



**HAL**  
open science

# A polygonal mesh partitioning algorithm based on protrusion conquest for perceptual 3D shape description

Sébastien Valette, Ioannis Kompatsiaris, Michael G Strintzis

## ► To cite this version:

Sébastien Valette, Ioannis Kompatsiaris, Michael G Strintzis. A polygonal mesh partitioning algorithm based on protrusion conquest for perceptual 3D shape description. SVE, 2005, Villars, Switzerland. hal-02272274

**HAL Id: hal-02272274**

**<https://hal.science/hal-02272274>**

Submitted on 27 Aug 2019

**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.

# A polygonal mesh partitioning algorithm based on protrusion conquest for perceptual 3D shape description

Sébastien Valette  
ITI, Thessaloniki, Greece  
valette@iti.gr  
www.iti.gr/~valette

Ioannis Kompatsiaris  
ITI, Thessaloniki, Greece  
ikom@iti.gr  
www.iti.gr/~ikom

Michael G. Strintzis  
ITI, Thessaloniki, Greece  
strintzi@eng.auth.gr

## Abstract

This paper introduces a novel partitioning algorithm for 3D polygonal meshes. The proposed approach is based on protrusion conquest which, for a given model, takes into account both the computed protrusion and the connectivity. The only constraint on the input mesh is that it must consist of one connected component. Our algorithm provides a good way to decompose the mesh into perceptually significant parts. The parts are further modeled by ellipsoids and a connectivity graph between them. This semantic representation is compliant to the perceptual shape description defined by the emergent standard MPEG-7.

**Keywords:** Part-based representation, protrusion, 3D meshes

## 1 Introduction

3D shape analysis is gaining more and more interest, with the growing number of 3D applications. Shape analysis can improve 3D processing from a low semantic level, such as indexation, to a higher semantic level such as shape recognition and classification. Furthermore, since shape is a key property of each ob-

ject, the advances in shape analysis can be particularly helpful for analyzing and understanding visual content. Following the recent advances in 2D multimedia analysis where domain knowledge, often defined by ontologies, is used to drive semantics extraction [1], a similar approach can be applied to 3D processing, as long as the appropriate concepts to account for the 3D space domain modeling are defined. In this paper, we focus on polygonal meshes, which is the most frequent representation for 3D shapes. The design of a mesh can be done with different approaches (3D scanners, hand-made) and as a result the produced meshes can have different properties. Hence, 3D mesh analysis approaches have to deal with meshes which can be non-closed, non-manifold, or even non-orientable. Most of the previously existing approaches assume that the input mesh is manifold, which is a severe restriction when considering the large number of models available on the internet that do not have this property. In this paper, we propose a robust polygonal mesh partitioning algorithm, able to process any 1-connected component polygonal mesh. Based on protrusion conquest, this scheme needs only one intuitive parameter which is robust, since all the experiments shown in this paper are done with the same pa-

parameter value. The paper is organised as follows: In Section 2 we describe some previous approaches for 3D mesh partitioning. In section 3, the protrusion computation for 3D polygonal meshes is explained. Section 4 explains how the protrusion was used by previous approaches, and why this way was not suitable for surface segmentation by itself. In Section 5 the protrusion conquest algorithm is described in details. Section 6 shows some experimental results obtained on a set of various meshes. Finally, section 7 concludes and gives some insight about future works.

## 2 Previous Works

The way human decomposes objects into parts is a complex process. According to Hoffman and Singh [2] minimal rule theory, three main factors are relevant for such a task: the relative sizes of the parts, their protrusion, and the strength of their boundaries. A lot of different works proposed solutions to decompose 3D meshes into visually significant parts.

Mangan and Whitaker [3] segment the regions with a watershed algorithm driven by local curvature. Some approaches followed the minima rule by identifying parts boundaries as concave regions (regions, with local negative curvature minima). Wu and Levine [4] simulated the repartition of 3D charges over the surface, to further localize parts boundaries in regions of local minimal charge distribution. Page et al. [5] also proposed a fast-watershed algorithm following the minima rule, in contrast with [3].

