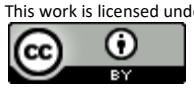
Available online at: https://ijact.in

Date of Submission	16/06/2020
Date of Acceptance	22/07/2020
Date of Publication	01/08/2020
Page numbers	3768-3774 (7 Pages)



ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

A BINARY SEGMENTATION APPROACH FOR 3D DUO-SPHERES

Saw Seow Hui[1], Ewe Hong Tat[2], Lee Byung Gook[3]

[1]Department of Computer Science, Faculty of Information and Communication Technology, Universiti Tunku Abdul Rahman, Kampar Campus, Perak, Malaysia, Email: shsaw@utar.edu.my

[2]Department of Electrical and Electronic Engineering, Lee Kong Chian Faculty of Engineering and Science, Sungai Long, Kajang, Selangor, Malaysia

[3]Division of Computer Engineering, Dongseo University, Busan, South Korea

Abstract: This paper is to present an interactive binary segmentation approach on a 3D duo-spheres. Our proposed approach has successfully partitioned the 3D mesh into two disjoint spheres. Six weight functions are added as soft constraints, together with the proposed max-flow min-cut to find the globally optimal binary segmentation. Some experimental results are included to demonstrate the effectiveness of our proposed binary segmentation approach.

Keywords: binary segmentation;3D duo-spheres; weight functions; improved max-flow min-cut

I. INTRODUCTION

Since a plain 3D mesh consists of only a set of simple primitives (vertices, edges and faces), the direct interpretation of 3D mesh by computer can be challenging. Hence, the exploitation of high-level semantics by dividing the 3D mesh into more manageable and corresponding segments for further analysis is required. It serves as a vital cornerstone in the pre-processing stage for various application such as 3D mesh unfolding [1], 3D printing [2-4], surveillance system, skeleton extraction [5,6], morphing [7,8], 3D model composition [9,10] etc. Additionally, the performance and quality of the segmentation are strongly dependent on the context in which it will be used.

In this paper, we proposed a binary segmentation approach for a 3D mesh in a strictly controlled setting, whereby it is composed of primitive geometric shapes: the spheres (details can be found in Table 1). The cutting edge for the tested 3D mesh is not obvious. Thus, it can be used to test the robustness of our proposed approach.

Generally, there are two significant phases in 3D mesh segmentation: the soft constraints definition and the partition process with hard constraints imposed by user. We have defined our own soft constraints on every edge based upon reciprocation on the existing cost functions extracted from 3D mesh simplification [14, 15]. Without soft constraints definition, a 3D mesh is just a collection of primitive structure without useful information for deciding which elements belong to the same segment.

The reciprocation is aimed to magnify the weightage of the edges of the 3D mesh in order to incorporate with the improved max-flow min-cut algorithm [17] during the partition process. Max-flow min-cut is one of the famous algorithm in combinatorial optimization by finding the largest amount of flow supported on every edge starting from the source, s to the sink, t. The saturated edges become the cutting edges of the 3D mesh, which partition the mesh into two disjoint parts $\{S, T\}$.

Overall, there are four stages in our implementation [18]: i) 3D mesh construction using the Korea Mathematical Methods for Curves and Surfaces (KMMCS) application [19], ii) Soft constraints definition, iii) Partition process with hard constraints and iv) The segmented 3D mesh.

The remainder of this paper proceeds as follows: Section 2 describes the existing and our newly formulated weight functions as the soft constraints definition. Then, the proposed partition algorithm with hard constraints is presented in Section 3. Section 4 demonstrated and discussed the results and performance. The conclusion and potential improvements of the proposed approach are described in Section 5.

Table 1. The Details for 3D Duo-spheres

Primitives	Quantity
No. of Vertices	3648
No. of Edges	2412
No. of Faces	2412

II. Proposed Soft Constraints

A common approach to define the soft constraints is assigning weight values based on the cardinality, geometric and topological characteristics [13] between adjacent vertices, faces or in dual graph of a mesh. In this paper, we present the newly formulated weight functions that are assigned onto all edges of the 3D duo-spheres. This comes from the notion of 3D mesh simplification, 2.1 Reciprocal Melax (RM) where the visual content of the object or scene is taken into account for the calculation. It takes high-detailed mesh with many polygons and generates the mesh with fewer polygons that looks reasonably similar to the original. The existing cost functions aim at minimizing the visual change to the 3D mesh during the simplification process. Thus, the small, coplanar triangles and insignificant portions of the mesh are designated to be the minimal cost and are favoured to be collapsed [15]. The visual-sensitive parts with higher cost tend to lie on the contours of concave discontinuity of the tangent planes and ridge-valleys, as stated in the minima rule [20].

