Semantic Interpretation of 3D Point Clouds of Historical Objects

Paul Cotofrei, Christophe Künzi, and Kilian Stoffel

Information Management Institute, University of Neuchâtel, Switzerland
{paul.cotofrei, christophe.kuenzi, kilian.stoffel}@unine.ch

Abstract. This paper presents the main concepts of a project under development concerning the analysis process of a scene containing a large number of objects, represented as unstructured point clouds. To achieve what we called the "optimal scene interpretation" (the shortest scene description satisfying the MDL principle) we follow an approach for managing 3-D objects based on a semantic framework based on ontologies for adding and sharing conceptual knowledge about spatial objects.

Keywords: ontology, scene interpretation, 3D point cloud

Introduction

During the last couple of years, point clouds have emerged as a new standard for the representation of largely detailed models. This is partly due to the fact that range scanning devices are becoming a fast and economical way to capture dense point clouds. These devices typically produce an unstructured cloud of sample points (possibly with noise), where each point encodes information on the shape attributes, such as 3D position, surface normal, surface color, material properties, etc. However, the huge amount of data captured during the acquisition phase may limit the applicability of the algorithms and methodologies currently developed for 3D computer vision. At the same time, it is obvious that the process of extracting useful knowledge or models from unstructured information spaces (and a point cloud is such a space) is a topic situated at the junction of several research fields, as spatial data mining, computer graphics, pattern recognition, machine learning, spatial reasoning, data bases/data warehousing, etc. Applications based on the manipulation and the analysis of such points are extensively used in many disciplines, such as mechanical engineering, architecture, bio-medicine, robotics, but also in other domains such as history and archeology.

The goal of this paper is to detail a framework developed to represent, to recognize and to retrieve objects in a point cloud, based on the generic idea published in the paper [1]. The database used to check the validity of our approach is a collection of multi-dimensional points having several characteristics along with the spatial location. This data stems from a large project conducted by the Kármán Center [2], for which the Pantheon in Rome was scanned (the result is a 3D digital model with more than 620,000,000 points, see **Error! Reference source not found.**). Therefore, if the performance of the final system is proved to be satisfactory, the application will be integrated into the Pantheon project at the disposal of the archaeologists and historians.