Kim et al. [6] convert the surface mesh into a volumetric representation. The volume is further decomposed using mathematical morphology.

Hilaga et al. [7] proposed a quantitative protrusion computing scheme for shape matching based on Multiresolution Reeb Graph. This protrusion criterion is used by Lin et al. [8] with boundary strength constraints to segment the mesh according to the minimal rule.

Katz and Tal. [9] decompose the mesh in a hierarchical approach, with the help of a fuzzy algorithm. Dey et al. [10] extract 2D and 3D features using flow discretization.

Note that a vast majority of previous works

process polygonal meshes assuming that they are 2-manifolds. Also, some of them can only process meshes without holes.

## 3 Protrusion Computation

In this paper, we approximate the local protrusion of 3D shapes as done in [7]. This approach is based on the computation of the criterion  $\mu(v)$ , defined as:

$$\mu(v) = \int_{p \in S} g(v, p) dS \quad (1)$$

for each vertex  $v$  of the mesh.  $g(v, p)$  is the geodesic distance between  $v$  and a point  $p$  on the surface  $S$ . An approximation of  $g(v, p)$  is computed with Dijkstra’s algorithm [11]. For a given vertex  $v$ ,  $\mu(v)$  will reflect its distance to all other points of the mesh; this is a usefull quantitative indication to know wether the vertex belongs to an ”extremity” of the mesh or to its ”center”. As proposed in [7], to reduce the computational cost of equation (1), we select a set of  $N$  base vertices  $b_i$  to compute the approximative protrusion defined by:

$$\mu(v) = \sum_i g(v, b_i) area(b_i) \quad (2)$$

where  $area(b_i)$  is the area of the mesh part associated to  $b_i$ . The base vertices selection is done using a uniform vertex clustering scheme similar to [12]. This clustering scheme distributes the base vertices uniformly over the surface to lower the error caused by such subsampling. For all the experiments shown in this paper, we chose  $N = 150$ . Figure 1 shows an example of clustering and protrusion computation for the dinosaur model.

Finally, for a given model, we compute the cube root of the computed protrusion (as equation (1) has dimension  $[l]^3$ ) and we normalize the result between 0 and 1. Computing the cube root has no effect on the protrusion conquest algorithm, but provides a good enhancement for the significant regions filtering step defined in section 5.3

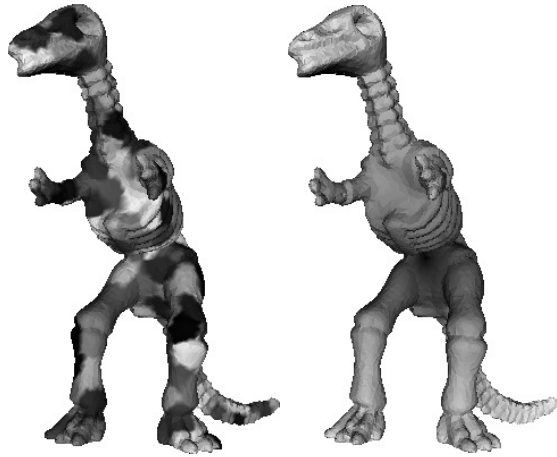


Figure 1: example of protrusion computing for the dinosaur model. Left: the results of the clustering scheme for the bases vertices selection. Right: the resulting protrusion (dark: low protrusion values, bright: high values)

#### 4 The protrusion quantization issue

Once the protrusion is computed, one can extract the different parts of the mesh. Most previous works using equation (1) such as [7] and [13] aim to index the 3D models for retrieval applications. These previous works used to quantize the protrusion values in a multiresolution approach, and create regions according to this quantization. From this clustering procedure, the authors create the multiresolution connectivity graph, the so-called Multiresolution Reeb Graph (MRG), which is further processed for 3D retrieval.

