A Framework for Mesh Segmentation and Annotation using Ontologies

Thomas Dietenbeck [a,b,c,d,e], Ahlem Othmani[a,b], Marco Attene[f], Jean-Marie Favreau[a,b]

[a]Clermont Université, Université d'Auvergne, ISIT, BP10448, F-63000 Clermont-Ferrand
[b]CNRS, UMR6284, BP10448, F-63000 Clermont-Ferrand
[c]Sorbonne Universités, UPMC Univ Paris 06, UMR 7371, UMR_S 1146, LIB, F-75013, Paris, France
[d]CNRS, UMR 7371, LIB, F-75013, Paris, France
[e]INSERM, UMR_S 1146, LIB, F-75013, Paris, France
[f]CNR-IMATI, Italy

Abstract. Mesh segmentation and annotation using semantics has received an increased interest with the recent democratisation of 3D reconstruction methods. The common approach is to perform this task in two steps, by first segmenting the mesh and then annotating it. However, this approach does not allow one part to take advantage of the other. In image processing, some methods are combining segmentation and annotation, but they are not generic and require implementation adjustments or rewritings for each modification of the expert knowledge. In this work, we describe an original framework that mixes segmentation and annotation while minimizing the required geometric analysis and we give preliminary results showing its feasability.

Our framework provides a generic ontology describing object feature concepts (geometry, topology, etc.) and algorithms allowing to detect these concepts. This ontology can be enlarged by any expert to formally describe a specific domain. The formalized domain description is then used to automatically perform the joint segmentation and annotation of objects and their features, by selecting at each step the most relevant algorithm given the previously detected semantics. This methodology has several advantages. Firsly it allows to segment and annotate objects without any knowledge in mesh or image processing by simply describing the object features in terms of ontological concepts. Secondly this framework can be easily reused and applied to different contexts by simply building on our generic ontology. Finally performing the joint segmentation and annotation allows to use in an efficient way the expert knowledge, reducing possible segmentation errors and the computation time by always launching the most efficient algorithm.

1 Introduction

During the last two decades, an important work has been done in the data mining and mesh processing communities to integrate a semantic dimension to their work. One of the main goal is to be able to extract an abstract description of the manipulated data, using some semantic descriptors. Bridging the gap between raw data and semantic concepts is a very complicated task, usually implying a good knowledge about the specific applicative domain the systems are working on. This link between the expert knowledge (*i.e.* the semantics) and the raw data can be achieved using learning techniques or by designing a deterministic system, expressing the knowledge of the expert in a language of computer science.

Techniques for semantic extraction have been explored more precisely for mesh segmentation, where the objects and their subparts can be described very precisely using semantic terms, to describe the shape, the structure or the functionality. A classical problem is to be able to identify an object given its shape, and to recognize each of its subparts by first segmenting them, then by labelling these subparts using the concepts available on the semantic domain of this object class. Some of these approaches (Hudelot et al., 2008; Hassan et al., 2010; Fouquier et al., 2012) are able to question the partial semantic description of the scene and to adjust their behaviour to the context. However, we noticed that all these approaches contain in their implementation, algorithms or procedures very specific to the applicative domain.

In this article, we describe an original framework for mesh segmentation that push up the semantic approach, creating a bridge bewteen an expert knowledge description and the segmentation algorithms. This framework allows an expert in a specific domain to formally describe his own domain in terms of a fundamental ontology, without any skill in geometric algorithms. This formalized domain description is then used by the system to automatically recognize objects and their features within that domain.

The genericity of our approach is insured by a multi-layer ontology modeling the expert knowledge. The first layer corresponds to the basic properties of any object, such as shapes and structures. The next layers are specific to each application, describing the functionalities and possible configurations of the objects in this domain. The segmentation and identification mecanism is hidden behind the concepts of the first layer, which are associated to the segmentation algorithms. We wish to highlight here that this work is exploratory and only studies the feasability of the proposed approach on a simple context.

After an overview of the existing methods in semantic-driven expert systems and for mesh segmentation using semantics, we introduce our framework for semantic-driven mesh segmentation and annotation, starting with the expert knowledge modelization, followed by a synthetic catalog of the segmentation algorithms, and finally with the description of the proposed expert system. The fourth section present some experimental results to illustrate the relevancy and feasability of our approach. Finally, we present possible extensions of this work and future improvements.

