



**HAL**  
open science

## 3D-Mesh Models: View-Based Indexing and Structural Analysis

Mohamed Daoudi, Tarik Filali Ansary, Julien Tierny, Jean-Philippe Vandeborre

► **To cite this version:**

Mohamed Daoudi, Tarik Filali Ansary, Julien Tierny, Jean-Philippe Vandeborre. 3D-Mesh Models: View-Based Indexing and Structural Analysis. First International DELOS Conference, Pisa, Italy, February 13-14, 2007, Revised Selected Papers, Feb 2007, Pisa, Italy. pp.298-307, 10.1007/978-3-540-77088-6\_29 . hal-00794698

**HAL Id: hal-00794698**

**<https://hal.science/hal-00794698>**

Submitted on 26 Feb 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# 3D-mesh models: view-based indexing and structural analysis

Mohamed Daoudi, Tarik Filali Ansary,  
Julien Tierny, and Jean-Philippe Vandeborre

TELECOM Lille 1 / LIFL / USTL (France)  
mohamed.daoudi@lifl.fr, tarik.filali@lifl.fr,  
julien.tierny@lifl.fr, jean-philippe.vandeborre@lifl.fr

**Abstract.** 3D-mesh models are widely used to represent real objects in synthesized scenes for multimedia or cultural heritage applications, medical or military simulations, video games and so on. Indexing and analyzing these 3D data is a key issue to enable an effective usage of 3D-mesh model for designers and even for final users.

The researches of our group mainly focus on these problems. In this paper, we present the work of our group for the DELOS NoE during the year 2006. We have worked on two approaches for 3D-model indexing and analyzing: view-based approach and structural approach. View-based approaches for 3D-model indexing are a very intuitive way to retrieve 3D-models among wide collections by using 2D natural views (as a human uses to represent 3D-objects). Structural analysis of a 3D-mesh model gives a structural decomposition of the object from a raw boundary representation of it, then this decomposition can be used to segment or index 3D-models.

**Keywords:** 3D-model, 3D-mesh, indexing, view based approach, Bayesian indexing, topological analysis, Reeb graph.

## 1 Introduction

Exploiting the information contents of digital collections poses several problems. In order to create added-value out of these collections, users need to find information that match certain expectations – a notoriously hard problem due to the inherent difficulties of describing visual information content.

In recent years, as a solution to these problems, many systems have been proposed that enable effective information retrieval from digital collections of images and videos. However, solutions proposed so far to support retrieval of images and videos – which are intrinsically bidimensional – are not always effective in application contexts where the information is intrinsically three-dimensional.

To satisfy this need, a variety of retrieval methods have been proposed that enable the efficient querying of model repositories for a desired 3D shape, many of which use a 3D-model as a query and attempt to retrieve models with matching shape from the database.

In our group, we have worked on two approaches: view based approach and structural approach. This paper presents, in the next two sections, these approaches, achieved work of our group for the DELOS NoE during the year 2006.

## 2 View-based Approach

The human visual system has an uncanny ability to recognize objects from single views, even when presented monocularly under a fixed viewing condition. The issue of whether 3D-object recognition should rely on internal representations that are inherently three-dimensional or on collections of two-dimensional views has been explored by Risenhuber and Poggio [1]. They show that, in a human vision system, a 3D-object is represented by a set of 2D-views.

The process of comparing two 3D-objects using their views can be separated into two main steps: 3D-model indexing (also known as the off-line step) and 3D-model retrieval (the on-line step). In the 3D-model indexing step, each 3D-object is characterized by a set of 2D views. For each view, a 2D descriptor is calculated. In the 3D-model retrieval step, the same descriptor is applied to the 2D request views. A matching process is then started to match the request 2D views to the views from the 3D-objects of the database.

### 2.1 Characteristic view selection

The main idea of view-based similarity methods is that two 3D-models are similar, if they look similar from all viewing angles. This paradigm leads to the implementation of query interfaces based on defining a query by one or more views, sketches, photos showing the query from different points of view.

A point which will have an impact on the effectiveness and the speed of the view based methods is the number and the position of views used to describe a 3D-object. This number is directly related to the performances of the system. It is necessary to decrease to the number of views to have a fast method, all in preserving the descriptive capacity of the views. Two approaches can be used to choose the number of characteristic views: using a fixed and restricted number of views; or using a dynamic number of views, that correspond the number of views to the geometrical complexity of views.

