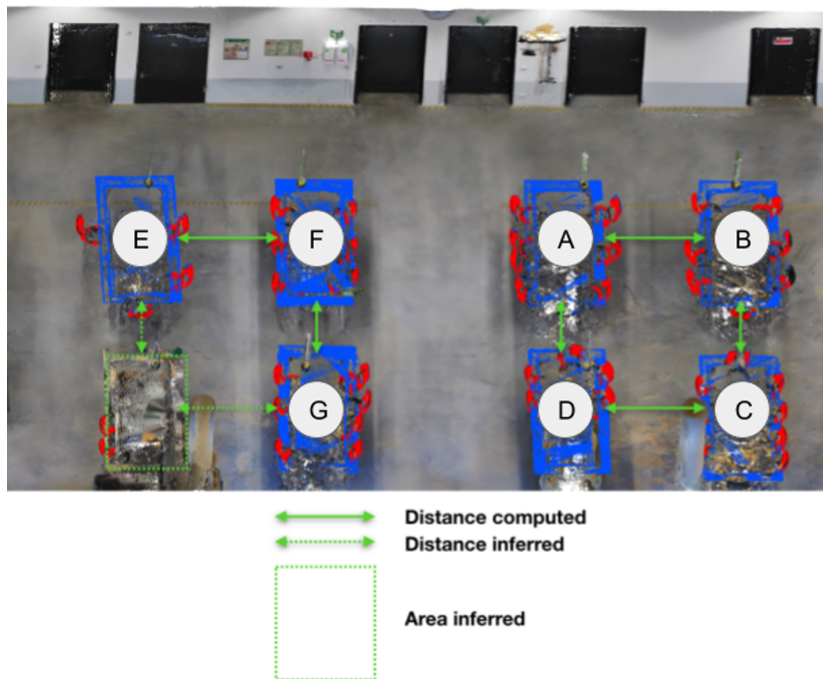


**Figure 12.** Example of a table that was not detected, highlighted in orange rectangle, after the classifications using geometry, topology and characteristics.

2.3.3.2. Step 2: Hypothesis Formulation

The analysis of characteristics common to segments representing the same object type (e.g., segments or sets of segments representing tables) aims at formulating hypotheses automatically. This formulation of hypothesis identifies and gathers the common points characterizing segments of the same object type and does so for each object type.

Let us take the previous example illustrated by Figure 12. In this use case, tables are aligned with another table (relation according to orientation north-south in Figure 13, for example, A is aligned with D) and parallel to other tables (relation according to orientation west-east in Figure 13, for example, A is parallel to B).



**Figure 13.** Illustration of topological relationships between tables.

The characteristics estimated from the previous step aims at grouping objects with the same characteristics into subsets of an object type (e.g., “tables aligned with a table,” “tables parallel to a table”). Grouping similar elements into a subset is applied through an aggregation rule using the characteristics that characterize the subset. This rule identifies elements having the characteristics to belong to the subset. Characteristics are divided between numerical (e.g., distance, height) and non-numerical (e.g., parallel, aligned) characteristics. The aggregation rule of groups based on non-numerical characteristics uses a relationship property corresponding to the characteristic.

In the previous example, let us call the group based on the alignment criterion “AlignedWithTable.” The following aggregation rule creates this group:

$$\text{Segment}(?s) \wedge \text{Table}(?t) \wedge \text{alignedWith}(?s, ?t) \Rightarrow \text{AlignedWithTable}(?s) \quad (5)$$

This rule gathers all segments having a relationship “aligned with” to an object identified as a table, into this group. Similarly, the following aggregation rule (6) gathers elements parallel to a table into the group called “ParallelToTable.”

$$\text{Segment}(?s) \wedge \text{Table}(?t) \wedge \text{parallelTo}(?s, ?t) \Rightarrow \text{ParallelToTable}(?s) \quad (6)$$

The aggregation rules of groups based on numerical characteristics use an interval of values. The value interval aims at providing more flexibility than an average value to assess the integration of an element according to its characteristic value. This value interval is computed through a statistical study of all values representing the characteristic, whose interval is searched. Each studied characteristic does not characterize all elements representing an object type but only a subset of an object type. Likewise, each value interval representing a studied characteristic characterizes only a subset of an object type. That is why it is calculated according to a confidence interval [46] from a set of value samples. Thus, the interval gives a confidence level. This level corresponds to the percentage of the proportion that the sample represents according to the complete set (e.g., 99%, 75%). The Equation (7) defines the confidence interval with  $\bar{x}$  the values mean,  $\delta$  the standard deviation,  $\eta$  the number of values and  $t_\alpha$  the confidence coefficient.

$$I_c = \left[ \bar{x} - t_\alpha \frac{\delta}{\sqrt{\eta}}; \bar{x} + t_\alpha \frac{\delta}{\sqrt{\eta}} \right] \quad (7)$$

Let us pursue the previous example on the proximity criterion between the nearest parallel tables. The group gathered according to this criterion is called “NearParallelTable” and is characterized by a value interval representing the shortest distance between two parallel tables. Thus the shortest distances between parallel tables provide a set of values used to calculate the confidence interval. In this example, the calculated confidence interval is [1.45;1.89]. It means that a segment belongs to this group if it has a distance value with a segment identified as a parallel table between 1.45 m and 1.89 m. Similarly, the height, width and length of tables and the shortest distance between aligned tables are each used as a criterion to apply the same process and create groups of “TableHeight,” “TableWidth,” “TableLength,” “NearAlignedTable,” respectively. These groups are used to formulate a hypothesis on characteristics relevant to characterize the object type in the sense that it improves the detection of objects belonging to this type. Thus all segments belonging to all groups used in the hypothesis are classified as belonging to the object type targeted by the hypothesis. Let us pursue the example of table detection by considering the hypothesis that a table is a segment that belongs to “NearParallelTable,” “NearAlignedTable” “ParallelToTable” and “AlignedWithTable” groups. A logical rule represents the hypothesis. The hypothesis rule corresponding to this example combines four groups and is described by the Equation (8).

$$\begin{aligned} & \text{Segment}(?s) \wedge \text{AlignedWithTable}(?s) \wedge \text{NearAlignedTable}(?s) \\ & \wedge \text{ParallelWithTable}(?s) \wedge \text{NearParallelTable}(?s) \Rightarrow \text{Table}(?s) \end{aligned} \quad (8)$$

The combination of groups to formulate a hypothesis allows for formulating some complex hypotheses. The more groups are used in a hypothesis; the more specialized the hypothesis is. A too-specific hypothesis has the risk of not finding other segments classified other than the beginning set. Conversely, a too-general hypothesis would not use enough groups to characterize an object type and would be invalidated by creating inconsistency according to the first classification. That is why the improvement of identification requires formulating hypotheses according to a hierarchical order. The hierarchical order consists in the beginning of the formulation of general hypotheses using a single group. Then it consists of adding another group to specify further the hypothesis that was invalid because it was too general.

