

KNOWLEDGE-BASED OBJECT DETECTION IN LASER SCANNING POINT CLOUDS

F. Boochs *, A. Karmacharya, A. Marbs
i3mainz, Institute for Spatial Information and Surveying Technology,
University of Applied Sciences Mainz,
Lucy-Hillebrand-Str. 2, 55128 Mainz, Germany,
(boochs, ashish, marbs)@geoinform.fh-mainz.de

Commission III, Working Group III/2

KEY WORDS: Point Cloud, Ontology, Modeling, Understanding, Processing, Classification, Algorithms

ABSTRACT:

Object identification and object processing in 3D point clouds have always posed challenges in terms of effectiveness and efficiency. In practice, this process is highly dependent on human interpretation of the scene represented by the point cloud data, as well as the set of modeling tools available for use. Such modeling algorithms are data-driven and concentrate on specific features of the objects, being accessible to numerical models. We present an approach that brings the human expert knowledge about the scene, the objects inside, and their representation by the data and the behavior of algorithms to the machine. This “understanding” enables the machine to assist human interpretation of the scene inside the point cloud. Furthermore, it allows the machine to understand possibilities and limitations of algorithms and to take this into account within the processing chain. This not only assists the researchers in defining optimal processing steps, but also provides suggestions when certain changes or new details emerge from the point cloud. Our approach benefits from the advancement in knowledge technologies within the Semantic Web framework. This advancement has provided a strong base for applications based on knowledge management. In the article we will present and describe the knowledge technologies used for our approach such as Web Ontology Language (OWL), used for formulating the knowledge base and the Semantic Web Rule Language (SWRL) with 3D processing and topologic built-ins, aiming to combine geometrical analysis of 3D point clouds, and specialists’ knowledge of the scene and algorithmic processing.

1. INTRODUCTION

A recent development in scanning technology is the ability to provide very precise data through the generation of highly dense 3D point clouds. Such high-density point clouds provide a digital replica of the scanned scene. From the early days in 3D point cloud processing, the research has been focused on investigating the reconstruction and recognition of geometrical shapes (Wessel, Wahl, Klein, & Schnabel, 2008), (Golovinskiy, Kim, & Funkhouser, 2009). More complex strategies try to reconstruct complete sites. They can be broadly categorized into two categories (Pu, 2010): data driven and model driven. Data driven methods (Beker, 2009), (Frueh, Jain, & Zakhor, 2002) extract selected geometries from the point cloud and combine them into a final model. With these methods, the huge redundancy of point cloud data produces difficulties due to corresponding high ambiguity. Model driven methods try to take this into account. They use predefined primitive templates and information (as detected geometries) from the data to map them against the most likely templates (Ripperda, 2008).

We present a knowledge driven method in which knowledge about the scene and the algorithmic processing is formalized logically for the generation of algorithmic sequences that detect objects automatically. The developments are part of the project WiDOP (*Knowledge based detection of objects in point clouds for engineering applications*), which implements this method in order to detect and identify objects in 3D point clouds. Knowledge driven methods for object detection are relatively new. Pu (2010) detected exterior structures (mainly facades) of buildings through knowledge, and Maillot & Thonnat (2008)

also detected complex features using a knowledge base. Both works separate detection and qualification into two independent steps. Detection is based on predefined algorithmic sets, while qualification uses knowledge to classify objects according to their nature. Our work avoids such a separation and provides a semantic bridge between scene knowledge and algorithmic knowledge. Knowledge is part of an Algorithm Selection Module (ASM), which guides the processing independently of a particular scene. As a generic solution, it is extendable to any scene or algorithm.

The project uses environments from the Deutsche Bahn (German Rail**) and Fraport (operator of Frankfurt Airport)*** to demonstrate effectiveness and versatility. Terrestrial laser scanning technology is used to capture point cloud data. These datasets are then used on demand to create models of the objects inside of the installations. To date, the tasks of creating and evaluating the 3D object models are solely manual, and hence are costly in terms of both time and resources. The existing tools do not provide significant assistance either, as they are mostly data driven and concentrate on specific features of the objects to be used for numeric models. Algorithms have limited flexibility and can provide adverse effects when deviated. Knowledge of algorithms and their limitations during implementation could limit such adverse effects. In the meantime, it provides flexibility to algorithmic manipulation for different scenarios.

