

Review of 3D Objects Segmentation Methods

Maciej Wencka and Krzysztof Walas^(✉)

Institute of Control and Information Engineering,
Poznan University of Technology, Poznań, Poland
maciej.wencka@gmail.com, krzysztof.walas@put.poznan.pl
<http://www.cie.put.poznan.pl/>

Abstract. This paper presents a review of segmentation methods of basic shapes represented by polygonal meshes. For a fair algorithms comparison, common training data was used. In this work, 11 methods of 3D Mesh segmentation were tested using four different measures of segments similarity. Namely, Cut Discrepancy, Hamming Distance, Rand Index, Consistency Error were used. All measures mentioned above were characterised in the paper. The results of the comparisons provide means of understanding strengths and weaknesses of the tested algorithms and provide the foundation for the further developments of 3D Objects segmentation methods.

Keywords: Shape characteristics · Machine learning · 3D models · Object segmentation

1 Introduction

The main goal of this paper is to provide the reader with an extensive comparison of existing methods of 3D objects segmentation. Such algorithms are intensively developed in computer graphics community, where the segmentation might be useful for classification purposes but also it could provide some semantic meaning to the object. The robotic systems could also exploit this information. For example, if one is able to segment a handle of a mug, the robot could use this information to perform a successful grasp. The proliferation of the methods presented in this paper to the robotics community will be beneficial, as it will ease the development of semantic robotics. Most of the methods presented in this review are based on machine learning (ML) techniques making 3D object segmentation a very promising field of application for this group of algorithms. The main contribution of this paper is an extensive comparison of existing methods of 3D objects segmentation. The following section of the paper describes a dataset. Then, the overview of used methods is given, together with a short presentation of each algorithm. Next, the description of the measures used for comparing the algorithms is provided. Subsequently, the results are presented and discussed. Finally, concluding remarks are given.

2 Materials and Methods

2.1 Dataset Used for Testing

We will start our presentation of the dataset used for algorithms comparison from the short description of 3D shapes representations. Mainly, there are three possibilities: voxels, point clouds and meshes. The first one is primarily used in robotics, and its main advantage is a possibility of representing a volume of an object. The second one only provides the information about the points on a surface of the object, e.g. from object scanning. However, the information about the topology of the surfaces is lost, no connectivity data available. Hence, two points close to each other in a Euclidean space are always close to each other in this representation, when in reality they belong to two different surfaces. The third representation addresses such situation. The use of a mesh allows us to represent inner and outer part of the object and provide the means of representing a volume of watertight objects implicitly. Thus, it provides advantages of both previous representations; we are able to represent real objects scanned with the appropriate device (Point Clouds) and to some extent account for the volume of the object (Voxel Representation).

Taking into account the properties of meshes we chose it as a most convenient representation for our tests and selected a dataset which provides such data. Namely, it is Watertight Track of SHREC 2007 [1]. It contains a large number of categories (20). Additionally, the dataset provides segmentation file. Line number in this file represents the vertex number in the object file. The information given in each line is a segment number which this vertex belongs to.

2.2 Tested Methods

Having the dataset 11 methods of 3D objects segmentation were tested. In the following paragraphs, each of them is shortly described.

- Learning 3D Mesh Segmentation and Labelling (LMS) [5] – for segmentation Conditional Random Fields (CRF) with conditions assessing conformity of neighbouring surfaces constituting an object were used. LMS is supervised learning algorithm. It uses already prepared templates with appropriate geometrical features.
- Unsupervised 3D Shape Segmentation and Co-Segmentation via Deep Learning (Deep Learning) [11] – this algorithm initially divides the objects into segments formed of the surfaces of the object. From each segment, the information about its centre of gravity is found. Subsequently, using Shape Diameter Function, which measures the area diameter, being in a neighbourhood of this centre point and an average geodesic distance to other centres, most similar surfaces are found.
- Mesh Segmentation with Concavity-aware Fields (Isoline Cuts) [7] – in this algorithm, it was assumed that the highest probability of segment borders is assigned to concave parts of the objects. Such borders are found through solving a Laplacian which is sensitive to such areas.

