RESEARCH ARTICLE OPEN ACCESS

Algorithm Designing for Edges and Surfaces Extraction from Mesh Models

AbdAlrazzakDabbour*, RabieHabib**, Mariam Saii***
*(Department of Mechatronics Engineering, Tishreen University, Syria)
** (Department of Mechatronics Engineering, Tishreen University, Syria)
*** (Department of Computer and Automatic Control Engineering, Tishreen University, Syria)

--************************----------------------------------

Abstract:

Many industrial tasks need geometry information from CAD files (STL, IGES etc.), to acquire these information, 3D mesh should be segmented to multi surfaces and edges should be extracted. TriFace's Growing (TRIFGR) method has been proposed for surfaces and edges extraction from triangular mesh based on the orientation angle between two adjacent triangles. Depending on orientation angle value, the adjacent triangles will be marked as belong to the same surface or not. The experimental results show that the proposed method is performed efficiently to extract surfaces and edges on both computer graphics and CAD mesh models and segmentation result could be easily changed according to the aim of the application. Besides that, to allow users importing mesh model from STL file with ASCII or Binary format and getting geometry information about each extracted surface, a simple windows application has been designed based on C#.

Keywords **—3D Mesh Segmentation, Surface Extraction, Edges Extraction, STL File.**

--************************----------------------------------

INTRODUCTION

In computer graphics, 3D mesh models are basically used to represent objects that are acquired by 3D scanning process or CAD/CAM software.

Besides, Many 3D mesh model processing application need 3D mesh segmentation as a major step which defined as dividing 3D mesh model into multi surfaces, each surface contains multi-connected triangles.

In the past few years, many approaches have been proposed for 3D mesh segmentation and edges extraction.

Guo, et al used deep Convolutional Neural Networks (CNNs) for 3D mesh segmentation, presented method can handle many types of 3D mesh and consists of multi stages: stage (1): training stage to represent each triangle in 3d mesh based on features extraction from triangle. Stage (2): generate label vector for each triangle. Stage(3) each triangle assigned to specified model's segments depending on label vectors [1]. Gu et al. Proposed a method to extract concave and convex feature regions via segmenting surface mesh of a mechanical part [2]. Wu et al. proposed mesh segmentation method depending on interactive approach with users, first step was over-segmenting the model, next step allow user to merge similar patches into same surface and assigning labels to each segment [3]. Hajari et al. suggested to use animal anatomy as the ground truth [4]. Zheng, Tai, and Au present a very easy-to-use interactive tool, for mesh segmentation. The directional search strategy was used [5]. Benhabiles, Lavoúe, Vandeborre, &Daoudi Produced method from 3d mesh segmentation and feature recognition depending on large manually segmented 3d mesh dataset based on boundary edges extraction, tested feature automatically recognized as its edges most relevant to features edges in dataset [6]. Kalogerakis, Hertzmann, & Singh Produced algorithm for 3d meshes segmentation and labeling , proposed algorithm used segmented meshes database for training, besides the proposed algorithm does not have any parameter should be manually tuned [7]. In this paper, surfaces and edges extraction algorithm for both 3D CAD mesh and Graphics mesh models is proposed. Besides, edges for each segment will be marked depending on the comparing result of the angle between each adjacent triangle pair and thresholding value.

STL FILE FORMAT

Each STL file describes one 3D model by using vertices and normal vector (perpendicular line to the triangle) for each triangle without any information about color.

STL file could be saved in (ASCII or Binary) format, in both format mesh model consist of triangular facet, each triangle shares two vertices with its neighbours which is called vertex-to-vertex rule.

3D VISUALIZATION OF STL

In this paper, an application using C# is designed to view STL files (both Binary and ASCII formats) with many options as shown in Fig. 1 such as:

- translation, scaling, rotation of the model.
- applying various lights, material color.
- options for solid, wireframe, points, lines.
- viewing various views of an object (front, back, left, right, top, bottom, isometric).