Unfortunately, such a construction is not well suited for 3D segmentation, because the computed quantization may produce regions non suitable for segmentation. Figure 2 shows two examples of protrusion quantization (8 levels) for the Dinosaur model, and for a Left Ventricle model. While the Left Ventricle is a geometrically simple model with a relatively uniform sampling, the constructed graph has a complex connectivity. Note that here, the graph construction is not theoretically correct, as for a genus 0 surface, no loops should occur in the graph. But we constrained the graph construction to use only the connectivity between vertices, to avoid problems with non manifold meshes.

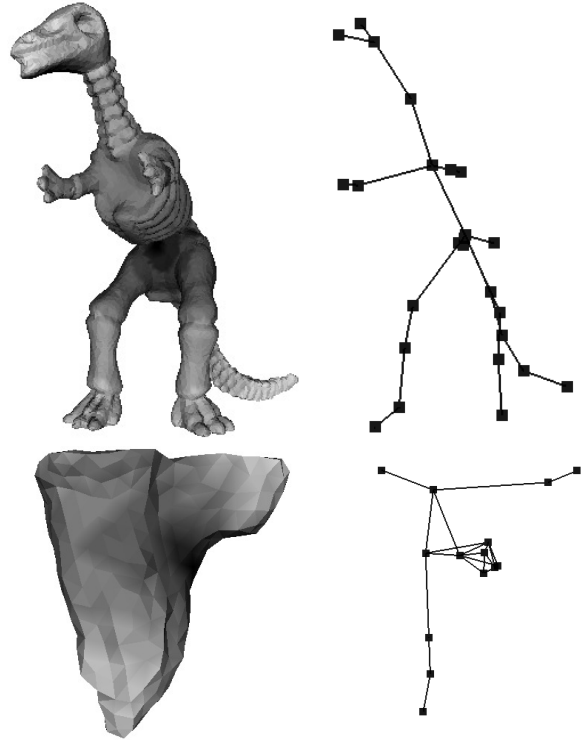


Figure 2: examples of protrusion quantization (left) and Graph construction (right): the case of the Dinosaur model (top) is simple, and its corresponding graph matches the visual aspect of the shape. On the other hand, the second model (bottom) has a simple shape, but its graph representation exhibits a complex connectivity which may be problematic for further processing

Moreover, the constructed mesh partition may not be faithful to human visual perception. As an example, the graph constructed from the Dinosaur model exhibits oversegmentation on the legs (several nodes for each leg) and the boundaries of the created regions do not match the boundaries that one human being would visually expect (see figure 3 for a close-up view of the dinosaur legs). This problem comes from the fact that the protrusion quantization is uniform. To solve this problem, one could try to adjust the quantization steps adaptively to the model characteristics, but this could be a hard task. In sharp contrast with previous works, we propose a new approach, the *protrusion conquest*. The novelty of this approach is that it avoids a quantization step, and the only tuning parameter is a very intuitive threshold.



Figure 3: close-up view of the Dinosaur legs segmentation (uniform quantization)

## 5 Protrusion Conquest

### 5.1 Outline

To solve the quantization issue, we propose an algorithm which segments the mesh more similarly to the smooth Reeb graph definition [14] than the discrete one. This algorithm is also related to the split tree construction defined in [15], which is an intermediate step to build contour trees. Contour trees have been defined for scalar fields and are related to Reeb graphs. Our approach is decomposed into three steps: protrusion conquest, selection of significant regions and splitting the center regions. These steps are detailed in the following sections. Figure 4 shows an example of our approach on a 2D synthetic object.