In the Adaptive Views Clustering (AVC) method, presented by our group [2], we use an initial set of 320 views to select positions for the views, which must be equally spaced, we use a two-unit icosahedron centered on the origin. We subdivide the icosahedron twice by using the Loop-subdivision schema to obtain a 320 faceted polyhedron. To generate the initial set of views, we place the camera on each of the face-centers of the polyhedron looking at the coordinate origin. To represent each of these 2D views, we use 49 coefficients of Zernike moment descriptor [3]. Finally, the initial set of views has to be reduced as we want from the method to adapt the number of characteristic views to the geometrical complexity of the 3D-model.

As every 2D view is represented by 49 Zernike moment coefficients, choosing a set of characteristic views that best characterise the 3D-models (320 views) is equivalent to choose a subset of points that represent a set of 320 points in a 49-dimension space. Choosing  $X$  characteristic views which best represent a set of  $N = 320$  views is well known as a *clustering problem*.

This kind of problem is generally solved by using *K-means* algorithm [4]. Its attractiveness lies in its simplicity and in its local-minimum convergence properties. However, it has one main shortcoming: the number of clusters  $K$ , no *a priori* known in our problem, has to be supplied by the user. To avoid this problem, we adapted a method derived from K-means – called X-means [5] –, and we also used a *Bayesian Information Criteria* (BIC) [6], which scores how likely the representation model fits the data. More details about this process can be found in [2].

## 2.2 Probabilistic approach for 3D-model indexing

The main idea of our probabilistic approach is that *not* all views of a 3D-model have the same importance. There are views which represent the 3D-model better than others. On the other hand, simple objects (e.g. cube, sphere) can be at the root of more complex objects, so they have a higher probability to be relevant. In this section, we present a probabilistic approach that takes into account that views do not have the same importance, and that simple objects have higher probability to appear than more complex one.

According to the previous step (characteristic view selection), each 3D-model of the collection is represented by a set of characteristic views  $V = \{V_1, V_2, \dots, V_C\}$ , with  $C$  the number of characteristic views. Moreover, to each characteristic view corresponds a set of represented views called  $V_r$ .

We want to find the 3D-models that corresponds to one or more request photos – which could also be the characteristic views of a 3D-model given as a query. We assume that in a query  $Q = \{I_1, I_2, \dots, I_K\}$  all  $K$  images represent the same object.

Considering a query  $Q$ , we wish to find the 3D-model  $M_i$  from the collection which is the closest to the query  $Q$ . This model is the one that has the highest probability  $P(M_i|Q)$ . A query is composed of one or more images, then this probability can be written:

$$P(M_i|Q) = \sum_{k=1}^K \frac{1}{K} P(M_i|I_k),$$

with  $K$  the number of images in the query  $Q$ .

The Bayesian retrieval framework we developed, takes into account the number of characteristic views of the 3D-models and the importance (amount of information) of their views. More details about our probabilistic approach can be found in [2] for a general explanation of our Bayesian framework, and [7] especially for 3D-model retrieval system with photo-oriented queries.

## 2.3 Experimental results

In this section, we present the experimental process and the results we obtained. The algorithms we described in the previous sections have been implemented using C++ and the TGS Open-Inventor libraries. The system consists of an off-line characteristic view extraction algorithm and an on-line retrieval process.

To experiment our algorithms, we also developed an on-line 3D search engine<sup>1</sup>. Our search engine can be reached from any device having compatible web browser (PC, PDA, SmartPhone, etc.). Depending on the web access device he/she is using, the user face two different kind of web interfaces : a rich web interface for full-featured web browsers, and a simpler interface for PDA web browsers. In both cases, the results returned by the 3D search engine are the same. The only difference lies in the design of the results presentation. Figure 1 shows three screenshots of our 3D retrieval system in its rich web interface version. Request could be expressed as a 3D-model, a 2D-sketch or one or more photos.

To evaluate our method, we used the *Princeton Shape Benchmark database*<sup>2</sup> (known as PSB), a standard shape benchmark widely used in shape retrieval community. PSB appeared in 2004 and is one of the most exhaustive benchmarks for 3D shape retrieval. It contains a database of 1814 classified 3D-models collected from 293 different Web domains. There are many classifications given to the objects in the database. During our experiments we used the finest granularity classification, composed of 161 classes. Most classes contain objects with a particular function (e.g cars). Yet, there are also cases where objects with the same function are partitioned in different classes based on their shapes (e.g, round tables versus rectangular tables). The mean number of views for the PSB is 23 views per model. The mean size for a 3D-model descriptor is 1,113 bytes.

