PFG 2013 / 3, 0221 – 0237
Stuttgart, June 2013

Report

Automatic Detection and Classification of Objects in Point Clouds using multi-stage Semantics

Hung Quoc Truong, Helmi Ben Hmida, Frank Boochs, Mainz, Adlane Habed, Christophe Cruz, Yvon Voisin & Christophe Nicolle, Dijon, France

Keywords: 3D processing, point clouds, knowledge modelling, ontology, scene object classification

Summary: Due to the increasing availability of large unstructured point clouds obtained from laser scanning and/or photogrammetric data, there is a growing demand for automatic processing methods. Given the complexity of the underlying problems, several new methods try to use semantic knowledge in particular for supporting object detection and classification. In this paper, we present a novel approach which makes use of advanced algorithms to benefit from intelligent knowledge management strategies for the processing of 3D point clouds and for object classification in scanned scenes. In particular, our method extends the use of semantic knowledge to all stages of the processing, including the guidance of the 3D processing algorithms. The complete solution consists of a multi-stage iterative concept based on three factors: the modelled knowledge, the package of algorithms, and the classification engine. Two case studies illustrating our approach are presented in this paper. The studies were carried out on scans of the waiting area of an airport and along the tracks of a railway. In both cases the goal was to detect and identify objects within a defined area. With our results we demonstrate the applicability of our approach.

Zusammenfassung: *Automatische Detektion und Klassifikation von Objekten in Punktwolken unter Nutzung mehrschichtiger Semantik*. Infolge der zunehmenden Verfügbarkeit großer unstrukturierter Punktwolken aus Laserscanning und Photogrammetrie entsteht wachsender Bedarf für automatisierte Auswerteverfahren. Angesichts der häufig hohen Komplexität der in den Punktwolken enthaltenen Objekte stoßen rein datengetriebene Ansätze an ihre Grenzen. Es entstehen vermehrt Konzepte, die auf verschiedene Weise auch Gebrauch von der Semantik machen. Semantik und Algorithmik sind dabei oft eng miteinander verwoben und führen zu Limitationen in Art und Umfang der nutzbaren Semantik. Mit der vorgestellten Lösung werden Algorithmik und Semantik klar getrennt und mit den exakt auf diese Domänen zugeschnittenen Werkzeugen behandelt. Deren prozedurale Verknüpfung führt dann zu einem neuen Verarbeitungskonzept, das eine nach unserem Kenntnisstand bislang nicht erreichte Flexibilität und Vielseitigkeit in der Nutzung unterschiedlichster Semantiken besitzt und auch die Steuerung der Algorithmen integriert. Die iterative Gesamtlösung fußt auf drei Säulen, nämlich dem modellierten Wissen, dem Pool der Algorithmen und dem Identifikationsprozess. Erreichbare Resultate werden an zwei Beispielen dokumentiert. Ein Beispiel befasst sich mit der Analyse von Punktwolken aus dem Bereich der Lichtraumvermessung an Bahntrassen, das zweite mit Räumlichkeiten in einem Flughafen. In beiden Fällen müssen bestimmte Objektarten aufgefunden und klassifiziert werden.

1 Introduction

Object detection, recognition and reconstruction from digitized data, typically images and point clouds, are important tasks that find applications in many fields. Because such processing tasks are extremely laborious and difficult when carried out manually, it is of the utmost importance that they benefit from the support – or even be entirely performed

www.schweizerbart.de
DOI: 10.1127/1432-8364/2013/0172
1432-8364/13/0172 $ 0.00

through – numerical algorithms. Most existing algorithms are data-driven and rely both on extracting discriminating features from the dataset, and also on numerical models characterizing either geometric, e.g. flatness and roughness, or physical, e.g. colour and texture, properties of the sought objects. The numerical model and the extracted features are combined to form a decision. These methods are generally affected by the nature of dataset and the behaviour of the algorithms. Instead, it is up to the user to decide, often subjectively but generally based on one's experience, which algorithms are better suited for any particular kind of objects and/or the datasets. It goes without saying that the success of these approaches is significantly compromised by the increasing complexity of the objects and the decreasing quality of the data. Furthermore, relying on only a restricted set of features and individual algorithms to process the data might lead to unreliable results. One way to overcome the drawbacks of the data-driven approaches is to resort to the use of additional knowledge. For instance, knowledge characterizing the objects to be detected with respect to the data at hand or their relationships to other objects may generally be derived beforehand. Such knowledge not only allows for a systematic characterization and parameterization of the objects but also supports the quantification of the effectiveness of the algorithms to be used.

The work presented in this paper precisely aims at efficiently exploiting additional knowledge in the processing of point clouds. In particular, our work bridges semantic modelling and numerical processing strategies in order to benefit from knowledge in any or all parts of an automatic processing chain. Our approach is based on structuring various knowledge components into ontology containing a variety of elements taken from multiple sources such as digital maps and geographical information systems. However, we do not only rely on information about objects potentially present in the scene, i.e. their characteristics, a hierarchal description of their sub-components, and spatial relationships, but also on the characteristics of the processing algorithms at hand. During processing, the modelled knowledge guides the algorithms and supports both the analysis of the results and the object classification. Knowledge is also used to support the choice among different algorithms, the combination of these, and the adopted strategies. Our main contribution is a comprehensive set-up to model and use knowledge from various domains and to let it interact and contribute to all steps of an object detection process. This starts with inferring steps controlling algorithms based on object and scene related knowledge in order to select adapted algorithmic strategies and ends with a knowledge-based object classification and simultaneous extension and updating of the knowledge base (KB).

Our paper is structured as follows. An overview of the relevant literature on the topic is presented in section 2. Our proposed solution is outlined in section 3. Knowledge building and knowledge management are discussed in section 4. Section 5 is dedicated to our knowledge-based strategy for object detection and classification. This is followed by two case-studies involving real-world examples in section 6. Our conclusion and future work are given in section 7.

2 State of the Art

Early 3D processing techniques were either data-driven or model-driven and often based on statistical approaches. Many such methods, generally based on fitting techniques employing local or global optimization and statistical regression, often in conjunction with the random sampling consensus (RANSAC) algorithm for robustness, have attracted and continue to attract significant attention (Nurunnabi et al. 2012). However, many data-driven methods, in particular those relying on the segmentation of data into primitive shapes, are known to be highly sensitive to noise as well as to local deformations (Tarsha-Kurdi et al. 2007). Model-driven approaches, while less sentitive to local irregularities, require reliable geometrical models which are often difficult to obtain especially when dealing with complex scenes (Huang et al. 2011). However, despite of the robustness and efficiency of many such processing algorithms, they cannot resolve ambiguities when assigning semantic labels to objects in a scene. Such ambiguities can be efficiently dealt with when integrating semantic knowledge with data processing (see for instance BUSCH et al. 2005, HELMHOLZ et al. 2012).

As far as feature-based object recognition is concerned, some of the approaches have been used both in 2D images and in 3D data. For instance, VOSSELMAN & DIJKMAN (2001) made use of higher level 3D features such as simple roof shapes, i.e. flat roofs, gable roofs and hip roofs, which are generally present in building structures. The authors relied on the use of the 3D Hough transform to detect planar roof faces in point clouds, and hence reconstructed the scene in a higher level of abstraction. Their segmentation strategy was based on detecting intersecting lines and "height jump edges" between planar faces. PU & VOSSELMAN (2006) used segmentation and feature extraction algorithms to recognize building components such as doors, walls, windows from point clouds. Based on constraints on these components, they were able to determine the categories to which each extracted feature belonged. However, the results were not satisfactory if the data did not clearly describe an object due to either the presence of noise or occlusions.