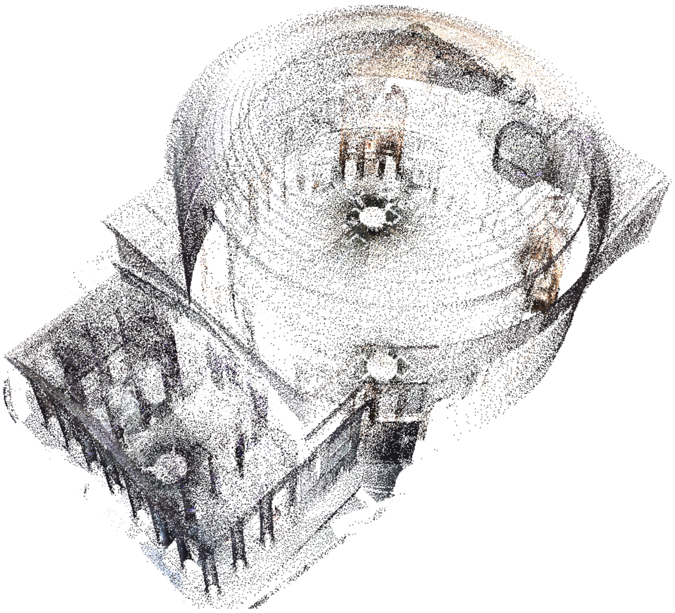

Fig. 1. 3D point cloud of the Pantheon in Rome

One of the main issues with this type of datasets (aside the basic data management) is the *interpretation of a scene*. The interpretation of a scene (query data consisting of partial 3D point clouds of (un)known 3D objects) is normally defined as knowing *which* model is located *where* in the scene. Such an interpretation binds the entities in the scene to the models which exist in form of prior knowledge. Scene interpretation represents one of the most known concepts which emphasize the evolution over time of the research relation between Computer Vision (CV) and Knowledge Representation (KR). If, in the early days, research in CV often involved development of semantic representations and inference mechanisms [3], the difficulties of this approach (scene interpretation tending to be highly unreliable) pushed the research toward a quantitative approach, based on statistical methods. In the lasts years, these methods apparently proved their limits and Computer Vision became again one of the goal of Artificial Intelligence ([4], [5]), especially when data are incomplete or ambiguous. There is no universally accepted definition of what scene interpretation means, especially as other similar concepts are used in the literature (as scene understanding or scene modelling), but at least one aspect may be considered as acquired: the process implies a knowledge representation framework. Reiter and Mackworth [6] considered the interpretation as an instantiation of a conceptual knowledge base consistent with evidence, i.e. with information about the scene delivered by sensors and low-level image analysis. Because a scene interpretation may contain arbitrary propositions (for example about objects outside the field of view), further criteria are required to narrow down the interpretation space and select a "best" interpretation [7]. In [8] scene interpretation is modelled as a stepwise process which exploits taxonomical and compositional relations between aggregate concepts (represented in a ALCF(D) Description Logic), while incorporating visual evidence and contextual information. The same approach, but using an automatic process, is described in [9]. Finally, different scene interpretation systems were developed, as RACER [10] (based on a particular DL system) or SCENIC [11] (based on a configuration system), which provides powerful retrieval mechanisms along with other inference processes.

Following the recent trend consisting in applying the AI point of view on Computer Vision problems and tasks, we propose an extended definition of the "interpretation" task (closed to what was called "high-level scene interpretation" [7]): the task consists in the construction of a symbolic description including scene elements (objects or higher-level entities) and predicates (class memberships and spatial relationships between these elements). This extension allows the acquisition of a new kind of knowledge, concerning the possible repeated, regular patterns of objects spatial distributions. Furthermore, if the set of models and the set of spatial relationships are the elements of a spatial description language, then the concept of *optimal scene interpretation* is well defined, expressing the shortest description (in this language) of the scene in terms of known objects and simplest neighborhood relations between them.

Therefore, the goal of our project is the establishment of a flexible approach (including a framework, a methodology, processing methods and finally a working system) allowing the optimal interpretation of a scene (according to our extended definition), containing a large number of objects. To reduce the complexity of the interpretation process in the perspective of the large diversity of real-world situations, the project's framework is based on the following assumptions:

- the scene or the model point clouds are not uniformly sampled nor overlapping;
- the objects of interest are rigid, free-form objects;
- the models exists in the database prior to recognition;
- a description language, based on the models from database and a selected set of spatial relationships, is defined and encoded as a set of fixed ontologies

Stated succinctly, the design of the scene interpretation system (in the following denoted RRR system) involves a three stage processing:

1. *Representation*: The objective is to derive from the point cloud a rich, compact yet meaningful description of the object for efficient storage and for fast and accurate retrieval during recognition.
2. *Recognition*: The derived spatial and geometric descriptions of the partial point cloud from the scene are compared with stored models of objects in order to identify which of those objects are present in the scene. This involves the tasks of

instance classification, determination of alignment parameters (rotation, translation) and localization.
3. *Retrieval*: The spatial relationships existing between the objects in the scene are discovered. In a first phase, these relations are analyzed by a pattern finding algorithm to extract possible regular patterns implying the models. In a second phase, a specific ontology describing the scene (which includes as instances the previous discovered relations), is processed to extract an optimal (according to the Minimum Description Length Principle) scene description.

The first two processing stages of the RRR system (described in the Section 2) are well documented in the literature (methodologies and algorithms) and therefore, during this project, we are only conducting performance-comparison studies in order to select the best solution regarding the data type we analyze (point clouds). On the other hand, the Retrieval phase (detailed in Section 3) represents - in our opinion, after a deep as far as possible bibliography study – an innovative idea which is directly linked to a new approach in computer graphics, the use of techniques and methodologies from Artificial Intelligence and Knowledge Management for scene understanding (see [12], [13], [14]). The Reference and the User ontologies (described in Section 4) are designed to support the semantic integration of architectural structures (an example of how the Corinthian column concept is represented as an ontological definition by a composition of Basic3DShapes objects is presented in detail).

