Eurographics Workshop on 3D Object Retrieval (2012)
M. Spagnuolo, M. Bronstein, A. Bronstein, and A. Ferreira (Editors)

SHREC'12 Track: 3D mesh segmentation

G. Lavoué[1], J-P. Vandeborre[2], H. Benhabiles[3], M. Daoudi[2], K. Huebner[4], M. Mortara[5], M. Spagnuolo[5]

[1]Université de Lyon, CNRS, INSA-Lyon, LIRIS UMR 5205, France.
[2]LIFL (UMR Lille1/CNRS 8022), University of Lille 1, France.
[3]Le2i UMR 5158 CNRS, Université de Bourgogne, France.
[4]Computational Vision & Active Perception Lab, Royal Institute of Technology (KTH), Stockholm, Sweden.
[5]CNR-IMATI Genova, Italy.

Abstract

3D mesh segmentation is a fundamental process in many applications such as shape retrieval, compression, deformation, etc. The objective of this track is to evaluate the performance of recent segmentation methods using a ground-truth corpus and an accurate similarity metric. The ground-truth corpus is composed of 28 watertight models, grouped in five classes (animal, furniture, hand, human and bust) and each associated with 4 ground-truth segmentations done by human subjects. 3 research groups have participated to this track, the accuracy of their segmentation algorithms have been evaluated and compared with 4 other state-of-the-art methods.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling— I.2.10 [Artificial intelligence]: Vision and Scene Understanding—Shape

1. Introduction

Three-dimensional models are mostly represented as polygonal meshes; this kind of representation has the advantage of being perfectly adapted to 3D display with the help of modern 3D accelerated hardware. But the main drawback of this representation is the lack of a structure or a hierarchical description that could be very useful for many applications. Hence, the automatic segmentation of 3D mesh models is very often a necessary pre-processing tool for applications such as compression, texture mapping, animation and particularly shape retrieval. Mesh segmentation consists in subdividing a polygonal surface into patches of uniform properties either from a strictly geometrical point of view or from a perceptual / semantic point of view.

According to recent states-of-the-art [Sha08], mesh segmentation techniques can be classified into two categories: surface-type (or geometric) methods and part-type (or semantic) methods. In the first case, the algorithms are based on low level geometric information (e.g. curvature) in order to define segments (i.e. regions) with respect to geometric homogeneity, while in the latter case, the algorithms aim at distinguishing segments that correspond to relevant features of the shape, by following higher level notions such as defined in human perception theory. This kind of approach is particularly suited for object animation/ deformation and shape retrieval applications, where the decomposition has to be meaningful.

The recent creation of ground-truth databases for the segmentation of 3D meshes [CGF09, BVLD09, BVLD10], has given to the computer graphics community the opportunity to quantitatively evaluate the segmentation algorithms; however, in spite of the success of these benchmarks, only few results from a limited number of algorithms are currently publicly available whereas such quantitative performance data are crucial for the emulation of this field of research. In that context, the objective of this SHREC 2012 segmentation track is to provide researchers with the opportunity to directly compare their methods with their counterparts, offering the community a fair evaluation and comparison.

2. Data

For this track we use the grund-truth corpus from [BVLD10]; it contains 28 3D models (as triangle meshes) grouped in five classes, namely animal, furniture, hand, human and bust. Each 3D model of the corpus is associated with 4 manual segmentations which give a total of 112 ground-truth segmentations done by 36 volunteers. Figure 1 illustrates the models of the corpus with one manual seg-

Figure 1: *Models of our corpus associated with one ground-truth.*

mentation per model. All the selected models are manifold, connected, and do not have intersecting faces; hence they are supported as an input by any segmentation algorithm. The volunteers have freely segmented the models and no condition was imposed on the manner with which they have segmented them.

3. Evaluation methodology

For a given segmentation algorithm to evaluate, the evaluation protocol is the following:

1. The algorithm is launched on the 28 models of the database, thus producing 28 segmentations.
2. For each model, the segmentations S_a from the algorithm is compared to the 4 ground-truth segmentations S_k using a mesh segmentation similarity metric.
3. We obtain a segmentation quality score for each model; these 28 scores can then be averaged over all models, all categories or presented in increasing order in the form of a performance curve.