An important processing approach, which partly solves some limitations of data-driven methods, makes use of artificial intelligence techniques to enforce the robustness of the processing and to allow for the recognition of more complex objects. A typical work in this category is the one presented by ANGUELOV et al. (2005) in which object segmentation and classification are obtained by a learning procedure employing Markov random fields and quadratic programming. Such methods generally require a large number of training datasets in order to obtain good results.

Building on the above results, significant improvements have been achieved in 3D data processing by additionally incorporating semantic aspects. The method proposed by CANTZLER et al. (2002) relies on a semantic network defining the relationships between objects in a scene such as walls being perpendicular to the floor and rules which the extracted features must obey. However, problems arise when dealing with complex indoor scenes possibly including many types of objects. HEDAU et al. (2010) located objects of a specific geometry in an indoor scene. The detector computes the 3D location of an object along with its orientation using the geometry and the mutual arrangement of the object and the scene as well as a single image. Although quite useful for scene understanding, such an approach is limited to dealing with the case of a single object in the scene.

Localizing multiple objects in a scene has proved to be a difficult and challenging problem that often requires considering spatial and/or semantic relationships between objects. One way to address such problem is to resort to the use of semantic knowledge. The ability to exploit semantic knowledge is limited when the number of objects becomes large as it requires an adequate way of structuring properties of objects and relationships between them. In some approaches, this is carried out through a hierarchical description of the attributes of each object and those of the scene. For instance, TEBOUL et al. (2010) segmented building facades using a derivation tree representing the procedural geometry, and connected knowledge representation by grammars with machine learning. Furthermore, this approach proposed a dynamic way of performing a search through a perturbation model. RIPPERDA & BRENNER (2006) also extracted building facades using a structural description and used reversible jump Monte Carlo Markov chains (GREEN 1995) to guide the application of derivation steps during the building of the tree. Another application of using knowledge is to infer the missing parts with detected parts. For example, PU & VOSSELMAN (2009) reconstructed building facades from terrestrial laser scanning data. Knowledge about size, position, orientation and topology is used to recognize features, e.g. walls, doors and windows, and also to hypothesize the occluded parts. In a similar work (SCHOLZE et al. 2002), a model-based reconstruction method was proposed. In this method, semantic knowledge is also used to infer missing parts of the roof and to adjust the overall roof topology. These approaches use knowledge to evaluate results of numerical processes, but do not integrate it into the processing as such.

Since the use of knowledge is also useful within the processing chain, other works have focused on knowledge management within computation. For example, Maillot & Thonnat (2008) used a visual concept ontology composed of visible features such as spatial and relationships, colour and texture to recognize objects by matching numerical features and visual concepts. Durand et al. (2007) proposed a recognition method based on an ontology which has been developed by experts of the domain; the authors also developed a matching process between objects and the concepts of ontology to provide objects with a semantic meaning. Interest also grows in developing knowledge-based system for various data processing tasks such as data segmentation and registration but also for scene understanding and interpretation. For instance, Trinder et al. (1998) proposed a knowledge-based method which automatically extracts roads from aerial images. The description of roads includes radiometric, geometric properties and spatial relationships between road segments, all formulated as rules in PROLOG. The knowledge base stores structures of roads and relationships between them extracted from images. By using topological information of road networks, the method is able to predict missing road segments. However, the used semantic model is limited to one type of objects (roads). Growe & Tonjes (1997) presented a knowledge-based approach for the automatic registration of remotely sensed images. Knowledge is explicitly represented using semantic nets and rules. Prior knowledge about scene objects and a geographic information system (GIS) are used to select and match the best set of features. Matsuyama (1987) proposed a method for automatic interpretation of remotely sensed images. The approach emphasises the use of knowledge management and control structures in aerial image understanding systems: a blackboard model for integrating diverse object detection modules, a symbolic model representation for 3D object recognition, and integration of bottom-up and top-down analyses. Two kinds of knowledge are considered in their expert system: knowledge about objects and knowledge about analysis tools, e.g. image processing techniques. Rost & Münkel (1998) proposed a knowledge-based system that is able to automatically adapt image processing algorithms to changes in the environment. The method uses expert knowledge that is explicitly formulated by rules. Depending on a given task, the system selects a sequence of relevant image processing tools and adjusts their parameters to obtain results with some predefined quality goals. Results on object contour detection, carried out in various conditions, show the benefit of taking into account expert knowledge for adjusting the parameters of various image processing operators. However, knowledge in these approaches has not been fully exploited: other capabilities, such as processing guidance, have not been explored.

Knowledge-based methods have the ability to not only manage and exploit geometric and/or topological relations between objects, but also to embed scene structures into semantic frameworks. Such knowledge is often translated into geometric constraints that can be used to improve object detection. Various kinds of knowledge-based methods have appeared for applications in object detection, demonstrating a clear and increasing interest for such approaches. This expresses a certain expectation about the role of semantics in future processing solutions. A step forward towards benefiting from the use of knowledge in such solutions would be a comprehensive approach that exploits knowledge in all processes, i.e. in guiding the numerical processing, evaluating, and classifying detected objects. Such an approach is proposed in this paper.

3 System Overview

When attempting to build an integrated approach with knowledge directing all parts of the process, several aspects have to be considered. At first, the whole process needs to be incorporated into a knowledge management tool. Therefore, it is necessary to have a process guiding all individual steps, leading from an initial situation to the final result. Inside this overall process, one part has to cover the numerical processing and another part has to handle the processing results. This latter part has to evaluate the results, draw conclusions about what has been found, and also what this means for further processing. This includes the need to update the content of the database with the objects that have been found. This database has to be managed in a way that every detected object is transferred from some initial state to a final one within the framework of a rule-based system.

The main components of our system are illustrated in Fig. 1. The adopted strategy is applied to the analysis of 3D point clouds, but can also be extended to other data sources. It is based on explicitly formulating prior knowledge of the scene, on spatial relations of objects and on processing algorithms. It is a multi-stage concept based on three components: the modelled knowledge (Fig. 1 left), the package of algorithms (Fig. 1 top-right) and the classification engine (Fig. 1 bottom-right). In the initial stage, the available knowledge is transferred into a KB. Starting from this initial stage, an update process, which invokes the algorithms and the classification engine, is launched. Here, the algorithm selection module (ASM) guides the processing via selecting a set of processing algorithms based on the nature of the target objects, and produces new elements which can be identified. These elements are passed on to the classification engine, which, based on the existing knowledge expressed in the ontology, attempts to apply Semantic Web Rule Language (SWRL) (Horrocks et al. 2004) rules and description logic (DL) constraints in order to identify the nature or object category of the elements. This classification handles the output obtained from the algorithms. The result of the classification step updates the KB by inserting newly classified or updating already existing elements before running the next stage of processing. The process ends either when all objects are detected and classified or in absence of any change in the annotation process for a predetermined number of iterations (whose values remain at the discretion of the user).