Our method provides more accurate results with the use of Bayesian probabilistic indexing. The experiment shows that our method Adaptive Views Clustering (AVC) gives better performance than 3D harmonics, Radialized Spherical Extent Function, and Gaussian Euclidean Distance Transform on the *Princeton 3D Shape Benchmark database*. Light Field Descriptor gives better results than our method but uses 100 views, does not adapt the number of views to the geometrical complexity, and uses two descriptors for each view (Zernike moments and Fourier descriptor), which makes it a slower and more memory consuming compared to the method we have presented.

Overall, we conclude that AVC gives a good compromise between quality (relevance) and cost (memory and on-line comparison time) for the 3D-models of the *Princeton 3D Shape Benchmark*.

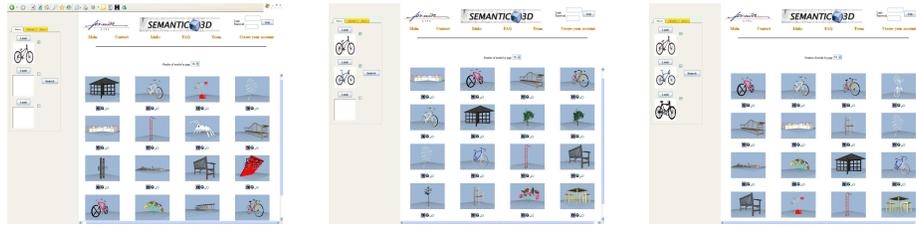
To evaluate the algorithms in the case of 3D retrieval by a set of photos, we have selected 50 images from the Internet. The images correspond to 10 classes of the *Princeton Shape Benchmark* (five images per class): Airplanes, Bicycles, Chairs, Dogs, Guns, Hammers, Humans arms out, Helicopters, Pots

---

<sup>1</sup> <http://www-rech.telecom-lille1.eu/3dretrieval/>

<sup>2</sup> <http://shape.cs.princeton.edu/benchmark/>

and Swords. The images are composed of six sketches, six synthesized images and 38 real photos of different sizes.



**Fig. 1.** 3D retrieval results using (from left to right) one, two and three photos.

Figure 1 shows the results of a query using (from left to right) one, two and three images of a bicycle. The left side of the figures represent the queries and the right side represent the 16 top retrieved 3D-models. From the seven 3D-models representing a bicycle in the database, three are in the fifteen top retrieved 3D-models when using only one photo. This number raises to four out of seven when two photos are used. Using three images, we retrieved five out of seven in the top sixteen retrieved 3D-models. The recall/precision curves obtained in [7], show that two or three photos makes a good compromise between time-effort and accuracy.

To extend his/her search, the user can click on a 3D-model from the result window. This 3D-model is then sent to the search engine and used as a query. In this case, the whole set of characteristic views of the 3D-model is considered as the set of query photos. So, the retrieval principle is exactly the same as the one described in the previous section.

At last, to give an idea of our framework performance, in the off-line process, the characteristic view selection takes about 18 seconds per model on a PC with a Pentium IV 2.4 GHZ CPU. In the on-line process, the comparison takes less than 1 second for 1814 3D-models from the PSB.

### 3 Structural Approaches

Structural indexing of 3D shapes is an interesting direction of research. It enables to retrieve similar 3D shapes that have different postures [8] and provides a promising framework for partial shape retrieval [9]. Our group focuses on the development of 3D mesh structural description algorithms based on topological skeletons [10].

These approaches study the properties of real valued functions computed over triangulated surfaces. Most of the time, those functions are provided by the application context, such as scientific data analysis. When dealing with topological skeletons, it is necessary to define an invariant and visually interesting

