# 3D Object Search Through Semantic Component

Chunjing Xu[1], Zhengwu Zhang[1], Jianzhuang Liu[1,2], and Xiaoou Tang[1,2]
[1]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[2]Department of Information Engineering, The Chinese University of Hong Kong, China
{ cj.xu, zw.zhang }@sub.siat.ac.cn, { jzliu,xtang }@ie.cuhk.edu.hk

## ABSTRACT

In this paper, we present a novel concept named semantic component for 3D object search which describes a key component that semantically defines a 3D object. In most cases, the semantic component is intra-category stable and therefore can be used to construct an efficient 3D object retrieval scheme. By segmenting an object into segments and learning the similar segments shared by all the objects in the same category, we can summarize what human uses for object recognition, from the analysis of which we develop a method to find the semantic component of an object. In our experiments, the proposed method is justified and the effectiveness of our algorithm is also demonstrated.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*

## General Terms

Algorithms, Design

## Keywords

3D object search, Semantic component

## 1. INTRODUCTION

Effective representation and visualization of an object in 3D are of great importance in current multimedia applications. Though we can obtain 3D raw data[1] much more easier due to the advance of technology, machine understanding and characterization of the data are still challenging tasks. Some basic concepts, such as how similar one object is to the other, are still hard to define mathematically such that they can match our visual perception. These difficulties result in deficiency in content based searching, indexing, and classification of 3D objects.

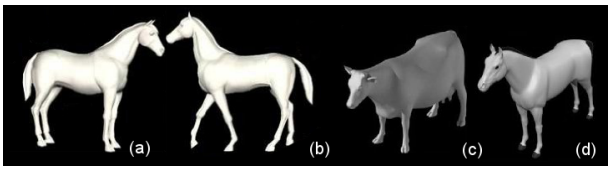


**Figure 1: Recognition of objects by learning resemblance and dissimilarity.**

By tackling 3D objects from different aspects, many description features and similarity measures have been proposed in the literature. Some applications can be tried online (such as Princeton 3D model search engine[2]). Regarding the characteristics used, most methods that handle 3D objects can be categorized into five groups: (i) tackling an object as a whole [9, 10], (ii) local features [1, 11], (iii) spatial maps and distributions [12, 8, 13], (iv) structures [7, 2, 16], and (v) 2D view based [3, 14]. To deal with more shape variations and to achieve better performance, a mixture of methods from different groups is a reasonable approach often used by researchers.

In a common scenario, a feature is extracted from an object and then is used in a specific application, such as object retrieval, to justify its effectiveness. In this application, usually the features from two 3D objects are compared by a similarity measure. This procedure has been followed many years with some fundamental flaws ignored. In these flaws, the most inconspicuous is that an extracted feature depends on its corresponding object in some specific shape/pose. The feature extraction in this way is far from the approach involving in human recognition of 3D objects.

Take the recognition of the objects in Figure 1 as an example. By giving a feature $f$ and feature data $f_a$, $f_b$, $f_c$, and $f_d$ extracted from the four objects. Machine takes two feature data as input and then gives a similarity estimation. Consider the same horse with different poses shown in (a) and (b). Ideally, $f_a$ and $f_b$ should be exactly the same. However, they usually should exhibit some difference if we want the feature $f$ has more discriminative power, such that in the case of comparing (c) and (d), $f_c$ and $f_d$ can be separated from each other. Therefore, a proper tradeoff between invariance and discriminativity has to be carefully considered when we design a feature. Unfortunately, determining such

[1]Here 3D raw data mean the data of an object represented by meshes, voxels, or range values, etc.

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


*MM'10,* October 25–29, 2010, Firenze, Italy.

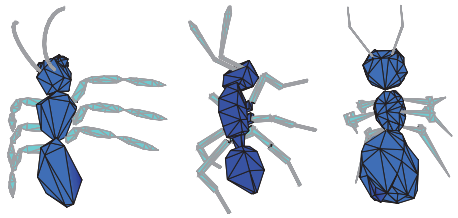

**Figure 2: Examples of semantic components (the blue bodies).**

a tradeoff is a very tricky task. However, human can easily overcome the difficulty by learning from given objects, perceiving different and similar components between objects and then pay distinct attentions to various aspects of objects. Can machine learn in a similar manner?