### 5.2 A priority and conquest algorithm

The main step of our approach is the protrusion conquest. This algorithm is a variation of Carr et al. split tree construction [15]. The split tree is constructed by analysing how the components of  $C$ , which is defined as

$$C = \{p \in S | \mu(p) \geq x\}, \quad (3)$$

evolve while decreasing the parameter  $x$ . In this paper,  $\mu(p)$  denotes the protrusion computed in section 3. Note that although this tree is called a split tree, while decreasing  $x$ , the components of  $C$  will actually merge. A tree node  $N_i$  is a point where several components merge for a given value  $\mu(N_i)$ . Note that here we associate a tree node to a geometrical vertex of the surface, assuming that several regions can simultaneously

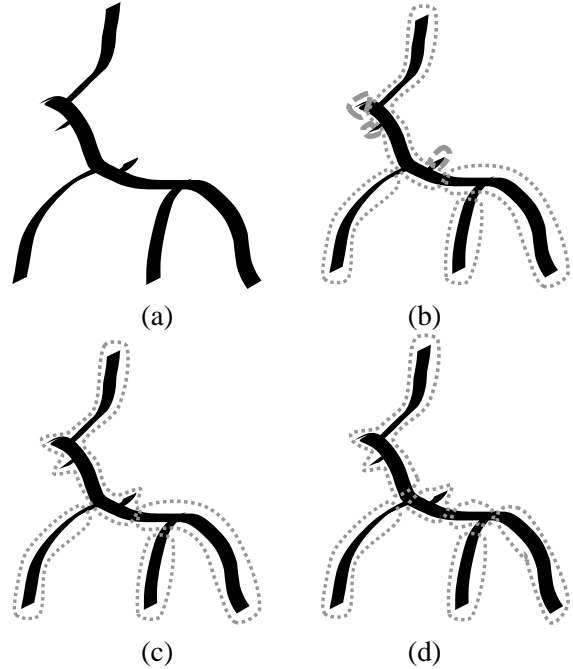


Figure 4: the algorithm on a 2D example: (a) original shape, (b) after protrusion conquest (7 regions), (c) after significance filtering (4 regions), (d) after center regions splitting (6 regions)

merge at only one vertex. This is not an issue, as the proposed algorithm is discrete and we process the vertices of the mesh one at a time. A tree edge  $J_\alpha^\beta$  corresponds to a connected part of the mesh, adjacent to  $N_\alpha$  and  $N_\beta$  for which every point  $p$  satisfies  $\mu(N_\alpha) \leq \mu(p) \leq \mu(N_\beta)$  (assuming that  $\mu(N_\alpha) \leq \mu(N_\beta)$ ). The surface can then be segmented according to the edges  $J_\alpha^\beta$ . Figure 5(a) shows such a segmentation on the horse model. Although the model is geometrically smooth, a lot of small regions are constructed. This is mainly due to the protrusion computation, which is only an approximation and exhibits a large number of local maxima.

We solve this issue by modifying the previous algorithm: while decreasing parameter  $x$  of equation (3), if several components meet at a given point  $p$ , we call  $n(p)$  the set of these regions. Instead of creating a new region and stopping the evolution of the set  $n(p)$ , we select the biggest region  $r_{big}$  of  $n(p)$  in terms of area, and stop the evolution of the other ones. The area of  $r_{big}$  will be increased by the sum of the other regions contained in  $n(p)$ . This provides a good means to merge the different created regions.

```

for all vertices  $v_i$  do
  push  $v_i$  into Queue with priority =  $\mu(v_i)$ 
end for
while Queue not empty do
  pop a vertex  $v$  from Queue
  if  $|n(v)| = 0$  then
    Create a new region  $r_{new}$ 
    Set  $Focus[r_{new}] = r_{new}$ 
    Set  $Class[v] = r_{new}$ 
  end if
  if  $|n(v)| = 1$  then
    Set  $Class[v] = focus(n(v))$ 
  end if
  if  $|n(v)| > 1$  then
    Find the region  $r_{big}$  in  $n(v)$  with the
    biggest area
    Set  $Class[v] = Focus[r_{big}]$ 
    for all regions  $r_j \in n(v)$  do
      Set  $Focus[r_j] = r_{big}$ 
    end for
    Update area of  $r_{big}$ 
  end if
end while