The RRR Model (Representation, Recognition, Retrieval)

2.1 The Representation Stage

Despite the different application contexts of free-form object models, some criteria apply to representations regardless of the domain. According to Brown [15], the general mathematical properties exhibited by object representation schemes are *ambiguity*, *conciseness* and *uniqueness*. The ambiguity or completeness measures the representation's ability to completely define the object in the model space, the conciseness represents how efficiently or compactly the description defines the object, whereas the uniqueness is used to measure if there is more than one way to represent the same object, given the construction methods of representation.

The choice of the object representation is one of the most important decisions for the performance of our RRR system, and must be accompanied by robust techniques for extracting compatible features from both the object model and the input point cloud. Between the two fundamental categories of representation, object-centered and view-centered, the nature of input data and the objective of our project clearly impose techniques from the first category, which attempt to describe the entire 3D volume occupied by the object.

As we already mentioned, the choice of the object model is based on a performance-comparison study of the known various 3D object representations (see [16]) (boundary-based methods [17], volumetric descriptors [18] or spherical representations based on generalized cones [19]) with a particular attention to the capacity to deal with missing data (undersampling of the surface), with noisy data and with the lacking of connectivity information (unstructured point cloud). Our implicit option is the polygonal mesh representation, especially adapted for point cloud in [20] [21].

2.2 The Recognition Stage

Whereas the problem of finding and identifying objects in single-object scenes with no occlusion has been well studied and many systems designed show good results [22], the same problem, but for multiple objects with the possibility of occlusion and background clutter is much harder. Recognition is performed by matching features derived from the scene with those stored in the model database. Some of the most popular and important approaches to the recognition and localization of 3D objects are:

- *Graph matching*. This approach captures the structural properties of objects. The scene and the model are described using attributed graphs or shock graphs, where each vertex characterizes a scene or a model feature and the edge between vertices represents the relation between two features.
- *Information-theoretic matching*. To align the scene image with the model image, a first proposal was the search of the transformation which maximizes the mutual information between the scene and the model. Another proposal was the entropic methods (through an entropic graph), which have the advantage of capturing non-linear relations between the features in order to improve the discrimination power.
- *Hypothesize and test*. In the hypothesize and test paradigm, a transformation (a set of non-linear equations) from the model's coordinate frame to the scene's coordinate frame is hypothesized. The alignment of the scene features with the model features is accepted or rejected based on matching error.
- *Iterative model fitting*. This approach is used when 3D objects are described using parametric representations. There is no feature computation or correspondence determination between model and scene features. Object recognition and pose estimation are reduced to estimating the orientation parameters of the model from the scene data, and matching with the stored parametric representations.

Given the nature of data, our choice for the recognition process points to the matching algorithm proposed in [23], a thermo-dynamically inspired algorithm designed to determine a correspondence between the scene and the model point clouds by combining the goodness of the graph-based structural approaches and the entropy-based spatial matching approaches. The maximization of the proposed objective function which captures the structural and spatial differences between point sets, leads to the desired correspondence.

The Retrieval Stage

A real useful and valuable functionality of an *intelligent* computer vision system would be its capacity to describe an unknown scene as concisely as possible in terms of known objects, transformations of them, and of their mutual spatial relationships. Therefore, we extend the meaning of the scene interpretation process by considering that an *optimal* interpretation of a scene is a description (based on a specific spatial language) which explains the scene in terms of the smallest number of known objects (i.e. known models) and simplest neighborhood relations between them, according to the Minimum Description Length Principle [24]. A simple description language allowing the scene interpretation must include at least rigid, opaque 3D objects, and a set of spatial relationships. From a technical viewpoint, our approach is to "encode" such description language inside a dynamically created semantic layer, added to our 3D point cloud, and expressed as a set of ontologies (in the following denoted as the *reference ontology*) comprising the description of different systems and representation models that might be used.