2 Related work

The design of enhanced vision systems has benefit from semantic knowledge such as ontology-driven strategy. The ontology paradigm in information science constitutes one of the most diffused tools to make people from various backgrounds work together (Seifert et al., 2011). In addition, ontology-based interfaces are a key component of an ergonomic (Seifert et al., 2011), adaptive computer system (Seifert et al., 2011), especially in biological/clinical fields (Othmani et al., 2010) for which concepts and standards are constantly shifting. Furthermore, reasoning capabilities embedded in the logical framework on which ontology softwares are built up should be a definite bridge between computer vision scientists and knowledge engineers. Even though still brittle and limited, reasoning inferences out of visual data may enhance the vision system experience (Othmani et al., 2010) as well. Computer graphics is one of the different domains which benefits from the semantic knowledge through ontology strategy and for that, different applications are designed to ensure the articulation between ontologies and mesh processing and to ensure a semantic representation in different domains including anatomy (Hassan et al., 2010), product design in e-manufacturing (Attene et al., 2009), robotic (Albrecht et al., 2011; Gurau and Nüchter, 2013).

In (Camossi et al., 2007), a system to support a user in the retrieval and the semantic annotation of 3D models of objects in different application contexts is presented. The ontology provides a representation of the knowledge needed to infer object's shape, functionality and behavior. Then, the annotation and the retrieval is performed based on the functional and behavior characteristics of the 3D model.

In (Attene et al., 2009), the ontology was used to characterize and to annotate segmented parts of a mesh using a system called "ShapeAnnotator". To this aim, the ontology is loaded according the type of the input mesh and the user can link segments to relevant concepts expressed by the ontology. While the annotation of the mesh parts is done by a simple link with the "ShapeAnnotator", Gurau and Nüchter (2013) and Shi et al. (2012) proposed to feed an ontology with a set of user-defined rules (*e.g.* geometric properties of objects, spatial relationships) and the final annotation is created according to them.

In (Hassan et al., 2010), an ontology including an approximation of the geometric shape of some anatomical organs was used to guide the mesh segmentation. The parameters needed to segment the input mesh were provided by the ontology. For the case of semantic classification, an ontology was used also in (Albrecht et al., 2011) into a SLAM-generated 3D point cloud map. After the reconstruction of the surface planes in the point cloud, the ontology is used to generate hypotheses of possible object locations and initial poses estimation and the final result is a hybrid semantic map, in which all identified objects have been replaced by their corresponding CAD models. Recently, Feng and Pan (2013) proposed a unified framework which bridges semantics and mesh processing. The mesh is divided into a fixed number of parts corresponding to the number of concepts in the ontology. The parts are then annotated based on some rules defined in the ontology (*e.g.* the head is very dissimilar with limbs).

3 Proposed method

Segmenting an object following its geometry is a non trivial problem, as well as associating semantic concepts to each object and its subparts. Both questions need a very specific processing. Previous works (Hudelot et al., 2008; Attene et al., 2009; Hassan et al., 2010; Fouquier et al., 2012) on this topic are more and more going in the direction of mixing the two problems, in order to help the segmentation using the already extracted semantics, and by extracting the semantics from the partial segmentations.

The framework we present in this section addresses the same goal, with a strong focus on the separation between segmentation algorithms and the semantic reasonings. The benefits of this approach will be experimented in the next section, and the possible extensions discussed in the last one, but we can already underline one direct benefit of this approach: using this framework to address a new applicative domain will only require the user to design the corresponding ontology, without code modification, assuming that the basic concepts he will need are part of the original core of the framework.

The only bridge between algorithms and semantics remains here on the classification and the description of each algorithm. In a first part, we describe our multi-layer ontology paradigm and how the expert knowledge on a specific domain is implemented on top of elementary concepts, then we give the specifications of the elementary segmentation algorithms. Finally, we will describe the way our framework is using these two modules to address the question of semantic-driven mesh segmentation.

3.1 Expert knowledge description

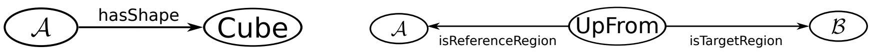
We propose in our framework to model the expert knowledge in 2 steps: 1) the semantic concepts associated to the applicative domain are grouped into a multi-layer ontology; 2) the possible combinations of concepts of a given layer to form concepts of a higher level. These combinations correspond to equivalent concepts in the ontology and will be classified at run-time by the ontology reasoner. Since the core of our framework needs to connect segmentation algorithms with the applicative domain ontology, we designed a core ontology, called $\mathtt{S}_0$ that contains all the elementary concepts required for a segmentation process of objects.