```

Table 1: protrusion conquest pseudo-code

Table 1 presents the pseudo-code for the protrusion conquest, assuming that the protrusion  $\mu$  has already been computed. *Queue* is a priority queue.  $n(v)$  is the set of regions adjacent to a vertex  $v$ .  $|n(v)|$  is the number of regions adjacent to  $v$ . As an example, if no direct neighbour of  $v$  is associated to a region, then  $n(v) = \emptyset$  and  $|n(v)| = 0$ .  $Class[]$  is an array which associates every vertex to a specific region (it is the output of the algorithm).  $Focus[]$  is an array storing the associations of the regions. It is useful to stop the growth of some regions and to let some others grow. Figure 5(b) shows the segmentation created with our algorithm. Note that there are still small regions due to the noise present in the protrusion, but the removal of these regions is easy, as explained in the following section.

### 5.3 Filtering out non significant regions

To keep only the significant segmented regions, we apply a simple filtering step, which is the only step needing an input parameter. Basically, we remove all the regions whom significance is lower than a certain percentage of the most sig-

nificant region. The significance is chosen according to the protrusion: For each created region  $R_k$ , we locate the vertices  $v_k^{max}$  and  $v_k^{min}$  having respectively the maximal and minimal protrusion value. Then for each region, we compute

$$\Delta_k = \mu(v_k^{max}) - \mu(v_k^{min}) \quad (4)$$

$\Delta_k$  gives a good estimation of the significance of the region  $R_k$ . The filtering consists in removing all the regions for which

$$\Delta_k < P_{Ratio} \cdot \Delta^{max} \quad (5)$$

where  $\Delta^{max}$  is the maximum among the values  $\Delta_k$ .  $P_{Ratio}$  is the input parameter. Note that for all the experiments shown in this paper (figure 6 excluded) we used  $P_{Ratio} = 10\%$ . After this cleaning step, the vertices belonging to deleted regions are no more associated to any region. We associate them to the nearest regions with a simple region growing algorithm. Figure 5(c) shows the results on the horse model after the filtering step.

### 5.4 splitting center regions

Since the protrusion conquest algorithm can be considered to be a merging algorithm for regions created with the split tree construction, some "extremities" of the mesh will be merged with "center" regions. As an example, for the horse model shown in figure 5(c), the rear left leg has been merged with the back of the horse, and the head and the neck has been merged with the horse body. We have to split these regions, in order to make the difference between "extremities" (the head+neck and the rear left leg) and "center" (body and back). The regions to be split are the regions with more than one neighbour region. To choose how to split them, we provide a simple approach: for a region  $R_k$  to be split, several vertices are boundary vertices (they are neighbours to other regions). Among these vertices, we pick the vertex  $v_k^{split}$  with maximal protrusion. Then all the vertices  $v$  in  $R_k$  satisfying

$$\mu(v) > \mu(v_k^{split}) \quad (6)$$

are associated to a new region. Figure 5 shows the segmentation after this final step for the horse model. Finally, the extremities of this model are well separated from the center regions.