- Spectral 3D Mesh Segmentation with a Novel Single Segmentation Field (SSF Seg) [12] – the operation of this algorithm is based on hierarchical grouping of surfaces constituting an object. Firstly, each surface is a separate segment, next they are joined based on appropriate transformation of eigenvectors of a matrix found using Laplace operator.
- Consistent Mesh Partitioning and Skeletonisation Using the Shape Diameter Function (Shape Diam) [9] – similarly to [11] this algorithm used Shape Diameter Function. In the beginning, this function is used for finding a centre of gravity of each surface of an object. The surfaces are joined to form a segment based on appropriate energy function.
- Normalized Cuts for 3D Mesh Analysis (Norm Cuts) [4] – is based on the hierarchical grouping. It begins with the assumption that each surface is a separate segment. Next, the surfaces with the least joint circumference divided by their joint area are merged to form a single segment. The stopping criterion is set by the user by providing the number of segments the object should contain.
- Randomised Cuts for 3D Mesh analysis (Rand Cuts) [4] – the method presented in the same work as in a previous paragraph. Its operation is the opposite to the Norm Cuts. It starts with the assumption that the whole object is a single segment and next it is divided into smaller segments till the appropriate number of segments is achieved.
- Mesh Segmentation Using Feature Point and Core Extraction (Core Extra) [6] – this is another hierarchical segmentation method. It does not require the information about the number of segments. The algorithm process the data as long as it is able to find characteristic points in a given segment. Such characteristic points are high convexities or concavities.
- Fast Mesh Segmentation Using Random Walks (Rand Walks) [8] – first, all the surfaces of the object are assumed to be separate segments and next based on random walks and the highest probability of reaching next surface the segments are joined to give a larger segment. The stopping criterion is based on the number of segments provided by the user.
- Hierarchical Mesh Segmentation Based on Fitting Primitives (Fit Prim) [2] – is based on grouping hierarchically and relies on matching adjacent surfaces to the arbitrary given geometrical primitives (surface, sphere, and cylinder). The segmentation starts with the assumption that all the surfaces of the object are separate segments and ends when the number of segments set by the user is reached.
- Metamorphosis of Polyhedral Surfaces Using Decomposition (K-Means) [10] – is based on computing centroids. For a given number of k-segments, the algorithm initially sets the k-centres of concentration producing k-segments and based on Euclidean distance assigns subsequent vertexes to the closest centres. Next, the centres of concentration are recomputed, and the algorithm stops when there are no changes in centres positions.
- Human – is given here as a reference. It provides the information how the inexperienced user will segment objects in provided dataset, without knowing the proper annotation.

3 Measures

Cut Discrepancy – it is a distance calculated along the segment borders provided in the reference file. The geodesic distance was used here. The distance of point p_i , forming a border of a segment and belonging to a set of points C_1 , to a border of a reference segment represented by a set of points C_2 is given by:

$$d_G(p_i, C_2) = \min\{d_G(p_i, p_j), \forall p_j \in C_2\}, \tag{1}$$

where:

$d_G(p_i, p_j)$ – geodesic distance between points p_i and p_j .

Directional Cut Discrepancy is computed as an average distance of all points from C_1 to the border of reference segment represented as a set of points C_2 :

$$DCD(S_1 \Rightarrow S_2) = \text{mean}\{d_G(p_i, C_2), \forall p_i \in C_1\}. \tag{2}$$

where:

S_1 - segmentation provided by the algorithms,

S_2 - reference segmentation.

Finally, the discrepancy of the borders is given by:

$$CD(S_1, S_2) = \frac{DCD(S_1 \Rightarrow S_2) + DCD(S_2 \Rightarrow S_1)}{\text{avgRadius}}, \tag{3}$$

where:

avgRadius - average distance of points from the object's Centre of Gravity [3].