Objects are represented by a point cloud or possibly data from other sources. Such data depend on many factors such as the type of the sensing system and the measuring/capturing conditions. This representation has to be handled by algorithms which also depend on many additional factors, e.g. noise, other data characteristics, and already existing objects. Strong interrelationships among these factors have a direct influence on the efficiency of the detection and classification processes. The more flexibly these factors and interactions are controlled, the better results are to be expected. For these reasons, knowledge from different domains is required and the quality of these various knowledge sets has significant impact on the results (Ben Hmida et al. 2011). Our solution relies on four main knowledge categories to construct the core of the KB: the scene knowledge, the spatial knowledge, the data knowledge and the algorithm knowledge. Each field of knowledge is represented by circles in Fig. 1, and relationships between these concepts are represented by directed edges. The scene knowledge contains information related to the content of the scene to be processed, important characteristics of objects, e.g. geometric features, appearance

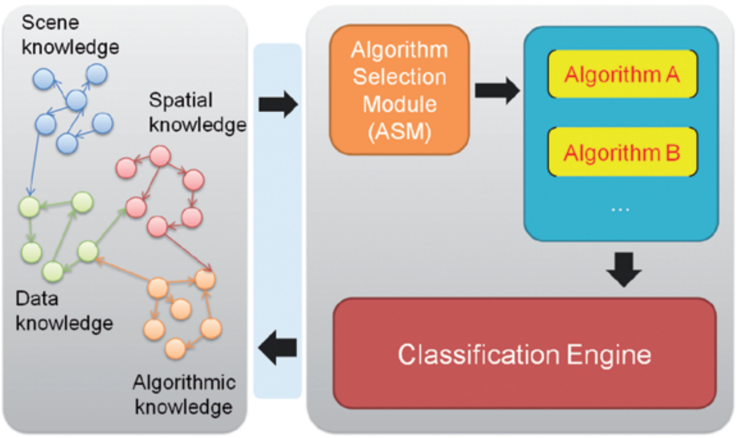

Fig. 1: System architecture.

and texture, and the geometry that composes its structure. Such knowledge is not only important for identification and classification processes but also supports the selection and guidance of the algorithms. The spatial knowledge models the relationships between objects in the scene. It is an important factor for the classification process because it supports an object's state disambiguation based on its relationship with the common environment. The data knowledge expresses important characteristics of the data itself. Finally, algorithm knowledge characterizes the behaviour of algorithms and determines which purpose they fulfil, which input is expected, which output is generated, and which geometries they are designed for. Based on this knowledge, a dynamic algorithm selection is possible allowing for a dynamic adaptation to processing situations given from other domains (Fig. 1).

4 Building Knowledge

The concept requires efficient methods for knowledge representation, management and interaction with algorithms. Efficient knowledge representation tools are available from the semantic web framework, which expresses knowledge through the web ontology language (OWL) (Bechhofer et al. 2004). The encapsulation of semantics within OWL through description logics (DLs) axioms has made it an ideal technology for representing knowledge from almost any discipline. We use the OWL to represent expert knowledge about the scene of interest and for algorithmic processing. With OWL ontology, we are able to describe complex semantics of a scene. For instance, the statement "A railway track is a linear feature with two linear structures running parallel to each other within a certain distance" can be expressed through logical statements. Likewise, we define the semantics of algorithmic processing within OWL. For example, the *CheckParallel* algorithm is designed for detecting a *Signal*, which contains parallel linear structures.

$$CheckParallel \exists\ isDesignedFor.Signal \sqcap Signal.hasParallel.\{true\} \quad (1)$$

As an additional technology, SWRL is available. It is a program which infers logic from the KB to derive a conclusion based on observations and hypotheses. For instance, the following rule (2) asserts that a detected element of class *Geometry* which has a distance from *DistanceSignal* of 1000 m, has a height equal to or greater than 4 m, and which has a linear structure, will be inferred as a *MainSignal*.

Geometry(?x) ^ hasLine(?x, ?l)
^ line(?l) ^ DistanceSignal (?y)
^ DistanceFrom(?x, ?y, ?dis)
^ swrlb:GreaterThan (?dis,1000)
^ hasHeight(?x, ?h) ^ swrlb:GreaterThan
(?h, 4) → MainSignal(?x) (2)

Variables are indicated by the standard convention in which they are prefixed by a question mark symbol (e.g. *?x*). An important SWRL feature is its ability to allow user-defined built-ins, i.e. user-defined predicates, such as, *swrlb:equal* and *swrlb:lessThan*, that can be used in SWRL rules, which help in the interoperation of SWRL with other formalisms and provide an extensible infrastructure for knowledge-based applications.

The techniques mentioned above serve as tools to formalize the identified and acquired knowledge. As explained, the actual solution handles four separate domains: the scene knowledge, the spatial knowledge, the data knowledge and finally the algorithm knowledge. All these knowledge domains have their representations in the domain ontology and participate in the whole processing cycle. The graphical structure of the top-level concepts of the ontology is given in Fig. 2, where we find four main concepts, called *Classes* in the next paragraphs. In order to proceed, these classes have to describe the different actors used during the detection and the classification process in a structured hierarchical way. The main factors that have to be modeled are: processing algorithms, point cloud data or image resources, and target objects with their geometry and characteristics. The class *DomainConcept* represents the different objects found in the target scene and can be considered the main class in this ontology. This class is further specialized into classes representing the different detected objects. The other classes are used to either describe the object geometry through the *Geometry* class by defining its geometric component or to describe its characteristics through the *Characteristics* class. Ultimately, the system selects algorithms for the processing chain based on their compatibility with the object geometry and characteristics read from the *Algorithm* class.

Knowledge of different domains is acquired from the relevant sources. Domain experts are the most reliable knowledge sources. However, information sources such as CAD, GIS data, or other available documents in the case of detailed input can also be used to extract knowledge. In our case, the algorithm knowledge is acquired by experts in numerical processing and the scene knowledge is acquired from existing digital documents as a CAD drawing or GIS dataset.

The scene knowledge is described in the schema of ontology and includes semantics of the objects such as properties, restrictions, relationships between objects and geometries. The more information about an object is created and used, the more accurate the detection and classification process is. An example of defining a semantic object is the following: an electric pole (type 2) along a railway track has a height above ground between 4 m and 6 m; it comprises a vertical structure that connects to a cube on the ground; at the top, there are two parallel linear structures; and along the tracks, the distance from an electric pole (type 2) to a signal column is 1000 m within the bounds of a predefined tolerance, for example ± 0.5 m, depending on the quality of data, measurement uncertainty and noise.

Knowledge about 3D spatial relationships is used to enhance the classification process. Information about how objects are scattered in a 3D scene makes the detection and classification easier. For instance, given the detection of a wall, the probability of detecting doors or windows is higher. 3D spatial knowledge includes standards like the 3D topologic knowledge, 3D metric knowledge and 3D processing knowledge. Spatial knowledge contains relationships such as *disjoint*, *contain*, *inside*, *cover*, *equal*, *overlap*. The terms represent the geometric relations between components of an object or between objects. Each of the mentioned types of spatial knowledge contains a variety of relations modelled in the ontology structure. The top level ontology is designed to include the topological relationships. This is used to enrich an existing KB to make it possible to define topological relationships be-

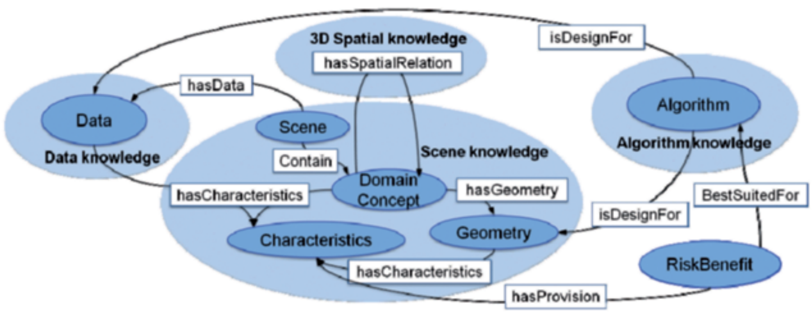

Fig. 2: General ontology schema overview.

