Ontology Based Shape Annotation and Retrieval

Olga Symonova and **Minh-Son Dao** and **Giuliana Ucelli** and **Raffaele De Amicis** [1]

Abstract. In this paper, 3D shape retrieval methodology suited for search in special category of 3D shape is presented. The proposed approach employs a fully unsupervised segmentation algorithm to decompose 3D models into components. Shape distribution vectors describing the resulting components are extracted and together with connectivity relations identify a 3D model. The 3D-shapes we are interested in this paper are models of furniture. Ontology of furniture that we started building will be used in annotation and then key word based retrieval of furniture models. A mapping between low level features extracted by the above mentioned algorithm and ontology concepts is performed. The proposed approach bridges the gap between keyword-based approaches and query-by-example approaches by using not only the low-level features but also a domain ontology.

1 INTRODUCTION

Shape description and retrieval problem arose with the growth of available information in Internet and development of technologies allowing easy creation of 3D models. However modern search engines allow only textual search of information in Internet. This approach is not effective for graphical objects [13]. Special structures describing geometrical and/or topological characteristics were suggested to substitute verbal description of a shape. The authors of [15] group shape descriptors into three large groups: feature based methods, graph based methods and other methods which can be as well compositions of the former two approaches. We refer interested reader to [14], [11], [17], [8], [3]. In our work we use the shape descriptor proposed in [14]. The descriptor is a vector of the distribution of the function defined over the shape. As the authors of [14] examined $D2$ function of the distance between two random points of the shape gives the best results. The shape distribution based descriptor can be used for categorizing 3D models into wide classes, because it is able to detect major differences between shapes, but cannot capture detailed features.

Although geometry and topology based descriptors have improved content based 3D shape retrieval, they still deal with low-level features and this leads to a big gap between low-level and high-level features. Moreover, geometrical-based matching does not consider the semantics of the object to be retrieved [12].

The research in the field of knowledge structuring suggests to use ontology for describing knowledge of a chosen domain. [5]. The author of [7] defines ontology as a specification of a representational vocabulary for a shared domain of discourse which may include definitions of classes, relations, functions and other objects. Therefore, if we know the domain in which the 3D shapes are constructed, the ontology of the domain can be built. Then mapping between low level features and ontology concepts is performed. Finally, 3D shapes are annotated and become well-defined structure under human-perspective.

[1] GraphiTech, 38050 Villazzano (TN), Salita Dei Molini 2, Italy

2 PROPOSED METHODOLOGY

2.1 Problem of shape similarity and appropriate assumptions

Content retrieval is a difficult task which is affected by the problem of ambiguity of words and shapes. It can be explained by variety of words, images and 3D models which have equal or similar spelling, shape but different meaning in different domains. This task became even more complicated while dealing with 3D models. The file containing a 3D model often lacks any description, its name can be ambiguous, misleading or not carrying any useful information. As a result descriptors containing geometrical and/or topological information are defined to be used in shape retrieval. The research in this field has proved that searching 3D graphical objects using words has worse results than while using shape descriptors [13]. However shape descriptors do not solve the problem of shape ambiguity. According to the domain where a model is used, it can have different semantic meaning, e.g. the model with the shape of human hand can be considered as a part of human body in the domain of human models or as a glove in the domain of clothing models. To solve similar problems existing in natural language (like words with different meanings) the current research suggests to build ontologies of different domains and interpret a word within the chosen domain. In order to transfer this approach to the field of shape retrieval we build ontology for 3D models, assuming that a model can be completely described by connectivity relations between its constituents and their shape. Restricting our models' domain to the one of furniture, we explain how we build the ontology of furniture, how we extract feature vectors from shapes, and using the latter, how we annotate the model and retrieve 3D objects within the same category.

According to the chosen furniture domain we can assume that models are created using Constructive Solid Geometry (CSG) approach. Thus the furniture models are assemblies of meaningful atoms that are similar to geometric primitives. To prove that this assumption does not constrict too much the number of 3D models which can be used in the proposed approach we performed a search of 3D furniture models in Internet. We downloaded 98 furniture models from Princeton Shape Benchmark [2] and Free Stuff of 3D Cafe [1]. After analysis we found that 63% of furniture models are compound models (here we notice that 88% of models from Princeton Benchmark are compound), and 75% of compound models are models composed from geometrical primitives. We suppose that these figures can increase when a 3D database is created by designers from the same industrial domain. As consequence our assumptions will be valid for the majority of CAD models, because assembly modelling is effective approach, which allows designers to work together