A reasoning engine processes the low-level knowledge structures captured in the reference ontology. The goal of this reasoning is to deduce new, high-level knowledge and to signal inconsistencies in the conceptualizations. Two main approaches can be applied: using general logic based inference engines or using specialized algorithms (Problem Solving Methods). The logic based inference engines may be classified1 by the expressivity of the logic they can reason with, from Higher Order Logics (HOL) down to different subsets of First Order Logic (including fuzzy or probabilistic approaches). In general, more expressive logics are more difficult to reason with, where in the worst case scenario there exist no strategies that could ensure the termination of the reasoning process. Concerning the second approach, each PSM represents a declarative, reusable description of reasoning for solving a particular type of problem.

Based on the information learned during the recognition stage (objects instances found in the scene and their exact localization), the system send queries to the reasoning engine concerning the possible binary spatial relationships between the found instances. The set of queries evaluated as true (positive queries) together with the set of object instances in the scene form then the raw data from which the optimal scene interpretation is generated. According to the MDL Principle, the optimal description minimizes the length of the set {theory, data encoded using the theory}. Consequently, to different types of theories corresponds different optimal descriptions, even if all are based on the same data. In our opinion, two types of theories must be considered as appropriate:

- *Rules*. Significant rules between the classes (models) of the object's instances are extracted using the spatial association rules approach [25]. Another approach, derived from similar applications on natural language [26], generates the grammatical rules underlying the language described by the set of positive queries (considered as correct sentences) using a semi-supervised learning algorithm. The

---

[1] http://www.semanticweb.org/inference.html

optimal description minimizes the length of rules, together with the exceptions to these rules.

- *Ontology*. As one of the form expressing the paradigm of concept formation (together with clustering or Concept Lattices), an ontology is a conceptual structure used for concisely characterizing data. If we define the length of an ontology as the number of nodes plus the number of links of type *is-a* or *has-a*, then an algorithm for creating an ontology, starting from the set of positive queries (statements) and using MDL as guiding principle, can be designed [27]. The algorithm must iteratively search analogies or isomorphism in *contiguous* sets of statements, where two statements are connected if they share a common symbol (instance or spatial relation), and a set of statements is contiguous if there's a path within the statement set from every statement in it to every other statement in the set.

Each type of optimal description (based on rules or on ontology) has its own advantages/drawbacks regarding the comprehensibility of the final result (an ontology is more adapted for a visual representation than a set of rules), and therefore must be chosen according to the interest of the final user of the RRR system.

As we already mentioned, the description language is encoded as a semantic layer over the raw data recorded as unstructured clouds of sample points. For architectural data (the Pantheon project) we defined [28] a system based on two components - an efficient storage module for 3D data and a concept-based representation module. The second module (detailed in the next section) is in fact the reference ontology designed to support the semantic integration of architectural structures (often based on very complicated models, taking into consideration technical and aesthetic aspects).

The Reference and User Ontology

4.1 The Reference Ontology

The semantic layer is mainly composed of two groups of ontologies: the upper (basic) group and the lower (user) group. This difference between the used ontologies has been shown already in [29]. Without losing in generalization, the user can describe a spatial object in a specific environment by actually constructing the 3D object from elementary shapes. Each elementary shape is described mainly by a transformation (scaling, translation, rotation), one or more positions and one or more dimensions. Since each transformation can be expressed in different ways or is shape–dependent, the upper ontologies comprise the description of different systems and mathematical models that might be used [28]:

1. The *Coordinate Systems Ontology*. This ontology defines a few systems for describing a position in space: cartesian, spherical and cylindrical — the ontology being easily extensible with other systems. Each coordinate system has properties that map its specific characteristics: e.g. the *CartesianSystem* has three length–based properties (corresponding to the x–, y– and z–coordinates), while the

CylindricalSystem has two metric–like properties and a degree–like property that correspond to the radial, vertical and azimuth values, respectively.