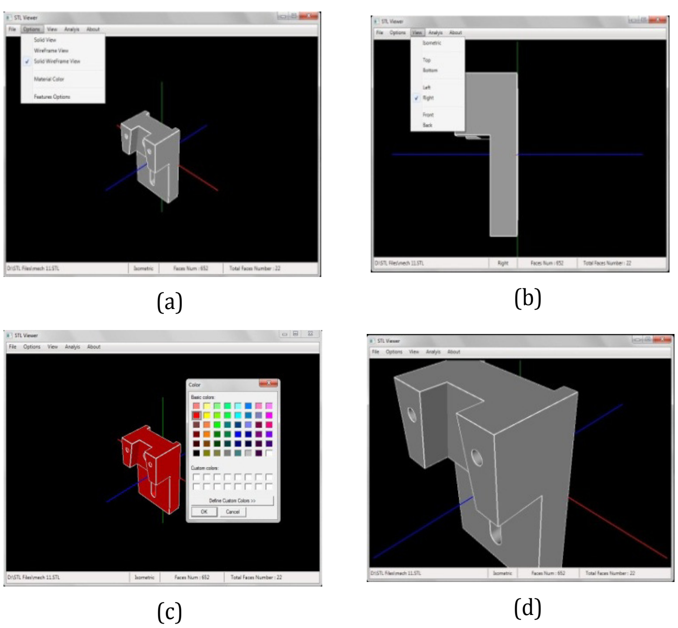


Fig. 1Some of viewing Options in proposed C# Application, (a) Display Style, (b) View Orientation, (c) Model's Color Option, (d) Zoom In/Out

MESH SEGMENTATION

3D mesh models can be classified into two main groups: computer graphics mesh and CAD mesh. CAD mesh models are characterized by planes and sharp edges among them as shown in Fig. 2.a, while the computer graphics mesh models consist of complex surfaces and likely with no hard edges as shown in Fig. 2.b. That is why segmentation methods for computer graphics mesh models may not be applied to the CAD mesh models directly and vice versa.

The basic challenge is to design a method can be used on both CAD mesh and computer graphics mesh models segmentation, and must be configured by few parameters could be changed relatively to mesh model specifications.

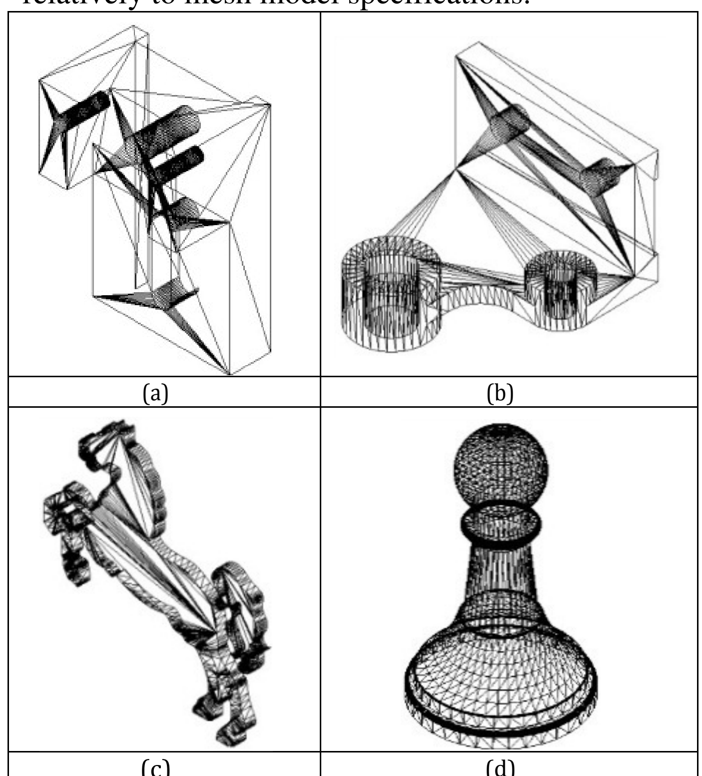


Fig. 2(a)(b) Samples of CAD Mesh Models, (c)(d) Samples of Computer Graphics Mesh Models

SURFACES AND EDGES EXTRACTION

The proposed segmentation method is shown in Table 1 divided the input mesh model into several surfaces, the angle between each adjacent triangle pair in mesh model will be measured and compared with thresholding value, according to this comparison we have two possible states:

- Measured angle is less than thresholding value: then the two tested adjacent triangles belong to the same surface.
- Measured angle is larger than thresholding value: then the two tested adjacent triangles belong to different surfaces and the line shared between them is an edge.