3.1.1 Elementary semantic concepts

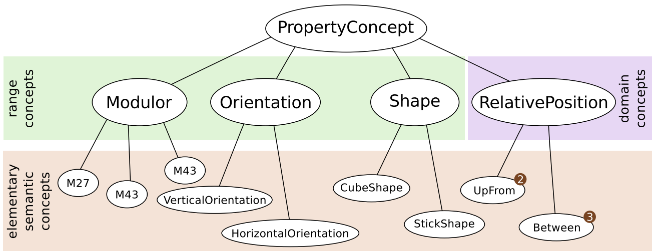
A segmentation and semantic labelling process implies that the algorithmic part is able to identify and label regions with specific properties, such as geometrical properties (*e.g.* stick, board, cube, vertical region), color or texture properties (*e.g.* color uniformity, reflectance, texture patterns), but also properties linked to the position and configuration of subparts with regards to others (*e.g.* parallel regions, $\mathcal{A}$ is up *wrt* $\mathcal{B}$, $\mathcal{A}$ is between $\mathcal{B}$ and $\mathcal{C}$). We will refer to these properties as unary (linking one region of the mesh to one concept, *e.g.* geometry, color, texture) or n-ary (linking several regions together through one concept, *e.g.* topology, distance).

In the expert knowledge description contained in our framework, these properties are constituting the first layer $\mathtt{S}_0$ of our ontology. In this modelization, we choose to group separately the geometrical and chromatical unary properties and the topological n-ary properties. More specifically, an object property and a range concept is associated to each unary property in the ontology (the domain being the object parts). The *range concept* (*e.g.* shape, orientation, color) is then specialized into *elementary semantic concepts* (*e.g.* shape $\rightarrow$ cube, cylinder, sphere, ...) which will correspond to the actual value of the object property. Ontologies cannot directly model n-ary properties and we thus represent them using two object properties and a domain concept. The *domain concept* (*e.g.* position, distance) is also specialized into *elementary semantic concepts* (*e.g.* distance $\rightarrow$ connected, close, far, ...) which are then connected to object parts through 2 object properties (isReferenceRegion and isTargetRegion).

Fig. 1a shows an unary and a n-ary property while Fig. 1b represents a subpart of a possible core ontology with domain / range concepts and their corresponding elementary semantic concepts.

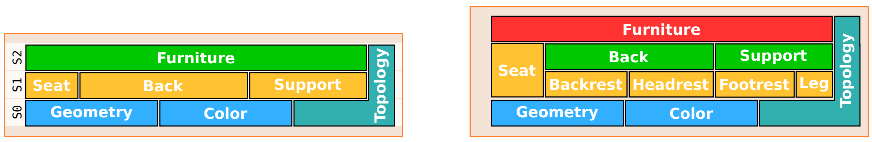


(a)

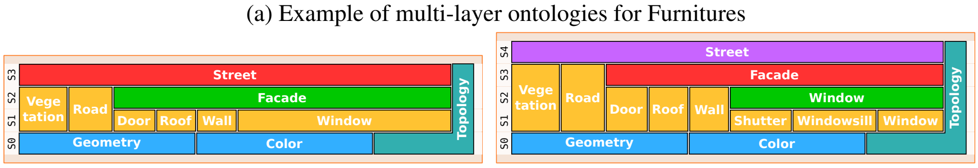


(b)

FIG. 1: (a) Examples of unary and n-ary property. $\mathcal{A}$ and $\mathcal{B}$ correspond to region of the object. (b) A subpart of the elementary semantic concepts of the core ontology. The number beside some concepts indicate that they link several regions together (*e.g.* UpFrom relates 2 regions $\mathcal{A}$ and $\mathcal{B}$).