### 2.3.3.3. Step 3: Hypothesis Verification

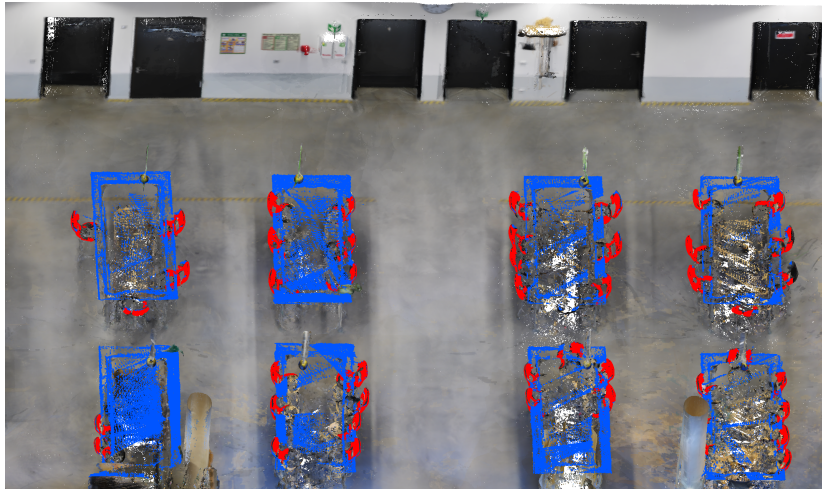
This approach of object detection is driven by knowledge. Thus, the new knowledge produced by the formulation of hypotheses impacts the behavior of the detection process for each considered use case. An impact can be positive by improving the detection process or negative by regressing its quality in the case of an incorrect hypothesis. That is why the hypotheses must be verified to warrant the improvement of the detection process or at least be of equivalent quality.

Such verification requires the measuring of the consequences of the knowledge change or adding on the object detection results. The measurement of consequences is performed by comparing results obtained before and after the integration of the hypothetical knowledge.

A hypothesis is validated if and only if the results obtained after the integration of the hypothetical knowledge do not cause any inconsistency with previously obtained results. Knowledge modeled through constraints and logical rules, and checking the hypothesis validation, is equivalent to checking the consistency of the knowledge base (as explained in the approach in Reference [47]).

Let us continue with the example used in the Section 2.3.3.2 and illustrated in Figure 13. In this example, several groups as “ParallelToTable,” “NearParallelTable” and “TableHeight” are used for hypothesis formulation. By considering the first hypothesis using only “ParallelToTable” and the second hypothesis using “ParallelToTable” combined with “NearParallelTable,” some segments previously classified as a chair are classified as a table after these two hypotheses. As each object type is defined as disjointed from other object types that are not a specification of this type, a segment classified both as a chair and as a table creates an inconsistency inside the knowledge base. Therefore, results obtained by an invalidated hypothesis are removed and a new hypothesis is reformulated.

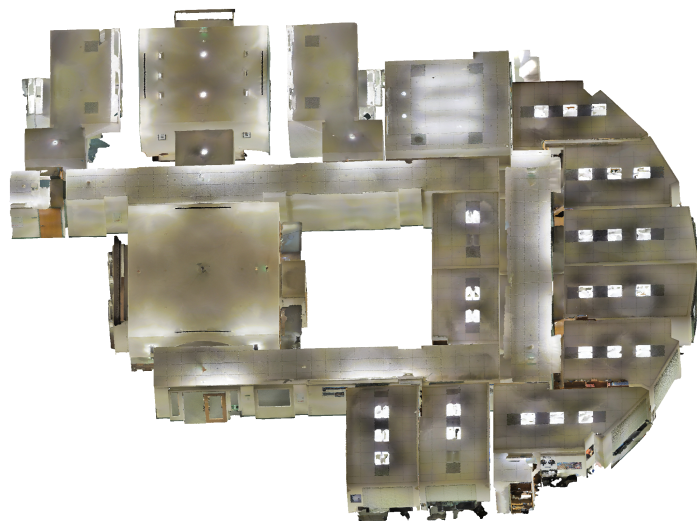
By following the principle of the group adding to reformulate a more specific hypothesis until finding a consistent result of hypothesis or no more possible hypotheses (cf. Section 2.3.3.2), the third hypothesis formulated adds the group “TableHeight” to the two previous ones. In this example, the reformulated assumption using the three groups of characteristics allows for the identification of segments belonging to the eighth table that was not previously detected (surrounded in orange in Figure 12) without inconsistency. The learning process based on the adding of new knowledge specific to the processed use case allows for improving the robustness and the quality of the object detection process, as shown by the result of table detection after learning in Figure 14.



**Figure 14.** Result of table detection after learning.

### 3. Results

This section presents a use case to illustrate the processing of the proposed solution. The results obtained in this use case aim at being compared with other approaches in Section 4. This use case is based on a test data set prepared for semantic scene understanding. This data set is a vast point cloud (more than 22 million points) of a “modern” building, as shown in Figure 15.



**Figure 15.** Illustration of a building area of Stanford point cloud.

The data is slightly noisy, dense, colorful and composed of regular shapes, which means it has a low roughness value. However, this data set contains many discontinuities, mainly for walls and floor due to their occlusions by elements inside rooms. Moreover, some parts of the walls are missing when the walls are composed of windows or reflective surfaces. The use case focuses on the detection of rooms, allowing to compare the results with two other strategies. This use case requires the knowledge base to contain the knowledge specific to the use case. This definition of knowledge follows the needs of the example and the structure explained in Section 2.2. The defined knowledge allows for selecting the appropriate algorithms to detect the elements according to the specificity of this example, as explained in Section 2.3.1. Based on this selection, the processing continues by following the detection process explained in Section 2.3. It identifies each element required to detect the objects sought.

The objective of this section is not to be exhaustive (i.e., to give all steps followed by the process) but instead to give a summarized view of the most generic steps of the proposed approach. For the application case studied here, only six data characteristics are sufficient and necessary: density, resolution, size, dimension (e.g., 2D, 3D), occlusion and noise. Some of these characteristics are more decisive than others. In particular, the occlusion, noise and density characteristics, which have a significant influence on the choice and configuration of algorithms. Besides, these characteristics often vary within the same dataset. For example, one portion of the data may be very dense and low-noise, while another portion may be low-density and high-noise. This information can be defined by the user but is mainly inferred from acquisition knowledge. In other cases of application, other characteristics must be taken into account.

### 3.1. Object Modeling

In the approach presented, all the elements that compose the data are defined as “objects.” The knowledge of each of these objects must be modeled, as explained in Section 2.2.2. Object modeling first describes for each object the specific characteristics of the object that can influence the acquisition of data (e.g., roughness, color, materials, height, length, width). Secondly, it describes the characteristics that a segment (a portion of data) must have to be classified in this type.

Let us take the example of modeling knowledge about the floor and walls (i.e., portions of data having a specific particularity). The geometry of the walls corresponds to a vertical planar surface. They have the specific characteristic of having a height greater than 2 m. Finally, their topological characteristics are to be perpendicular to the floor and to be in contact (“is connected to”) with at least one other wall. Their semantic description in OWL2 under the Manchester syntax is described in Listing 2.

Listing 2: Example of Wall modeling.

```

1 Wall :
2 Object that
3 (hasGeometry some (Plane and hasOrientation only Vertical) )
4 and (hasHeight exactly 1 xsd:double [>2.0])
5 and (isPerpendicular min 1 Floor)
6 and (isConnectedTo min 1 Wall)

```

Similarly, the geometry of the floors corresponds to a horizontal planar surface. Their specific characteristic is that they have a surface area greater than . Finally, their topological characteristics are that they can be perpendicular to walls and parallel to the ceiling. Their semantic description in OWL2 under the Manchester syntax is described in Listing 3.