Notice that the objects from (a) to (d) in Figure 1 share some very similar components — their limbs. They are not distinctive among the objects and play a minor role in the recognition of these objects, i.e., we may not notice the difference of limbs but still can easily conclude that both (a) and (b) are horses, and at the same time distinguish (c) from (d). Ignoring components like the limbs leads to the following benefits for machine recognition of 3D objects:

i) The intra-category **invariance** is increased. For example, for the two objects (a) and (b) in Figure 1, considering the main bodies of them without the limbs, they can be easily matched.

ii) The inter-category **variance** is increased too. Take (c) and (d) in Figure 1 as an example. Similar limbs between them could mess up the obtained feature data and raise the similarity between them.

Through extensive observation, we know that most objects contain a key component, by which they are semantically defined. The fact that we can recognize a horse and distinguish it from a cow is not because of their limbs but their main bodies. For a general object, determining its visual key component is not trivial. In what follows, we discuss how to define the key component, called **semantic component**, for a given object and develop an algorithm to locate it. We also explore its applications and discuss its pros and cons.

## 2. SEMANTIC COMPONENT

As what we mentioned above, we try to understand an object by its key component which carries the main semantic meaning to identify the object. It should be noticed that semantic component for an object can be composed of more than one basic segments. For example, the semantic component of a chair includes its back and seat base. The semantic component can also change due to different context and the learning data offered. For example, if we consider the category of horses, then the semantic component is the trunk of a horse. However, if we consider the category of quadrupeds, then the limbs become the semantic component.

Suppose that there are a set of 3D objects $\mathcal{O}_1, \cdots, \mathcal{O}_M$ composed of $K$ categories, a decomposition scheme $\Psi$ of each object, and a similarity scheme $\Gamma$. For each category, say, the $k$th, we have a set of segments $C_1, \cdots, C_{M_k}$, which can be obtained by some over-segmentation algorithm such as [5] from all the objects in the $k$-th category. Regard $C_i$ as an object and let $\mathscr{C}_{ij}$ be the similarity between $C_i$ and $C_j$, where

$$\mathscr{C}_{ij} = \begin{cases} 0 & C_i \text{ and } C_j \text{ from the same object,} \\ \Gamma(C_i, C_j) & \text{otherwise.} \end{cases} \quad (1)$$

We can have a similarity matrix $\mathscr{C} = [\mathscr{C}_{ij}]$. By feeding it to a clustering algorithm such as spectral clustering, we can easily have a 2-cluster partition of the segments. The cluster with higher intra-similarity represents the collection of the segments whose shapes are commonly shared by the objects in the $k$th category. The collection of the segments in this cluster is defined as semantic components.

An experiment that follows the above method is carried out by randomly choosing a set of 3D objects from the Princeton shape benchmark. The set consists of 50 objects from 5 categories. Some examples of semantic components derived in this experiment are shown in Figure 2. By repeating the experiment many times (more than 100) and observing the obtained semantic components for the objects, we notice that the semantic components are usually bulky and unwieldy parts of the objects. It is reasonable because articulated components usually can change sharply, leading to lower intra-category similarity. Therefore, articulated parts generally are not included in the cluster that defines the semantic components.

In practical applications, the category labels of the objects in a database are usually unknown. Therefore, we cannot follow the method above to obtain semantic components. Next, we develop a method that can separate an object into two components, one of which is the semantic component.