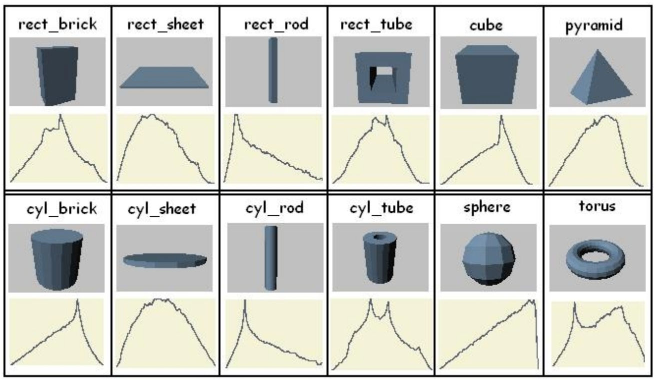

Figure 1. Geometrical primitives used for composition of 3D models and their shape distributions.

on a complex model and gives a possibility of the further reuse of designed objects.

2.2 Feature vector extraction

Given a query 3D model we start analyzing it. Considering the assumption that models are compositions of conceptual parts, we start the model analysis from its decomposition into the constituents. We load the triangle mesh, representing the given model, and then we perform its decomposition into connected components. The decomposition process has the complexity $O(|V| + |E|)$. Then we analyze the shape of each constituent of the model using the approach suggested in [14]. For each constituent we construct the vector of shape distribution. The choice of using a descriptor based on shape distribution is determined by simplicity of construction, invariance to affine transformations and good discriminative results for the models similar to geometrical primitives, like cubes, spheres, cylinders, etc [9]. According to the assumptions stated before, we consider models that are the compositions of geometrically simple objects. As a consequence we can build the finite set of the geometric primitives, which can be used to construct CAD models. For each of such geometrical primitives we extract the distribution based shape descriptor, and we label primitives with the corresponding name. At this stage the construction of the database of geometrical primitives and labeling them with corresponding names are done manually. The number of geometrical primitives which can be used for the composition of furniture model is finite, thus once the database has been constructed it can be used later without user intervention. Figure 1 illustrates which geometric primitives we have considered along with their shape descriptions. Having decomposed the given 3D model into constituents, we start to compare each part with geometrical primitives from Figure 1. The smallest distance between the vectors of shape descriptors identifies the shape of the analyzed constituent. In the current work we calculate Euclidean distance; however the other types of metrics, like Earth Mover and the Kolmogorov-Smirnov distances [14] can be used. The constituent inherits the label with the name of the most similar geometric primitive. The process continues for all parts of the model. As a result we output the vector, which has as components the names of constituent parts of the query model. For better description of the model we also analyze the connectivity relations between the parts. We compute the principal eigenvectors of the triangulations representing each part of the model and we calculate the angle between them in pairs. In this way we obtain $n \times (n-1)$ values of angles between model constituents where n is the number of connected components.

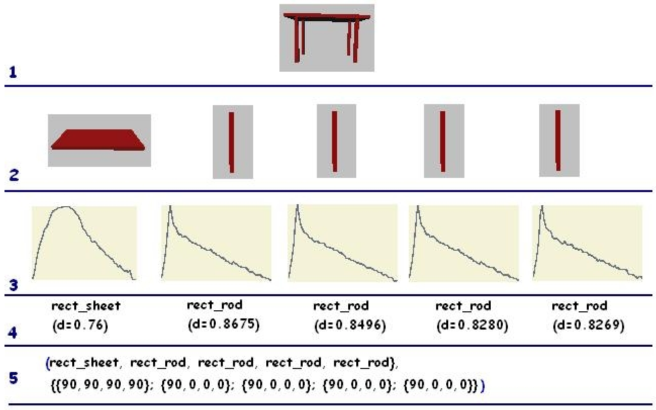

Figure 2. Feature vector extraction. 1) Input model. 2) Model decomposition. 3) Shape distributions of the model's constituents. 4) Labelling model's constituents as shape primitives. 5) Output feature vector.

To clarify the shape analysis process we consider the example of a 3D model of a table. Figure 2 shows this process. As a result we pass the feature vector identifying the query model to the Table 1, which describes the ontology of the domain . In the next chapter we explain how having the feature vector we obtain the vector of semantic labels.

2.3 Mapping feature vector to semantic labels using knowledge domain

In order to map geometrical and topological features of an object from a specific domain to semantically meaningful constituents of the object we should create a database describing all models of the domain. Table 1 illustrates the description of the models of the table of Figure 2.

Table 1. Mapping from low-level features to semantic labels.

Component	Geometrical primitive	Connectivity (pairwise angle between components	Semantic label
1	rect_sheet	{90,90,90,90}	top
2	rect_rod	{90,0,0,0}	leg
3	rect_rod	{90,0,0,0}	leg
4	rect_rod	{90,0,0,0}	leg
5	rect_rod	{90,0,0,0}	leg

The database should describe all concepts present in the ontology of the domain. Thus, querying it by the feature vector we can output as a result the vector of semantic labels. For instance taking the model of the table of the previous example, we get {*top,leg,leg,leg,leg*}, and we can pass the given semantic vector to the domain ontology in order to identify the category the model belongs to.