With the benefits offered by these existing cost functions, we have reformulated these equations into our weight functions by employing the principle of reciprocation. Therefore, the edges along the concave seam would have the minimal cost and higher chance to be the cutting edge during the partition process. This is the new attempt among the existing criteria definitions that has successfully facilitated the segmentation. Three existing cost functions from 3D mesh simplification are selected, studied and implemented: Melax (M) [16], Kims-Levin (KL) [21] and Lindstrom-Turk (LT) [22]. We have reciprocated these existing cost functions as our newly proposed weight functions. Table 2 exhibits the comparison results between the existing cost functions and our newly defined weight functions for the binary segmentation approach.

A. Reciprocal Melax (RM)

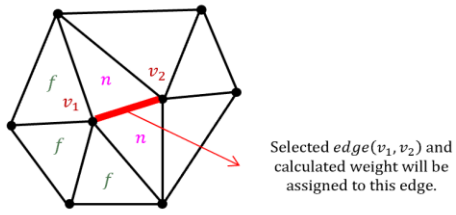


Figure 1. Geometric attributes to compose Eq. (1) and Eq. (2).

Stan Melax [16] has developed an algorithm to reduce the polygons via a sequence of edge collapses. In the operation of collapsing the edges of the model, two vertices v_i and v_{i+1}, or an $edge(v_i, v_{i+1})$ are/is selected and one of them (v_i) iscollapsed onto the other side (v_{i+1}).

The author defined the cost of collapsing an edge based on the local features such as the length of the edge multiplied by a curvature term. The curvature term for collapsing an $edge(v_i, v_{i+1})$ is determined by comparing dot products of face normalin order to find the triangle adjacent to v_{i+1}that faces furthest away from the other triangles that are along the $edge(v_i, v_{i+1})$as denoted in Eq. (1). The meaning for each respective parameter is illustrated in Figure 1.

$$weight\ M\big(edge(v_i, v_{i+1})\big) = \|v_i - v_{i+1}\| \times \max_{f \in T_{v_i}} \left\{ \min_{n \in T_{v_i v_{i+1}}} \left\{ \frac{(1 - f \cdot normal \cdot n \cdot normal)}{2} \right\} \right\} \quad (1)$$

where T_{v_i} is the set of triangles that contain v_i, while $T_{v_i v_{i+1}}$ is the set of triangles that contain both v_i and v_{i+1}.

Edges along a trough, coplanar surfaces, short and with fewer triangles surrounding vertices v_i and vertices $v_{i+1}$are assignedwith some values (Eq. (1)) and these minimal edges are chosen to collapse; hence, the overall shape of the mesh is remained.

However, our goal is to magnify the obvious cutting edge to facilitate the partition process later. Hence, we reformed the existing equation into a new weight function by reciprocating it as described in [18]. It is written as:

$$weight\ RM\big(edge(v_i, v_{i+1})\big) = \frac{1}{weight\ M\big(edge\ (v_i, v_{i+1})\big) \text{ in Eq. (1)}} \quad (2)$$

B. Reciprocal Kims-Levin (RKL)

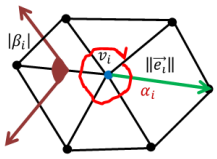


Figure 2. A vertex v_i and related variables: dihedral angle β_i, angleα_i and the length of the edge $||\vec{e}||$.

Apart from the use of basic geometric information such as the distance and normals mentioned in section A, we have also tested the other geometric attributes: curvatures on discrete surfaces. According to [21, 23, 24], 3D triangle meshes do not have any curvature at all, since all faces are flat. However, by assuming a 3D mesh as a piecewise linear approximation of an unknown smooth surfaces, the value of curvature can be estimated.