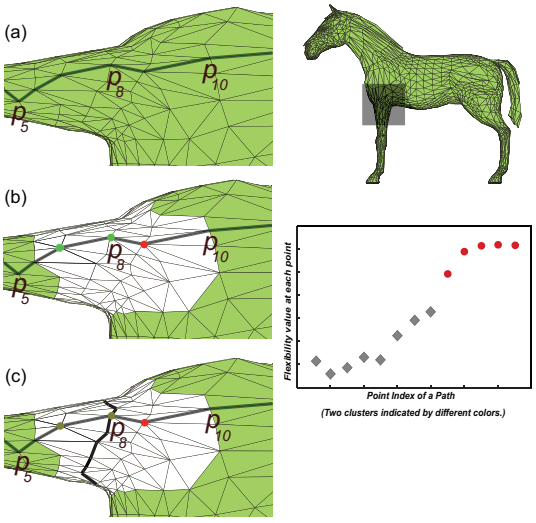


**Figure 3: Illustration of the segmentation procedure on a horse. (a) A walk on the mesh to generate a path $p_1, \cdots, p_n$, with the "flexibility" along the path computed. Only one path is shown here. (b) Using the flexibility values for the points to categorize the points into two classes, and marking the m-step neighborhood. For convenient observation, here we set $m = 1$. (c) Using the randomized cut to obtain a refined segmentation cut. The white region denotes the candidates.**

In the discussion above, we ignore the details of the decomposition scheme $\Psi$. How to segment a 3D object into several segments is a traditional topic in computer graphics. Numerous methods have been proposed in the literature [4, 15]. However, most methods cannot be used in our task because they serve for graphics processing purpose, instead of providing semantic information of objects. Moreover, these methods usually provide over-segmentation which is too delicate to shape variations. To meet our requirement that an object is partitioned into two components with one being semantic, we develop a new segmentation approach in the following.

Inspired by [17, 6], we find that the *flexibility* feature can be used to represent the bulkiness of object parts. The flexibility gives each point on an object a value which denotes the deformable potential at that point. To use the flexibility to guide segmentation, we define a random walk path on an object surface mesh, $p_1 \cdots p_n$, where vertices $p_i$ and $p_{i+1}$ $(1 \leq i < n)$ are neighbors on the mesh and $n$ is set to one half of the number of vertices on the mesh. By separating the flexibility values at $p_1, \cdots, p_n$ into two clusters, we can derive a labeling for vertices on the path. Select those vertices whose m-step neighbors[3] are with different labels and mark the triangle patches containing the vertices as segmentation candidates[4]. By repeating the process many times (100 in our case), multiple paths and their corresponding candidates can be found. With these candidates, we then use the randomized cut [5] to obtain the segmentation. The procedure can be seen from Figure 3.

By the above segmentation scheme, $L$ segments, $C_1, \cdots, C_L$, are obtained from an object. We can have a flexibility vector $H_i$ for each $C_i$, where $H_i$ is the histogram of the flexibility values for the points on $C_i$. Taking each $H_i$ as a sample and feeding them into a clustering algorithm (such as K-means), we obtain two clusters that classify the segments into semantic and non-semantic components.

We compare the semantic components obtained by the steps as shown in Figure 3 with those derived by the learning process as we discuss at the beginning of this section. A number of objects with their semantic components are computed by the two approaches. We find that they generate very similar results. This indicates that the later method is effective without the need of a group of labeled objects, even though it handles one object each time.

In above discussion, we do not consider the topological structure among the segments $C_1, \cdots, C_L$. To obtain a more robust similarity measure, we take the connecting relationship between components as another feature. The structure can be represented by a graph and the matching between two graphs defined in [16] can be used to gauge the structural similarity between two objects.

## 3. IMPLEMENTATION & EXPERIMENTS

In this section, we describe the implementation of the method proposed in Section 2. We also show the effectiveness of our method with a set of experiments.

In these experiments, one critical element is to define a similarity measure between two given objects $\mathcal{O}_a$ and $\mathcal{O}_b$.

[3]Two vertices are $m$-step neighbors when the minimum number of edges between them is less than or equal to $m$.
[4]Segmentation candidates are meshes confining that the future segmentation can only occur on them.

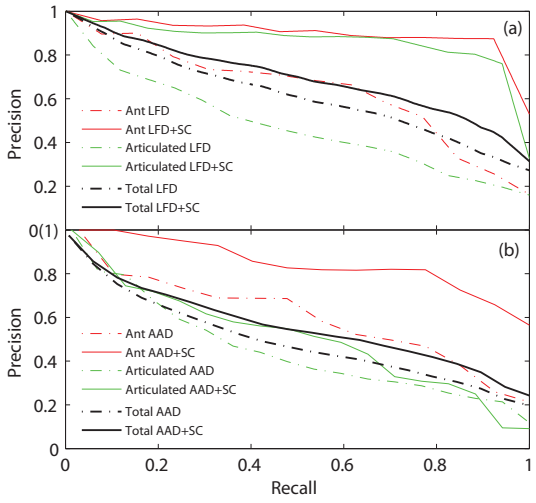


**Figure 4: The improvements by exploiting semantic component(SC) on the McGill 3D shape benchmark. (a) Compared with LFD. (b) Compared with AAD.**