Listing 3: Example of Floor modeling.

```

1 Floor :
2 Object that
3 (hasGeometry some (Plane and hasOrientation only Horizontal) )
4 and (hasSurface exactly 1 xsd:double [>1.0])
5 and (isPerpendicular some Wall)
6 and (isParallel some Ceiling)

```

Objects can also be defined as being composed of several other objects. For example, a room is defined as a floor parallel to a ceiling and both are connected to at least three common walls. The more precise the description of objects is, the better their detections are. However, the representation of objects in the data is influenced by multiple factors (e.g., context, light, instrument, acquisition method) and can therefore be very diverse. Consequently, it is not possible to model the knowledge of objects in such a way as to describe all possible representations of objects. Thus, it is necessary to understand and model the reasons that cause these various representations of objects. That is why knowledge

about the factors that influence the object characteristics must be also described using logical rules and queries that allow deducing the different possible representations of objects or to delimit the search field of this type of object as explained in Section 2.2.1.

### 3.2. Processing

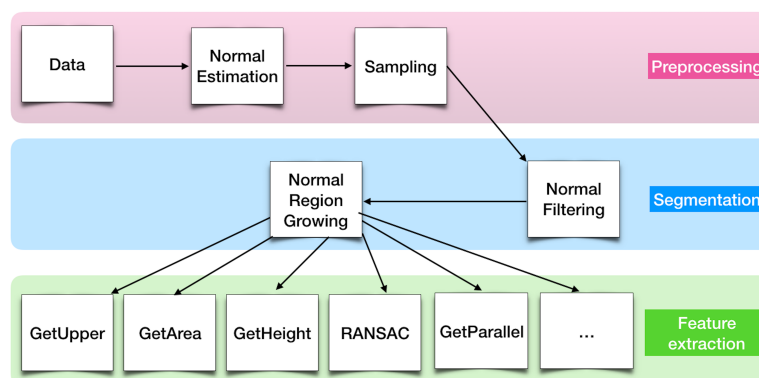
This section presents the results produced for the main detection process steps on the studied use case for room detection. Since rooms are defined as consisting of floor, wall and ceiling, these three types of objects are the main objects to be detected to identify rooms. The generic description logic of these three types of objects is summarized in Table 1.

**Table 1.** Summary of the wall, floor and ceiling characteristics (“NC” means unknown).

Object	Geometry	Orientation	Height	Area	Topology
Wall	Plane	Vertical	$\geq 2$ m	NC	$\perp$ floor and connected to walls
Floor	Plane	Horizontal	NC	$\geq 1$ m	under walls
Ceiling	Plane	Horizontal	NC	$\geq 1$ m	upper walls

#### 3.2.1. Algorithms Selection And Configuration

It is necessary to process the data and identify objects, to automatically select, configure and execute the algorithms most suitable for the application case considered (as explained in Section 2.3.1). Figure 16 shows the execution graph of the selected algorithms for the application case studied.



**Figure 16.** Execution graph of the selected algorithms for the application case.

The three objects, walls, ceilings and floors, have a geometry described by an orientation (vertical or horizontal). This orientation requires that the normal at every point be assessed. The relevance of the algorithms is determined according to the characteristics of the objects being searched, the data used and the prerequisites of the algorithms (as explained in the Equation (9)).

$$\begin{aligned}
 & \text{Algorithm}(?a) \wedge \text{Object}(?o) \wedge \text{Characteristics}(?c) \wedge \\
 & \text{hasCharacteristics}(?o, ?c) \wedge (\text{produces}(?a, ?c) \vee \text{produces}(?a, ?o)) \\
 & \Rightarrow \text{isSuitableFor}(?a, ?o)
 \end{aligned} \tag{9}$$

The Equation (9) specifies that if an object (?o) has a characteristic (?c) and an algorithm (?a) produces such characteristic (?c) or produces the object (?o) itself, then it (?a) is suitable for the object (?o). Moreover, if the algorithm (?a) has a specific prerequisite or input (?p) and another algorithm

(?b) satisfies such prerequisite (?p), then the latter algorithm (?b) is suitable for the first algorithm (?a) (as explained in the Equation (10)).

$$\begin{aligned} & \text{Algorithm}(?a) \wedge \text{Algorithm}(?b) \wedge \text{Object}(?o) \wedge \\ & \text{isSuitableFor}(?a, ?o) \wedge \\ & (\text{hasPrerequisite}(?a, ?p) \vee \text{hasInput}(?a, ?p)) \wedge (\text{produces}(?b, ?p)) \\ & \Rightarrow \text{isSuitableFor}(?b, ?a) \end{aligned} \quad (10)$$

Furthermore, the “suitable” property is transitive, meaning that if one algorithm (?b) is suitable for another algorithm (?a) that is suitable for the object (?o), then the algorithm (?b) is suitable for (?o) by transitivity. These equations work on the description logics of the algorithms.

In the case studied, the characteristics of the objects sought (walls, floors and ceiling) are: planes, normals (horizontal and vertical), height and area (as shown in the Table 1), as well as the parallelism relationship (between ceilings and floors), the perpendicular relationship (between walls and floors or ceilings) and other topological relationships (such as upper, under). The selection of algorithms according to the characteristics of the objects is summarized in Table 2.

**Table 2.** Summary of the matching between algorithms and characteristics detected by these algorithms.

Algorithms	Characteristic
RANSAC	Plane
Normal Estimation	Normal
GetHeight	Height
GetArea	Area
GetParallel	Parallel
GetPerpendicular	Perpendicular
GetUpper	Upper
GetUnder	Under

Among these algorithms, only the “Normal Estimation” algorithm works on a complete point cloud. The other algorithms work on segments that are only part of a point cloud. Therefore, the first algorithm executed is the algorithm (“Normal Estimation”) (as shown in Figure 16). The other algorithms require segments. The algorithm requirements are highlighted by the properties “hasPrerequisite” and “worksOn” in the description logic of each algorithm. For example, the description logic of the “RANSAC” algorithm states that the algorithm requires segments by the “worksOn” property (as shown in the Listing 4).

Listing 4: RANSAC algorithm modeling in Manchester syntax.

```

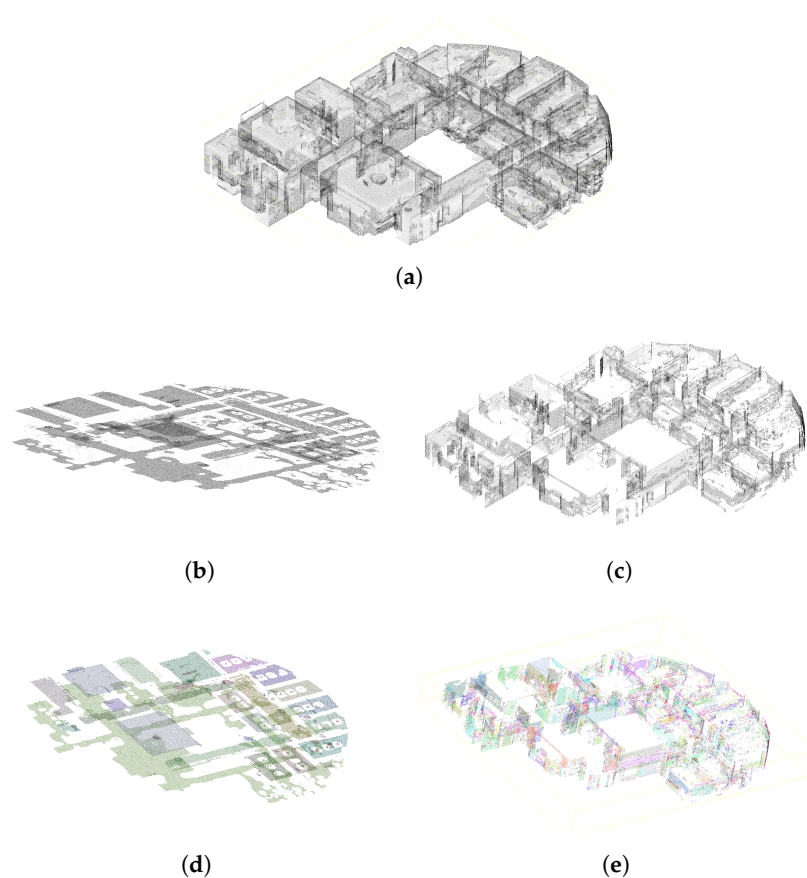
1 RANSAC:
2 Algorithm
3 and (worksOn some (Segment) )
4 and (hasParameter exactly 1 Tolerance)
5 and (hasParameter exactly 1 IterationNumber)
6 and (generates some (Plane that comesfrom exactly 1 Segment
7 and hasFeature exactly 1 Precision))