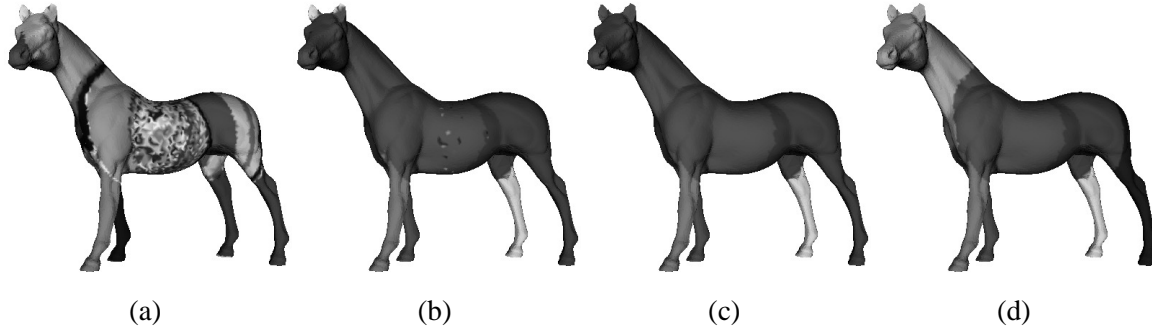


Figure 5: segmenting the horse model: (a) regions created with the split tree approach of Carr et al., (b) protrusion conquest, (c) after filtering out regions with low significance, (d) after the splitting step

### 5.5 Complexity

The algorithm complexity is in  $N \log(N)$ ,  $N$  being the number of vertices of the processed model. Most of the processing time is dedicated to the computation of the protrusion  $\mu(v)$ . As an example, our implementation segments a model with about 50k vertices in 40 seconds with an Intel Pentium III running at 1GHz.

## 6 Results

Figure 7 shows the results obtained on a set of various meshes, from synthetic ones (Octoflower, Mushroom) to models acquired with 3D scanners.

For each mesh, the left image corresponds to the segmentation, and the right image corresponds to a perceptual 3D shape description of the model, proposed in [16]. To compute this representation, we take each segmented region independently, and represent it by an ellipsoid, which features can easily be computed with the 3D covariance matrix of the region vertices. By providing the connectivity graph of the constructed regions, we build a description of the models which is compliant to the descriptors defined in [16], with only one difference: our description is surface-based while the initial descriptors are volume-based. This representation has also proven its efficiency for 3D retrieval applications [17].

Visually, the provided representations follow what we could expect from a 3D shape segmentation algorithm. Although this segmentation is not the most precise one, it is very robust, and can be applied to a wide range of polygo-

nal models, may they have holes, handles or non manifold vertices or edges.

Figure 6 shows 4 different ellipsoidal representations for the cow model. These results were obtained with 4 different values of  $P_{Ratio}$ : 0.04, 0.12, 0.2, 0.28. The number of regions (or ellipsoids) is respectively 16, 12, 10, 8. One can see that the main features of the model are kept, and while increasing  $P_{Ratio}$ , the small parts (horns, ears) disappear.

## 7 Conclusion and perspectives

We proposed in this paper a new segmentation algorithm for 3D polygonal meshes for perceptually significant parts representation. Our approach is driven by only one intuitive input parameter and can process polygonal meshes without any constraint on their topology.

Further works include the extension of this method for 3D models made of several disconnected components, by selectively connecting the components altogether.

We also explore the possibility of extending the presented approach to an ontology-driven 3D shape classification and parts recognition framework. The knowledge infrastructure would consider both kinds of information associated with visual content, namely the low-level features that can be automatically extracted and the underlying semantics. The ontology would formally define the syntax and semantics of the examined visual features and additionally associate them with the conceptualization of the 3D objects of interest. Enriching the knowledge base with appropriate prototypes for the mod-

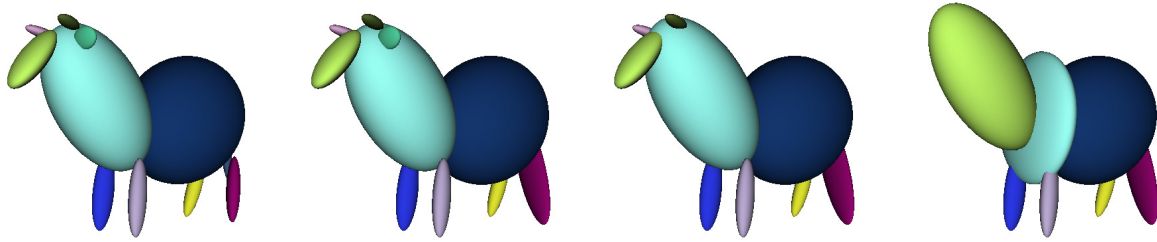


Figure 6: the cow model represented by ellipsoids. The values of  $P_{ratio}$  are respectively : 0.04, 0.12, 0.2, 0.28

eled concepts is adequate for extracting meaningful descriptions with respect to the examined domain.