2. The *Transformation Systems Ontology*. The same approach has been used for the transformations ontology, i.e. each instantiable rotation system has predefined attributes (e.g. roll angle, vector, etc.) that match their corresponding mathematical elements. For example, the *EulerAxisRotation* defines properties for rotation vector and angle, while the *TaitBryanRotation* has a degree–like property for each dimension. Both coordinate systems and transformation systems ontologies have been designed bottom–up [30], the superclasses being constructed as the union of their subclasses (see Fig. 2). By this approach, we force the notion of abstract classes (we cannot have instances of *CoordinateSystem*, it has to be an instance of *CartesianSystem* or *SphericSystem*) without losing the type-of relation and the inheritance mechanism between concepts. Furthermore, the cardinality constraints defined on the properties (such as radial, coordinates, etc.) makes those properties mandatory.

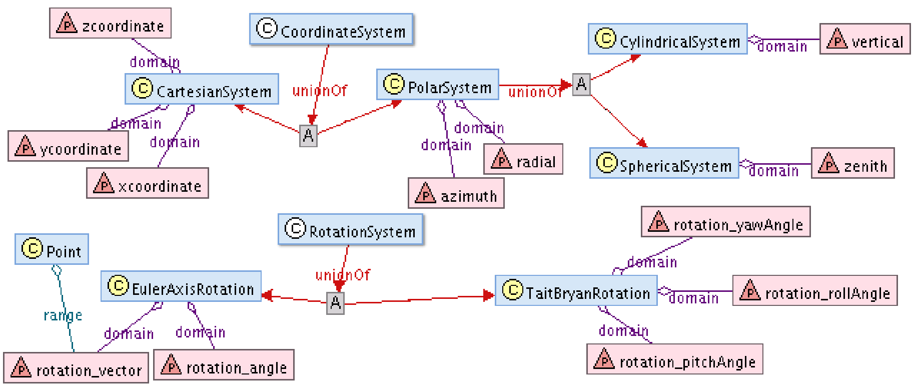

Fig. 2. Excerpt from the Coordinate Systems and Transformation Systems Ontologies

3. The *Geometrical Shapes Ontology*. Inspired by [31], the shapes ontology is the most complex one and it formalizes the fundamental geometrical shapes such as cuboids, sphere, etc. Its dependency with the previously described ontologies gives it more flexibility in the positioning and transformation of the spatial shapes. The central concept of this ontology is the *SpatialObjet*, all basic shapes as well as any user–defined spatial object being subclasses or instances of the *SpatialObjet* concept. In the spatial ontology, each shape is described mainly by a transformation (e.g. *rotatedBy*), a position (*hasPosition*) and by its dimensions (*hasDimensions*), or it can be identified by one or more points of reference (*definedBy*). As can be seen in Fig. 3, the spatial relationships between objects are expressed as a hierarchy of properties (*RelativePositioning*), but it's also possible to be defined as a distinct ontology.

For all of the upper ontologies, the system considers that two parameters are implicit: the distance unit expressed in meters and the degree unit in radians.

The topological and compositional properties defined on *SpatialObjet*'s let the user to construct iteratively more complex *SpatialObjet*'s. When he starts working with the initial system, the user can essentially use *Basic3dShape* and its associated basic operations to define his queries that correspond to its basic objects. By composing these simple *Basic3dShape* objects, the user can describe new, more complicated shapes. These new spatial objects have to be defined in the *User Ontology*. We will illustrate now an example of how a user might proceed to define its own objects and make them available as an ontological definition.

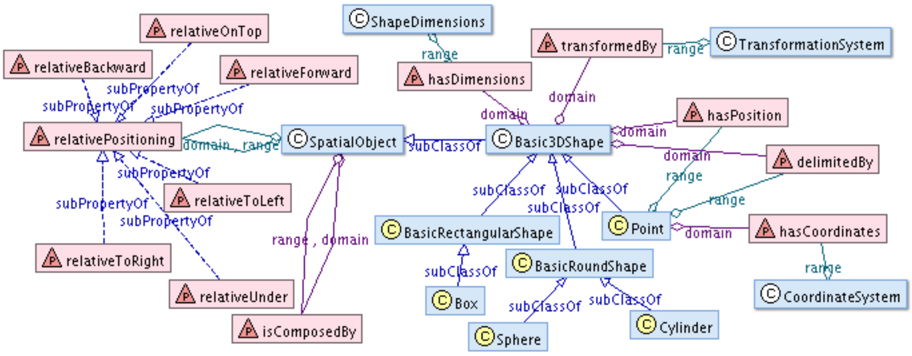

Fig. 3. Excerpt form from the "Geometrical Shapes" Ontology.