To provide a relevant evaluation of the performance, it is critical to choose an accurate mesh segmentation similarity metric (see step 2 above); several metrics have been proposed so far, they have been extensively studied and compared in [BVLD10] and the 3D-NPRI metric has shown to provide the best results (in term of correlation with the human opinion). This metric is derived from the Rand Index (RI) also used in [CGF09], it was introduced in [BVLD10] by transposing the 2D image version of Unnikrishnan et al. [UP07]. Let S_a be the automatic segmentation to be compared to a set of ground-truth segmentations S_k; we denote the corresponding label of a vertex i (label of the region which the vertex belongs to) by $l_i^{S_a}$ in segmentation S_a and by $l_i^{S_k}$ in a ground-truth segmentation S_k. The 3D Probabilistic Rand Index (3D-PRI) is defined as:

$$3DPRI(S_a,\{S_k\}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i<j}} e_{ij}p_{ij} + (1-e_{ij})(1-p_{ij}) \tag{1}$$

where e_{ij} is a binary number which denotes the event of a pair of vertices i and j belonging to the same region in the automatic segmentation:

$$e_{ij} = \mathbf{I}(l_i^{S_a} = l_j^{S_a}) \tag{2}$$

and p_{ij} is the probability of the vertices i and j belonging to the same region in the ground-truth segmentations $\{S_k\}$:

$$p_{ij} = \frac{1}{K}\sum_k \mathbf{I}(l_i^{S_k} = l_j^{S_k}) \tag{3}$$

This 3D-PRI takes values ranged in [0,1], where 0 indicates no similarity between S_a and S_k, and 1 indicates a perfect match. To provide more meaningful results and increase the dynamic range of the metric, the authors [BVLD10] provided also a version normalized by a baseline of random segmentations: the 3D-NPRI, which takes values in [-1,1] with 0 standing for an average similarity as a random segmentation could produce and 1 for a perfect similarity.

4. Methods

Three segmentation methods have been registered to the track:

- *Box Approximation and Decomposition* [Hue12] from the Computer Vision & Active Perception Lab, KTH, Stockholm, Sweden.
- *Plumber* [MPS*04] from the CNR-IMATI, Genova, Italy.
- *Boundary Learning* [BLVD11] from LIFL, Université de Lille and LIRIS, Université de Lyon, France.

These algorithms are detailed in the next subsection. We have also launched four algorithms from the state of the art on our dataset to strengthen the results. They are quickly described in subsection 4.2.

4.1. Registered methods

4.1.1. Box Approximation and Decomposition

This method [Hue12] is proposed by Kai Huebner from the Computer Vision & Active Perception Lab, KTH,

Stockholm, Sweden.

The aim of this segmentation algorithm is to iteratively split an oriented bounding box (starting from one root box) in such a way that the new point sets yield a better box approximation of the shape. Iterative splitting of a root box corresponds to the build-up of a hierarchy of non-axis aligned minimum volume bounding boxes (MVBBs). As the motivation of the algorithm originates from our research in robot grasping of arbitrary object, we want to conveniently approximate a shape with as few boxes as possible. Though the internal fitting algorithm [BHP01] is efficient, a fitting step after each splitting consumes valuable computation time. A heuristic to find a *good* split of the point cloud is needed. In our case, we define a *good* split by consulting the relation of the box volume before and after performing a split. A split of the parent box is the better, the less volume the two child MVBBs include. Intuitively, this is clear, as shape approximation is better with highly tight-fitting boxes. To efficiently split the point set in a box, we project the points to the 6 box surfaces and compare the best 2D splits. According to the best split, the original point cloud is divided into two subsets of the data points. Using these as new inputs to the fitting algorithm represents the complete iterative fit-and-split technique. As an iteration breaking criterion, we subsequently test the real 3D volume gain θ of the resulting best 2D split. Therefore, we compute the gain in volume defining:

$$\theta = (volume(C_1) + volume(C_2) + volume(A^{\setminus P}))/volume(A) \quad (4)$$