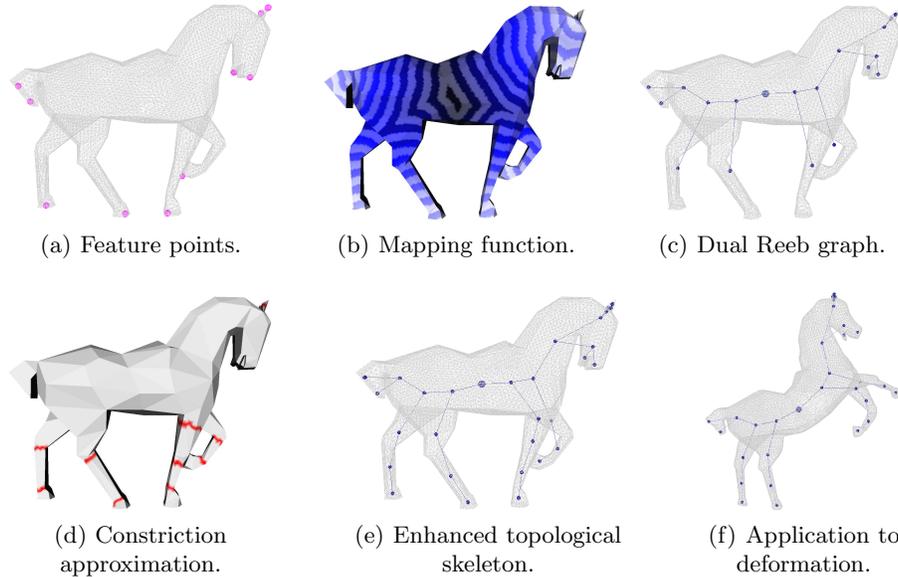
mapping function, which remains an open issue [10]. Moreover, traditional topological graph construction algorithms assume that all the information brought by the mapping function is pertinent, while in practice, this can lead to large graphs, encoding noisy details.

Finally, topological approaches cannot discriminate visually interesting sub-parts of identified connected components, like the phalanxes of a finger. This is detrimental to certain applications, such as mesh deformation.

Our group proposed a novel and unified method which addresses the above issues [11, 12].

### 3.1 Method overview

Given a connected triangulated surface  $T$ , we propose a unified method to decompose  $T$  into visually meaningful sub-parts, connected components, considering the topological and geometrical characteristics of *discrete contours*.



**Fig. 2.** Main steps of the decomposition on a standard model.

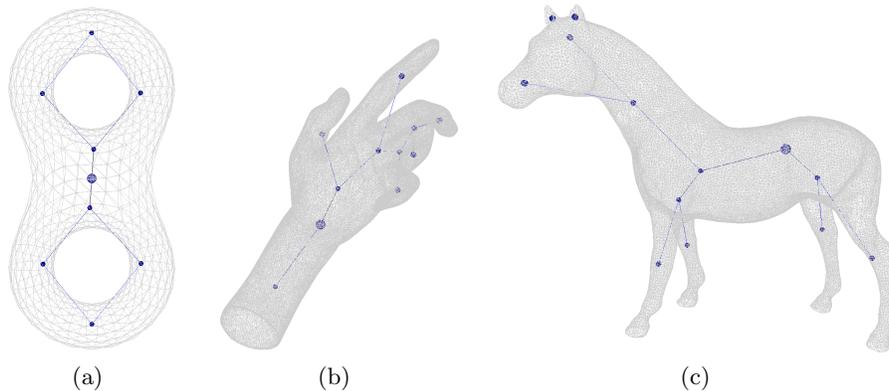
The algorithm proceeds in three stages. Firstly, mesh feature points are extracted (fig. 2(a)) in order to compute an invariant and visually interesting mapping function (fig. 2(b)), denoted  $f_m$  in the rest of the paper, revealing the most relevant parts of the mesh. Secondly, for each vertex in the mesh, we compute its *discrete contour*, a closed curve traversing it which is carried by the edges of  $T$ , and approximating  $f_m$  continuous contour. Thirdly, as the set of *discrete*

*contours* recovers the entire mesh, it is possible to analyze each contour characteristics, either to detect topological changes (enabling a pertinent Reeb graph construction, fig. 2(c)) or to detect geometrical transitions (enabling an approximation of constrictions, fig. 2(d)). The constriction approximation enables the refinement of Reeb graphs into enhanced topological skeletons (fig. 2(e)). The enhanced topological skeleton could then be used in different ways: 3D-mesh deformation (fig. 2(f)), indexing, etc.

Detailed methods and algorithm as well as extended results can be found in [11, 12].