## Acknowledgements

Part of this work was supported by the EU project SCHEMA “Network of Excellence in Content-Based Semantic Scene Analysis and Information Retrieval” (IST-2001-32795). The first author was founded by the Marie Curie Fellowship Association. The authors thank Stamatia Dasiopoulou and Frédéric Precioso for their help improving the paper. The authors also thank the anonymous reviewers for their valuable theoretical and practical comments.

## References

- [1] S. Dasiopoulou, V. K. Papastathis, V. Mezaris, I. Kompatsiaris, and M. G. Strintzis. An ontology framework for knowledge-assisted semantic video analysis and annotation. In *proceedings of 4th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2004) at the 3rd International Semantic Web Conference (ISWC 2004)*, pages 1–10, November 2004.
- [2] D. D. Hoffman and M. Singh. Saliency of visual parts. *Cognition*, 63(1):29–78, April 1997.
- [3] A. P. Mangan and R. T. Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):308–321, October/December 1999.
- [4] K. Wu and M.D. Levine. 3d part segmentation using simulated electrical charge distributions. *IEEE Transactions on PAMI*, 19:1223–1235, 1997.
- [5] D. L. Page, A. Koschan, and M. A. Abidi. Perception-based 3d triangle mesh segmentation using fast marching watersheds. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 27–32, June 2003.
- [6] D. H. Kim, I. D. Yun, and S. U. Lee. Shape decomposition scheme by combining mathematical morphology and convex partitioning. In *Proceedings of Fifth Asian Conference on Computer Vision*, pages 418–423, January 2002.
- [7] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Nii. Topology matching for fully automatic similarity estimation of 3d shapes. In *ACM SIGGRAPH conference proceedings*, pages 203–212, 2001.
- [8] H. Sean Lin, H. M. Liao, and J. Lin. Visual-saliency-guided mesh decomposition. In *Proceedings of IEEE Multimedia Signal Processing Workshop*, pages 331–334, October 2004.
- [9] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics (proceedings of SIGGRAPH 2003)*, 22(3):954–961, July 2003.
- [10] T. K. Dey, J. Giesen, and S. Goswami. Shape segmentation and matching with flow discretization. In *Proceedings*



of Workshop Algorithms Data Structures (WADS 03), pages 25–36, 2003.

- [11] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [12] S. Valette and J.-M. Chassery. Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. *Computer Graphics Forum (proceedings of Eurographics 2004)*, 23(3):381–389, September 2004.
- [13] T. Tung and F. Schmitt. Augmented reeb graphs for content-based retrieval of 3d mesh models. In *proceedings of International Conference on Shape Modeling and Applications (SMI'04)*, pages 157–166, June 2004.
- [14] G. Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendus de l'Académie des Sciences (Paris)*, 222:847–849, 1946.
- [15] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.
- [16] D. H. Kim, I. K. Park, I. D. Yun, and S. U. Lee. A new mpeg-7 standard: Perceptual 3-d shape descriptor. In *proceedings of 5th Pacific Rim Conference on Multimedia PCM(2)*, pages 238–245, 2004.
- [17] D. H. Kim, I. D. Yun, and S. U. Lee. A comparative study on attributed relational graph matching algorithms for perceptual 3d shape descriptor in mpeg-7. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 700–707, 2004.