Hamming Distance – is a measure of divergence of two sets (the segmentation provided by the algorithm and the reference one). Therefore, the sum of the differences between two sets is obtained. Since the tested objects were of different size, the distance was normalised by dividing the result by the area of the object, as it was described in [3].

A directional form of a Hamming distance is given by:

$$D_H(S_1 \Rightarrow S_2) = \sum_i \|S_{2i} \setminus S_{1it}\| \tag{4}$$

where:

\setminus - segments difference operator,

$\|x\|$ - size of x (overall area of the surfaces constituting an object),

S_1 - segmentation provided by the algorithms,

S_2 - reference segmentation,

i - given segment,

$it = \max_k \|S_{2i} \cap S_{1k}\|$.

Hamming distance is a directional average measured in both directions, scaled by the object area:

$$HD(S_1, S_2) = \frac{D_H(S_1 \Rightarrow S_2) + D_H(S_2 \Rightarrow S_1)}{2\|S\|} \quad (5)$$

where:

S - overall surface of the object.

In the tests provided in Sect. 4, the results for each direction are given:

$$R_m(S_1, S_2) = \frac{D_H(S_1 \Rightarrow S_2)}{\|S\|}, \quad (6)$$

$$R_f(S_1, S_2) = \frac{D_H(S_2 \Rightarrow S_1)}{\|S\|}. \quad (7)$$

Rand Index – it is a measure of a membership of the same object surfaces (computed and reference one) to the same segment. For S_1, S_2 representing computed and reference segmentations respectively and for s_i^1, s_i^2 representing segment number of i -th surface from S_1 and S_2 and assuming coefficients:

$$C_{ij} = 1 \iff s_i^1 = s_j^1, \quad (8)$$

$$P_{ij} = 1 \iff s_i^2 = s_j^2, \quad (9)$$

Rand Index could be computed as follows:

$$RI(S_1, S_2) = \binom{2}{N}^{-1} \sum_{i,j,i < j} [C_{ij}P_{ij} + (1 - C_{ij})(1 - P_{ij})], \quad (10)$$

where:

N - is the overall number of surfaces within the object.

Because the coefficients C_{ij} and P_{ij} are equal to 1 when compared surfaces belong to the same segment, RI is greater for better segmentations, in contrary to previous measures. For this reason the results in Sect. 4 were provided as $1 - RI(S_1, S_2)$ [3].

Consistency Error – measures similarities and discrepancies between computed and reference segments. It consists of two results – global (GCE) and local (LCE), which are calculated in the following way: Assuming S_1, S_2 to be segmentations (computed and reference respectively), \setminus – difference operator, $\|x\|$ – size of x and f_i - i -th object surface, local error for a single surface is given by:

$$E(S_1, S_2, f_i) = \frac{\|R(S_1, f_i) \setminus R(S_2, f_i)\|}{\|R(S_1, f_i)\|} \quad (11)$$

where:

$R(S, f_i)$ - segment (set of connected surfaces) inside segmentation S , which contains surface f_i .

Finally:

$$GCE(S_1, S_2) = \frac{1}{N} \min \left\{ \sum_i E(S_1, S_2, f_i), \sum_i E(S_2, S_1, f_i) \right\}, \quad (12)$$

$$LCE(S_1, S_2) = \frac{1}{N} \sum_i \min \{ E(S_1, S_2, f_i), E(S_2, S_1, f_i) \}, \quad (13)$$

where:

N - is the overall number of surfaces within the object [3].

4 Results

Taking the measures described in Sect. 3 a comparison of 11 different approaches presented in Sect. 2.2 (plus human annotator) is given. In most cases, the performance of the algorithms was tested by using the code provided by the research groups who developed each method. Only for [7, 11, 12], the results were taken directly from the papers. First, we focused our attention on Cut Discrepancy measure. The results of the comparison are shown in Fig. 1. As it can be observed from this figure the best results are achieved by LMS method.