The proposed algorithm has the advantage that segmentation result could be easily and fast tuned using only one parameter according to the aim of the application.

TABLE I
Algorithm 1 shows pseudo-code for proposed approach

Algorithm 1 TriFace's Growing (TRIFGR)	
Input:	Set *Normals* (Normal Vectors for each Triface)
	Set *Points* (Points Coordinations for each Triface)
	Variable Angle_thresholding ()
Output:	Set *triFaces_number* (Default value -1)
1	Faceindex = 0
2	Waitlist = Empty
3	**for** i = 0 → i ≤ Triface:count**do**
4	**If** (Triface value ≠ -1) **then**
5	Adding 1 to Faceindex's value
6	Faceindex value → *triFaces_number[i]*
7	Adding i to *Waitlist*
8	**While** (*Waitlist* ≠ Empty) **do**
9	Find Set *Adjacent_triface* (their numbers form triFaces_number = -1)
10	**For** j = 0 → j ≤ *Adjacent_triface*.count**do**
11	**If** (angle(current_triface, *Adjacent* _triface[j]) <Angle_thresholding) **then**
12	Assign Faceindex value to current_triface from *Adjacent_triface*
13	Adding current_triface to *Waitlist*
14	**Else**
15	Mark Segment between two as Edge
16	**End if**
17	**End for**
18	Delete *current_triface* from *Waitlist*
19	End while
20	End if
21	End for

A. Determining thresholding value

In proposed algorithm, thresholding value has high effective on segmentation results, that is way it should be determined carefully according to mesh model specifications.

If the thresholding angle value was large and mesh model had no hard edges then many surfaces will be merged together and labeled as one surface.

In another hand, if the thresholding angle value was small and mesh model had hard edges, then one surface may be divided into many surfaces.

RESULTS AND DISCUSSION

The proposed algorithm (TRIFGR) was tested on CAD mesh and computer graphics mesh models which are imported from STL files by using C# Application, Fig. 7. shows some of testing mesh models.

Table 2 shows each STL file properties (number of trifaces, the number of segmented surfaces according to thresholding angle) as the results of proposed method (TRIFGR).

TABLE II
tested mesh models properties (using TRIFGR method)

Mesh Name	Thresholding Angle	Trifaces number	Total Surface Number	Time (Sec)
mech11	12°	652	22	0.1
mech12	12°	1034	18	0.15
horse	12°	3736	38	0.4
Chess-Pawn	12°	5250	7	0.6

In another hand, Fig. 3 shows how the number of extracted surfaces are changed according to thresholding angle value for CAD mesh and computer graphics mesh models.