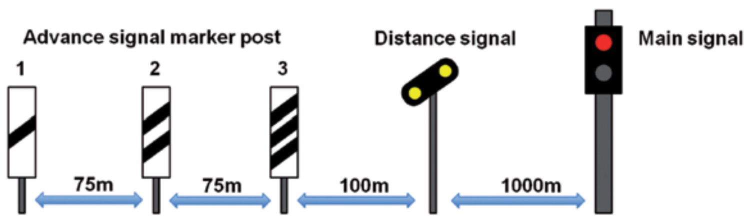

Fig. 3: Metric rules.

tween objects in a specific case. Metric knowledge presents important information, because the different elements fulfil very strict metric rules which can also be used for the detection and classification process. In the example of scenes specific for railways, Fig. 3 shows an ontological structure supported by the SWRL rules which can automatically specify that an object with certain characteristics that has a distance of 1000 ± 0.5 m from *Distance signal,* can be a *Main signal.*

Regarding the numerical processing algorithms, effectiveness depends on the quality of the data (resolution, noise), the characteristics of the object that needs to be detected, or other factors depending on a specific case. Algorithms are modelled under specialized classes of algorithms, sharing certain taxonomical and relational behaviour. The hierarchical representation of the algorithms is addressed by dividing the algorithms according to the context in which they are executed. Classes, including *GeometryDetection*, *AppearanceDetection*, *ImageProcessing* and *NoiseReduction*, follow such a hierarchal structure. Likewise, relational semantics are represented by properties. In broader terms, there are two types of relationships: one which applies to the geometry that an object in *Domain Concept* possesses, and one which relates distinct objects. The first category of relationships is used for detecting geometries. The object property *isDesignedFor* maps algorithms to the respective geometries. For example: *LineDetection1 isDesignedFor lines*. The second set of algorithm properties *hasInput/hasOutput* are inter-relational properties to connect algorithms based on the compatibility of output from an algorithm to the inputs of others.

It is necessary to adapt the processing parameters depending on data, scene and characteristics of objects to enable a well focussed detection and classification. The concept allows for these interactions, as it is able to automatically change the strategy based on a compromise between quality and risks. A part of the KB is dedicated to risk-benefit factors that have an influence on the algorithms. This was derived from "trial and error" simulations with every individual algorithm. Since an algorithm may perform best with some given parameters in one setting, and fail to deliver the same quality in other settings, it is important to assess the risk-benefit factors of every algorithm with various possible settings. The class *RiskBenefits* includes all identified risks and benefits. The class contains the four instances *Distinct*, *Illusive*, *Noise*, and *DetectionError*. The instances are the risks or the benefits with some influence on the algorithms or at least on the parameters of the algorithms. Note that the classes above form an ontology, which can also be used for other domains, such as creation of semantic annotated maps by a mobile robot, mobile mapping of street furniture or forests, and semantic place labelling from airborne laser data.

Knowledge modelling and human interaction: The process of modelling knowledge requires the user to collect "information" from related domains. This process is currently carried out manually. "Collecting information" can imply extracting knowledge from various sources or filling the ontology with objects corresponding to specific classes, object properties, algorithms, algorithmic properties, etc. Some of these tasks such as data extraction from technical documents have the potential to be done automatically using specialized processing tools borrowed from the document analysis community (Tang et al. 1996). Depending on the available tools and target application including its related domains, the knowledge modelling process may take a single person from one to several days of work (data extraction and ontology modelling) including interaction with domain experts and modelling all relationships. Examples for the length of this process and the amount of human interaction are given in section 6. However, although such figures may seem significant, one has to keep in mind that knowledge modelling for a given application is done only once and used for processing numerous point clouds with virtually very little or no changes to the ontology. Other approaches such as those based on machine learning would also require a significant amount of preparation to extract training data and carry out annotations generally from large amounts of scans, which may require at least as much time as modelling an ontology. This is especially true when dealing with special environments such as railways or industrial plants, which are often subject to regulations, which require a certain level of expertise.

5 Knowledge Guidance for the Object Detection and Classification Process

5.1 *Knowledge-driven Strategy*

The knowledge formalization is based on the understanding of the underlying semantics and processes it using technologies such as OWL. The top-level ontology presents the main knowledge framework and holds generic semantics for all addressed domains. Regarding the case studies, this framework contains the scene, object geometries, spatial relations and algorithms and originates from existing knowledge sources, such as information systems, or guidelines of the Deutsche Bahn (DB, German Railways), and an extensive study of the sample scenarios. Obviously, quality and completeness of such formalized knowledge strongly influence the quality of the results, and have to be adapted to the individual application. In the general case, such a framework only contains the abstract and general knowledge of object categories, the structure of a scene, geometric relations between objects, the structure of data, the nature of algorithms and the potential relationships between these components. In a simpler scenario with specific information about potentially existing objects, for example known through CAD or Industry Foundation Class (IFC) files, the detection strategy can be guided more easily and may be reduced to a change detection problem.

Starting from the initial situation, the process iteratively updates the KB at certain stages. At the beginning of each iteration, the content of the KB is used to detect new features, may it be a new object or a component of it. These new geometric features are passed on to the KB in order to extend the KB for the following classification. This classification is guided by the content and the structure of the KB, which has reasoning capabilities based on property restrictions or rule languages (such as SWRL) and refines the actual content. This refined content is used in the next iteration. The process is repeated until all entities have been completely annotated and meet the following convergence conditions: (1) All objects defined on the KB are detected and annotated (simple change detection). (2) A predefined number of iterations without refinement for any entity has been reached.

5.2 *Usage of Algorithms guided by Knowledge*

Object related knowledge does not influence classification only, but also algorithmic processing. Different algorithms are designed for different contexts. The differences can be addressed and properly modelled. The KB holds the algorithm knowledge in the class *Algorithm*. This class is related to other classes inside the KB, such as objects. This allows for the modification of the role of algorithms, e.g. parameter, sequences, corresponding to the KB details. The interrelationships among different algorithms are mapped through compatibility of their input and output characteristics (Fig. 4). Fig. 4 illustrates that more than one path from an initial algorithm to a desired one exist. We use the well-known Djikstra's algorithm (Djikstra 1959) for finding the shortest path in the graph leading to the desired algorithm. This approach has the advan-

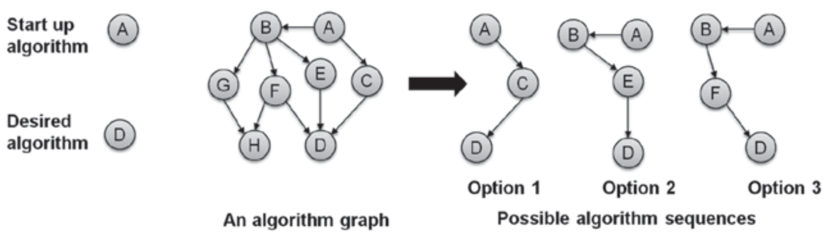

Fig. 4: Algorithm sequences extracted from the graph.

tage of preventing the sequence of algorithms to form an endless loop and allows for finding an appropriate sequence.