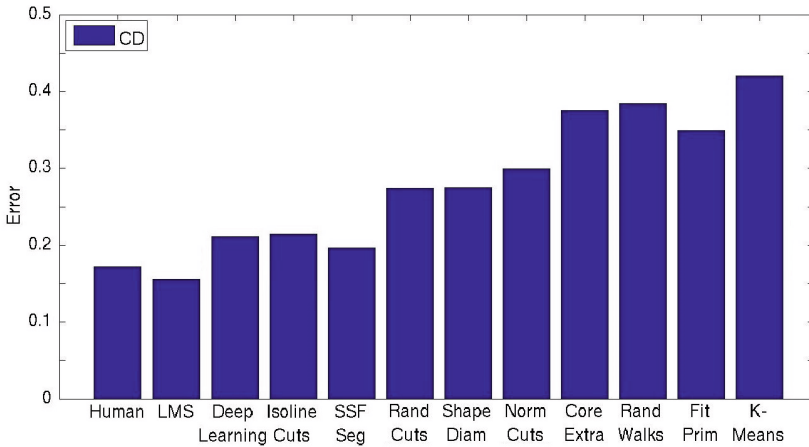


Fig. 1. Average Cut Discrepancy (CD) for each algorithm.

Second, we tested the algorithm against Hamming Distance measure. The results of the comparison are shown in Fig. 2. For each algorithm three measures are given. Namely, directional R_m , R_f and an average. As in the previous case the best results are achieved by LMS method.

Third, Rand Index measure was applied. The results of the comparison are shown in Fig. 3. Once more the best results were achieved for LMS method.

Taking into account Consistency Error the best results were obtained for LMS algorithm. The results of the comparison are shown in Fig. 4, where one can see both global and local error.

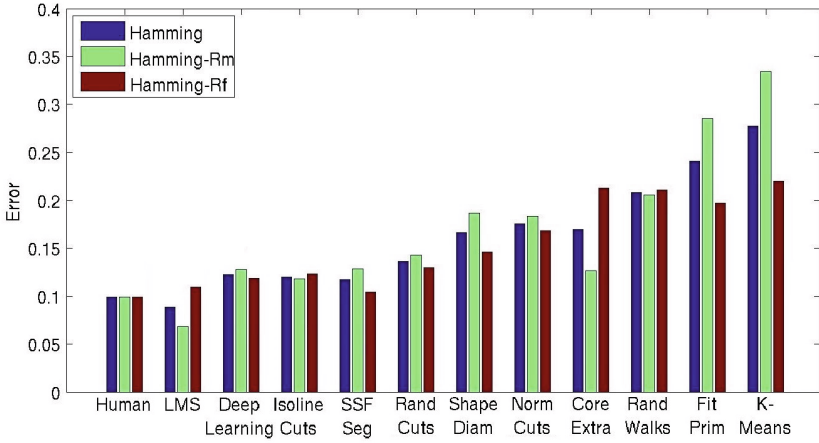


Fig. 2. Average Hamming Distance (HD) for each algorithm together with directional distances R_m and R_f .

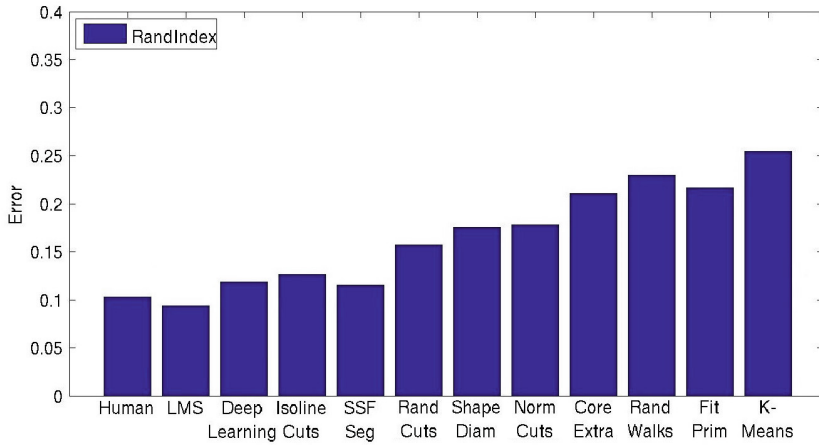


Fig. 3. Average Rand Index (RI) for each algorithm.