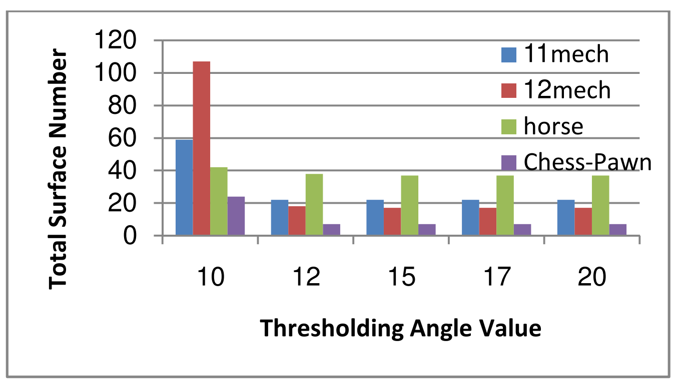
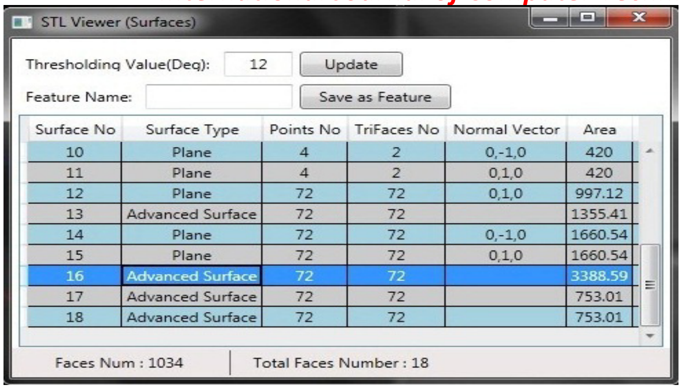


Fig. 3Changing Total Extracted Surfaces according to thresholding value (by using TRIFGR method)

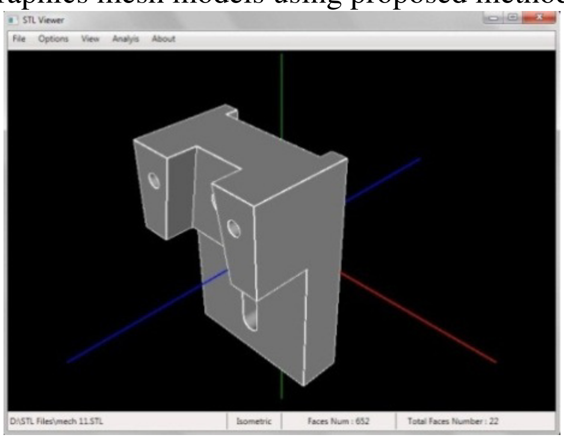
Fig. 4 shows extracted surface's properties such as (Surface Number, Surface Type, Points Number, TriFaces Number, Normal Vector and Surface's Area).



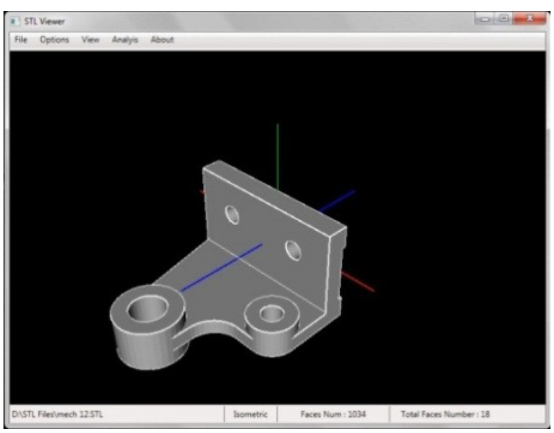
Surface No	Surface Type	Points No	TriFaces No	Normal Vector	Area
10	Plane	4	2	0,-1,0	420
11	Plane	4	2	0,1,0	420
12	Plane	72	72	0,1,0	997.12
13	Advanced Surface	72	72		1355.41
14	Plane	72	72	0,-1,0	1660.54
15	Plane	72	72	0,1,0	1660.54
16	Advanced Surface	72	72		3388.59
17	Advanced Surface	72	72		753.01
18	Advanced Surface	72	72		753.01

Fig. 4. Extracted Surface's Properties (by using TRIFGR method)

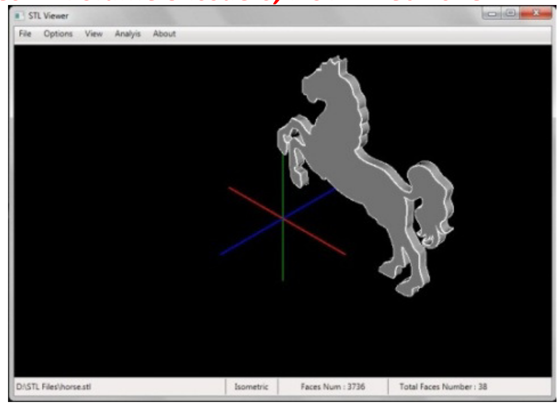
Fig. 5 shows samples of test 3D mesh model, Fig. 6 and Fig. 7 show samples of extracted surfaces from tested models using the TRIFGR method, while Fig. 8 and Fig. 9 show extracted edges and lines from CAD mesh models and computer graphics mesh models using proposed method.