(a) Example of multi-layer ontologies for Furnitures



(b) Example of multi-layer ontologies for Streets

FIG. 2: Examples of multi-layer ontologies for furnitures (a) and street (b) segmentation. Note how a more detailled expert knowledge descriptions of the same domain can be achieved by simply adding a layer (*e.g.* for the concepts of Back or Support in the case of Furnitures)

3.1.2 Multi-layer ontology

This first layer ($\mathtt{S}_0$, blue blocks in Fig. 2) is part of the core of our framework. It is enriched for each applicative context with specific semantic concepts. In section 3.3 we will describe the effective expert system that make the bridge between semantics and algorithms. It requires that the specific semantic concepts are grouped into two (or more) layers: one layer called $\mathtt{S}_1$ (yellow blocks in Fig. 2), using only references to concepts from the $\mathtt{S}_0$ layer, and that describe all the object configurations that can be combinatorially drawn up, *i.e* the result of a cartesian product between *elementary semantic concepts*. The supplementary layers ($\mathtt{S}_2, \ldots, \mathtt{S}_n$, green, red and purple blocks in Fig. 2) describe the combination rules to populate a complete scene.

3.1.3 Linking two layers: Equivalent concepts

In addition to the semantic concepts of the application domain, another important expert knowledge consists in how the concepts are linked with each other. This can be expressed as a set of equivalent concepts of the $\mathtt{S}_n$ layer describing possible or impossible combinations of concepts of the $\mathtt{S}_{n-1}$ layer. The work of the expert is thus strongly simplified since he can describe not only positive rules, but also negatives. For example in the furniture ontology, a chair leg can be described as a stick shape and a vertical orientation; on the other hand, the combination of a headrest without backrest is an incompatible configuration.

In practice, incompatible configurations are specialized in specific concepts, one for each incompatibility type. This specialization allows to perform some reasoning and classification on partially annotated individuals.

Once these equivalent concepts are given by the expert, they are used in two ways: either to *build a decision tree* (which will be detailled in Section 3.3) or to *suggest segmentation correction*. Indeed, during the segmentation / annotation process, incompatible configuration might appear due to either a segmentation error or a missing equivalent concept. In this case, the reasoner can be asked for the reason of the inconsistency which is then presented to the user for correction. The main advantage of this approach is that it allows us to ask the user to correct errors only in the regions that caused the classification as incompatible instead of having to explore the whole mesh / labelling.

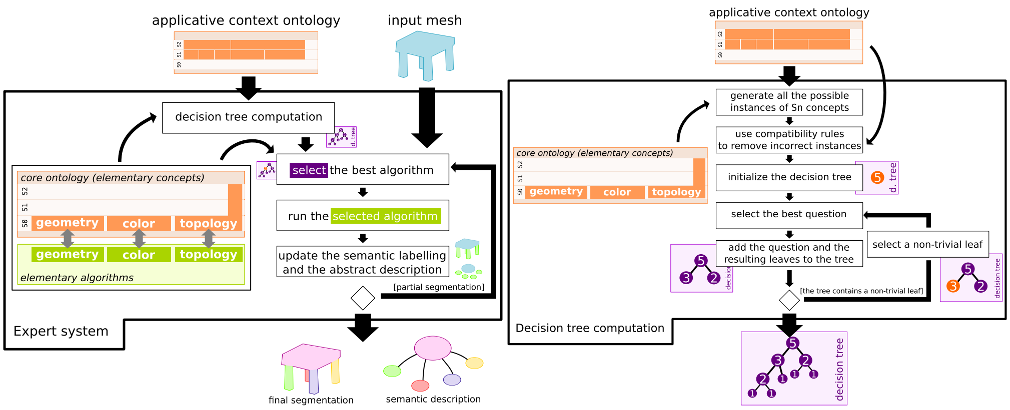
3.2 Semantic type signatures of algorithms

In this work, the core idea about the algorithmic part is to split the method into *elementary algorithms* that are dedicated to one of the elementary semantic concepts. But this consideration cannot be the only guide to produce a synthetic catalog of the algorithms: the type signature of these algorithms cannot be only defined by a single concept.

Each of these algorithms will be involved in the segmentation and labellization process by splitting the given region and by adding semantic descriptors to the subparts. Since a region can be described by more than one concept (for example, a part can be a stick and vertical), and since our goal is to have the more elementary algorithms as possible, we deduce that it also exist algorithms that are not splitting the given region, but only increasing the knowledge on this region. Finally, and because we want to deal with concepts that are not necessary involving a single region, we need to distinguish functions associated to unary properties, and functions associated to n-ary ones.