where A is the overall set of boxes in the current hierarchy, P is the current (parent) box, C_1,C_2 are the two child boxes that will be produced by the selected split, and volume being a volume function. θ is the main parameter of the algorithm, controlling depth of the splitting hierarchy and granularity of the produced segments. As mentioned, the algorithm has been developed for decomposition of object models into intuitive parts for generating robot grasp hypotheses [Hue12], i.e. to be robust towards noise, outliers, and incomplete models as they appear in real sensor settings. In this context it also has to mentioned that the algorithm takes only the vertices into consideration, but does not need any polygonal surface representation. Also, a random subset selection may result in different decomposition when re-processing a model. Though grasp evaluation was evaluated by the authors, the segmentation result itself has never been evaluated before. Minor changes had to be made to let the original open source code (available at `http://www.csc.kth.se/~khubner/badgr/`) to provide the output data needed for the SHREC 2012 dataset. Some results with boxes are shown in figure 2.

4.1.2. Plumber

This method [MPS*04] was registered for the track by Michela Mortara and Michela Spagnuolo from the CNR-IMATI Genova, Italy.

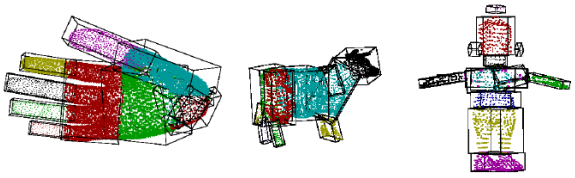

Figure 2: *Some examples of segmentations / box decompositions from [Hue12].*

Plumber is a specialized shape segmentation method for detecting tubular features of 3D objects represented by triangle meshes. The *Plumber* algorithm segments a surface into connected components that are either body parts or elongated features, that is, handle-like and protrusion-like features, together with their concave counterparts, i.e. narrow tunnels and wells (more details in [MPS*04]). Thus, *Plumber* identifies primitives of the object with a specific structure, i.e. generalized cones and cylinders, and not only related to a curvature and concavity analysis [MPS*03, KT03]. The segmentation can be done at single or multi-scale, and produces a shape graph which codes how the tubular components are attached to the main body parts.

Intuitively, tubes are identified by parts of the shape whose intersection with a sphere of appropriate radius produces two intersection curves. The section of the tube and its axis can be arbitrarily shaped; however, chosen a level of detail R_i, tubes of diameter R_i or smaller will be identified. Small radii determine details, while bigger ones are used to analyse the global characteristics of the model.

Chosen a level of detail R_i, *Plumber* performs the following steps:

1. identify *limb-regions* composed by vertices generating two intersection curves by the sphere centred on them, and check that the limb-regions have two boundaries on the object surface (so that the same topology of a cylinder is guaranteed);
2. shrink each of the two selected boundary components along the surface to its medial-loop, whose points are nearly equidistant from the two border loops;
3. expand-back the medial-loop by sweeping the extent of the shape in both directions. More precisely, at each iteration we place a sphere of radius R in the barycentre of the new medial loops. If the intersection between the sphere and the surface generates two loops, mesh vertices inside the sphere are marked as visited;
4. the procedure is iterated in both directions until:
   - no more loops are found, or more than one loop is found on not-visited regions;
   - the new loop lies on triangles that are already part of another tube, or the length of the new loop exceeds a pre-defined threshold.

5. the tube skeleton may be extracted by joining the loops' barycentres.

For this track, we have run *Plumber* at 8 increasing radii, determined by the the average edge length over the input mesh multiplied by even factors from 2 to 16.

For some models, the sampling density on the radius values was not sufficiently fine to capture all the tubular features, or the maximum radius value was not big enough to detect very thick tubes. Moreover, thick tubes need large spheres to be detected, and the corresponding tube growing step makes intersections with other tubes frequently occurr, which makes the sweeping stop prematurely (this is the case for the Homer legs), whereas the method performs very well on thin and elongated features even in complex configurations (e.g. the furniture models). Being targeted to tubular shaped primitives, the method is not suitable for certain shape classes without any elongated features, like the busts.