2.4 Ontology for shape annotation

Before building an ontology we should define its scoping, that is its domain, and its purpose, that is its intended usage [16]. In our case the domain that our ontology will formalize is that of furniture. In the first phase the intended usage of the furniture ontology is the annotation of models in the database with ontology concepts. In a second phase we want to investigate the possibility of retrieving the models by textual queries. We regard the 3D models as a syntactic domain and the ontology language as a semantic domain. An interpretation function will assign to each ”3D model” a concept from ontology. In this way we can say that a certain 3D model is a ”Chair”, while another 3D model is a ”Table”, where ”Chair” and ”Table” are concepts in our ontology. Since the classes of models are distinguished at the syntactic level by the feature vectors extracted and explained in the above sections, there are two interesting questions that an ontology based shape annotation system should answer:

1. What is the system precision? The precision of the system is defined in the well know way:

$$P = \frac{M_{Cadn}}{M_{adn}} \times 100\% \qquad (1)$$

   where M_{Cadn}- is the number of correctly annotated models and M_{adn} is the number of annotated models. Since the system is still not fully operational we cannot quantify its precision, but we can make an interesting observation. The upper boundary of what can be achieved is already known. If the properties that distinguish two ontology concepts cannot be mapped to distinct sets of syntactic features that the above component can extract then the system will fail to correctly annotate the models. Let’s suppose for example that there are in our ontology two concepts named ”YellowChair” and ”BlueChair”. Both concepts have as their superclass the concept ”chair” and they are distinguished only by the color they have: respectively yellow and blue. Because the above mentioned algorithm cannot extract the color of an object the system will fail to correctly annotate ”BlueChair” and ”YellowChair” models. However, assuming that for designers the shape of a model is a more important matter than its color, we suppose that the feature vector extracted on the previous step completely describes a model.
2. The second relevant parameter is the recall of the system.

$$R = \frac{M_{adn}}{M_T} \times 100\% \qquad (2)$$

   where M_T is the total number of models we have. If all models are well formed the recall will be 100%.

We started building the furniture ontology using Wordnet Domains [4]. Developed at IRST, Wordnet Domains, is PWN (Princeton Wordnet) 1.6 [6] augmented with a set of Domain Labels. PWN 1.6 synsets have been semi-automatically linked with a set of 200 domain labels taken from Dewey Decimal classification, the world most widely used library classification system. The domain labels are hierarchically organized and each synset received one or more domain labels. We are interested in the synsets that are annotated with the domain ”furniture”. Because PWN is a linguistic resource and many concepts found there are not suitable for building an ontology of furniture we want to make use in our work of other ontologies and specialized thesauri.

We decided to encode our ontology in OWL language. At the moment the ontology is a simple taxonomy enriched with a relation ”hasPart” that specifies the parts of objects in the furniture domain. We make use also of cardinality restrictions as the following example, which describes the entry for the concepts ”BackRestChair” and Back Rest Chair ”BackRestChairWithFourLegs”, shows:

```
<owl:Class rdf:ID="BackRestChair">
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:onProperty>
         <owl:ObjectProperty rdf:
                         ID="hasPartLeg"/>
       </owl:onProperty>
       <owl:someValuesFrom rdf:
                         resource="#Leg"/>
     </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:someValuesFrom rdf:
                    resource="#BackRest"/>
       <owl:onProperty>
         <owl:ObjectProperty rdf:
                    ID="hasPartBackRest"/>
       </owl:onProperty>
     </owl:Restriction>
   </rdfs:subClassOf>
 </owl:Class>
 <owl:Class rdf:
           ID="BackRestChairWithFourLegs">
   <rdfs:subClassOf rdf:
                resource="#BackRestChair"/>
   <rdfs:subClassOf>
     <owl:Restriction>
       <owl:onProperty>
         <owl:ObjectProperty rdf:
                      about="#hasPartLeg"/>
       </owl:onProperty>
       <owl:cardinality rdf:datatype=
       "http://www.w3.org/2001/XMLSchema#int"
       >4</owl:cardinality>
     </owl:Restriction>
   </rdfs:subClassOf>
 </owl:Class>
```

The above OWL representation says that a ”BackRestChair” has as a part exactly one ”BackRest” and that a ”BackRestChairWithFourLegs” IS-A ”BackRestChair” and has exactly four legs. The only kind of inference needed in the example is the ”inheritance” of properties from super-classes to their subclasses.