The following synthetic catalog is a proposal to identify each segmentation algorithm in the context of semantic labelling: semantical questions starting with *find all* (`SF`), with unary concepts (*e.g.* find all rectangles in region $\mathcal{A}$), semantical questions starting with *is it a* (`SI`), with unary concepts (*e.g.* is $\mathcal{B}$ a flat region), topological questions (`T`), identifying n-ary relations between regions (*e.g.* are $\mathcal{B}$ and $\mathcal{C}$ connected), topological questions starting with *find all* (`TF`), using an n-ary relation and 1 to $n-1$ regions, to find regions satisfying the relation *wrt* the input regions (*e.g.* find all regions connected to the region $\mathcal{B}$; find all regions between the regions $\mathcal{B}$ and $\mathcal{C}$).

Each algorithm gets as input one or more regions of the original object and returns a set of regions, each of them enriched by a semantic description generated by the function, plus a score in $[0;1]$ to illustrate the matching between this region and the associated concept. This



(a) Overview of the proposed method applied on the furniture domain.

(b) Offline decision-tree generation from the ontology of the application domain.

FIG. 3: Overview of the proposed method.

first synthetic catalog of possible semantic type signatures covers all the useful algorithms for mesh segmentation using semantic description, as illustrated in the following sections.

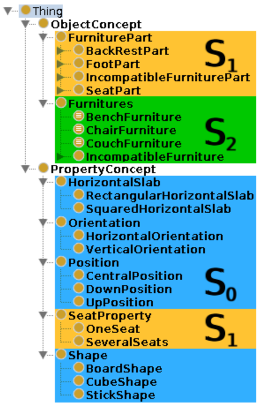
3.3 Expert system

In this section, we describe how the expert knowledge is used in our algorithm to efficiently segment and annotate an object. Fig. 3a gives an overview on this framework.

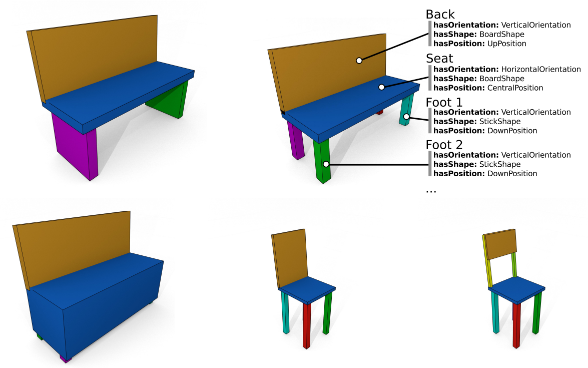
One of the advantage of our approach is that it gives the possibility to easily compute a tree containing the order of the questions to ask to reach the solution in the most efficient way. To build this *decision tree*, we first use the equivalent concepts of each layer to build the set of possible configurations. For each layer and starting at the S_0 layer, the Cartesian product between all the properties of the layer is performed. The reasoner is then used to classify the instances in equivalent concepts and incompatible candidates are removed. The remaining ones are then used in the above layer as semantic concepts in the Cartesian product to compute the new list of possible configurations.

Once the set of all possible configurations Ω is created, the idea is to split it according to the concept maximizing a criterion C. The choice of this concept gives us the question to ask and thus a node of the tree. For each possible answer, we then get the corresponding subset and look for the next concept maximizing C. This operation is iterated until only one possibility is left in each subset. In the resulting tree, the root thus corresponds to Ω and stores the first question to ask, each leaf is a possible configuration and intermediary nodes are subsets of Ω and store the next question to ask. Note that this step can be performed only once and offline in order to speed up the process. This procedure is illustrated in Fig. 3b.

The first inline step of our expert system is to browse the decision tree: starting from the root, the system gets the question to ask and run the associated elementary algorithm on the input object. The system then selects the child node corresponding to the algorithm's result



(a) Multi-layer ontology used to segment the furnitures.



(b) Result of the segmentation and identification on 5 objects from the furniture domain. Result of the identification: orange ⇒ backrest, blue ⇒ seat, other colors ⇒ feet.

FIG. 4: Ontology and meshes used for our experiments on furniture segmentation and annotation.

and the process iterates until a leaf of the tree is reached meaning that the semantics associated to the object is known as only one possible configuration remains.