** <http://www.bahn.de>

*** <http://www.fraport.com/content/fraport-ag/en.html>

2. THE WIDOP PLATFORM

Integrating knowledge into a processing strategy provides much needed flexibility. However, it is clear that knowledge varies at different stages. This variation depends mainly on type, amount, and the quality of knowledge available during processing, as well as on the ability to connect different sources and domains of knowledge (related to objects, algorithms, scenes, data and so on). Additionally, knowledge should increase step by step based on the quality of results collected from concrete applications. Success will also clearly increase with increasing amounts of available knowledge. We therefore distinguish different scenarios.

2.1. The Scenarios

2.1.1. Known objects, known positions

Detailed knowledge (exact positions and characteristics) of objects already exists in such cases. The knowledge base ("KB") supports the processing for verification.

2.1.2. Known objects, unknown positions

This case reflects a typical situation, in which knowledge about scene objects exists but not their location in the data. The KB that provides the scene knowledge interacts with the processing knowledge to determine the probable sequences that detect the objects and derive their location.

2.1.3. Unknown objects, unknown positions

This is the most complex case, in which only generic knowledge about the scene exists. In such cases, the KB recommends the detected geometries to their object types through examining the semantics defined against them.

2.2. The Iterative Approach of Classification

This approach is used to derive concrete detection from a generic base. We call it the "Iterative Semantic Classification Method" or *ISCM*. Semantic Figure 1 illustrates the iteration method. Details on it will be presented in further sections.

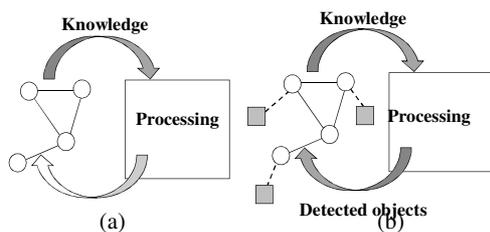


Figure . ISCM (a) Basic knowledge framework (b) Knowledge population

The initial knowledge is mainly a schema that represents the scene and the processing knowledge. It is hence not a concrete knowledge source (fig.1a). It has to be enriched with real objects in the course of the iterative process. The knowledge is refined after every step of processing, through the population of the results into the knowledge schema. It thus transforms the knowledge schema into a concrete and comprehensive knowledge base (fig. 1b).

2.2.1. Knowledge Domains

Building on the works of Pu (2010) and Maillot & Thonnat (2008), knowledge of algorithmic processing is related to that of objects in the scene in order to support their detection. In this manner, the mapping of algorithmic knowledge to the scene and objects can infer processing, and determines which algorithms are best suited for any particular characteristic of the objects. This process makes the methodology scene independent.

The knowledge domains of *Algorithms* and *Scene* are mapped through rules, which are related to geometry, topology etc. These mappings infer best suitable algorithms or algorithmic sequence for detecting geometries. Once detected, they are related to their corresponding objects inside the KB. The preexisting scene knowledge is then used for verification. Beside these two, other supporting knowledge domains provide significant supports. They are seamlessly integrated within the knowledge schema through their semantic interpretation and relationship to the main knowledge domains.

The solution is based on knowledge technologies of the Semantic Web (Berners-Lee, 1998) framework. The WiDOP platform uses knowledge technologies like Web Ontology Language (Bechhofer, et al., 2004), (Patel-Schneider, Hayes, & Horrocks, 2004) or the Semantic Web Rule Language (SWRL) (Horrocks, et al., 2004). The knowledge equations used here are based on Description Logics (DL) which is core in the rapid development of the knowledge technologies. The next section discusses the ontology schema of the WiDOP platform (expressed in OWL) to demonstrate its robustness to adapt into any structural domain.

2.2.2. Ontology Schema

The top level knowledge is illustrated in figure 2. The top level classes of algorithmic and scene knowledge are represented through the top level classes *Algorithms* and *DomainConcept* respectively. The class *Algorithms* constitutes the algorithmic knowledge through a taxonomical hierarchy, and semantic rules through restrictions. Similarly, the class *DomainConcept* presents the scene knowledge through a hierarchical structure reflecting the objects in the scene and semantic rules.