2.5 Retrieval through annotations

After we annotated the 3D models with ontology concepts users have two possibilities. First they can make retrieval of 3D objects by textual query. A query can be typed by the user or can be formed by ontology browsing. For example a user interested in barber chair models can input the concept in a text box. Alternatively he can browse the ontology and select the appropriate concept. The system will answer the user query by returning all the models annotated with the input concept or with a subconcept of the input concept. An enhanced retrieval system based on textual queries can take advantage of Boolean operators.

The second possibility is to query by an example model. Here a user can browse all models within the category of the input model and autonomously search for more similar models. Such approach groups all objects into quite large classes. The other way to search for similar models is to find the smallest dissimilarity measure (i.e. the smallest distance value) between feature vectors of the constituents of a sample model and corresponding parts of models from the same category. Such approach reduces the number of comparisons needed to retrieve similar models. Furthermore, as was pointed out in [10] the descriptors based on shape distribution do not give good discriminative results for models with detailed shape properties. Decomposition of the model and shape understanding allows to perform comparison between each constituent part separately. As a result the overall dissimilarity measure will be the sum of dissimilarities between corresponding constituent parts.

3 CONCLUSIONS AND FUTURE WORK

In the current work we presented the methodology for the the new synthesis of shape description and ontology-based annotation and retrieval. Performing shape analysis we decompose a 3D model into its constituent and we analyze the shape and connectivity between each of the parts of the model. As a result we output the feature vector describing the 3D model. Using a database defining all concepts of the ontology of the given domain (here furniture), we map the extracted feature vector to the vector of semantic labels. Finally, the ontology of the considered domain will be used in model annotation and then key word based retrieval of furniture models. The proposed method offers two options to the user: textual query and query by a sample model. As a result, the proposed method succeeded in term of shape-to-text (shape annotation) and text-to-shape (query shape by text) schemes. In the future, the database will be enriched not only in terms of number of 3D models but also by number of other specific domains. Beside that, the ontology will be constructed in more details to improve the accuracy of the query process.

ACKNOWLEDGEMENTS

We thank Eduard Barbu for the useful discussions and comments on ontology construction.

This work has been supported by the CFP6 IST NoE 506766 AIM@SHAPE.

REFERENCES

[1] 3d cafe. `http://www.3dcafe.com`.

[2] Princeton shape benchmark. `http://shape.cs.princeton.edu/search.html`.

[3] Marco Attene, Silvia Biasotti, and Michela Spagnuolo, 'Shape understanding by contour driven retiling', *The Visual Computer*, **19**(2-3), 127–138, (2003).

[4] Magnini Bernardo and Cavaglia Gabriela, 'Integrating subject field codes into wordnet', in *2nd International Conference on Language Resources & Evaluation*, Athens, Greece, (May-June 2000).

[5] Balakrishnan Chandrasekaran, John R. Josephson, and Richard V. Benjamins, 'What are ontologies, and why do we need them?', *IEEE Intelligent Systems*, **14**(1), 20–26, (1999).

[6] Christiane Fellbaum (Ed.). Wordnet: An electronical lexical database. MIT Press, May 1998.

[7] Thomas R. Gruber, 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, **5**(2), 199–220, (1993).

[8] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii, 'Topology matching for fully automatic similarity estimation of 3d shapes', in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, (August 2001).

[9] Taesik Hong, Kunwoo Lee, Sungcnan Kim, Chongnam Chu, and Hyunchan Lee, 'Similarity comparison of mechanical parts', *Computer-Aided Design and Applications*, **2**(6), 759–769, (2005).

[10] Cheuk Yiu Ip, Daniel Lapadat, Leonard Sieger, and William C. Regli, 'Using shape distributions to compare solid models', in *SMA '02: Proceedings of the seventh ACM symposium on Solid modeling and applications*, New York, NY, USA, (2002).

[11] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz, 'Rotation invariant spherical harmonic representation of 3d shape descriptors', in *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, Aire-la-Ville, Switzerland, Switzerland, (June 2003).

[12] Patrick Min, *A 3D Model Search Engine*, Ph.D. dissertation, Princeton University, January 2004.

[13] Patrick Min, Michael Kazhdan, and Thomas A. Funkhouser, 'A comparison of text and shape matching for retrieval of online 3d models', in *Research and Advanced Technology for Digital Libraries: 8th European Conference*, Bath, UK, (September 2004).

[14] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin, 'Matching 3d models with shape distributions', in *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, Washington, DC, USA, (2001).

[15] Johan W.H. Tangelder and Remco C. Veltkamp, 'A survey of content based 3d shape retrieval methods', in *Shape Modeling International*, Genoa, Italy, (June 2004).

[16] Mike Uschold and Martin King, 'Towards a methodology for building ontologies', in *IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, (August 1995).

[17] Dejan V. Vranic and Dietmar Saupe, '3d shape descriptor based on 3d fourier transform', in *ECMCS '01: Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services*, Budapest, Hungary, (September 2001).