4.2 The User Ontology

To illustrate the user's point of view, let's imagine for this example that he would like to find Corinthian columns in the Pantheon data. By looking at the image of the entrance (see Fig. 4), one user may try to retrieve the points defining a column using the basic definition of a *box*, another user may prefer to retrieve the same points using a *cylinder*, whereas a third user may realize that a combination of the two previous approaches might be more appropriate. This mainly depends on the specific point of view. An architect using the data might have a completely different approach than a historian.

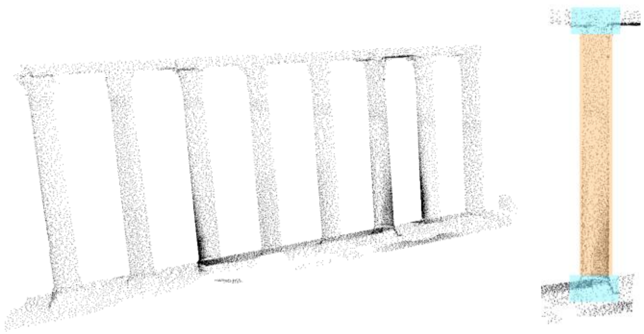

Fig. 4. The entrance of the Pantheon in Rome and abstract representation of a Corinthian Column

These new spatial objects have to be defined in the ontology, more precisely in the part reserved for user definitions. To do this he can use the Ontology Web Language (OWL) or the tools provided by the system. A user can define new 3D objects or redefine existing objects (e.g. by changing the coordinate system).

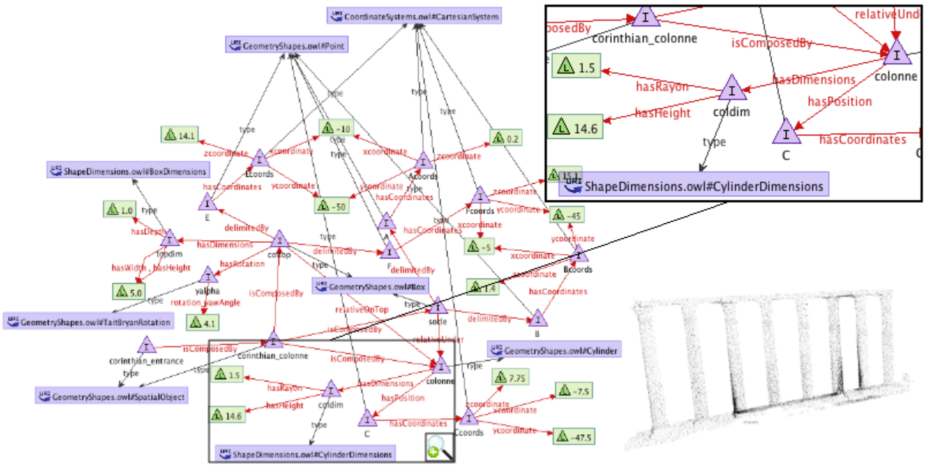

Fig. 5. Extended User Ontology for *CorinthianColumn*

We will present now an example of how a user might proceed to define its own objects and make them available as an ontological definition. Let’s say he would use a box for the base element, then a cylinder for the middle part of the column, and another box for the top of the column, all of them being combined to define a Corinthian column. For these types of complex shapes, a new concept can be added to the user ontology, named *CorinthianColumn* (see Fig. 5). Furthermore, another concept *CorinthianEntrance* can be defined as composed of *CorinthianColumn*’s. The *Basic3dShape*’s used to define the new concepts have precise coordinates and ontological descriptions

Based on the extended ontology another user could add his own concepts and make them dependent on the newly introduced concepts of the *CorinthianColumn*. As known from the history of architecture, Corinthian columns might consist of identical base and middle element, but they could differ in their top element. The ontological definitions should also allow this refinement of the basic definitions of a *CorinthianColumn*.