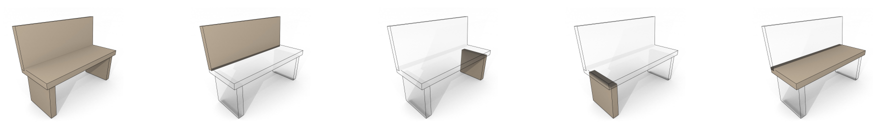
In some cases, the expert system might reach a leaf before every part of the mesh is segmented or annotated (because some concepts might be inferred from the presence / absence of others). Some supplementary algorithms are then run on the mesh to confirm the global semantics and annotate the missing parts. This can again be done very efficiently by questionning the reasoner, which will give the expert system the missing concepts and thus the elementary algorithm to run.

4 Experiments on Furnitures Segmentation and Annotation

The experiments have been done on furnitures segmentation and annotation, using basic shapes relevant to this domain (see Fig. 4b). In this work, we focus our interest on the expert system in order to study the feasability of the presented framework. Finely selecting and adjusting elementary segmentation algorithms will be one of the future work we mention in section 5. We implemented the expert system detailed in section 3.3 using `Java` and the OWL API, and designed our prototype such that the purely mesh manipulations are written in `C++`, using `CGAL`. The connection between these two parts is done using a client/server paradigm via sockets.

To address the specific context of furnitures, we designed a dedicated ontology (see Fig. 4a) using Protégé [1] on top of a simplified version of the elementary concepts introduced in section 3.1.1. The decision tree associated to this ontology is generated following the procedure described in section 3.3 where the criterion C was chosen to be the dichotomy, *i.e.* we look for concepts allowing to split the set into two subset of same size.

1. `http://protege.stanford.edu/`



(a) R0 (b) R1 (c) R2 (d) R3 (e) R4

1. `SF` `OneSeat` in R0: false
2. `SF` `VerticalOrientation` in R0
   — R1: true
   — R2: true
   — R3: true
3. `SI` `DownPosition` R1: false
4. `SI` `DownPosition` R2: true
5. `SI` `DownPosition` R3: true
6. `SI` `StickShape` R2: false
7. `SI` `StickShape` R3: false
8. `SI` `DownPosition` R1: false
9. `SI` `BoardShape` R1: true
10. `SI` `BoardShape` R2: true
11. `SI` `BoardShape` R3: true
12. `SI` `BoardShape` R0 - {R1, R2, R3}
    — R4: true

FIG. 5: List of the elementary algorithms and corresponding answers generated by our expert system to segment and annotate a bench mesh. `SF`: semantic *find all*, `SI`: semantic *is it a*.

Object	#subparts	S. then I.		Naive S&I		Our method	
		preprocessing		-		-	
		# `SF`	# `SI`	# `SF`	# `SI`	# `SF`	# `SI`
bench 1	4	0	21	9	0	2	10
bench 2	6	0	31	9	0	3	13
couch	6	0	31	9	0	3	14
chair 1	6	0	31	9	0	2	15
chair 2	8	0	41	9	0	2	19

TAB. 1: Number of segmentation steps for a complete identification and segmentation.

Once the decision tree is generated, we can run our expert segmentation system on meshes. Fig. 5 gives the list of questions that are computed in order to segment and recognize the first bench in Fig. 4b. The other images in Fig. 4b are illustrating the segmentation and identification process using the same expert ontology with various meshes.

We have compared our segmentation and identification method with other equivalent approaches, and we summarize the results in Tab. 1. The first column, titled *S. then I.* corresponds to an approach where a first segmentation preprocessing is done to split regions, then each region is labeled using the semantic concepts. The second column titled *Naive S&I* corresponds to an approach where the segmentation algorithms dedicated to specific kind of shapes are run independently to identify the regions. The last column corresponds to our approach. For each method, we detailed the number of semantic *find all* (`SF`) and semantic *is it a* (`SI`) algorithms required to segment and annotate the 5 objects shown in Fig. 4b.

We choose to distinguish the 2 kinds of algorithms, because the complexity of each algorithm family is significantly different: a `SF` algorithm will require to browse all the given regions (possibly the whole mesh), and will have to extract subparts from it. In comparison, a `SI` algorithm will only have to validate or not a feature on a given region. Minimizing the number of `SF` runs is thus the main goal of a segmentation and identification process.