The basic ontology schema provides an overview of the scene and processing knowledge, defining *what* knowledge exists in different domains and *how* they are interrelated. They are defined by rules which facilitate selecting the algorithms and define the strategy to detect the objects in the scene.

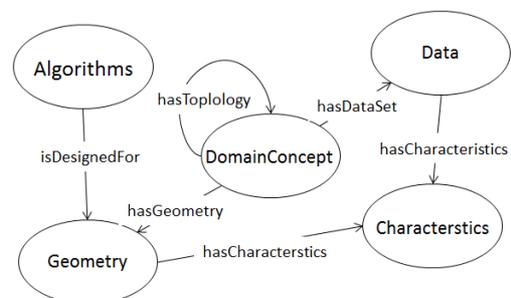


Figure . General overview of ontology schema

The top level classes represent knowledge in their specific areas. We can see there are other top level classes beyond *Algorithms* and *DomainConcept* in figure 2. There are other knowledge domains having influences on these two main domain classes. Actually, WiDOP uses two such top level knowledge classes, however it is possible to extend if required. The interrelation between the knowledge domains completes the knowledge set. This provides the foundation for enabling various knowledge sources to interact and to use it for different purposes, like deciding upon a processing strategy (sequence of algorithms) or upon the nature of elements being found during processing. These inter-relations in knowledge terms are object properties and shown through the arcs in figure 2. They are basically catalysts for knowledge generation, as they act in the knowledge framework and appear in property restrictions or domain rules realized by a rule language like SWRL. Such rules are important, as they allow for the integration of external processing components, called built-ins, which are necessary to do conventional numerical processing. This happens when invoking algorithms or topological operations (Ben Hmida, Cruz, Nicolle, & Boochs, 2011).

Figure 3 illustrates the taxonomical hierarchical structure of the class *Algorithms*. The algorithms are currently classified into three major sub-classes. This could be extended if there are requirements. Most of the algorithms used here belong to one of those three classes.

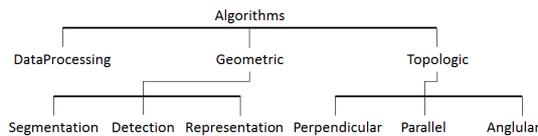


Figure . Taxonomical hierarchical semantics of class *Algorithms*

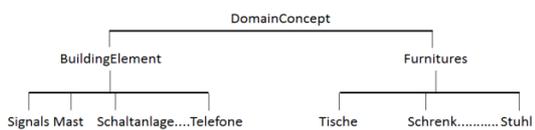


Figure . Taxonomical hierarchical semantics of class *DomainConcept* (scene Deutsche Bahn)

Other classes in the top level ontology schema (figure 2) function as supporting classes.

The objects from the scene are categorized to their respective classes under *DomainConcept* (figure 4). This structure is designed for the objects found in the Deutsche Bahn (“DB”) scene. It can easily be replaced by other domains provided that the top level structure is respected.

They represent knowledge domains defining the characteristics of the main classes. Two top level classes are currently defined, although this is extensible. Class *Geometry* (fig. 5a.) is defined to represent geometric characteristics. Class *Data* (5b) represents data characteristics. Both these classes are used within rules which map algorithms to scene knowledge.

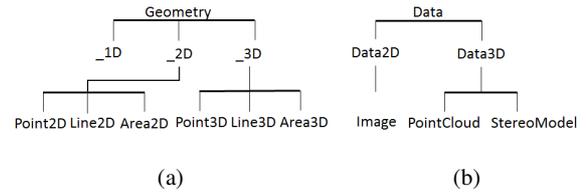


Figure . (a) Class hierarchy of geometry (b) data

2.2.3. The Iteration

ISCM detects and refines the detection process through new gained knowledge at every step of the iteration. In most cases the degree of knowledge is limited initially (barring case 2.1.1).

The knowledge schema provides a basis for formulating the processing strategy, and provides a platform to define inference rules. These rules are based on the expert interpretation of the scene and the algorithmic behavior. The knowledge schema presents the prominent rule defining the scene, with which the ASM uses to begin the algorithmic processing. This prominent rule is inferred against the semantic rules of algorithms to evoke the most suitable algorithm or algorithmic sequence.