Once the user ontology is completely populated, the Retrieval phase may be applied in order to obtain the optimal scene interpretation. The system’s functionalities for this phase are still in the implementation phase, but we may illustrate what it could be considered as an optimal description: for the scene representing the entrance, this could be the following set:

{ RULES:
R1: IF *instance*$_1$ of CorinthianColumn THEN *instance*$_2$ of CorinthianColumn *relativeToLeft atDistance d*
DATA:
instanceA # instance of the last column at right, satisfying the rule R1
instanceB # instance of the first column at left, exception for rule R1
}

The retrieval algorithm designed to search a specific object in a spatial scene receives as input an instance of a concept from the user ontology (usually a composition of basic shapes, defined by some properties and related by some spatial relations). The main goal is to navigate from a conceptual scene representation to a deterministic scene composed by points. Because the cloud points contain often a huge number of points, the complexity of operations executed by shape extraction algorithms (subroutines of retrieval algorithm) represents a very important issue. From this viewpoint, the ontology layer is facilitating the search because it allows focusing only on some kind of primitives.

The Reference ontology mainly specifies pure mathematical definitions of the shapes as well as their spatial relations. In certain cases, the interpretation of the scene cannot be done without adding a human (i.e. uncertain and vague) perspective of the scene (as example, for searching a *small* sphere *nearby* a *big* box, where the terms {*small, nearby, big*} might have a specific meaning only for the scene under consideration). Therefore, the user must have the possibility to extend the reference ontology with a set of relations or properties inspired from linguistic variables defined by the fuzzy logic theory. This linguistic variables are treated as relations and can be integrated in exactly the same was as all other spatial relations.

The details of the implementation of the retrieval algorithm based on this extended ontology are beyond the scope of this paper and are presented in [22], [23].

Conclusions

Following a novel direction in computer vision during the last years - the use of techniques from Artificial Intelligence and Knowledge Management for scene understanding – we started the development of a complex project designed to generate an optimal scene interpretation starting from 3D unstructured point clouds. The novelty of our approach is given, in our opinion, by the definition of the concept *optimal interpretation*, seen as the shortest description (according to the MDL principle) of a scene, expressed in a spatial language encoded as a dynamically created semantic layer, implying 3D object instances and spatial relationships. For architectural data (the Pantheon Project) we designed this layer as a semantic framework for adding and sharing conceptual knowledge about spatial objects, by starting from a reference ontology that describes the basics of the spatial aspects. The description also includes extension mechanisms allowing adapting the basic reference systems to specific user needs. Finally we were able to outline the basic structure that would allow querying in a simple way point clouds especially well adapted for presenting the data as well as the query results on the WWW.