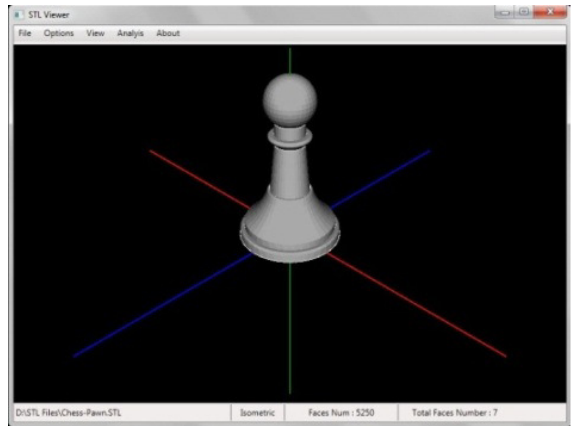
(a)



(b)



(c)



(d)

Fig. 5Samples of tested mesh models, (a) mech11 model, (b) mech12 model, (c) horse model, (d) Chess-Pawn

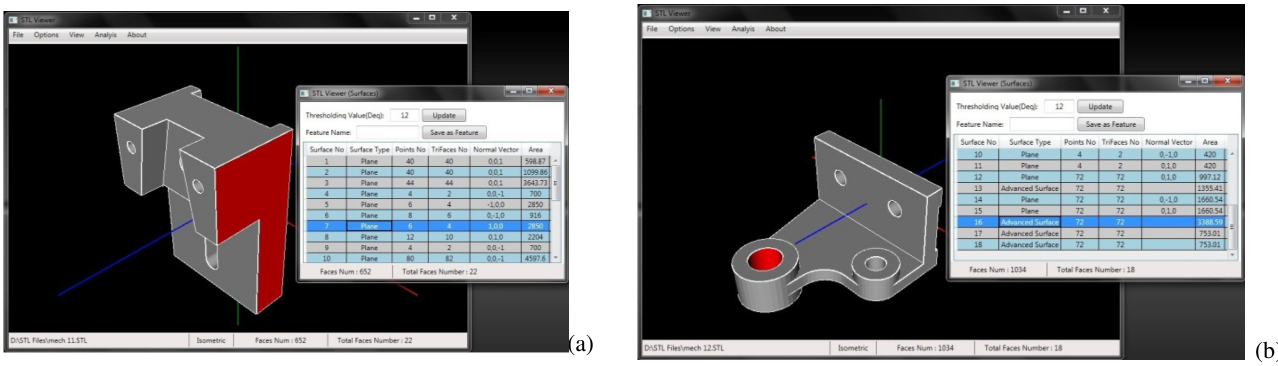


Fig. 6 Samples of Surfaces extraction from CAD mesh by using TRIFGR method, (a) mech11 model, (b) mech12 model