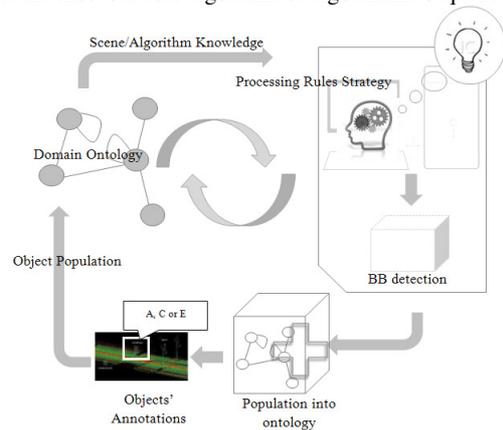


Figure 6. Detailed Iterative Semantic Classification Method

Again using the DB example, first the knowledge schema determines that the objects are vertical in nature, and algorithms suitable for vertical geometry detection are selected. After the algorithmic results are populated into the KB, the detected geometries are qualified to their respective objects. As such, the prominent rule evokes the first set of algorithms best suited to detect the simple and dominant objects, then the more complex objects are detected through their relationship to the simple ones through further iterations.

Qualification follows the population. The detected geometries are inferred against the geometric semantics of the objects to initiate the first set of approximations. The current version of WiDOP provides this approximation through semantic annotation. Labeling mechanisms for 3D point clouds have been researched by Anguelov et al. (2005) and Munoz, Vandapel, & Hebert (2009), who use an associative Markov network to label. Shapovalov, Velizhev, & Barinova, (2010), Golovinskiy, Kim, & Funkhouser (2009), and Shapovalov,

Velizhev, & Barinova (2010) instead label them through capturing geometric classes in context by designing node features. However, these works do not incorporate semantics. Koppula, Anand, Joachims, & Saxena (2011) present a more semantic approach, whereby a graphical model that captures features and their contextual relationships is presented. WiDOP presents semantic rules to semantically annotate the objects (Ben Hmida, Cruz, Nicolle, & Boochs, 2011). These rules are executed through the extended SWRL, and use semantic annotation to match the detected geometries with their probable objects. The following example (equation 1) will detect all the vertical geometries and annotate them as *Mast* if they are higher than 6m.

$$3DProcessing_swrlb:VerticalElementDetection(?vtr, ?dir) \wedge Height(?x, ?ht) \wedge swrlb:greaterThan(?ht, 6) \rightarrow Mast(?vrt) \quad (1)$$

The domain ontology schema now hosts the first impressions of the semantically annotated geometries. At this point the annotations are still rough, and can be one of three types: unambiguous, ambiguous or unknown.

Unambiguous: Geometries annotated to a single object.

Ambiguous: The same geometry can be qualified as two or more objects.

Unknown: Geometries unclassifiable at this level of iteration.

The first iteration is likely to have a large number of ambiguously annotated objects or even unknown objects. The second iteration is needed to improve the result, wherein the KB will now host more semantics. During the second iteration, ASM uses unique characteristics to remove the ambiguity. The mechanism under ASM investigates the rules which are unique to each object in such ambiguity. It then uses these unique rules to infer an algorithmic sequence for each of them. More precise geometries are thus detected during this iterative stage and are populated into the KB. The qualification through extended geometries then repeats (equation 2).

$$BasicSignal(?y) \wedge BoundingBox_3D(?x) \wedge hasHeight(?x, ?h) \wedge swrlb:greatThan(?h, 1) \wedge swrlb:lessThan(?h, 3) \wedge 3D_swrlb_Topo:distance(?x, ?y, 100, 10) \rightarrow SecondarySignal(?x) \quad (2)$$

The iteration continues until all the ambiguity is removed and objects are finally recognized and stable. In case of unknown or ambiguous annotations, new knowledge about the scene or the processing activities is fed into the KB. The first case (section 2.1) resembles the unambiguous annotations. The first level of iteration is therefore unnecessary for this scenario. As objects and their positions are known, the platform executes the iteration from the second step and verifies.