The main attributes in computing discrete curvature is the sum of absolute principle curvatures $|k_1|$ and $|k_2|$ at every vertex ofa mesh as follows:

$$|k_1| + |k_2| = \begin{cases} 2|H|, & \text{if } K \geq 0, \\ 2\sqrt{H^2 - K}, & \text{otherwise.} \end{cases} \quad (3)$$

K represents the Gaussian curvature while $|H|$ represents the absolute mean curvature. These two curvatures have to be derivedin order to compute the discrete curvature. Gaussian curvature K of a vertex is related to angles and faces that are connected tothat vertex. As the absolute mean curvature $|H|$ is related to dihedral angles and edge lengths

as shown in Figure 2. Since the computation has to involve the total curvature attributed to a vertex v with respect to the area $S = S_v$, the integration for both K and the integral absolute mean curvature $|\bar{H}|$ can be obtained by the following equation:

$$\bar{K} = \int_S K = 2\pi - \sum_{i=1}^{n} \alpha_i$$

$$|\bar{H}| = \int_S |H| = \frac{1}{4}\sum_{i=1}^{n} ||\vec{r_i}||\,|\beta_i| \tag{4}$$

[21, 23] assumed the curvatures are uniformly distributed around the vertex, simply normalize the curvatures at vertex v from these integral values as defined in Eq. (5) by the area:

$$K = \frac{\bar{K}}{S}$$

$$H = \frac{|\bar{H}|}{S} \tag{5}$$

This existing cost function (Eq. (6)) is then reciprocated by the sum of absolute principle curvatures denoted in Eq. (3) for all vertices of the 3D mesh in Eq. (7). Then, the mean value for the length of the $||edge(v_i, v_{i+1})||$ is multiplied by the mean of theprinciple curvatures at adjacent vertices v_i and v_{i+1}.

$$weight\ KL\big(edge(v_i, v_{i+1})\big) = \big||v_i - v_{i+1}|\big| \times \left(\frac{v_i(|k_1|+|k_2|)+v_{i+1}(|k_1|+|k_2|)}{2.0}\right) \tag{6}$$

$$weight\ RKL\big(edge(v_i, v_{i+1})\big) = \frac{1}{||v_i - v_{i+1}|| \times \left(\frac{v_i(|k_1|+|k_2|)+v_{i+1}(|k_1|+|k_2|)}{2.0}\right)} \tag{7}$$

C. Reciprocal Lindstrom-Turk (RLT)

The previous weight functions involved local features of the 3D mesh. While [22] introduces a positioning strategy that focuses on the preservation of volume and boundary area. They include consideration of the local and global features of the3D mesh in the calculations.

[12] and [22] explained that there are basically two factors that are crucial for the quality of the 3D mesh simplification. The first factor is to determine the new vertex position for an edge that is intended for collapsing, while the second factor is to decide the next edge to be collapsed. This is usually done by evaluating the edges and sorting them in a priority queue. By processing and removing the element at the top of the queue each time, the smallest error is applied in every single step.

The objective function $f(e, v)$ defined by [22] optimizes the volume and the boundary area information about a 3D mesh and it is written in linear combination as follows:

$$\lambda f_V(e, v) + (1 - \lambda)L(e)^2 f_\beta(e, v) \tag{8}$$

Where, V represents the objective function for volume, while β represents the objective function for boundary.

Based on the setting by [19], the parameter λ can be used to weight the partial costs differently and decided to set $\lambda = 0.5$. The squared edge length $L(e^2)$ as additional factor provides identical units for both terms and makes the edge weight function invariant under scaling. Each respective function is defined as follows:

- Volume Optimization

The derived objective function aims at minimizing the unsigned volume of each individual tetrahedron as follows:

$$f_V(e, v) = v^T H_V v + 2c_V^T v + k_V$$
$$= \bar{v}^T \begin{pmatrix} H_V & c_V^T \\ c_V^T & k_V \end{pmatrix} \bar{v}$$
$$= \frac{1}{36}\bar{v}^T \sum_i \bar{G}_{V_i}^T \bar{G}_{V_i} \bar{v} \tag{9}$$

Where, $\bar{v}$ denotes corresponding homogeneous coordinates of v and$\bar{G}_{V_i}$ represents the 1-by-4 block matrix:

$$\bar{G}_{V_i} = \Big(\big(v_0^{t_i} \times v_1^{t_i} + v_1^{t_i} \times v_2^{t_i} + v_2^{t_i} \times v_0^{t_i}\big)^T - \big[v_0^{t_i}, v_1^{t_i}, v_2^{t_i}\big]\Big) \tag{10}$$

associated with triangle t_i.