```

Segmentation algorithms divide data into segments. Thus, the Equation (10) defines segmentation algorithms as suitable for the “RANSAC” algorithm and other algorithms (“GetHeight,” “GetArea,” “GetPerpendicular,” “GetParallel,” “GetUpper” and “GetUnder”). Among the different segmentation algorithms, “Normal Region Growing” is selected for its fit with the orientation characteristics of the objects sought. Therefore “Normal Region Growing” is selected, as shown in Figure 16. Nevertheless, this algorithm needs to be applied to small data. It cannot, therefore, be applied to the original data of the use case, which is too large (several million points).

This means the original data must be reduced before applying the “Normal Region Growing” algorithm. Thus, the “Sampling” algorithm satisfies the prerequisite of the “Normal Region Growing” algorithm by reducing the size of the data. The configuration of this algorithm depends on the minimum size of the objects to be detected. In this case, its execution depends on the size of the walls, ceilings and floors.

Walls, floors and ceilings are defined as having a single orientation (vertical for walls and horizontal for floors and ceilings). Based on this information, the detection process executes a “Normal filtering” as shown in Figure 16. Indeed, this algorithm satisfies the specific prerequisite of the vertical or horizontal orientation of the planes that compose the walls, floors and ceilings. This algorithm aims at filtering the data according to the orientation of the objects being searched. It allows for identifying the areas of point clouds to search for an object. These areas are identified by configuring the algorithm with the orientation of the objects sought to retrieve data having only points which have the orientation sought. Figure 17 presents the results of these algorithms according to each orientation characteristic.



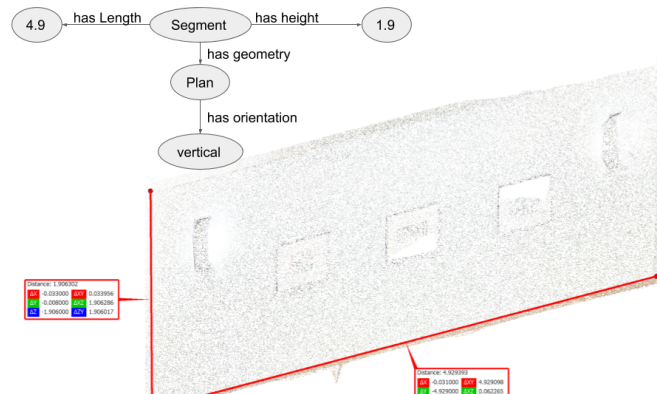
**Figure 17.** Results of the main algorithm. (a) Results of the sampling algorithm; (b) Results of the horizontal filtering algorithm; (c) Results of the vertical filtering algorithm; (d) Results of the segmentation applied on the horizontal filtered data; (e) Results of the segmentation applied on the vertical filtered data.

The semantic description of walls, floors and ceilings defines them through a planar geometric shape. The identification of such geometry for the detection of these objects requires the execution of a plane detection algorithm. Finally, the object detection requires the execution of algorithms to estimate object characteristics (e.g., height, width, length, volume) and topological relationships (e.g., distance, perpendicular, in contact) that characterize segments. Thus the reasoning process configures and executes these algorithms. The results of these algorithms are interpreted to add the knowledge that



they provide on segments into the knowledge base. Figure 18 illustrates an example of knowledge about a segment obtained from the data understanding.

The classification process presented in the next section is based on this knowledge about data segments to classify them as objects.



**Figure 18.** Example of knowledge about a segment and its corresponding view in the data.

### 3.2.2. Classification

The use of OWL2 to define the knowledge allows for classifying segments as an object or geometry type through logical reasoning. Constraints that specify their characteristics define object and geometry types. Thus a segment is classified in the type when it satisfies the constraints. The satisfaction of a constraint type means the segment has all characteristics that characterize this type.

Such classification is carried out by translating the description logics (mainly the “class construct” part) of objects into a rule of inference. For example, the description logic of a wall is presented in listing 2 and translated into the rule of inference presented by the Equation (11). This translation is carried out by an automatic process whose pseudo-code is shown by the Listing 5.

Listing 5: Pseudo-code of the automatic translation of “class construct” into “rule of inference”.

```

1 v := 0;
2 occ:= the class construct to be translated
3 query := "CONSTRUCT { ?ind rdf:type <" + this.origin.getURI() + "> . ";
4 where := "";
5 foreach restriction of occ
6 property := property of occ;
7 resource := resource of occ;
8 if restriction = owl:hasValue
9 where += "?ind <" + property+ "> <" + resource + "> . ";
10 else
11 v++;
12 if restriction := owl:Datatype
13 where += "?ind <" + property.getURI() + "> ?v" + v + " . ";
14 where += "FILTER (?v";
15 if property = MIN INCLUSIVE
16 where += ">="
17 else if property = MAX INCLUSIVE
18 where += "<="
19 ...
20 where +=")";
21 else

```

```

22 if restriction := owl:allValueFrom
23 where += "?ind <" + property + "> ?v" + v + " . ";
24 where += "?v" + v + " rdf:type <" + resource + "> . ";
25 else if restriction := owl:someValueFrom
26 where += "OPTIONAL{ ?ind <" + property + "> ?v" + v + " . ";
27 where += "?v" + v + " rdf:type <" + resource + "> . }";
28 ...
29
30 query += "} WHERE { " + where + " }";
31 return query;

```

Thus, walls are classified in the studied use case, by the rule shown in the Equation (11).

$$\begin{aligned}
 & \text{Segment}(?s) \wedge \text{hasHeight}(?s, ?h) \wedge \text{greaterThan}(?h, 2) \wedge \\
 & \text{Plane}(?p) \wedge \text{isVertical}(?p) \wedge \text{hasGeometries}(?s, ?p) \wedge \\
 & (\text{Wall}(?w) \wedge \text{connected}(?s, ?w) \vee ( \\
 & \text{Segment}(?s2) \wedge \text{hasHeight}(?s2, ?h2) \wedge \text{greaterThan}(?h2, 2) \wedge \\
 & \text{Plane}(?p2) \wedge \text{isVertical}(?p2) \wedge \text{hasGeometries}(?s2, ?p2) \\
 & \wedge \text{connected}(?s, ?s2))) \\
 & \wedge \text{Floor}(?g) \wedge \text{perpendicular}(?g, ?s) \Rightarrow \text{Wall}(?s)
 \end{aligned} \tag{11}$$

This rule states the geometry, the characteristics and topological relationship that a segment (?s) have to be classified as a wall. It must have a vertical plane (?p) and a height (?h) of at least 2 m. In addition to this geometry and characteristic, it must satisfy the two following topological relationships. The first topological relationship is to be connected to at least one wall (?w) or to another segment (?s2) having the geometry of a wall (?p2 and ?h2). The second topological relationship is to be perpendicular to at least one segment classified as a floor (?g) or having the geometry of a floor. The last part of the second topological relationship (“or having the geometry of a floor”) does not appear in the rule above to facilitate its understanding.