3. IMPLEMENTATION

Figure 7 and 8 illustrate a typical site in the DB railroad system and its 3D scan. The complexity in detecting objects in the point cloud is not only due to the complex nature of the objects but also due to the scan nature. The area is scanned using a moving train; the objects are scanned only in one direction, presenting challenges through occlusions.



Figure 7. A typical site of Deutsche Bahn system (source: Christophe Leimkühler, Metronom Software)



Figure 8. The 3D scan data of the site

It must be emphasized here that we do not use the algorithmic knowledge to estimate what objects should appear in the scene, but rather the opposite. The characteristics of objects in the scene evoke the appropriate algorithms for that object. In this sense, the objects in the class *DomainConcept* are in the center of the top level ontology schema (fig. 2). It hosts the semantics of the objects first through hierarchical taxonomy (fig. 4) and then through semantic rules for each specialized class under the hierarchy. If we examine carefully, *DomainConcept* is a bridge through which other knowledge domains can be explored. For instance, the geometric characteristics of an object under *DomainConcept* are related to the knowledge domain of *Geometry* through the relationship defining it (fig. 2). This can also be extended to other knowledge domains, as we did with data through class *Data*. The top level ontology in figure 2 provides a glimpse of such bridging and is not restricted to it.

3.1. Illustration

This section illustrates how underlying ASM within WiDOP infers rules to derive an algorithmic sequence. We basically will illustrate the principles discussed in section 2 through a case of Deutsche Bahn (DB) with the underlying ASM in focus.

The property restriction rules play a major role in determining the best algorithm. ASM determines this through inferring the rules defined in *DomainConcept* (termed as *DC* in the DL equations) to that defined in the class *Algorithms*. The platform starts with the dominant rule of the scene. We presume the dominance through the number of occurrences of the rules, with the higher the number, the more dominant the rule. An example of this could be the scene of a lecture room where most of the objects have planar surfaces. In such cases the horizontal plane detection algorithm will be preferred as a starting algorithm.

This rule when inferred against specialized algorithms in class *Algorithms* yield that algorithm *HeightApproximation* (presented by HAA in the equation 4) is best suited for this case.

It is because the algorithm constitutes the rule stating it is designed for data having height as shown in equation 3.

$$HAA \equiv \exists isDesignedFor.Data3D \sqcap Data3D.hasHeight.>0 \quad (3)$$

The execution of the algorithm detects prominent geometries of dominant and simple objects in the scene. Qualification follows detections. This is carried out through extended SWRL. Examples can be seen in equation 1 and 2.

As stated, three possible qualifications are possible: unambiguous, ambiguous and unknown. For simplicity we carry forward this discussion with the first case where the objects are qualified distinctly. The WiDOP platform utilizes the semantic rules (defined through property restrictions) to infer the algorithm or algorithmic sequence for verification. We illustrate this with the two prominent objects in the DB scene: *Mast* (poles carrying cables for powering trains) and *Signal*. Figure 9 presents point cloud subsets for them.

The classes of types *DomainConcept* is related to that of *Geometry* through specialized properties of *hasGeometry* (see fig 2).

$$DC \sqsubseteq \exists hasGeometry.Geometry \quad (4)$$

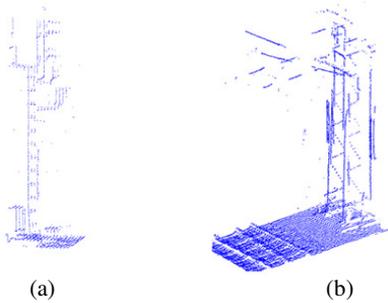


Figure 9. Point cloud sets (a) Signal (b) Mast

The specializations of class *Geometry* are possible geometry types (fig. 5a). For instance *Line3D* is a type of *_3D* which is a specialization of the class *Geometry*.

$$Geometry \sqsubseteq _3D \sqsubseteq Line_3D \quad (5)$$

Putting together equation 4 and 5, we can conclude that *Mast* has *Line_3D*. Furthermore, we can also say *Mast* has *Line_3D* with dense, linearly arranged points (we term them as *thick* lines for simplicity) as shown in equation 6, and *Signal* has *Line_3D* with a low density of points (called *thin* lines for simplicity), as shown in equation 7. Here thick and thin are characteristics of the line. These characteristics (termed *hasChar* in DL equations) are helpful in determining the input parameters for the algorithms. It will be discussed later but for now we present how ASM uses these simple rules in algorithmic selection.