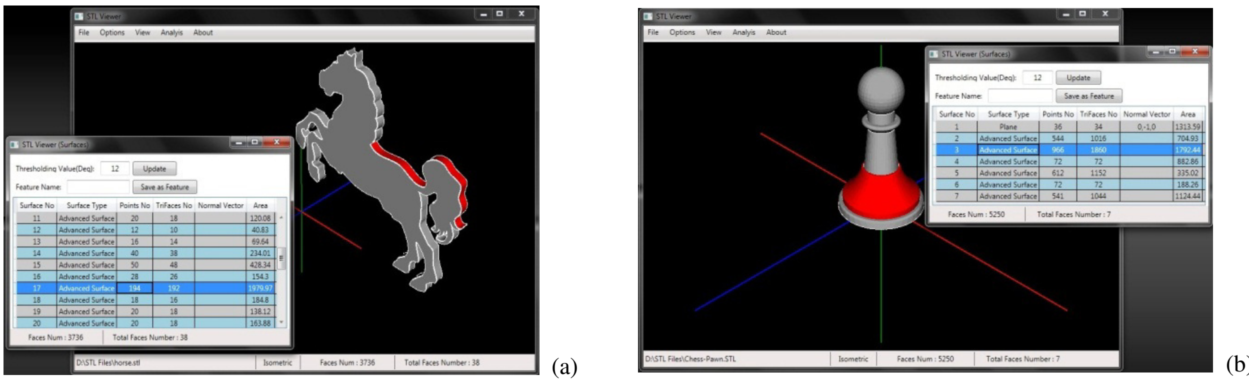


Fig. 7Samples of Surfaces extraction from Computer Graphics mesh by using TRIFGR method, (a) horse model, (b) Chess-Pawn model

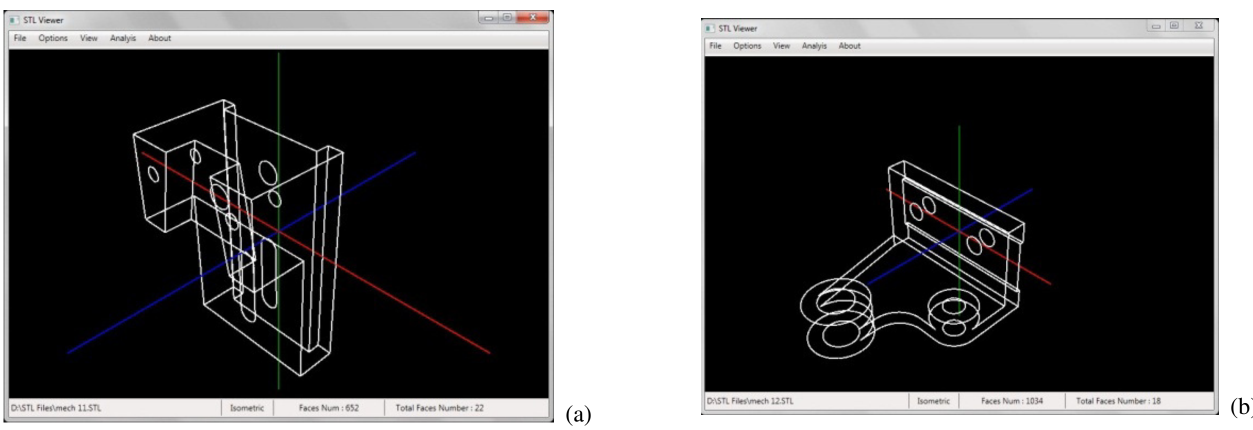


Fig. 8Samples of Edges extraction from CAD mesh by using TRIFGR method, (a) mech11 model, (b) mech12 model