4.1 Discussion of the Results

From the previous section, one can get the impression that the LMS algorithm is the best of all presented algorithms. However, it is not always the case. All the Figures presented so far shows the average error, but taking a closer look at class by class comparison given in Table 1 one can see that Deep Neural Networks performs better for selected classes. LMS and Deep Neural Networks are the best of all algorithms for seven object classes each.

Table 1. Rand Index (RI) for each class separately. The best result for each class indicated by bold typeface.

Class	Human	LMS	Deep Learn.	Isoline Cuts	SSF Seg	Rand Cuts	Shape Diam	Norm Cuts	Core Extra	Rand Walks	Fit Prim	K-Means
Human	0.135	0.112	0.116	0.123	0.128	0.158	0.179	0.182	0.225	0.295	0.171	0.203
Cup	0.136	0.100	0.096	0.211	0.146	0.224	0.359	0.236	0.307	0.334	0.409	0.439
Glasses	0.101	0.136	0.173	0.098	0.113	0.097	0.204	0.142	0.301	0.316	0.230	0.178
Airplane	0.092	0.079	0.150	0.127	0.132	0.115	0.092	0.186	0.256	0.261	0.166	0.227
Ant	0.030	0.019	0.001	0.039	0.028	0.025	0.022	0.047	0.065	0.068	0.086	0.131
Chair	0.089	0.054	0.040	0.121	0.084	0.189	0.111	0.093	0.187	0.167	0.214	0.213
Octopus	0.024	0.019	0.036	0.041	0.026	0.067	0.045	0.063	0.051	0.069	0.103	0.100
Table	0.093	0.061	0.040	0.065	0.061	0.374	0.184	0.098	0.244	0.140	0.200	0.379
Teddy	0.050	0.032	0.024	0.053	0.036	0.045	0.057	0.121	0.114	0.127	0.132	0.181
Hand	0.091	0.105	0.135	0.115	0.110	0.097	0.202	0.156	0.155	0.222	0.206	0.164
Plier	0.071	0.053	0.151	0.073	0.085	0.109	0.375	0.183	0.093	0.230	0.169	0.263
Fish	0.155	0.128	0.288	0.243	0.215	0.297	0.248	0.399	0.273	0.406	0.421	0.414
Bird	0.062	0.088	0.171	0.097	0.078	0.114	0.115	0.212	0.124	0.280	0.206	0.199
Armadillo	0.083	0.070	0.073	0.106	0.091	0.081	0.090	0.120	0.141	0.107	0.094	0.116
Bust	0.220	0.238	0.275	0.244	0.286	0.251	0.298	0.332	0.315	0.335	0.327	0.352
Mech	0.131	0.110	0.073	0.122	0.126	0.283	0.238	0.175	0.387	0.244	0.272	0.426
Bearing	0.104	0.091	0.056	0.177	0.148	0.129	0.119	0.179	0.398	0.271	0.234	0.260
Vase	0.144	0.147	0.212	0.168	0.154	0.160	0.239	0.268	0.226	0.287	0.289	0.401
Fourleg	0.149	0.136	0.140	0.181	0.165	0.177	0.161	0.189	0.191	0.208	0.183	0.195
Average	0.103	0.094	0.118	0.127	0.116	0.157	0.176	0.178	0.211	0.230	0.216	0.255