$$Mast \equiv \exists hasLine3D.Line3D \sqcap Line3D.hasChar.\{Thick\} \quad (6)$$

$$Signal \equiv \exists hasLine3D.Line3D \sqcap Line3D.hasChar.\{Thin\} \quad (7)$$

The reasoning engine of the underlying ASM infers the rules against the rules within sub-classes of *Algorithms*. Algorithm *LineDetectionin3DbyRANSAC* is recommended, considering that it is designed for 3D lines.

$$LineDetectionin3DbyRANSAC \equiv \exists isDesignedFor.Line3D \quad (8)$$

The reasoning engine inside ASM implements the same principle to infer algorithms for other knowledge domains. We have implemented it against the data knowledge under class *Data*. Presuming that the standard deviation of a dataset establishes a noise value for that dataset, ASM then infers the algorithms best suited for datasets containing noise. It shows the use of universal knowledge through combining different knowledge domains (related to the scene, to classes of objects, to instruments and so on), allowing the ASM to interact with them. This interaction helps in providing answers for detecting objects in extreme situations. However, it is necessary to define appropriate rules to determine the usage of these knowledge domains. Likewise, the underlying knowledge schema (fig. 2) provides freedom in choosing its data source. We use a 3D point cloud from the DB scene for our case, however it is also possible to use images or other data formats.

3.2. Simulation Knowledge

Algorithms behave differently in different situations, for instance reflecting differences in geometry or data. Even two characteristics of the same geometry might need to be addressed in the detection algorithm. As shown in Equation 6 and 7, the ASM chooses *LineDetection3DbyRANSAC* for detecting the geometries, but using the same parameters for both cases might not yield best results. In principle, it should use different parameters for different point densities of the linear structure to capture most of the points within the linear structure (fig. 10). We thus need a higher radius value for thick lines and a lower radius value for the thin.

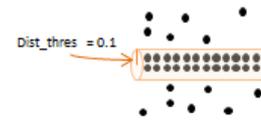


Figure 10. Cylinder radius for detection

This is exactly the intention behind obtaining and modeling the simulation knowledge into the KB. Each observation of the execution of an individual algorithm is induced in the KB. These simulations are based on the results as they are tested against different data, geometries, and other characteristics. The clear benefit is that in above given situations for *Mast* and *Signal*, the ASM of the WiDOP platform selects a different radius threshold for the *LineDetectionin3DbyRANSAC* algorithm. Furthermore, the ASM can evaluate the rules defined by the scene to select different algorithms for different cases. Instead of *LineDetectionin3DbyRANSAC*, ASM recommends *2DHoughTransformation* for example, if the detection process uses images or any 2D data as source data.

3.3. The Result

Our approach was tested with a 500 m long 3D point cloud of the Nuremberg main station (Nürnberg Hbf). The KB consists

of the objects found in the scene along with the algorithms that could possibly be used to detect them.

Table 1 presents the detection and qualification of objects in the scene through *ISCM* within the *WiDOP* platform. There were 105 geometries detected and among them 34 were semantically annotated. 71 detected geometries are not annotated because the KB did not contain enough rules to classify them. Although currently the results are based on unsophisticated data and rule sets, we believe that those objects would be correctly annotated after further enrichment of the KB. Mismatches visible from Table 1 show the necessity of improvements by addition of modification of rules within the KB. However, results already show the general functioning of such a flexible approach.

Objects	Annotated	Reality
Masts	13	12
Signal	18	20
SwitchGear	3	5

Table 1. The first set of results

The result seen in table 1 through the knowledge driven method is satisfactory considering the complexity of the scene. Rule modifications would improve results, and further development in this regard is ongoing.

4. CONCLUSIONS

The knowledge driven approach for selecting algorithms suitable to detect objects has been presented. Building on previous research, technologies within the Semantic Web have been advanced, and SWRL in particular has been extended in the qualification process. Keeping the essence of the knowledge based approach in processing, this solution uses a methodology that fuses knowledge from different knowledge domains for suitable algorithmic selection, which leads to a flexible processing chain for detecting objects. Furthermore, the integration of simulation knowledge, representing behavioral knowledge of algorithms in different situations and patterns, adds more flexibility.