4.1.3. Boundary Learning

This method [BLVD11] is proposed by Halim Benhabiles, Guillaume Lavoué, Jean-Philippe Vandeborre and Mohamed Daoudi from LIFL, Université de Lille and LIRIS, Université de Lyon, France.

The algorithm is carried out using two main steps: the offline step in which an objective boundary function is learned using a set of manually segmented models (ground-truths), and the on-line step in which the learned function is used to segment the input mesh. The problem of learning the boundary edge function is formulated as a classification problem, resolved with the Adaboost classifier. This classifier takes as input a training dataset (we used the Princeton segmentation benchmark [CGF09] as training set) and generates the boundary edge function. The training dataset is composed of a set of feature vectors F_e computed for each edge of the ground-truth meshes. A feature vector F_e of a given edge contains a set of geometric criteria and is associated with its proper class label L so that $L = +1$ if the edge is a boundary (according to the manual segmentations of the mesh containing this edge) and $L = -1$ if the edge is not a boundary. Once the learning is done, the classifier produces the boundary edge function. This function is a weighted combination of the set of geometric criteria. It takes as input a feature vector from any given edge and outputs a signed scalar value whose sign will provide the estimated classification of the edge. The use of the boundary edge function on a given mesh leads to produce a set of interest regions (see figure 3.a), each of which is represented by a set of connected edges of the mesh. Hence it is not possible to directly consider these regions as the final segment boundaries of the mesh. To overcome this problem we propose a processing pipeline that transforms these non-connected fuzzy regions into thin, closed and smooth contours, by using the edge function. This processing pipeline comprises three stages. In the first stage of the process, for each interest region, a thinning algorithm [HG01] is applied. This latter algorithm gives as output a set of open linear contours (figure 3.b). Next, each open contour is completed using an improved version of the algorithm proposed by Lee et al. [LLS*05] using the edge function (figure 3.c). At this step we have created a set of closed contours which represent a first version of the segmentation boundaries. However, these boundaries are often not smooth nor precise since in the thinning stage we do not consider any geometric information. To overcome this drawback, we apply an improved version of the snake movement algorithm proposed in [JK04] based also on the learned edge function. The snake movement allows to improve the quality of the boundaries in term of smoothness and precision without changing the mesh connectivity (figure 3.d). This set of improved boundaries defines the final segmentation (figure 3.e).

4.2. State-of-the-art methods

To strengthen the evaluation we have also launched the following algorithms on the dataset, using authors' implementations:

- *Curvature Classification*. Lavoué et al. [LDB05] propose a segmentation algorithm based on curvature classification and then region growing and merging. It extracts connected regions associated with similar curvature. This algorithm is especially suited for CAD models.
- *Fitting Primitives*. Attene et al. [AF06] base their algorithm on an iterative hierarchical face clustering. Starting with one cluster per face, clusters are then iteratively merged into larger patches that best fit some predefined primitives like plane, sphere or cylinders.
- *Topology Driven*. Tierny et al. [TVD07] have proposed a hierarchical segmentation mostly based on the topology of the model. They first extract an enhanced topological skeleton using Reeb graph and constriction detection. Then this skeleton is used to extract the core of the object and to identify the junction areas. The result of this operation is a fine segmentation which can then be simplified by merging the nodes of the skeleton.
- *Shape Diameter*. Shapira et al. [SSCO08] compute a highly relevant scalar field over the mesh vertices: the *Shape Diameter Function*. Their algorithm classifies this scalar field into several clusters and then uses graph-cut to provide a smooth partition of the 3D mesh.

5. Evaluation results

Tables 1 and 2 present respectively the 3D-NPRI values and ranking for all methods, averaged for each category and for the whole corpus; several remarks can be drawn regarding these results:

- The best method is the Boundary Learning one [BLVD11]. This appears logical since this algorithm has