At any given iteration, each entity in the KB may be assigned a new label: *identified*, *unknown* or *ambiguous*. This label may change in the course of an iteration. Based on this information, the ASM chooses the best algorithm for generating new characteristics, which will help in the next classification step. This selection also integrates the choice of an optimal sequence out of several possible ones (routes) of algorithms (or nodes). Various knowledge components can have an impact here, e.g. data (noise, point density, point of view), object (size, shape, orientation), and scene (possible objects, neighbourhood).

5.3 *Classification Step*

As discussed in section 4, the ontology schema holds the semantics of the objects such as its geometries and other spatial characteristics. This information supports identifying detected entities and is used in the inference process. The complexity of the required rules directly depends upon the complexity of the processed situation. In simple cases, even very simple rules are sufficient to produce a correct result. However, this concept also allows to handle more complex situations. A simple classification of an entity (*Geometry*) based on a SWRL rule annotates an electric pole (type 2), as found along railway tracks:

Geometry (?x) ^ hasHeight(?x, ?ht) ^ swrlb:greaterThan(?ht, 4) ^ swrlb:lessThan(?ht, 6) → ElectricPole2(?x) (3)

The use of spatial relations (Metric, Topologic, and Directional) between the detected entities is one possible extension of such simple geometry (Ben Hmida et al. 2012). It only requires the appropriate algorithms and then provides the result for the topological operation. Zlatanova et al. (2002) gives a survey of different 3D models and relations. The spatial operators available for a spatial query language consist of 3D topological operators (Borrmann & Rank 2008), 3D metric operators (Borrmann et al. 2009), 3D directional operators (Borrmann & Rank 2009) and finally 3D Boolean operators (Borrmann et al. 2006). In a simplified example, the following rule specifies that a "Building" that overlaps a "Railway" (both defined in the ontology), is a "RailwayStation".

Building(?b) ^ Railway(?r) ^ topo: overlaps(?b, ?r) → RailwayStation(?b) (4)

Fig. 5 shows our process guided by various knowledge domains in object detection and classification. In this figure, object classes are referred to as A, B, C, D, and E. We recall here that the process iterates until convergence, i.e. all objects are labelled, or stopping conditions, i.e. maximum number of iterations without refinement, are met.

6 Case Study

Two case studies illustrating our approach are presented in this section: Deutsche Bahn (DB) and Frankfurt Airport (Fraport). The goal in both cases was to detect and check relevant objects inside a defined work area.

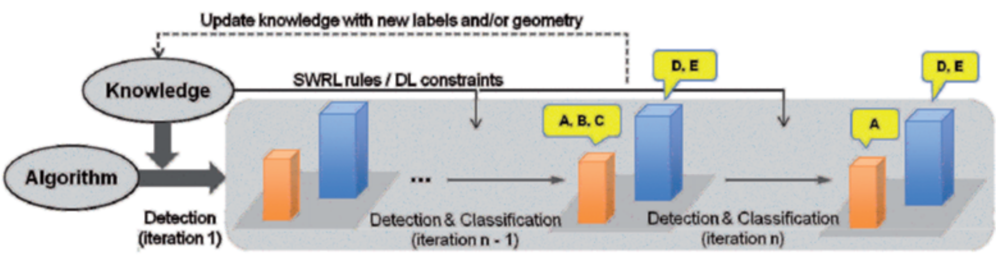

Fig. 5: Knowledge-driven method for object detection and classification process.

6.1 *Object Classification in the Railway System (DB)*

In the DB example, we used scans in the vicinity of the tracks. Data were captured from LIMEZ III, a surveying train equipped with a laser scanner mounted at its front-end. Two non-domain experts worked for approximately 20 days to build the DB example ontology. They were supported by experts of the German railway (DB). The available knowledge is used to classify the entities as:

- *Identified*: as soon as a feature value is in the range of a class. This annotation has to be supported by subsequent classifications and remains valid as long as no conflict is detected.
- *Ambiguous*: as soon as a feature value satisfies more than one class. Both annotations are stored and have to be separated by subsequent classifications and remain doubtful as long as no separation is possible.
- *Unknown*: indicates that a feature value does not match any existing class. Further processing requires the ASM to select other properties in order to continue the process.

Although a simple example, this nevertheless shows the general logic, which can then be further extended with other considerations among entities. Success is directly related to the ability to detect entities and the significance of the feature values chosen. Less characteristic features can also be used. However, these will require more iterations and additional rules in order to achieve a stable classification.

The aspect of quality can also be incorporated into the concept. This may either be realized by thresholds modelling data noise or by changing the strategy of selecting a path through the graph. The latter case handles situations in which features are sensitive to noise and corresponding algorithms might fail. For instance, an electric pole (type 2) is represented by parallel vertical supports. ASM searches and selects the relevant algorithm – *CheckParallel* – from the algorithmic library. This library is described by a graph (see Fig. 6) representing all allowed connections, based on input and output between algorithms. Based on some data quality thresholds, the sequence may or may not include pre-processing algorithms (e.g. *NoiseReduction*). On the path from the starting algorithm (in this case, *PositionDetection*) to the desired algorithm (*CheckParallel*), ASM infers and invokes all concerned algorithms based on the *hasInput/hasOutput* property. *Segmentation*, *NoiseReduction* and *LineDetection1* are the selected ones. Afterwards, ASM links them together to create a proper sequence: it then looks as follows (result illustrated in Fig. 7c):

PositionDetection → *Segmentation* → *NoiseReduction* → *LineDetection1* → *CheckParallel*.

The execution of this sequence provides a list of recognized object entities, which then

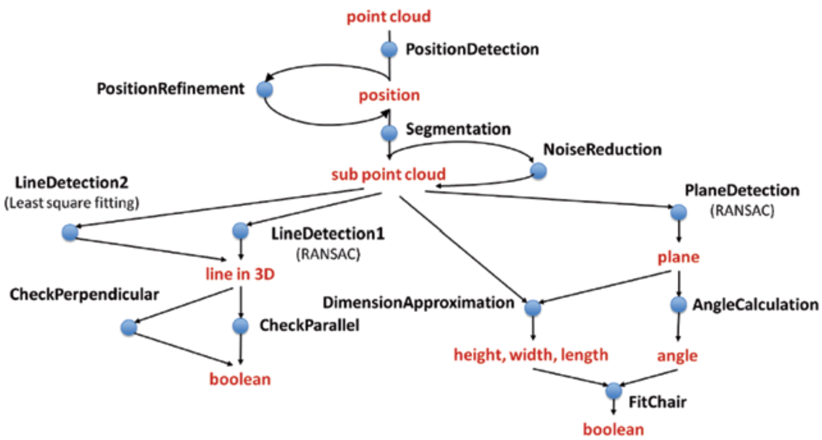

Fig. 6: Graph of possible algorithmic paths generated by ASM and used for detecting objects in both DB and Fraport cases.

are classified. Further sequences are used to improve the quality and to reduce the ambiguity within the results (Fig. 7d). Iterations are repeated until a complete annotation for all entities is performed. The convergence conditions are applied to terminate the detection process for entities.