5. ACKNOWLEDGEMENTS

This paper presents work performed in the framework of a research project funded by the German ministry of research and education under contract No. 1758X09. We also acknowledge the contributions from Helmi Ben Hmida, Hung Quoc Truong, Christophe Cruz and Adlane Habed.

References

Anguelov, D., Taskarf, B., Chatalbasev, V., Koller, D., Gupta, D., & Ng, A., 2005. Discriminative learning of Markov random fields for segmentation of 3D scan data. In: *Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society*, San Diego, USA, IEEEExplore, pp. 169 -176.

Bechhofer, S., Harmelen, F. v., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., & Stein, L. A., 2004. „OWL Web Ontology Language”, from W3C Recommendation: <http://www.w3.org/TR/owl-ref/> (28 Nov 2010)

Beker, S., 2009. Generation and application of rules for quality dependent facade reconstruction. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, pp. 640 - 653.

Ben Hmida, H., Cruz, C., Nicolle, C., & Boochs, F., 2011. Toward the Automatic Generation of a Semantic VRML Model from Unorganized 3D Point Clouds. In: *The Fifth International Conference on Advances in Semantic Processing*. Lisbon Portugal.

Berners-Lee, T., 1998. “Why RDF model is different from the XML model.”, from W3C: A roadmap to the Semantic Web. <http://www.w3.org/DesignIssues/Overview.html> (22 Oct 2010).

Frueh, C., Jain, S., & Zakhor, A., 2002. Data processing algorithms for generating textured 3D building facade meshes from laser scans and camera images. *Interantion Journal for Computer Vision*, pp. 834 - 847.

Golovinskiy, A., Kim, G. V., & Funkhouser, T., 2009. Shape-based Recognition of 3D Point Clouds in Urban Environment. In: *International Conference on Computer Vision (ICCV)*, Kyoto, Japan IEEEExplore, pp. 2154 - 2161.

Horrocks, I., Schneider, P. f., Boley, H., Tabelt, S., Grosf, B., & Dean, M., 2004. “SWRL - A Semantic Web Rule Language - Combining OWL and RuleML”, SWRL: A Semantic Web Rule Language - Combining OWL and RuleML. <http://www.w3.org/Submission/SWRL/> (22 May 2009)

Koppula, S. H., Anand, A., Joachims, T., & Saxena, A., 2011. Semantic Labeling of 3D Point Clouds for Indoor Scenes. In: *Twenty-Fifth Annual Conference on Neural Information Processing Systems*. Granada, Spain.

Maillot, E, N., Thonnat M., 2008. Ontology based complex object recognition. In. *Image and Vision Computing*. Vol. 26 Issue 1, Butterworth-Heinmann Newton, MA, pp. 102 - 113

Munoz, D., Vandapel, N., & Hebert, M., 2009. Onboard contextual classification of 3-d point clouds with learned high-order Markov random fields. In: *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, IEEE Press, Kobe, Japan , pp. 4273 - 4280.

Patel-Schneider, P. F., Hayes, P., & Horrocks, I., 2004. “OWL Web Ontology Language”, W3C Recommendation: <http://www.w3.org/TR/owl-semantics/> (12 July 2010)

Pu, S., 2010. *Knowledge based building facade reconstruction from laser point clouds and images*. Twente: Optima Grafische Communicatie B.V.

Ripperda, N., 2008. Grammer based facade reconstruction using RjMCMC. In: *Photogrammetrie Fernerkundung Geoinformation*.

Shapovalov, R., Velizhev, A., & Barinova, O., 2010. Non-associative markov networks for 3d point cloud Classification. In: *Photogrammetric Computer Vision and Image Analysis - ISPRS Commission III symposium*, Paris, France, PCV 2010, pp. 103 -108.

W3C. (1995, April 17). *VRML Virtual Reality Modeling Language*. Retrieved December 18, 2011, from <http://www.w3.org/MarkUp/VRML/>

Wessel, R., Wahl, R., Klein, R., & Schnabel, R. (2008). Shape recognition in 3D point clouds. *The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. Plzen - Bory: UNION Agency-Science Press.