Similar rules based on knowledge allow the classification of floors and ceilings. Figure 19 illustrates the result of floor, wall and ceiling classification.



Figure 19. Classification of floors in green, walls in blue and ceilings in red.

A room is described in the knowledge base as composed of a floor connected to at least three walls. Figure 20 shows the results of the room classification.



Figure 20. Results of the rooms classification

Rooms are not well-segmented due to the derivation between the general knowledge used for driving the detection process and the information of this specific application case. Therefore it is necessary to adapt general knowledge to the specificity of the application case. This adaptation is automatically carried out by the self-knowledge-based learning process.

### 3.3. Self-Knowledge-Based Learning

The classification step allows for identifying a majority of objects inside the data. The self-learning process begins by finding all possible characteristics of segments classified using available algorithms for characteristic extraction (cf. Section 2.3.3). It uses these new characteristics to make hypotheses of the better object type definition. The validated hypotheses enrich the knowledge base and improve the detection process. During the enrichment of the self-learning on this use case, the knowledge base is enriched by the topological relationship of parallelism between walls and the distance that separates them from each other. The self-knowledge-based learning process then brings together recurring and common characteristics of a set of objects of the same type to formulate hypotheses.

Thus, groups of walls have as recurring characteristics having the same length or width, being parallel to each other and being connected to the same wall. The hypothesis shows in the Equation (12), represented as a logical rule and is therefore automatically formulated based on these characteristics.

$$\begin{aligned}
 & \text{Segment}(?s) \wedge \text{Wall}(?w) \wedge \text{areParallel}(?s, ?w) \wedge ( \\
 & \quad (\text{hasLenght}(?s, ?l) \wedge \text{hasLenght}(?w, ?l2) \wedge \text{equals}(?l, ?l2)) \vee \\
 & \quad (\text{hasWidth}(?s, ?wd) \wedge \text{hasWidth}(?w, ?wd2) \wedge \text{equals}(?wd, ?wd2))) \wedge \\
 & \quad \text{Wall}(?w2) \wedge \text{arePerpendicular}(?s, ?w2) \wedge \text{arePerpendicular}(?w, ?w2) \Rightarrow \text{Wall}(?s)
 \end{aligned} \tag{12}$$

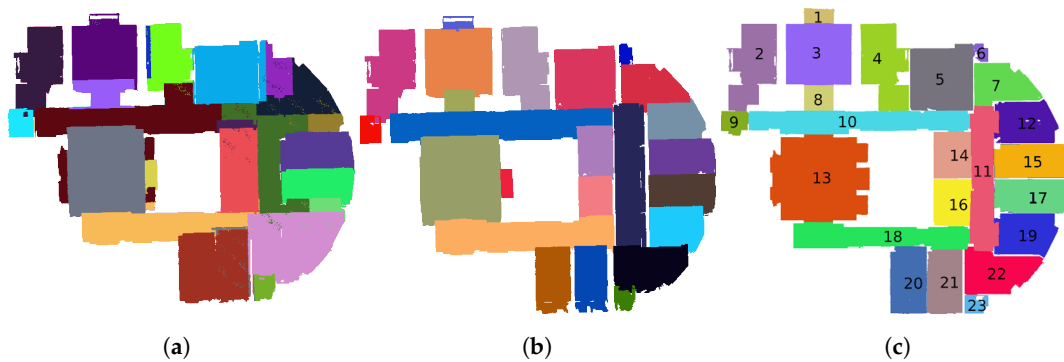
The new rule defines that if a segment (?s) is parallel to a wall (?w) and both have equal length (?l and ?l2) or equal width (?wd and ?wd2) and that a same wall (?w2) is perpendicular with both elements (the segment ?s and the wall ?w), then the segment is a wall.

This hypothesis is then validated automatically if its application does not produce any inconsistency in the knowledge (segment identified as corresponding to two different types of an object). If it is validated, then the produced knowledge enriches the knowledge base. This enrichment of knowledge about objects impacts the entire detection process. Hypothesis formulation allows objects to be detected even when their geometric representation in the data differs significantly from the geometry expected and defined in the knowledge.

Several hypotheses, in addition to the hypothesis illustrated in the Equation (12) allow improving the classification of objects. First, in this application case, every ceiling is upwards of 2.15 m. Meanwhile, all other objects defined in the knowledge base are lower than this value. Therefore, a new hypothesis is formulated and specifies that every element upwards of 2.15m in the data of this case studied is a

ceiling. The same type of analysis is also applied to the floor description. The description of floors, ceilings and walls are thus improved. Figure 21 shows the result of room detection before (a) and after (b) the iteration of new knowledge provided by these hypotheses.

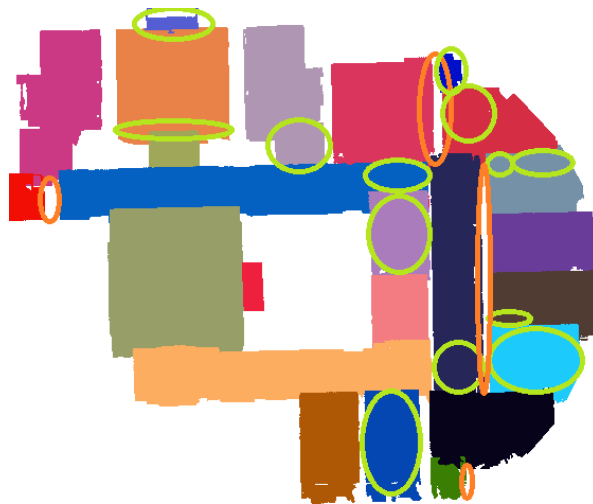
The analysis process of the learning enables the adapting of the description of objects according to the data dynamically, significantly improving the object detection process. The rooms are better-segmented thanks to the improved knowledge of the walls, floors and ceilings. The relationships that unify them as rooms are also better controlled.



**Figure 21.** Results of the room detection. (a) Results obtained before the self-knowledge-based learning process; (b) Results obtained after the self-knowledge-based learning process; (c) Ground truth point cloud from Reference [48].

#### Discussion of the Efficiency of the Self-Learning Process

A first object detection provides a base of information to apply a self-learning process. The learning analyzes the characteristics of detected objects to improve the accuracy of knowledge according to each application case. Obtaining accurate knowledge aims at improving the detection process, which is guided by knowledge. Thus the detection process is directly impacted by the accuracy of the knowledge. The benefit of this self-learning on the detection process efficiency is estimated through a comparison of results obtained before and after the application of the self-learning process.



**Figure 22.** Results obtained after the self-learning process: green circles highlight the data understanding improvement and orange circles highlight the loss of data understanding.

Figure 22 highlights the improvements (cf. green circles in Figure 22) and losses (cf. orange circles in Figure 22) generated by the self-learning process. These losses are a loss of accuracy for some rooms, for whom some points have not been classified. However, the improvements solve problems of non-detection on the one hand and misclassification on the other hand.