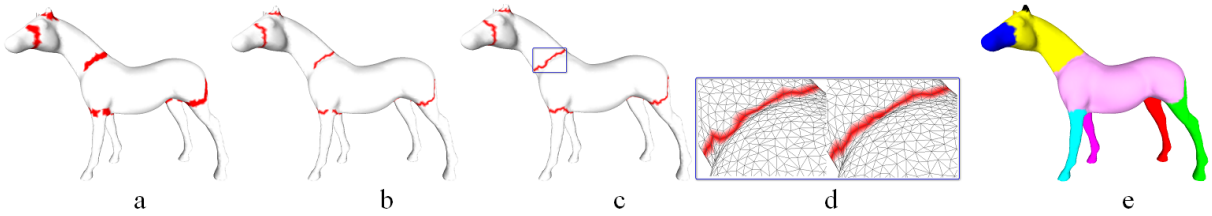

Figure 3: *Post processing pipeline; interest regions (a), thinning (b), contour completion (c), snake movement (d), final segmentation (e).*

Table 1: *3D-NPRI results for all tested methods.*

Method Name	Animal	Bust	Furniture	Hand	Human	Global Rank
Box Approximation and Decomposition [Hue12]	0.52	-0.08	0.08	-0.09	0.37	0.16
Plumber [MPS*04]	0.36	0.00	0.54	0.27	0.33	0.30
Boundary Learning [BLVD11]	**0.68**	**0.41**	0.79	0.68	**0.69**	**0.65**
Curvature Classification [LDB05]	0.43	0.10	0.38	0.45	0.29	0.33
Fitting Primitives [AF06]	0.45	0.09	0.56	0.52	0.61	0.45
Topology Driven [TVD07]	0.51	-0.07	0.36	**0.78**	0.50	0.41
Shape Diameter [SSCO08]	0.62	0.24	**0.85**	0.19	0.66	0.51

learned how people segment 3D models and thus produces results very close to the ground-truth; moreover it integrates several different features (including curvature, shape diameter function, dihedral angles, etc.) whereas most of other methods rely on only one type of feature. However we can notice that the method relying only on the Shape Diameter function [SSCO08] still produces very good results, which attests the power of this feature.

- We can notice that the peformance of the methods is highly class-dependent, indeed no method owns the same rank for all classes. The performance actually depends on the features used by each method. For instance, the Plumber algorithm [MPS*04] detects tubular parts, hence it provides good results for the *furniture* class; on this latter class, the Box Approximation and Decomposition approach [Hue12] behaves poorly since chairs or tables are not adapted to such decomposition (see figure 4).
- Certain classes are easier to segment than others; for instance all methods produce good results on the *animal* class (see figure 5), whereas the *bust* class seems much more difficult to segment by any automatic method (see figure 6). This is probably due to the high semantic aspect carried by a face, this semantic aspect influences the manual segmentations, and is difficult to capture using simple local geometric criteria.

Figure 7 represents 3D-NPRI indices for each model of the corpus, these indices are plotted in increasing order for each algorithm, hence i^{th} index does not refer to the same 3D model for each algorithm. These curves provide a good overview of the performances of the algorithms among the models of the corpus.

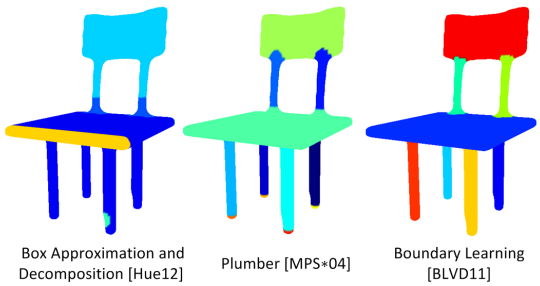

Figure 4: *Segmentation results for one object from the furniture class.*

6. Conclusions

In this paper, we have presented the database, evaluation protocol, involved methods and results of the 3D mesh segmentation track of the SHREC 2012 contest. The Boundary Learning method has logically demonstrated superior results; however there is still room for improvements since some classes remain difficult to segment.
All results from this track are available through our online 3D Mesh segmentation benchmark `http://www-rech.telecom-lille1.eu/3dsegbenchmark/` which allows an online evaluation of segmentation algorithms.

Acknowledgements

We would like to thank Marco Attene, Ariel Shamir, Shy Shalom, and Julien Tierny for providing us with their source code or the binary of their segmentation algorithms.