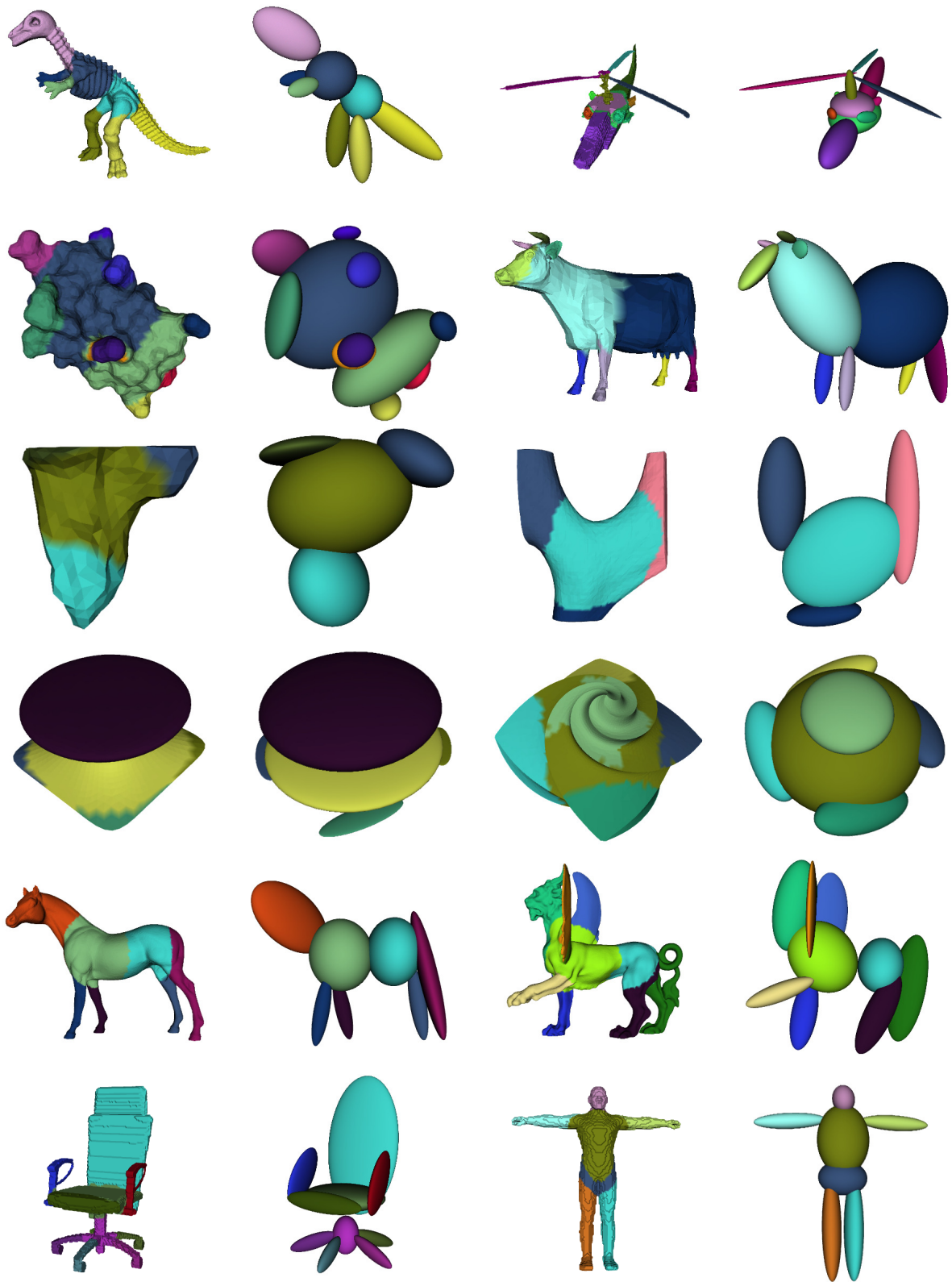


Figure 7: Results on a set of various meshes: segmentation and semantic representation with linked ellipsoids