The quality difference between the detection processes before and after the self-learning process is assessed according to the metrics of Recall, Precision and F1-score (metrics explained by Reference [49]).

**Recall:** represents the proportion of points considered as negatively well classified. It is computed from the number of points similarly classified in the assessed set and the reference set divided by the number of all points from the assessed set.

**Precision:** represents the proportion of points considered as positively well classified. It is computed from the number of points similarly classified in the assessed set and the reference set divided by the number of all points from the reference set.

**F1-score:** represents the harmonic average between the precision and recall (with a best value at 1 and a worst at 0). It is computed from the precision and the recall scores.

Table 3 presents the three metric scores obtained for each room of the dataset before and after the self-learning process execution.

**Table 3.** Comparison of metrics obtained before and after the self-learning process on the Stanford dataset.

Room	Recall		Precision		F1-Score	
	Before	After	Before	After	Before	After
Room 1	0.100	0.602	0.838	0.843	0.179	0.702
Room 2	0.989	0.964	0.815	0.820	0.894	0.886
Room 3	0.888	0.940	0.757	0.820	0.817	0.875
Room 4	0.985	0.925	0.517	0.722	0.678	0.811
Room 5	0.816	0.915	0.798	0.503	0.807	0.649
Room 6	0.187	0.974	0.559	0.707	0.280	0.819
Room 7	0.395	0.706	0.376	0.591	0.385	0.643
Room 8	0.489	0.887	0.845	0.779	0.619	0.829
Room 9	0.977	0.963	0.696	0.453	0.813	0.616
Room 10	0.664	0.898	0.588	0.768	0.624	0.828
Room 11	0.519	0.634	0.606	0.840	0.559	0.723
Room 12	0.984	0.976	0.226	0.542	0.367	0.697
Room 13	0.888	0.958	0.672	0.666	0.765	0.785
Room 14	0.505	0.639	0.122	0.545	0.197	0.588
Room 15	0.825	0.970	0.813	0.434	0.819	0.600
Room 16	0.498	0.956	0.763	0.464	0.603	0.624
Room 17	0.907	0.921	0.636	0.594	0.748	0.722
Room 18	0.932	0.768	0.640	0.819	0.760	0.793
Room 19	0.983	0.980	0.165	0.565	0.282	0.717
Room 20	0.535	0.968	0.798	0.520	0.641	0.677
Room 21	0.455	0.965	0.697	0.461	0.551	0.624
Room 22	0.564	0.962	0.837	0.576	0.674	0.720
Room 23	0.996	0.985	0.713	0.527	0.831	0.687
<b>Average</b>	0.699	0.889	0.629	0.633	0.605	0.722

According to the average evaluation of the metrics, the quality improvement is mainly obtained by an improvement of the Recall score with an increase of 0.190 (compared to an increase of 0.004 for the Precision score). Thus, in this application case, the self-learning process provides an average quality increase of 0.117 on its F1 score.

However, the self-learning impacts each room differently. Four rooms (rooms 1, 6, 14 and 19) obtain a significant increase in quality, about 0.5 in the F1 score. Seven other rooms (rooms 4, 7, 8, 10, 11, 12 and 21) also benefit from an increase of quality with an F1 score increase of about 0.2. Eight rooms (rooms 2, 3, 13, 16, 17, 18, 20 and 22) are hardly impacted by the self-learning. Finally, four rooms (rooms 5, 9, 15 and 23) suffer a loss of quality with an F1 score decrease of about 0.2. This quality decrease is due to a more significant decrease of precision than the increase of recall.

This decrease of precision for some rooms is caused by points that were classified as belonging to a room before the self-learning process and are unclassified after the self-learning process. A point

that could be classified into two disjoint object types would create an inconsistency in the knowledge. That is why such a point stays unclassified rather than be classified in the two object types or be classified arbitrarily in one of the object types to avoid the wrong classification.

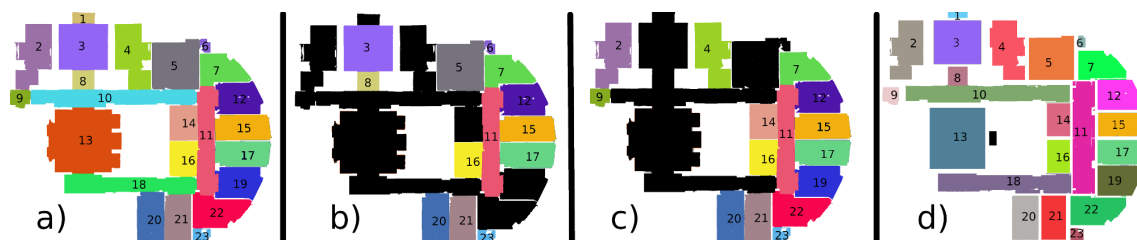
Therefore the improvement of knowledge slightly increases the classification ambiguity of points which are between two rooms but strongly increases the amount of correctly classified points.

Indeed, the loss of quality on these four rooms remains low compared to the benefits provided on the eleven rooms that obtained a quality increase between 0.2 and 0.6.

#### 4. Discussion

A “modern” data set acquired by laser scanner technology has been used in this paper as an application case to show the efficiency of the detection process proposed. The main difficulty of the object detection process in data acquired by laser scanner technology is to adapt the process of detection to the specific acquisition context. Indeed, the acquisition context is affected by various and complex factors. Thus, the representation of acquired objects inside the data can vary greatly from the representation of a real object. That is why the detection process has to consider various problems such as occlusions and incompleteness caused by transparent or reflective surfaces (e.g., windows or aluminum material). These problems transform the geometry of objects, inhibiting their detection.

This paper introduces an approach that addresses this issue through a dynamic and smart adaptation of the detection process. This adaptation takes into account the data variation and object representation variation. This approach first detects and classifies walls, ceilings and floors, as shown in Figure 19. Secondly, the reasoning and iterative classification process build rooms according to the links between walls, ceilings and floors. The comparison of this approach with References [28,29] approaches on the same data allow the illustration of its benefits. Reference [48] provides the data used for the comparison. This data is the third area corresponding to a point cloud composed of more than 22 million points. These approaches ([28,29]) fail to detect the overall rooms as shown in Figure 23.



**Figure 23.** Illustration of the result comparison, the black color represents rooms incorrectly detected; other colors represent good detection of rooms: (a) the floor truth; (b) the reconstruction result of Reference [28]; (c) the reconstruction result of Reference [29]; (d) the result of the presented approach.

The results of the two approaches presented in Figure 23b,c are respectively the results of the reconstruction of room areas for both approaches that are available in References [28,29].

The approach proposed by Armani et al. [28] is a “void-based” approach that requires the point cloud of the entire building. This “void-based” approach aims at localizing the area between two walls of neighboring rooms. To this end, this “void-based” approach creates a density histogram to identify areas which are empty (no point) and between two high-density areas representing two walls. Nevertheless, such an approach requires that walls are “roughly planar” and rooms are rectangular. Therefore, this approach cannot deal with irregular walls or rooms, which is the case for rooms 19, 22, 2 and 4 in Figure 23a. Moreover, such an approach cannot separate rooms which are not separated by walls, such as rooms 1, 9, 10, 13, 14 and 18. Therefore, the approach of Reference [28] does not detect room 1, poorly detects room 19 and room 22 (a part of room 19 is detected as room 22) and cannot properly segment rooms 2, 4, 14, 9, 10, 13 and 18 (rooms 9, 10, 13 and 18 are detected as the same room). Figure 23b shows in black every room which is not well detected by this approach.