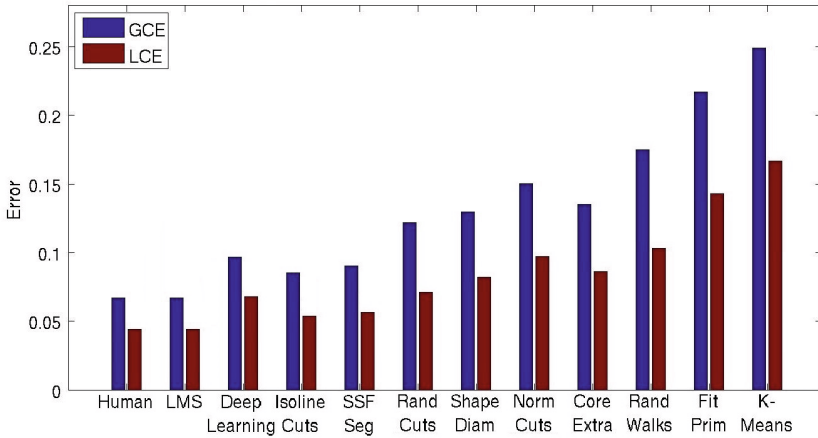


Fig. 4. Average global (G-) and local (L-) Consistency Error (CE) for each algorithm.

5 Conclusions

In this paper, we provided an extensive comparison of existing methods of 3D objects segmentation. The results clearly show that currently a method called LMS outperforms other approaches and for certain measures, it even surpasses inexperienced human annotator. However, if we perform class by class comparison, comparable results are achieved by Deep Learning method. This review is a basis for indicating future ways of improving existing approaches and also serves as a guide for using these algorithms in applications such as service robotics, where the segmentation might help in providing semantic meaning to the parts of the object.

Acknowledgments. Krzysztof Walas is supported by the Poznań University of Technology grant DSMK/0154-2016.

References

1. Watertight track of shrec (2007). dostpny w Internecie: <http://watertight.ge.imati.cnr.it/>
2. Attene, M., Falcidieno, B., Spagnuolo, M.: Hierarchical mesh segmentation based on fitting primitives. *Vis. Comput.* **22**(3), 181–193 (2006). <http://dx.doi.org/10.1007/s00371-006-0375-x>
3. Chen, X., Golovinskiy, A., Funkhouser, T.: A benchmark for 3D mesh segmentation. *ACM Trans. Graphics (Proc. SIGGRAPH)* **28**(3), August 2009
4. Golovinskiy, A., Funkhouser, T.: Randomized cuts for 3D mesh analysis. *ACM Trans. Graphics (Proc. SIGGRAPH ASIA)* **27**(5), December 2008
5. Kalogerakis, E., Hertzmann, A., Singh, K.: Learning 3D Mesh Segmentation and Labeling. *ACM Trans. Graphics* **29**(3) (2010)

6. Katz, S., Leifman, G., Tal, A.: Mesh segmentation using feature point and core extraction (2005)
7. Kin-Chung Au, O., Zheng, Y., Chen, M., Xu, P., Tai, C.L.: Mesh segmentation with concavity-aware fields. *IEEE Trans. Vis. Comput. Graph.* **18**(7), 1125–1134 (2012). <http://dx.doi.org/10.1109/TVCG.2011.131>
8. Lai, Y.K., Hu, S.M., Martin, R.R., Rosin, P.L.: Fast mesh segmentation using random walks. In: *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling, SPM 2008*, pp. 183–191. ACM, New York (2008). <http://doi.acm.org/10.1145/1364901.1364927>
9. Shapira, L., Shamir, A., Cohen-Or, D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput.* **24**(4), 249–259 (2008). <http://dx.doi.org/10.1007/s00371-007-0197-5>
10. Shlafman, S., Tal, A., Katz, S.: Metamorphosis of polyhedral surfaces using decomposition. In: *Computer Graphics Forum*, pp. 219–228 (2002)
11. Shu, Z., Qi, C., Xin, S., Hu, C., Wang, L., Zhang, Y., Liu, L.: Unsupervised 3D shape segmentation and co-segmentation via deep learning. *Comput. Aided Geom. Des.* **43**(C), 39–52 (2016). <http://dx.doi.org/10.1016/j.cagd.2016.02.015>
12. Wang, H., Lu, T., Au, O.K.C., Tai, C.L.: Spectral 3D mesh segmentation with a novel single segmentation field. *Graph. Models* **76**(5), 440–456 (2014). <http://dx.doi.org/10.1016/j.gmod.2014.04.009>