Table 2: *3D-NPRI ranking for all tested methods.*

Method Name	Animal	Bust	Furniture	Hand	Human	Global Mean
Box Approximation and Decomposition [Hue12]	3	5	7	7	5	7
Plumber [MPS*04]	7	5	4	5	6	6
Boundary Learning [BLVD11]	1	1	2	2	1	1
Curvature Classification [LDB05]	6	3	5	4	7	5
Fitting Primitives [AF06]	5	4	3	3	3	3
Topology Driven [TVD07]	4	5	6	1	4	4
Shape Diameter [SSCO08]	2	2	1	6	2	2

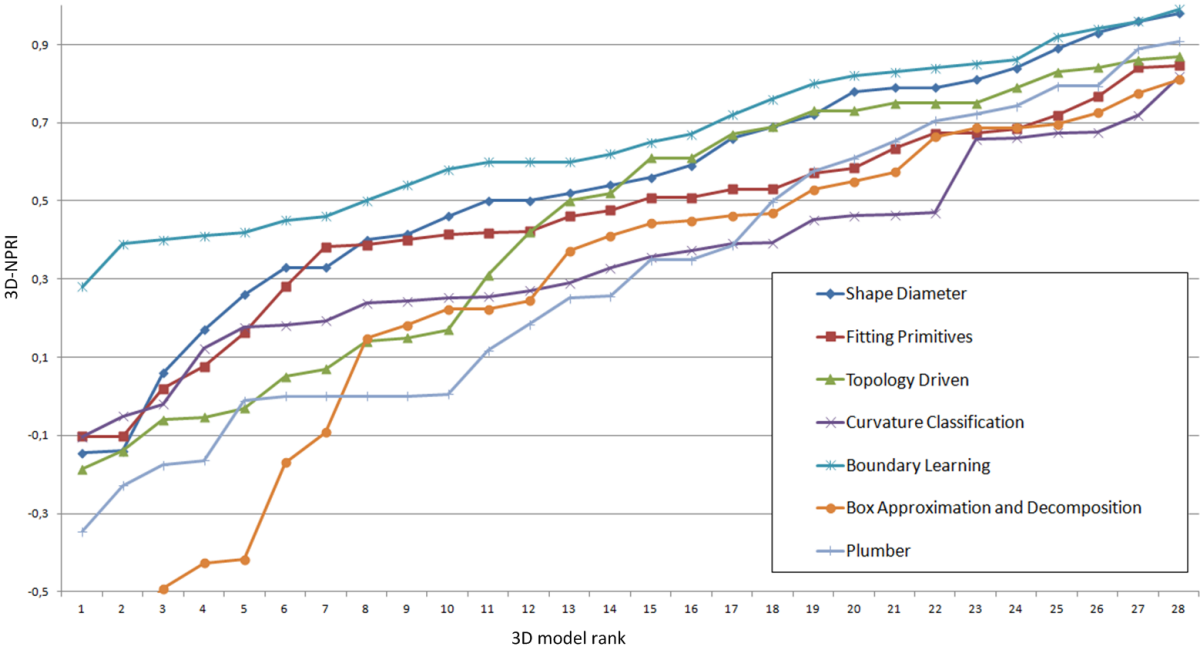

Figure 7: *3D-NPRI indices of the 28 models sorted in increasing order for all algorithms.*

References

[AF06] ATTENE M., FALCIDIENO B.: Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer* (2006). URL: http://www.springerlink.com/index/B3J373G864395712.pdf. 4, 5, 6

[BHP01] BAREQUET G., HAR-PELED S.: Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *Journal of Algorithms 38*, 1 (2001), 91–109. URL: http://dl.acm.org/citation.cfm?id=314536. 3

[BLVD11] BENHABILES H., LAVOUÉ G., VANDEBORRE J.-P., DAOUDI M.: Learning Boundary Edges for 3D-Mesh Segmentation. *Computer Graphics Forum 30*, 8 (2011), 2170–2182. URL: http://doi.wiley.com/10.1111/j.1467-8659.2011.01967.x, doi:10.1111/j.1467-8659.2011.01967.x. 2, 4, 5, 6

[BVLD09] BENHABILES H., VANDEBORRE J.-P., LAVOUÉ G., DAOUDI M.: A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D-models. In *Shape Modeling and Applications, 2009. SMI 2009. IEEE International Conference on* (2009), IEEE, pp. 36–43. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5170161. 1