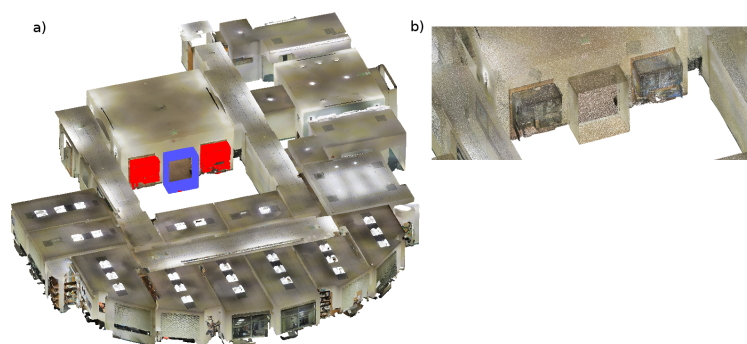
The approach of Reference [29] proposes computing the internal free space by segmenting the point cloud into voxel and labelling them as “free” or “busy.” From this voxel labelling, the approach computes the maximum detection in the potential field values. Then, the approach projects the potential field values computed in 3D in 2D space in the positive z-direction. Finally, this approach uses the HDB-SCAN approach [50] for clustering the point cloud and then labels it. Nevertheless, such an approach has difficulty to segmenting rooms which contain several elements due to the sensitivity of the volumetric signature estimation, which is the case for rooms 5 and 6. Moreover, the irregularities in the potential field map computed lead to miss-segmentation, especially for transitions between long corridor rooms, which is the case for rooms 10 and 18. Therefore such approach detects the six rooms 1, 3, 8, 10, 13 and 18 as the same room. Figure 23c shows in black every room not well detected by this approach.

The Table 4 numerically compares the results of the two approaches with the approach presented in this paper.

**Table 4.** Result comparison between the presented approach and two other approaches on the use case.

Approach	Correctly Classified Room (Max 22)	Success (%)
Presented approach	21	95%
[28]	12	55%
[29]	14	63%

The proposed approach detects the 22 rooms of the use case. However, among the 22 rooms, room 13 in Figure 23d is not fully detected and a sub-part of this room is detected as a room itself (the green part near room 13 in Figure 23d). Some parts of the room (small parts in room 13 in Figure 23d) are not classified due to their characteristics. They correspond to exterior elements of room 13, as illustrated in Figure 24. Thus, room 13 of the presented approach is considered not correctly detected.



**Figure 24.** (a) Illustration of room parts not classified in red and incorrectly classified in purple; (b) Zoom on the representation of these three room parts.

Moreover, the classification of the external elements of room 13 as belonging to room 13 can be discussed. In the presented approach, they are not classified due to different features than other rooms (e.g., ceiling not at the same height as the room). An addition of knowledge about this type of element would fix their classification. However, does the belonging of these elements to room 13 makes sense? These elements are outside room 13 and the center element (in purple in Figure 24) is classified as a room and has the same characteristics as room 1 in Figure 23. Therefore, despite a difference with the principle, the detection made by the presented approach follows logical reasoning to understand the data.

This logical reasoning leads to a limit of accuracy in the delimitation of rooms. Indeed, the reasoning process considers all points classified in more than one room as inconsistent. Thus, all points at the border of two rooms are not classified in any room (illustrated by the white gap

between rooms in Figure 23d). However, thanks to the flexibility and the automatic adaptation of the knowledge module of this approach, the addition of further knowledge about building elements could fix this issue.

The knowledge is modeled in OWL-description language. Thus, the various descriptions of an object, depending on the application context, are sure to be entirely consistent. Indeed, the OWL description language allows for checking the consistency of all information and description logic by the reasoner. The knowledge base is thus safe for use to drive the object detection process and then to be enriched by algorithms' execution and reasoning. According to results obtained (95% of rooms detected), we can claim that our approach is more robust than the other two approaches (55% and 63% of rooms detected). Its advantages come from its full management by the semantic. The semantic allows for adapting the process to the characteristics of the data and objects and dynamically enriching these characteristics by an understanding process of results obtained. Thus, this approach ensures proper object detection through a safe and smart classification process that uses reasoning and checking of consistency to avoid incorrect classification.

Future works will consist of detecting more complex objects by adding new object descriptions and new algorithms. In the context of an indoor point cloud, elements of buildings like doors and windows constitute the future set of objects to detect. It is necessary to extend the knowledge base by adding a description of these objects.

**Author Contributions:** Conceptualization, F.B., A.T., and J.-J.P.; writing original draft preparation, J.-J.P.; writing review F.B. and A.T.; editing, J.-J.P.; supervision, F.B. and A.T.; project administration, F.B.

**Funding:** This research received no external funding.

**Acknowledgments:** The authors would like to thank the NaVvis company and Fraunhofer IPM ([https://www.ipm.fraunhofer.de/de/presse\\_publicationen/Presseinformationen/messfahrzeug-3D-Daten-breitbandausbau.html](https://www.ipm.fraunhofer.de/de/presse_publicationen/Presseinformationen/messfahrzeug-3D-Daten-breitbandausbau.html)) for the permission to use their data.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

SPARQL	SPARQL Protocol and RDF Query Language
OWL	Web Ontology Language
SWRL	Semantic Web Rule Language
RANSAC	Random sample consensus
AEE	Algorithm Execution Engine
KAE	Knowledge Analysis Engine

## References

1. Buckley, S.J.; Howell, J.; Enge, H.; Kurz, T. Terrestrial laser scanning in geology: Data acquisition, processing and accuracy considerations. *J. Geol. Soc.* **2008**, *165*, 625–638. [[CrossRef](#)]
2. Nocerino, E.; Poiesi, F.; Locher, A.; Tefera, Y.T.; Remondino, F.; Chippendale, P.; Van Gool, L. 3D reconstruction with a collaborative approach based on smartphones and a cloud-based server. In Proceedings of the 5th International Workshop LowCost 3D—Sensors, Algorithms, Applications, Hamburg, Germany, 28–29 November 2017; pp. 187–194.
3. Weber, T.; Hänsch, R.; Hellwich, O. Automatic registration of unordered point clouds acquired by Kinect sensors using an overlap heuristic. *ISPRS J. Photogramm. Remote Sens.* **2015**, *102*, 96–109. [[CrossRef](#)]
4. Han, X.F.; Jin, J.S.; Wang, M.J.; Jiang, W.; Gao, L.; Xiao, L. A review of algorithms for filtering the 3D point cloud. *Signal Process. Image Commun.* **2017**, *57*, 103–112. [[CrossRef](#)]
5. Vosselman, G.; Dijkman, S. 3D building model reconstruction from point clouds and ground plans. *Int. Arch. Photogramm. Remote Sens.* **2001**, *34*, 37–44.