For each object $\mathcal{O}_i, i = a, b$, we can have a duple $\langle S_i, N_i \rangle$, where $S_i$ is the semantic component and $N_i$ is the non-semantic one. Note that $S_i$ or $N_i$ can contain multiple segments of an object. For $S_i$, which is usually the main body of the object without articulated segments, many matching schemes, including those using relatively simple geometrical or local features, can be used to handle the component. On the other hand, for non-semantic components, those methods suitable for semantic component matching may not work well any more. In this case, a similarity measure employing object structure may achieve better performance.

Therefore in our scheme, we compute two measures $\Gamma_s$ and $\Gamma_n$ and then combine them to obtain a similarity measure:

$$\ell_{\mathcal{O}_a,\mathcal{O}_b} = \Gamma_s(S_a, S_b)[\Gamma_n(N_a, N_b)]^{\nu}, \tag{2}$$

where $\Gamma_s$ and $\Gamma_n$ are the similarities for semantic and non-semantic components respectively, and $\nu$ is a constant to balance $\Gamma_s$ and $\Gamma_n$ ($\nu = 0.5$ in this paper). In the following experiments, $\Gamma_n$ comes from the connecting component structure as discussed in Section 2. For $\Gamma_s$, we employ two well-known similarity measures and test if our approach can improve their performance.

The experiments are carried out on two most commonly used data sets, McGill[5] and Princeton[6] 3D shape benchmarks. The two well-known measures are AAD shape distributions [13] and Lightfield (LFD) [3], which are used to compute $\Gamma_s$ in (2) after each object is decomposed into two components.

In the experiments on the McGill 3D shape benchmark, we use objects in i) a specific category (say ant), ii) an articulated object set, and iii) the whole database as queries to test the average retrieval performance. The results are shown in Figure 4 (a) and (b), where the dash lines are the P-R curves for using LFD and AAD only, and the solid lines are for our scheme with SC short for semantic component. The lines with the same color are a pair showing the performance when the queries are from the above three cases. The improvements of retrieval performance with both

[5]http://www.cim.mcgill.ca/∼shape/benchmark.
[6]http://shape.cs.princeton.edu/benchmark.

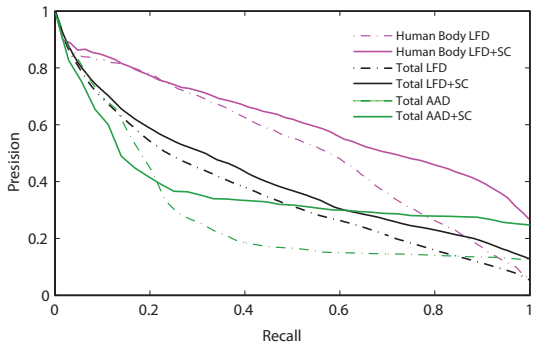


**Figure 5: Tests on non-artificial objects from the Princeton 3D shape benchmark.**

measures AAD and LFD on articulated objects (including ants) are most significant. There is about 20% precision increase at the same recall level on average. It is not surprising that there is no so distinct improvement for objects without articulated parts. However, we can still have nearly 10% precision improvement on the whole database.

We also test our method on Princeton 3D shape benchmark (PSB) with LFD and AAD. All non-artificial objects, which are generally considered as the toughest challenge to 3D object retrieval, are included in our experiments. In Figure 5, first we show the performance when taking human bodies, a class of typical objects in PSB, as queries. We can see that the precision rate can be 10%–20% better when we demand a high recall rate. On average we can achieve about 5% higher precision, compared to the original LFD method. Generally, the experiments on all non-artificial objects result in 4%–5% precision improvement. Similarly, our method can improve the precision of AAD about 5% to 8% when the recall rate is high.