References

1. Cotofrei, P., Künzi, C. Stoffel, K.: Optimal Scene Interpretation: Semantic Management of 3-D Object from a Point Cloud Scene. In: International Conference of Computer Graphics and Artificial Intelligence, pp. 189–196. Athens (2009)
2. The Institute of Advanced Study in the Humanities and the Social Science. http://www.iash.unibe.ch/content/archiv/krmn_center/index_ger.html. [Online]
3. The Psychology of Computer Vision. P Winston (eds). McGraw-Hill (1975)
4. Bennett, B.: Combining logic and probability in tracking and scene interpretation. In: Logic and Probability for Scene Interpretation. Dagstuhl Seminar Proceedings (2008)
5. Plemenos, D.: Using artificial intelligence techniques in computer graphics. In: International Conference Graphicon, (1999)
6. Reiter, R., Mackworth, A.: The logic of depiction. Technical report TR 87-23, Dept. Computer Science, Univ. of British Columbia. Vancouver (1987)
7. Neumann, B., Weiss, T.: Navigating through logic-based scene models for high-level scene interpretations. In: 3rd Int. Conf. on Computer Vision Systems (ICVS-2003), pp. 212–222, (2003)
8. Neumann, B., Moller, R.: On scene interpretation with description logics. In: Cognitive Vision Systems: Sampling the Spectrum of Approaches, vol. 3948, pp. 247–278, (2006)
9. Pangercic, D., Tavcar, R., Tenorth, M., Beetz, M.: Visual scene detection and interpretation using encyclopedic knowledge and formal description logic. In: Int. Conf. on Advanced Robotics, (2009)
10. Haarslev, V., Moller, R.: Racer system description. In: Int. Joint Conf. on Automated Reasoning (IJCAR 2001), pp. 701–705, (2001)
11. Hotz, L., Neumann, B.: Scene interpretation as a configuration task. In: Kunstliche Intelligenz, vol 19(3), pp. 59–73, (2005)
12. Plemenos, D.: Using Artificial Intelligence Techniques in Computer Graphics. In: International Conference Graphicon. (1999)
13. Golfinopoulos, V., Miaoulis, G., Plemenos, D.: A semantic approach for understanding and manipulating scenes. In: Int. Conf. 3IA. Limoges (2005)
14. Amitha Perera, A. G. et al.: Moving Object Segmentation using Scene Understanding. In: Conf. on Computer Vision and Pattern Recognition. (2006)
15. Brown, C. M.: Some mathematical and representational aspects of solid modeling. In: IEEE Trans Pattern Anal. Mach. Intell, vol. 3, pp. 444–453, (1981)
16. Campbell, J. R., Flynn, P. J.: A survey of free-form objects representation and recognition techniques. In: Computer Vision and Image Understanding, vol. 81, pp. 166–210. (2001)
17. Besl, P.J.: Surfaces in Range Image Understanding. Springer-Verlag (1988)
18. Suk, M., Bhandarkar, S.M.: Three Dimensional Object Recognition from Range Images. Springer-Verlag (1992)
19. Ikeuchi, K., Hebert, M.: Spherical Representations: From EGI to SAI. In: Object Representation in Computer Vision, pp. 327–345. (1992)
20. Oblonsek, C., Guid, N.: A fast surface-based procedure from object reconstruction from 3D. In: Computer Vision Image Understanding, vol. 69, pp. 185–195. (1998)

21.Jagannathan, A., Miller, E.L.: 3D Surface Mesh Segmentation using a Curvedness-Based Region Growing Approach. In: IEEE Trans. on Pattern Annals and Machine Int., vol. 29(12), pp. 2195–2204. (2007)
22.Murase, H., Nayar, S. K.: Visual learning and recognition of 3-D objects from appearance. In: Int. Journal of Computer Vision, vol. 14, pp. 5–24. (1995)
23.Jagannathan, A., Miller, E.L.: Unstructured 3D Point Cloud Matching within Graph-theoretic and Thermodynamic Frameworks. In: Computer Vision and Pattern Recognition. (2005)
24.Rissanen, J.: A universal prior for integer and estimation by minimum description length. In: Annals of Statistics, vol. 11, pp. 416–431. (1982)
25.Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In: Fourth Int. Symp. on Large Spatial Databases, pp. 47–66. (1995)
26.Petasis, G., et al.: Learning context-free grammars to extract relations from text. In: ECAI, pp. 303–307. (2008)
27.Pickett, M., Oates, T.: The Cruncher: Automatic concept formation using minimum description length. In: Sixth Int. Symposium SARA. Springer-Verlag (2005)
28.Ciorascu, C., Künzi, C., Stoffel, K.: Overview: Semantic Management of 3D Objects. In: SAMT Conference, S-3D Workshop. (2008)
29.Grenon, P., Smith, B.: SNAP and SPAN: Towards Dynamic Spatial Ontology. In: Spatial Cognition and Computation, vol. 4(1), pp. 69–104. (2004)
30. Ciorascu, C., Künzi, C., Stoffel, K.: Collaborative design of spatial ontologies. Technical report, Information management institute, University of Neuchâtel. (2008)
31.Perry, M., Hakimpour, F., Sheth, A.: Analyzing Theme, Space and Time: An Ontology-based Approach. In: 14th Int. Symposium AGIS, pp. 147–154. (2006)
32. IMI-Pantheon Site: http://imi.unine.ch/pantheon/
33.Künzi, C., Stoffel, K.: Ontology-driven shape detection in 3D point clouds. Technical Report, Information management institute, University of Neuchâtel. (2011)