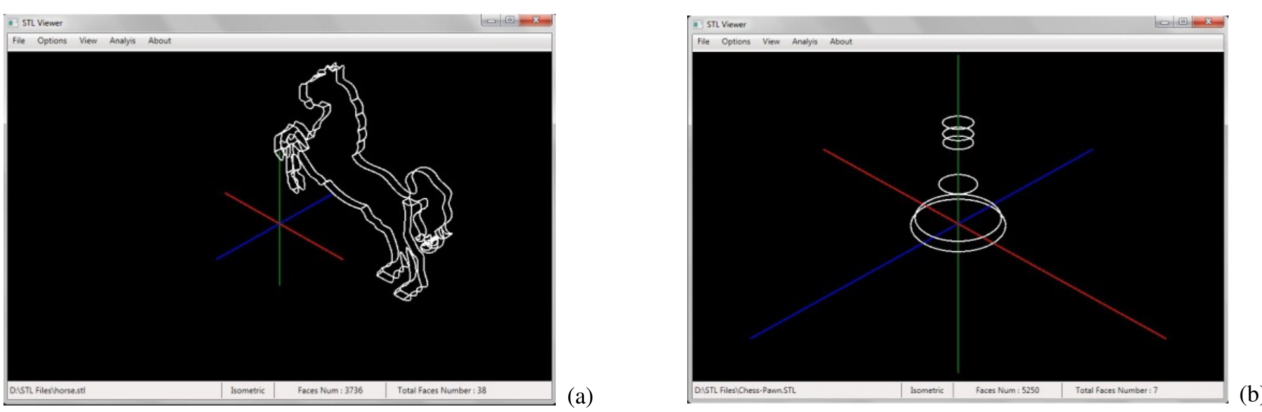


Fig. 9Samples of Edges extraction from Computer Graphics mesh by using TRIFGR method, (a) horse model, (b) Chess-Pawn model

CONCLUSION

This paper presents a method for surface segmentation and edge extraction from both CAD mesh and computer graphics mesh models, it is integrated into C# Application which allows the user to import mesh models from STL file (ASCII or BINARY formats) besides some options such as:

- translation, scaling, rotation of the model.
- applying various lights, material color's.
- options for solid, wireframe, points, lines viewing.
- various views of an object (front, back, left, right, top, bottom and isometric) .

Moreover, labelling segmented surface and shows information about it (area, normal vector if it was a plane, point's number, trifaces number).

Our Method divided input mesh into several surfaces according to angle's value between adjacent triangle pairs, the tested triangles belong to the same surface if the angle between them less than thresholding angle value's. Otherwise, the line which is being shared between will be marked as an edge.

The proposed method gives good results on CAD mesh models because they usually contain hard edges, but in computer graphics mesh models the segmented surfaces are deeply related to thresholding angle value's, that is why segmentation results are highly changed according to thresholding angle value's and should be specified by the human user or by another analyzing process related to mesh specifications and processing purpose.

Compared with other methods, our method can meet the engineering applications requirements in a better way for its capability of labeling both CAD mesh and computer graphics mesh models besides edges extraction in one stage, while other methods depend on multi-stages process for surfaces segmentation and edges extractions and some of them only work on CAD mesh model.

In the future, we intend to:

- Improve our application to import more types of mesh models (IGES, STEP, DXF, etc.)
- Analysing extracted edges as an important step for feature extraction from CAD model (cylinder, Hole, Sphere, cone, etc.).
- Improve our method to work with point cloud model not only mesh model.

REFERENCES

[1] K. Guo, D. Zou, X. Chen, 3D Mesh Labeling via Deep Convolutional Neural Networks, ACM Trans. Graph. 35 (2015).

[2] M. Gu, L. Duan, M. Wang, Y. Bai, H. Shao, H. Wang, F. Liu, An improved approach of mesh segmentation to extract feature regions, PLoS One. 10 (2015) 1–15. doi:10.1371/journal.pone.0139488.

[3] Z. Wu, R. Shou, Y. Wang, X. Liu, Interactive shape co-segmentation via label propagation, Comput. Graph. 38 (2014) 248–254. doi:10.1016/j.cag.2013.11.009.

[4] N. Hajari, I. Cheng, A. Basu, G. Lavou, Evaluation of 3D Model Segmentation Techniques based on Animal Anatomy, (2013).

[5] Y. Zheng, C.L. Tai, O.K.C. Au, Dot scissor: A single-click interface for mesh segmentation, IEEE Trans. Vis. Comput. Graph. 18 (2012) 1304–1312. doi:10.1109/TVCG.2011.140.

[6] H. Benhabiles, G. Lavoúe, J.P. Vandeborre, M. Daoudi, Learning boundary edges for 3D-mesh segmentation, Comput. Graph. Forum. 30 (2011) 2170–2182. doi:10.1111/j.1467-8659.2011.01967.x.