# 4. DISCUSSION AND CONCLUSION

In this paper, we have presented a novel concept named semantic component to describe the key component that semantically defines an object. In most cases, the semantic component is intra-category stable and therefore can be used to construct an efficient object retrieval scheme. We have also proposed a method to segment an object and then to find its semantic component. In the experiments, our theory is justified and the effectiveness of the proposed scheme is demonstrated.

From our experiments, we also notice that our scheme gives less improvement when dealing with objects without articulated parts. In many cases like this, the semantic component of an object is the object itself. Sometimes, we cannot find semantic component for an object. Take snake as an example. Given a set of learning samples in various configurations, We can hardly learn a component shared alike among the samples since the shape of a snake body is equally tenuous at any part. With our theory we can expect a structure involved measure, like those using skeleton, can achieve a better performance on these objects.

In our future work, we plan to build an object component warehouse including a comprehensive collection of everyday objects. With it we can learn which part or structure is semantically important to each object in the warehouse, from which we can obtain a better understanding of human perception of 3D objects.

# 5. ACKNOWLEDGEMENT

This work was supported by grants from Key Laboratory of Robotics and Intelligent System, Guangdong Province (2009A060800016), Natural Science Foundation of China (No. 60975029), Shenzhen Bureau of Science Technology & Information, China (No. JC200903180635A), and the Research Grants Council of the Hong Kong SAR, China (Project No. CUHK 414306, 415408).

# 6. REFERENCES

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 24(4):509–522, 2002.

[2] S. Biasotti, S. Marini, M. Mortara, G. Patane, M. Spagnuolo, and B. Falcidieno. 3D shape matching through topological structures. In *Discrete Geometry for Computer Imagery*, pages 194–203. Springer, 2003.

[3] D.Y. Chen, X.P. Tian, Y.T. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum*, volume 22(2), pages 223–232, 2003.

[4] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. In *ACM SIGGRAPH*, pages 73–85, 2009.

[5] A. Golovinskiy and T. Funkhouser. Randomized cuts for 3D mesh analysis. *SIGGRAPH ASIA*, 27(5), 2008.

[6] B. Gong, C. Xu, J. Liu, and X. Tang. Boosting 3D object retrieval by object flexibility. In *ACM Multimedia*, pages 525–528, 2009.

[7] M. Hilaga, Y. Shinagawa, T. Kohmura, and T.L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *ACM Annual Conference on Computer Graphics and Interactive Techniques*, pages 203–212, 2001.

[8] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 164–171, 2003.

[9] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Symmetry descriptors and 3D shape matching. In *Eurographics/ACM SIGGRAPH Symposium on Geometry processing*, pages 115–123, 2004.

[10] I. Kolonias, D. Tzovaras, S. Malassiotis, and MG Strintzis. Fast content-based search of VRML models based on shape descriptors. *IEEE Transactions on Multimedia*, 7(1):114–126, 2005.

[11] M. Kortgen, G.J. Park, M. Novotni, and R. Klein. 3D shape matching with 3D shape contexts. In *7th Central European Seminar on Computer Graphics*, volume 4, 2003.

[12] M. Novotni and R. Klein. 3D Zernike descriptors for content based shape retrieval. In *Proceedings of the eighth ACM Symposium on Solid modeling and applications*, pages 225-233. 2003.

[13] R. Ohbuchi, T. Minamitani, and T. Takei. Shape-similarity search of 3D models by using enhanced shape functions. *International Journal of Computer Applications in Technology*, 23(2):70–85, 2005.

[14] R. Ohbuchi, M. Nakazawa, and T. Takei. Retrieving 3D shapes based on their appearance. In *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 39–45. ACM, 2003.

[15] A. Shamir. A survey on mesh segmentation techniques. In *Computer Graphics Forum*, volume 27(6), pages 1539–1556. John Wiley & Sons, 2008.

[16] T. Tung and F. Schmitt. Augmented reeb graphs for content-based retrieval of 3d mesh models. In *Proc. IEEE Conf. on Shape Modeling and Applications*, pages 157–166. 2004.

[17] C. Xu, J. Liu, and X. Tang. 2D shape matching by contour flexibility. *IEEE TPAMI*, 31(1):180–186, 2009.