[BVLD10] BENHABILES H., VANDEBORRE J.-P., LAVOUÉ G., DAOUDI M.: A comparative study of existing metrics for 3D-mesh segmentation evaluation. *The Visual Computer 26*, 12 (Apr. 2010), 1451–1466. URL: http://www.springerlink.com/content/0000178472145172/, doi:10.1007/s00371-010-0494-2. 1, 2

[CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (TOG) 28*, 3 (2009), 73. URL: http://dl.acm.org/citation.cfm?id=1531379. 1, 2, 4

[HG01] HUBELI A., GROSS M.: Multiresolution feature extraction for unstructured meshes. In *Visualization, 2001. VIS'01. Proceedings* (2001), IEEE, pp. 287–294. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=964523. 4

Figure 5: *Segmentation results for some objects from the animal class.*

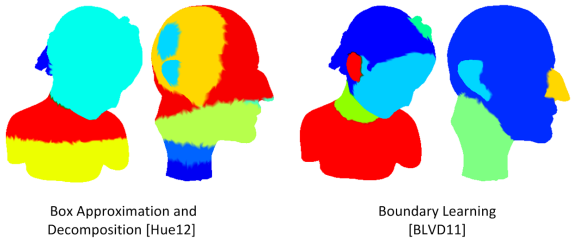

Figure 6: *Segmentation results for two objects from the bust class. Results from the Plumber method are not presented since this algorithm creates only one region for the objects of this class.*

[Hue12] HUEBNER K.: BADGr - A Toolbox for Box-based Approximation, Decomposition and GRasping. *Robotics and Autonomous Systems 60*, 3 (2012), 367–376. 2, 3, 5, 6

[JK04] JUNG M., KIM; H.:. In *12th Pacific Conference on Computer Graphics and Applications* (2004), IEEE, pp. 87–93. doi:10.1109/PCCGA.2004.1348338. 4

[KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph. 22*, 3 (2003), 954–961. 3

[LDB05] LAVOUÉ G., DUPONT F., BASKURT A.: A new CAD mesh segmentation method, based on curvature tensor analysis. *Computer-Aided Design 37*, 10 (Sept. 2005), 975–987. URL: http://dx.doi.org/10.1016/j.cad.2004.09.001, doi:10.1016/j.cad.2004.09.001. 4, 5, 6

[LLS*05] LEE Y., LEE S., SHAMIR A., COHEN-OR D., SEIDEL H.: Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design 22*, 5 (2005), 444–465. URL: http://www.sciencedirect.com/science/article/pii/S0167839605000427. 4

[MPS*03] MORTARA M., PATANÈ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica 38*, 1 (2003), 227–248. 3

[MPS*04] MORTARA M., PATANÉ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Plumber: a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies. In *Proceedings of the ninth ACM symposium on Solid modeling and applications* (2004), Eurographics Association, pp. 339–344. URL: http://dl.acm.org/citation.cfm?id=1217929. 2, 3, 5, 6

[Sha08] SHAMIR A.: A survey on Mesh Segmentation Techniques. *Computer Graphics Forum 27*, 6 (Sept. 2008), 1539–1556. URL: http://doi.wiley.com/10.1111/j.1467-8659.2007.01103.x, doi:10.1111/j.1467-8659.2007.01103.x. 1

[SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer 24*, 4 (Jan. 2008), 249–259. URL: http://www.springerlink.com/index/10.1007/s00371-007-0197-5, doi:10.1007/s00371-007-0197-5. 4, 5, 6

[TVD07] TIERNY J., VANDEBORRE J., DAOUDI M.: Topology driven 3D mesh hierarchical segmentation. In *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on* (2007), IEEE, pp. 215–220. URL: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4273384. 4, 5, 6

[UP07] UNNIKRISHNAN R., PANTOFARU C.: Toward objective evaluation of image segmentation algorithms. *Pattern Analysis and 29*, 6 (June 2007), 929–44. URL: http://www.ncbi.nlm.nih.gov/pubmed/17431294, doi:10.1109/TPAMI.2007.1046. 2