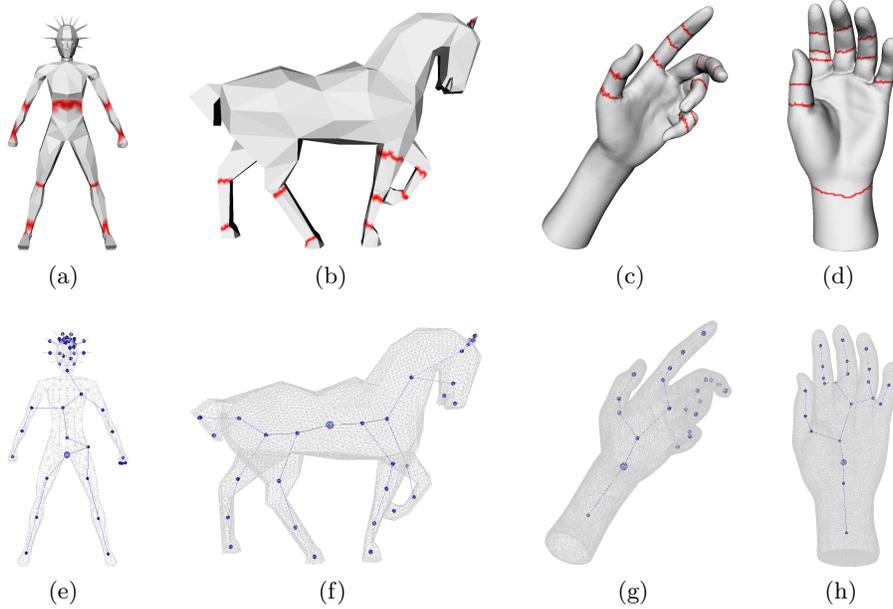
### 3.2 Results

Our scientific contribution resides in three points. (i) We propose an original and straightforward algorithm for feature point extraction. It can resolve extraction on constant curvature areas – such as spheres – and it is robust against variations in mesh sampling and model pose. (ii) We show that a *discrete contour* formulation enables, without re-meshing and without any input parameter, a pertinent Reeb graph construction providing visually meaningful graphs, affine-invariant and robust to variations in mesh sampling. (iii) We show that the geometrical information brought by *discrete contours* enables the approximation of constrictions on prominent components providing enhanced topological skeletons, and consequently Reeb graph refinement.



**Fig. 3.** Dual Reeb graphs of primitive and complex shapes ( $f_m$  function).

Figure 3 shows several dual Reeb graphs obtained with the very first steps of our method, without any refinement, with regard to  $f_m$ . Connected components are represented by the nodes located at their barycenter. Figure 4 shows several enhanced topological skeleton thanks to the constriction detection process. The



**Fig. 4.** Constriction approximations and enhanced topological skeletons of standard models.

enhanced topological skeleton is refined compared to classical Reeb graphs and shows more details in sub-parts.

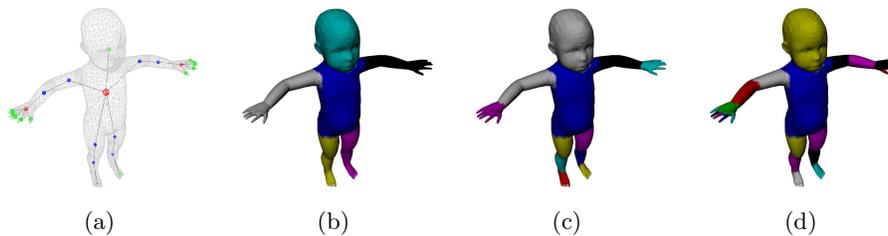
A video demonstration of this work is available in [http://www.telecom-lille1.eu/people/tierny/stuff/videos/tierny\\_pacific06.avi](http://www.telecom-lille1.eu/people/tierny/stuff/videos/tierny_pacific06.avi)

### 3.3 Applications of enhanced topological skeletons

Each node of the extracted skeleton references a surface patch on the mesh. This link between the surface and the skeleton enables applications like hierarchical semantic-oriented segmentation [13] or shape retrieval by part [14]. This is the next step of our work we are currently investigating.

Based on our previous work [11], we address the semantic-oriented 3D-mesh hierarchical segmentation problem using enhanced topological skeletons [13]. This high level information drives both the feature boundary computation as well as the feature hierarchy definition. Proposed hierarchical scheme is based on the key idea that the topology of a feature is a more important decomposition criterion than its geometry. First, the enhanced topological skeleton of the input triangulated surface is constructed. Then it is used to delimit the core of the object and to identify junction areas. This second step results in a fine segmentation of the object. Finally, a fine to coarse strategy enables a semantic-oriented hierarchical composition of features, subdividing human limbs into arms

and hands for example. An example of such a multi-level segmentation is shown in figure 5.