We have processed a 500 m section along the railway. Out of 12 algorithms modelled in the KB (Fig. 6), the following ones were used by the system to classify objects (Tab. 1): *PositionDetection*, *Segmentation* (cropping points surrounding a given position), *DimensionApproximation*, *NoiseReduction*, *LineDetection1* (using RANSAC) and *AngleCalculation*. Knowledge was collected carefully in order to provide a reliable KB related to objects, scene, the nature of the data, algorithms and relationships between them. The base was progressively extended with new knowledge gained either from the analysis of the detected geometries or from classification results. Initially, 17 classes were defined as subclasses of the 5 classes in Tab. 1. These classes represent different types of signals and electric poles that can be found along the tracks and are of interest to our study. A total of approximately 500 geometries such as 3D line segments, angles and points of interest were recognized, 10 SWRL rules are used and 63 entities (possible object positions) were identified after the initialization step shown in (Fig. 7b). All entities include possible objects in the scene but also noise and objects of no interest. The true number of railway objects was 13 (Tab. 2). With the second iteration, the process tries to refine the results and classify the objects. At the end, 10 out of 13 real railway objects were correctly classified, 50 entities which represent non-railway objects were classified as *unknown*, and 3 railway objects could not be unambiguously classified with the rules implemented. The results in Fig. 7d were obtained by our software system. Computation took about 10 minutes on an Intel Xeon 2.4 GHz with 12G RAM. Note that our software is a prototype and has not been optimized for performance. In our experiments, we used the "shortest path" criterion from starting the algorithm to the desired algorithm in order to find the optimal algorithm sequence. Our system assumes equal weights for all edges in the algorithms graph, i.e. factors that are intrinsic to algorithms such as time and memory requirements are not taken into account at this stage. Results can be improved by applying more complex rules, possibly using additional geometric constraints such as line or plane orientation, angle between lines or number of lines expressed in the rule (5):

Geometry(?x) ^ hasLine(?x, ?l) ^ line(?l) ^ DistanceSignal (?y) ^ DistanceFrom(?x, ?y, ?dis) ^ swrlb:GreaterThan (?dis,1000) ^ hasHeight(?x, ?h) ^ swrlb:GreaterThan (?h, 4) ^ hasVerticalLineNumber(?x, ?vn) ^ swrlb:lessThanOrEqual(?vn, 2) ^ hasObliqueLineNumber(?x, ?on) ^ swrlb:equal(?on, 0) → MainSignal(?x) (5)

In order to relate the classification to human interpretation the point cloud was presented to test persons. They identified 8 of 13 railway objects based on a visual inspection of the cloud and without taking into account topological or descriptive knowledge. This just shows the limited representation of objects inside such types of point clouds. One major reason for the poor quality of the point cloud is the fact that only the side of the object facing the tracks is captured due to the scanner on the train. However, this also shows the usefulness of additional knowledge.

Tab. 1: Classes and properties used in DB scenario.

Class	Object properties
Electric pole (type 1)	Vertical structure, height, perpendicular lines
Electric pole (type 2)	Vertical structure, height, parallel lines
Electric pole (type 3)	Vertical structure, height, oblique line
Main signal (mechanical)	Vertical structure, height, perpendicular lines, parallel line, number of lines
Main signal (light)	Vertical structure, height, perpendicular lines, parallel line, oblique line, number of lines

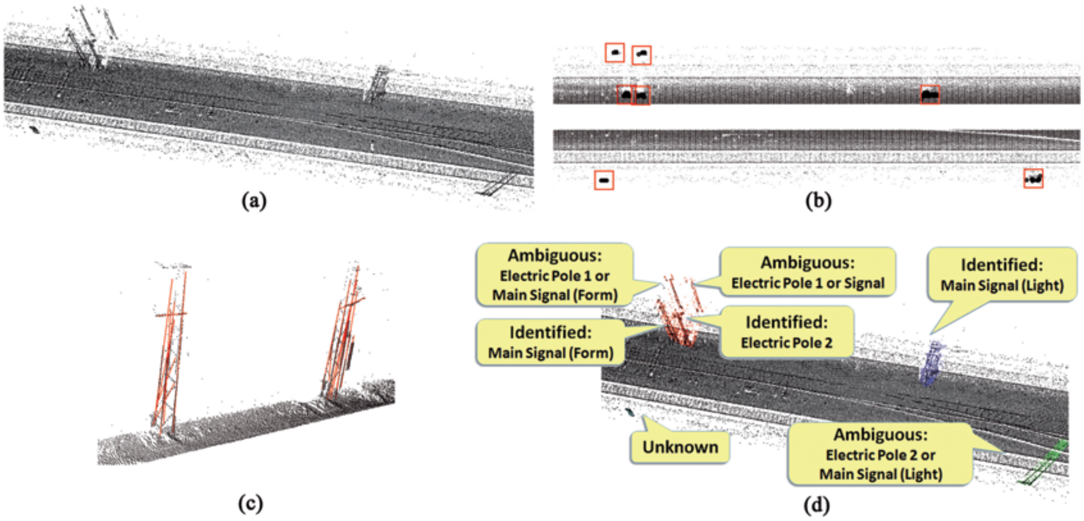

Fig. 7: (a) Point cloud representation of a section of a railway; (b) Results after executing the initialization step, projecting the point cloud to the ground plane, rectangles denote possible object positions; (c) Results from detecting 3D lines of a signal and electric pole (type 3) along the railway; (d) Positions of objects and annotation results after the first iteration.

Tab. 2: Experiment in a section of DB railway, comparison result between two approaches: Visual inspection using the standard software tool of DB, and knowledge-based data processing.

Object	Visual inspection	Knowledge-based data processing
Electric pole (type 1)	1/1*	1/1
Electric pole (type 2)	2/4	4/4
Electric pole (type 3)	1/1	1/1
Main signal (mechanical)	2/4	2/4
Main signal (light)	2/3	2/3
Total	8/13 (61.53 %)	10/13 (76.92 %)
(*) Number of detected objects over number of ground-truth objects.		

Results obtained after the processing along the tracks are shown in Tab. 2. Remarkable to see that the only failures using knowledge were Main Signals (light) that could also not be recognized by visual inspection. This is mainly caused by the poor quality of the data, especially in terms of point density, which made such structures hardly visible and undistinguishable. Some objects, the type 2 electric poles, were successfully identified using the automated detection and classification whereas visual inspection failed.

6.2 *Object Detection inside Airport Building (Fraport's Waiting Area)*

In the second case, we used scans from an environment inside the airport buildings, typically a waiting area. Changes in the technical infrastructure were of main interest. Data were obtained from classical terrestrial laser scanning. The Fraport scenario is different from the DB test example because a data base of expected objects in the scene exists and can be used as a-priori knowledge. Two persons worked for about 10 days to fill the ontology with knowledge such as properties of

Tab. 3: Classes and properties used in the Fraport scenario.