For each region of an object, the expert ontology uses 3 range concepts (shape, position, orientation) that implies 8 elementary semantic concepts. Each object is also characterized by a supplementary range concept (number of seats) that implies 2 elementary semantic concepts per object. The number of `SI` algorithms in the *S. then I.* has been estimated counting for each subregion of the mesh one `SI` per elementary semantic concept (minus 1 per range concept that can be deduced), plus the 1 elementary semantic concepts of the full object. The *Naive S&I* consist in running all the available `SF` algorithms. The number of `SF` and `SI` of our method comes from the trace of the experimental runs (see an example in Fig. 5).

We first compare our work to a *S. then I.* approach, where a segmentation preprocessing is applied before the identification. We cannot quantitatively compare this approach with ours, but since we use the expert knowledge to reduce the number of `SF` algorithms in our approach, we can deduce that our `SF` computations are almost equivalently expensive as the preprocessing stage of the *S. then I.* approach [2]. The number of `SI` algorithms is also strongly reduced with our approach.

The second considered approach for comparison is a *Naive S&I* approach, where all the `SF` algorithms are run. The number of `SF` algorithms is strongly reduced by our approach, and the comparison between the *Naive S&I* approach can be summarized with a comparison between the complexity of `SF` and the complexity of `SI` algorithms.

These first results are illustrating the relevancy of our method with respect to the existing approaches: mixing segmentation and identification steps is a good approach to reduce the complexity of the global algorithm.

5 Conclusion and future work

In this paper, we presented a new framework for efficient segmentation and annotation of meshes. It is composed of two blocks: a multi-layer ontology gathering the semantics on the application domain and a processing part allowing to detect elementary geometrical, chromatical and topological concepts. The main advantage of our method is that it separates the knowledge on the domain with the processing allowing an expert to segment and annotate an object without knowledge in image or mesh processing. Another advantage is that using the expert knowledge, we are able to build a decision tree to perform an efficient search amongst the set of possible objects while being able to suggest segmentation and annotation corrections to the user if an impossible configuration is reached.

The ontology we designed for this experiment is very basic, and we plan to experiment during the next months a more complete ontology with more concepts. In particular, the n-ary properties will be integrated in order to express more realistic constraints between subparts of the objects.

The simulation we presented in section 4 is using manual segmentations to mimic the result of the algorithms. Our next step will be to select and adjust elementary algorithms from the litterature, like geometrical extractors (Mortara et al., 2004; Li et al., 2011) or segmentation approaches driven by functionality features (Laga et al., 2013).

Introducing automatic segmentations will open many related questions we plan to handle in a near future. One of the next challenges will be to introduce better approaches to choose between algorithms than the dichotomical one. A first criterion to consider could be a weighting system that favour algorithms with a small computational time, or to include the exactness of the algorithms. These weights will be introduced in the decision tree computation in order to design an expert system that handle the question of efficiency.

2. The best strategy for a preprocessing can be to choose the smallest range concept, then run for each elementary concept a `SF` algorithm.

Applying our approach on meshes acquired from low resolution devices will complicate the job of the segmentation algorithms. It will probably generate incoherent subregions, with overlappings or unlabelled parts. One possible approach is to use fuzzy maps to describe the regions, but another question will have to be handled: how to adjust an existing partial segmentation? Our framework is a good candidate to provide a specific answer to this problem, since the expert knowledge contains information about the expected configurations. One possible extension of this work could be to introduce adjustment algorithms for each elementary concepts, that will be able to adjust a first segmentation using the global knowledge of a specific domain.

Finally, a long term extension of this work will be to introduce it into a machine learning approach, where the ontology will be deduced or extended from an existing one, using a set of shapes of the domain. This extended framework will be a possible challenger of the 3D Shape Retrieval Contest (SHREC) organized each year in the mesh segmentation community.

Acknowledgements Marco Attene thanks the EU FP7 Project VISIONAIR for having supported his contributions in this research.

References

Albrecht, S., T. Wiemann, M. Günther, and J. Hertzberg (2011). Matching CAD object models in semantic mapping. In *Workshop Semantic Perception, Mapping and Exploration (ICRA'11)*, pp. 1.

Attene, M., F. Robbiano, M. Spagnuolo, and B. Falcidieno (2009). Characterization of 3D shape parts for semantic annotation. *Computer-Aided Design 41*(10), 756–763.

Camossi, E., F. Giannini, and M. Monti (2007). Deriving functionality from 3D shapes: Ontology driven annotation and retrieval. *Computer-Aided Design and Applications 4*(6), 773–782.