**Fig. 5.** Hierarchical mesh segmentation based on skeleton extraction.

Reeb graph based segmentation can also be employed for the problem of arbitrary genus surface comparison [14]. In this work, a novel surface parameterization based technique that addresses the pose insensitive shape signature problem for surface models of arbitrary genus, is presented. It is based on the key idea that two surface models are similar if the canonical mappings of their sub-parts introduce similar distortions. First, a Reeb graph of the shape is computed so as to segment it into charts of controlled topology, denoted as Reeb charts, that have either disk or annulus topology. Next, we define for each Reeb chart a straightforward mapping to the canonical planar domain. Then, we compute a stretching signature of the canonical mapping based on an area distortion evaluation. Finally, the input shape is represented by the set of the stretching signatures. An application to pose-insensitive shape similarity is proposed by comparing the signatures of the different Reeb charts. The gain provided by this new signature as well as its interest for partial shape similarity are demonstrated in [14].

## 4 Conclusion

This paper briefly presents two of the main research interests of our group: view-based 3D-model indexing and 3D-mesh structural analysis.

In the first topic, we proposed a entire framework for the indexing of 3D-models: a dynamic characteristic view selection process that adapts the number of characteristic view to the complexity of the 3D-model, and a Bayesian approach to 3D-model indexing based on the characteristic views. The framework is capable to manage 2D-view queries (one or more photos) and 3D-model queries.

In the second topic, we presented our method to topologically analyze 3D-mesh model and obtain a structural decomposition named *enhanced topological skeleton*. This method uses a novel approach to feature point extraction as well as mapping function computation. Moreover, the enhancement of Reeb graphs

is obtained with a constriction evaluation process. The resulting enhanced topological skeleton is a major tool for our future works on 3D-model segmentation and 3D partial shape retrieval that already give promising results.

## References

1. Riesenhuber, M., Poggio, T.: Models of object recognition. *Nature neuroscience* **3** (2000) 50–89
2. Filali Ansary, T., Daoudi, M., Vandeborre, J.P.: A bayesian search engine using adaptive views clustering. *IEEE Transactions on Multimedia* **9**(1) (2007)
3. Khotanzad, A., Hong, Y.H.: Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(5) (1990) 489–497
4. Duda, R.O., Hart, P.E.: *Pattern classification and scene analysis*. John Wiley and Sons (1973)
5. Pelleg, D., Moore, A.: X-means: Extending k-means with efficient estimation of the number of clusters. In: *International Conference on Machine Learning*. (2000) 727–734
6. Schwarz, G.: Estimating the dimension of a model. *The Annals of Statistics* **6** (1978) 461–464
7. Filali Ansary, T., Vandeborre, J.P., Daoudi, M.: On 3D retrieval from photos. In: *3rd IEEE International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'06)*, Chapel Hill, North Carolina, USA (2006)
8. Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.: Topology matching for fully automatic similarity estimation of 3D shapes. In: *SIGGRAPH*. (2001) 203–212
9. Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., Dobkin, D.: Modeling by example. *ACM Transactions on Graphics* **23** (2004) 652–663
10. Biasotti, S., Marini, S., Mortara, M., Patanè, G.: An overview on properties and efficacy of topological skeletons in shape modelling. In: *Shape Modeling International*. (2003) 245–254
11. Tierny, J., Vandeborre, J.P., Daoudi, M.: 3D mesh skeleton extraction using topological and geometrical analyses. In: *14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006)*, Taipei, Taiwan (2006) 85–94
12. Tierny, J., Vandeborre, J.P., Daoudi, M.: Invariant high-level reeb graphs of 3D polygonal meshes. In: *3rd IEEE International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT'06)*, Chapel Hill, North Carolina, USA (2006)
13. Tierny, J., Vandeborre, J.P., Daoudi, M.: Topology driven 3D mesh hierarchical segmentation. In: *IEEE International Conference on Shape Modeling and Applications (Shape Modeling International 2007)*, Lyon, France (2007) short paper.
14. Tierny, J., Vandeborre, J.P., Daoudi, M.: Reeb chart unfolding based 3D shape signatures. In: *Eurographics, Prague, Czech Republic (2007) short paper*.