6. Overby, J.; Bodum, L.; Kjems, E.B.; Ilsøe, P.M. Automatic 3D Building Reconstruction from Airborne Laser Scanning and Cadastral Data Using Hough Transform. *Int. Arch. Photogramm. Remote Sens.* **2004**, *35*, 296–301.
7. Borrmann, D.; Elseberg, J.; Lingemann, K.; Nüchter, A. The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Res.* **2011**, *2*, 3. [[CrossRef](#)]
8. Anagnostopoulos, I.; Pătrăucean, V.; Brilakis, I.; Vela, P. Detection of walls, floors and ceilings in point cloud data. *Constr. Res. Congr.* **2016**, *2016*, 2302–2311.
9. Schnabel, R.; Wahl, R.; Klein, R. Efficient RANSAC for point-cloud shape detection. *Comput. Graph. Forum.* **2007**, *26*, 214–226. [[CrossRef](#)]
10. Moosmann, F.; Pink, O.; Stiller, C. Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009; pp. 215–220.
11. Himmelsbach, M.; Hundelshausen, F.V.; Wuensche, H.J. Fast segmentation of 3D point clouds for ground vehicles. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 21–24 June 2010; pp. 560–565.
12. Rabbani, T.; Van Den Heuvel, F.; Vosselmann, G. Segmentation of point clouds using smoothness constraint. *Int. Arch. Photogramm. Remote Sens.* **2006**, *36*, 248–253.
13. Khaloo, A.; Lattanzi, D. Robust normal estimation and region growing segmentation of infrastructure 3D point cloud models. *Adv. Eng. Inform.* **2017**, *34*, 1–16. [[CrossRef](#)]
14. Macher, H.; Landes, T.; Grussenmeyer, P. Point clouds segmentation as base for as-built BIM creation. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *2*, 191. [[CrossRef](#)]
15. Jung, J.; Stachniss, C.; Kim, C. Automatic room segmentation of 3D laser data using morphological processing. *ISPRS Int. J. Geoinf.* **2017**, *6*, 206. [[CrossRef](#)]
16. Hichri, N.; Stefani, C.; De Luca, L.; Veron, P. Review of the “as-built BIM” approaches. In Proceedings of the 3D-ARCH International Conference, Trento, Italy, 25–26 February 2013.
17. Dong, C.; Loy, C.C.; He, K.; Tang, X. Learning a deep convolutional network for image super-resolution. In Proceedings of the European conference on computer vision, Zurich, Switzerland, 6–12 September 2014; pp. 184–199.
18. Girshick, R. Fast r-cnn. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
19. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 379–387.
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:1506.01497.
22. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3D object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
23. Li, B. 3D fully convolutional network for vehicle detection in point cloud. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1513–1518.
24. Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; Jia, J. IPOD: Intensive Point-based Object Detector for Point Cloud. *arXiv* **2018**, arXiv:1812.05276.
25. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 3–9 November 2017; pp. 5099–5108.
26. Du, X.; Ang, M.H.; Karaman, S.; Rus, D. A general pipeline for 3D detection of vehicles. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3194–3200.

27. Ku, J.; Mozifian, M.; Lee, J.; Harakeh, A.; Waslander, S.L. Joint 3D proposal generation and object detection from view aggregation. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–8.
28. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D Semantic Parsing of Large-Scale Indoor Spaces. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
29. Bobkov, D.; Kiechle, M.; Hilsenbeck, S.; Steinbach, E. Room segmentation in 3D point clouds using anisotropic potential fields. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 727–732.
30. Ponciano, J.J.; Prudhomme, C.; Tietz, B.; Boochs, F. Detection and isolation of switches in point clouds of the German railway network. In Proceedings of the The 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Bangkok, Thailand, 23–27 November 2015; pp. 96–102, doi:10.1109/SITIS.2015.86. [[CrossRef](#)]
31. Antoniou, G.; Van Harmelen, F. Web ontology language: Owl. In *Handbook on Ontologies*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 67–92.
32. Omerovic, S.; Milutinovic, V.; Tomazic, S. *Concepts, Ontologies, and Knowledge Representation*; Springer: Berlin/Heidelberg, Germany, 2001.
33. Poux, F.; Neuville, R.; Van Wersch, L.; Nys, G.A.; Billen, R. 3D Point Clouds in Archaeology: Advances in Acquisition, Processing and Knowledge Integration Applied to Quasi-Planar Objects. *Geosciences* **2017**, *7*, 96, doi:10.3390/geosciences7040096. [[CrossRef](#)]
34. Poux, F.; Neuville, R.; Nys, G.A.; Billen, R. 3D Point Cloud Semantic Modelling: Integrated Framework for Indoor Spaces and Furniture. *Remote Sens.* **2018**, *10*, 1412, doi:10.3390/rs10091412. [[CrossRef](#)]
35. Dietenbeck, T.; Torkhani, F.; Othmani, A.; Attene, M.; Favreau, J.M. Multi-layer ontologies for integrated 3D shape segmentation and annotation. In *Advances in Knowledge Discovery and Management*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 181–206.
36. Hmida, H.B.; Cruz, C.; Boochs, F.; Nicolle, C. From 3D Point Clouds To Semantic Objects An Ontology-Based Detection Approach. *arXiv* **2013**, arXiv:1301.4783.
37. Tao, J.; Sirin, E.; Bao, J.; McGuinness, D.L. Integrity constraints in OWL. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 11–15 July 2010.
38. Karmacharya, A.; Boochs, F.; Tietz, B. Knowledge guided object detection and identification in 3D point clouds. In Proceedings of the Videometrics, Range Imaging, and Applications XIII, Munich, Germany, 22–25 June 2015; p. 952804.
39. Ponciano, J.J.; Karmacharya, A.; Wefers, S.; Atorf, P.; Boochs, F. Connected Semantic Concepts as a Base for Optimal Recording and Computer-Based Modelling of Cultural Heritage Objects. In *Structural Analysis of Historical Constructions*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 297–304. doi:10.1007/978-3-319-99441-3\_31. [[CrossRef](#)]
40. Prud, E.; Seaborne, A. *Sparql Query Language for RDF*; World Wide Web Consortium: Cambridge, MA, USA, 2006.
41. Punnoose, R.; Crainiceanu, A.; Rapp, D. SPARQL in the cloud using Rya. *Inf. Syst.* **2015**, *48*, 181–195. [[CrossRef](#)]
42. Hitzler, P.; Krötzsch, M.; Parsia, B.; Patel-Schneider, P.F.; Rudolph, S. OWL 2 web ontology language primer. *W3C Recomm.* **2009**, *27*, 123.
43. McGuinness, D.L.; Van Harmelen, F. OWL web ontology language overview. *W3C Recomm.* **2004**, *10*, 2004.
44. Ponciano, J.J.; Boochs, F.; Trémeau, A. Knowledge-based object recognition in point clouds and image data sets. *gis.Science-Die Zeitschrift für Geoinformatik* **2017**, hal-02047375.
45. Ponciano, J.J.; Boochs, F.; Trémeau, A. Identification and classification of objects in 3D point clouds based on a semantic concept. 18. Oldenburger 3D-Tage, 2019. hal-02014831.
46. Kalinowski, P. Understanding Confidence Intervals (CIs) and effect size estimation. *APS Obs.* **2010**, *23*, 4.
47. Wang, H.H.; Li, Y.F.; Sun, J.; Zhang, H.; Pan, J. Verifying feature models using OWL. *J. Web Semant.* **2007**, *5*, 117–129, doi:10.1016/j.websem.2006.11.006. [[CrossRef](#)]
48. Armeni, I.; Sax, A.; Zamir, A.R.; Savarese, S. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv* **2017**, arXiv:1702.01105.

49. Zheng, M.; Wu, H.; Li, Y. An Adaptive End-to-End Classification Approach for Mobile Laser Scanning Point Clouds Based on Knowledge in Urban Scenes. *Remote Sens.* **2019**, *11*, 186. [[CrossRef](#)]
50. Campello, R.J.; Moulavi, D.; Zimek, A.; Sander, J. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data* **2015**, *10*, 5. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).