Class	Object properties
Wall	Vertical plane, length, height
Separation panel	Vertical plane, length, height
Advertising panel	Vertical plane, length, height, number of planes
Chair	Horizontal plane, leaning plane, angle between planes, length of chair

objects, scene, nature of data and characteristics of buildings. The data sources were CAD plans, related documents from the experts and observations from the real scene. The process first attempted to validate the presence of static objects such as walls, and separation or advertising panels in the point cloud that were supposed to exist according to the data base (Tab. 3). After that, moveable objects like chairs, trash bins, were detected and also fed into the KB. The initialization was different from the DB case because of more complex objects and the prominent role of many vertical planes. Therefore, we first detected vertical planes (Fig. 8a, b). This was possible by a vertical projection of the point cloud followed by Hough Line detection to locate the static objects' position on the ground plane. *VerticalProjection* and *HoughLineDetection* are included in *PositionDetection* algorithm. Points with a vertical projection in the vicinity of these lines were used to define segments corresponding to vertical planes. The following step was used to verify walls, separation panels or advertising panels defined in the data base based on their particular length and height (Fig. 8e). However, there are also many moveable objects like chairs, tables, counters, or trash bins, which also need to be detected to update the KB. All objects already available from the first validation phase gave a geometric and semantic frame helping to support the detection of unknown moveable objects. For example, chairs were searched for in a specific area defined within a certain distance from the wall (5 m in our experiments) and 0.7 m above the floor. Note that the reference frame of our point cloud is attached to the floor such that the latter is simply determined by fitting a horizontal plane (initialized at height Z = 0) using the *PlaneDetection* algorithm. We focussed on detecting walls in the border region of the check-in area. The static structures obtained

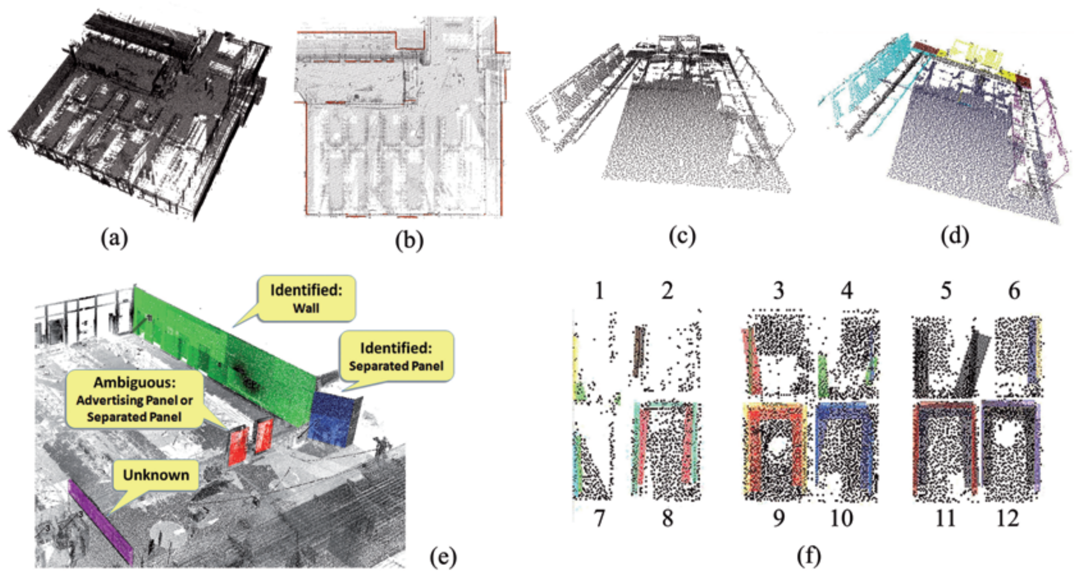

Fig. 8: Fraport scenario: (a) 3D scan of a check-in area, (b) detected walls, (c) point cloud exhibiting chairs, (d) detection results of a chair set, (e) annotated static objects, (f) identification results obtained on 12 chair sets in a waiting area (failures 1–2, partial detection 3–7, successful identification 8–12).

from the point cloud are shown in Fig. 8e. Only two walls exist in the scene and the remaining larger static structures are either separation or advertising panels, which are easily distinguishable from walls by their height. Both walls were successfully identified.

After the walls were detected, ASM generated, based on the properties of a chair, i.e. chair's length, horizontal plane, leaning plane, angle between two planes, an appropriate sequence of algorithms to invoke:

PositionDetection → *Segmentation* → *PlaneDetection* → *DimensionApproximation* → *AngleCalculation* → *FitChair*

FitChair is used to combine the detected geometries of a chair as depicted in Fig. 8d, f. A rule is also applied to classify chairs:

Geometry(?x) ^ hasCorrespondingGeo(?x, ?l) ^ LeaningPlane(?l) ^ hasCorrespondingGeo(?x, ?s) ^ HorizontalPlane(?s) ^ hasAngle(?x, 120) ^ hasLength(?x, ?len) ^ swrlb:greaterThan(?len, 370) ^ swrlb:lessThan(?len, 380) → Chair(?x) (6)

Chair sets are arranged parallel to the walls and represented by very sparse point clouds (Fig. 8c). Nevertheless, it is possible to detect, model and identify chair sets based on a sequence of algorithms making use of topological and geometrical constraints arising from previously detected elements. Six algorithms were used (out of the 12 in Fig. 6) such as: *PositionDetection*, *Segmentation*, *DimensionApproximation*, *PlaneDetection* (based on RANSAC), *AngleCalculation* and *FitChair* (which verifies a chair by two connected planes in an angle of 120°).

The results obtained are shown in Fig. 8f in which the five chair sets 8-12 were successfully identified, the five chair sets 3–7 were only partly detected and the two chair sets "1" and "2" could not be identified due to missing points. In the next stage of processing, objects were verified using topological constraints, such as a distance-based identification from the identified objects. Finally, 10 out of 12 chair sets could be correctly classified even in an insufficient dataset. The results reported here were obtained with an ontology that had been filled with approximately 350 detected geometries (planes, line segments…) and used 4 SWRL rules. The process took about 7 minutes on an Intel Xeon 2.4 GHz with 12G RAM when using our prototype software. The full process of detecting chair sets including wall identification is depicted in Fig. 9.

7 Conclusion

This paper presents a knowledge-driven approach to detect objects in point clouds. It is based on the semantics of different associated domains which assist in detecting and classifying objects. Knowledge supports all processing steps including the arrangement of the data processing. This allows inter-relating the characteristics of algorithms with those of the objects in the domain of the application. Our system also provides the flexibility to infer the strategy from existing knowledge, and to adapt the processing to the application-specific requirements. The permanent interaction between the algorithms and the KB allows for a smooth and gradual construction of the KB which contains at the end of the process all entities which can be detected and identified. Admittedly, it takes time to collect the knowledge at the beginning. However, it has only to be collected once and is later always available

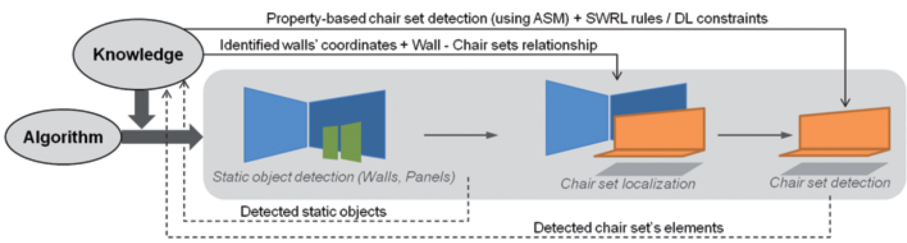

Fig. 9: Chair set detection process.

when needed. In addition, the KB can be iteratively extended by the operator at many practical waypoints. The quality of results depends on the robustness of the implemented algorithms, the selected strategy and the amount of knowledge integrated. In practice, the solution is oriented towards the requirements of a specific application. Further development is needed to make algorithms more robust to quality variations in the data, and to segment more complex objects. Furthermore, the knowledge sources (data features, object properties and scene characteristics) have to be extended in order to enhance the classification processing, especially regarding ambiguous cases. Lastly, both an expansion of the ontology and further implementation and testing of rules are currently considered and subject to investigation.