Feng, X. and X. Pan (2013). A unified framework for mesh segmentation and part annotation. *J. of Computational Information Systems 9*(8), 3117–3128.

Fouquier, G., J. Atif, and I. Bloch (2012). Sequential model-based segmentation and recognition of image structures driven by visual features and spatial relations. *Computer Vision and Image Understanding 116*, 146–165.

Gurau, C. and A. Nüchter (2013). Challenges in using semantic knowledge for 3D object classification. In *KI 2013 Workshop on Visual and Spatial Cognition*, pp. 29.

Hassan, S., F. Hétroy, and O. Palombi (2010). Ontology-guided mesh segmentation. In *FOCUS K3D Conf. Semantic 3D Media and Content*.

Hudelot, C., J. Atif, and I. Bloch (2008). Fuzzy spatial relation ontology for image interpretation. *Fuzzy Sets and Systems 159*, 1929–1951.

Laga, H., M. Mortara, and M. Spagnuolo (2013). Geometry and context for semantic correspondence and functionality recognition in manmade 3D shapes. *ACM Trans. Graphics (TOG) 32*(5).

Li, Y., X. Wu, Y. Chrysathou, A. Sharf, D. Cohen-Or, and N. Mitra (2011). Globfit: Consistently fitting primitives by discovering global relations. *ACM Trans. Graphics (TOG) 30*(4), 52.

Mortara, M., G. Patané, M. Spagnuolo, B. Falcidieno, and J. Rossignac (2004). Plumber: a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies. In *ACM Symposium on Solid modeling and applications*, pp. 339–344.

Othmani, A., C. Meziat, and N. Lomenie (2010). Ontology-driven image analysis for histopathological images. In *Int. Symposium on Visual Computing (ISVC'10)*.

Seifert, S., M. Thoma, F. Stegmaier, M. Hammon, M. Kramer, M. Huber, H. Kriegel, A. Cavallaro, and D. Comaniciu (2011). Combined semantic and similarity search in medical image databases. In *SPIE Medical Imaging*.

Shi, M., H. Cai, and L. Jiang (2012). An approach to semi-automatic semantic annotation on Web3D scenes based on an ontology framework. In *Intelligent Systems Design and Applications (ISDA'12)*, pp. 574–579.

Résumé

La segmentation et annotation de maillages utilisant la sémantique a été l'objet d'un intérêt grandissant avec la démocratisation des techniques de reconstruction 3D. Une approche classique consiste à réaliser cette tâche en deux étapes, tout d'abord en segmentant le maillage, puis en l'annotant. Cependant, cette approche ne permet pas à chaque étape de profiter de l'autre. En traitement d'images, quelques méthodes combinent la segmentation et l'annotation, mais ces approches ne sont pas génériques, et nécessitent des ajustements d'implémentation ou des réécritures pour chaque modification des connaissances expertes. Dans ce travail, nous décrivons un cadre de fonctionnement qui mélange segmentation et annotation afin de réduire le nombre d'étapes de segmentation, et nous présentons des résultats préliminaires qui montrent la faisabilité de l'approche.

Notre système fournit une ontologie générique qui décrit sous forme de concepts les propriétés d'un objet (géométrie, topologie, etc.), ainsi que des algorithmes permettant de détecter ces concepts. Cette ontologie peut être étendue par un expert pour décrire formellement un domaine spécifique. La description formelle du domaine est alors utilisée pour réaliser automatiquement l'assemblage de la segmentation et de l'annotation d'objets et de leurs propriétés, en sélectionnant à chaque étape l'algorithme le plus pertinent, étant données les information sémantiques déjà détectées. Cette approche originale comporte plusieurs avantages. Tout d'abord, elle permet de segmenter et d'annoter des objets sans aucune connaissance en traitement d'images ou de maillages, en décrivant uniquement les propriétés de l'objet en terme de concepts ontologiques. De plus, ce cadre de fontionnement peut facilement être réutilisé et appliqué à différents contextes, dès lors qu'une ontologie de domaine a été définie. Finalement, la réalisation conjointe de la segmentation et de l'annotation permet d'utiliser d'une manière efficace la connaissance experte, en réduisant les erreurs de segmentation et le temps de calcul, en lançant toujours l'algorithme le plus pertinent.