[7] E. Kalogerakis, A. Hertzmann, K. Singh, Learning 3D mesh segmentation and labeling, ACM Trans. Graph. 29 (2010) 1. doi:10.1145/1833351.1778839.

[8] X. Yang, Fei and Zhou, Fan and Wang, Ruo-mei and Liu, Li and Luo, A fast and efficient mesh segmentation method based on improved region growing, Appl. Math. J. Chinese Univ. 29 (2014) 468–480. doi:10.1007/s11766-014-3240-0.

[9] J. Lv, X. Chen, J. Huang, H. Bao, Semi-supervised mesh segmentation and labeling, Comput. Graph. Forum. 31 (2012) 2241–2248. doi:10.1111/j.1467-8659.2012.03217.x.

[10] H. Benhabiles, J.P. Vandeborre, G. Lavoué, M. Daoudi, A comparative study of existing metrics for 3D-mesh segmentation evaluation, Vis. Comput. 26 (2010) 1451–1466. doi:10.1007/s00371-010-0494-2.

[11] H. Benhabiles, G. Lavoué, J.P. Vandeborre, M. Daoudi, A subjective experiment for 3D-mesh segmentation evaluation, in: 2010 IEEE Int. Work. Multimed. Signal Process. MMSP2010, 2010: pp. 356–360. doi:10.1109/MMSP.2010.5662046.

[12] S. Gao, W. Zhao, H. Lin, F. Yang, X. Chen, Feature suppression based CAD mesh model simplification, CAD Comput. Aided Des. 42 (2010) 1178–1188. doi:10.1016/j.cad.2010.05.010.

[13] H. Benhabiles, J.P. Vandeborre, G. Lavoué, M. Daoudi, A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D models, in: 2009 IEEE Int. Conf. Shape Model. Appl. SMI 2009, 2009: pp. 36–43. doi:10.1109/SMI.2009.5170161.

[14] G. Lavoué, C. Wolf, Markov random fields for improving 3D mesh analysis and segmentation, Proc. 1st Eurographics Conf. 3D Object Retr. (2008) 25–32. doi:10.2312/3DOR/3DOR08/025-032.

[15] D.H. Kim, I.D. Yun, S.U. Lee, Boundary-trimmed 3D triangular mesh segmentation based on iterative merging strategy, Pattern Recognit. 39 (2006) 827–838. doi:10.1016/j.patcog.2005.11.022.

[16] M. Attene, B. Falcidieno, M. Spagnuolo, Hierarchical mesh segmentation based on fitting primitives, Vis. Comput. (2006) 181–193. doi:10.1007/s00371-006-0375-x.

[17] M. Attene, G. Patan??, S. Katz, M. Spagnuolo, M. Mortara, A. Tal, Mesh segmentation - A comparative study, Proc. - IEEE Int. Conf. Shape Model. Appl. 2006, SMI 2006. 2006 (2006) 7. doi:10.1109/SMI.2006.24.

[18] Y.-K. Lai, Q.-Y. Zhou, S.-M. Hu, R.R. Martin, Feature sensitive mesh segmentation, Proc. 2006 ACM Symp. Solid Phys. Model. - SPM '06. (2006) 17–25. doi:10.1145/1128888.1128891.

[19] H. Yamauchi, S. Lee, Y. Lee, Y. Ohtake, A. Belyaev, H.P. Seidel, Feature sensitive mesh segmentation with mean shift, Proc. - Int. Conf.

Shape Model. Appl. SMI'05. 2005 (2005) 238–245. doi:10.1109/SMI.2005.21.

[20] K. Hildebrandt, K. Polthier, M. Wardetzky, Smooth feature lines on surface meshes, ACM Int. Conf. Proceeding Ser. Vol. 255. (2005) 85. http://portal.acm.org/citation.cfm?id=1281935.

[21] Y. Sun, D.L. Page, J.K. Paik, A. Koschan, M. a. Abidi, Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing, Proceedings. Int. Conf. Image Process. 3 (2002) 825–828. doi:10.1109/ICIP.2002.1039099.