Acknowledgements

The work presented in this paper is part of the research project WiDOP – Wissensbasierte Detektion von Objekten in Punktwolken für Anwendungen im Ingenieurbereich, funded by the German Federal Ministry for Research and Education (grant no. 1758X09). Project partners are Metronom Automation GmbH, DB Netz AG (railway) and Fraport AG (facility management).

References

Anguelov, D., Taskarf, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G. & Ng, A., 2005: Discriminative learning of markov random fields for segmentation of 3d scan data. – Computer Vision and Pattern Recognition (CVPR) **2:** 169–176.

Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F. & Stein, L.A., 2004: OWL web ontology language reference. – W3C recommendation 10.

Ben Hmida, H., Cruz, C., Nicolle, C. & Boochs, F., 2011: From 3D Point Clouds To Semantic Objects An Ontology-Based Detection Approach. – International Conference on Knowledge Engineering and Ontology Development, Paris, France.

Ben Hmida, H., Cruz, C., Nicolle, C. & Boochs, F., 2012: From Quantitative Spatial Operator to Qualitative Spatial Relation Using Constructive Solid Geometry, Logic Rules and Optimized 9-IM Model. – International conference on computer sciences and automation engineering **2:** 453–458, Zhanjiajie China.

Borrmann, A., Van Treeck, C. & Rank, E., 2006: Towards a 3D spatial query language for building information models. – Computing and Decision Making in Civil and Building Engineering.

Borrmann, A. & Rank, E., 2008: Topological operators in a 3D spatial query language for building information models. – Conference on Computing in Civil and Building Engineering.

Borrmann, A. & Rank, E., 2009: Specification and implementation of directional operators in a 3D spatial query language for building information models. – Advanced Engineering Informatics **23:** 32–44.

Borrmann, A., Schraufstetter, S. & Rank, E., 2009: Implementing metric operators of a spatial query language for 3D building models: octree and B-Rep approaches. – Journal of Computing in Civil Engineering **23** (1): 34–46.

Busch, A., Gerke, M., Grünreich, D., Heipke, C., Liedtke, C. & Müller, S., 2005: Automatisierte Verifikation topographischer Geoinformation unter Nutzung optischer Fernerkundungsdaten. – PFG – Photogrammetrie, Fernerkundung, Geoinformation **2:** 111–122.

Cantzler, H., Fisher, R. & Devy, M., 2002: Quality enhancement of reconstructed 3D models using coplanarity and constraints. – Pattern Recognition: 34–41.

Dijkstra, E.W., 1959: A Note on Two Problems in Connexion with Graphs. – Numerische Mathematik **1:** 269–271.

Durand, N., Derivaux, S., Forestier, G., Wemmert, C., Gancarski, P., Boussaid, O. & Puissant A., 2007: Ontology-based Object Recognition for Remote Sensing Image Interpretation. – Tools with Artificial Intelligence (ICTAI 2007): 472–479.

Green, P.J., 1995: Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. – Biometrika **82** (4): 711–732.

Growe, S. & Tonjes, R., 1997: A knowledge based approach to automatic image registration. – International Conference on Image Processing **3:** 228–231.

Hedau, V., Hoiem, D. & Forsyth, D., 2010: Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry. – The European Conference on Computer Vision (ECCV): 224–237.

Helmholz, P., Becker, C., Breitkopf, U., Büschenfeld, T., Busch, A., Braun, C., Grünreich, D., Heipke, C., Müller, S., Ostermann, J., Pahl, M., Rottensteiner, F., Vogt, K. & Ziems, M., 2012: Semi-Automatic Quality Control of Topographic Datasets. – Photogrammetric Engineering & Remote Sensing **78** (9): 959–972.

Huang, H., Brenner, C. & Sester, M., 2011: 3D building roof reconstruction from point clouds via generative models. – International Conference on Advances in Geographic Information Systems: 16–24.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B. & Dean, M., 2004: SWRL: A semantic web rule language combining OWL and RuleML. – W3C Member submission **21:** 79.

Maillot, N. & Thonnat, M., 2008: Ontology Based Complex Object Recognition. – Image and Vision Computing (IVC) **26** (1): 102–113.

Matsuyama, T., 1987: Knowledge-based aerial image understanding systems and expert systems for image processing. – Geoscience and Remote Sensing **3:** 305–316.

Nurunnabi, A., Belton, D. & West, G., 2012: Diagnostic-robust statistical analysis for local surface fitting in 3D point cloud data. – ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences **I-3**, Melbourne, Australia.

Pu, S. & Vosselman, G., 2006: Automatic extraction of building features from terrestrial laser scanning. – International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences **36:** 25–27.

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

Ripperda, N. & Brenner, C., 2006: Reconstruction of Facade Structures Using a Formal Grammar and RjMCMC. – The **28th** conference on Pattern Recognition: 750–759.

Rost, U. & Münkel, H., 1998: Knowledge based configuration of image processing algorithms. – International Conference on Computational Intelligence & Multimedia Applications.

Scholze, S., Moons, T. & Van Gool, L., 2002: A probabilistic approach to building roof reconstruction using semantic labelling. – Pattern Recognition: 257–264.

Tang, Y., Lee, S.W. & Suen, C., 1996: Automatic document processing: A survey. – Pattern Recognition **29** (12): 1931–1952.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P. & Koehl, M., 2007: Model-driven and data-driven approaches using LIDAR data: analysis and comparison. – International Archives of Photogrammetry, Remote Sensing and Spatial Information Systems: 87–92.

Teboul, O., Simon, L., Koutsourakis, P. & Paragios, N., 2010: Segmentation of Building Facades Using Procedural Shape Priors. – Computer Vision and Pattern Recognition: 3105–3112.

Trinder, J.C. & Wang, Y., 1998: Knowledge-based road interpretation in aerial images. – International Archives of Photogrammetry and Remote Sensing **32:** 635–640.

Vosselman, G. & Dijkman, S., 2001: 3D building model reconstruction from point clouds and ground plans. – International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, **34** (3/W4): 37–44, Athens, Georgia, USA.

Zlatanova, S., Rahman, A. & Shi, W., 2002: Topology for 3D spatial objects. – International Symposium and Exhibition on Geoinformation: 22–24.

Addresses of the Authors:

M.Sc. Hung Truong & M.Sc. Helmi Ben Hmida, University of Applied Sciences Mainz und University of Burgundy, e-mail: {truong}{helmi.benhmida}@geoinform.fh-mainz.de und {quoc-hung.truong}{helmi.benhmida}@u-bourgogne.fr

Prof. Dr. Frank Boochs, i3mainz, University of Applied Sciences Mainz, Lucy-Hillebrand-Str. 2, 55128 Mainz, Tel.: +49-6131-628-1489, Fax: +49-6131-628-91489, e-mail: frank.boochs@geoinform.fh-mainz.de

Prof. Dr. Adlane Habed, Prof. Dr. Christophe Cruz, Prof. Dr. Yvon Voisin & Prof. Dr. Christophe Nicolle, University of Burgundy, Le2i, Route des Plaines de l'Yonne, BP 16, 89010 Auxerre cedex, France, Tel.: +33-3-86-49-28-51, Fax: +33-3-86-49-28-50, e-mail: {adlane.habed}{christophe.cruz}{yvon.voisin}{christophe.nicolle}@u-bourgogne.fr

Manuskript eingereicht: März 2012
Angenommen: März 2013