- Boundary Optimization

In order to minimize the sum of squared magnitudes of the directed area vectors. The function adds up to:

$$f_\beta(e, v) = \frac{1}{4}\bar{v}^T \sum_i \bar{G}_{V\beta_i}^T \bar{G}_\beta \bar{v} \tag{11}$$

with the 3-by-4 block matrix

$$\bar{G}_{\beta_i} = \Big(\big((v_i^{e_i} - v_0^{e_i}) \times\big) - v_1^{e_i} \times v_0^{e_i}\Big)$$
$$= \big((e_{1i} \times) - e_{2i}\big) \tag{12}$$

Note that in all cases where the collapsed e is a non-boundary edge, the value of $f_\beta(e, v)$ equals zero, so that it has no bearing on the total weight. Thus, we have defined two additional weight functions for our criteria: the existing and the reciprocated.

$$weight\ LT\big(edge(v_i, v_{i+1})\big) = \lambda f_V(e, v) + (1 - \lambda)L(e)^2 f_\beta(e, v) \tag{13}$$

$$weight\ RLT\big(edge(v_i, v_{i+1})\big) = \frac{1}{\lambda f_V(e,v) + (1-\lambda)L(e)^2 f_\beta(e,v)} \tag{14}$$

III. Proposed Partition Algorithm

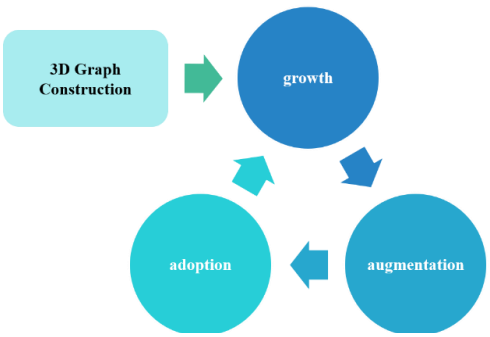


Figure 3. Our proposed algorithm for the partition process.

This paper has adapted the improved max-flow min-cut algorithm proposed by [17]. The authors have improved the empirical performance of the standard augmenting path techniques on graph. Maximum-flow minimum-cut (max-flow min-cut) is the combinatorial optimization approach on a graph to find the maximum amount of flow that begins from source, s to sink,t saturates several edges (augmenting paths), resulting the division of nodes into two disjoint sets S, T corresponding to aminimum cut [25]. The duality relationship between max-flow and min-cut has been proved in [26].

The augmenting path[1] found by the improved max-flow min-cut algorithm is not necessarily the shortest one unlike [27]. Thus, the time complexity of the shortest augmenting path is no longer valid.

The source vertex, s and sink vertex, t (also called terminal vertices) are treated as hard constraints imposed by user to facilitate the binary segmentation. Intuitively, these hard constraints provide hints on how the 3D mesh intends to segment.

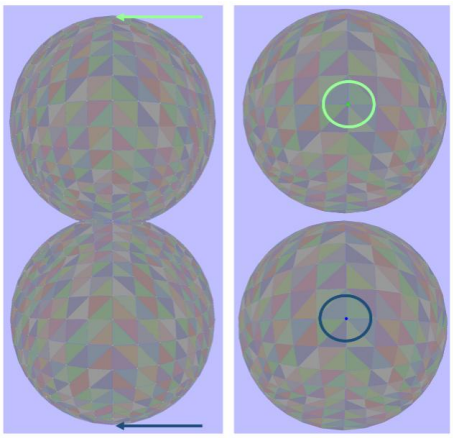
Figure 3 illustrates our adapted partition process. There are four involved modules: 3D graph construction, growth, augmentation and adoption. The description for each respective module is explained as follows:

D. 3D Graph Construction

We have to transform the semantic information of the 3D duo-mesh into a graph so that the improved max-flow min-cut algorithm [17] can be used as partition process. The steps can be outlined as follows:

1. Firstly, the terminal vertices (source vertex, s and the sink vertex, t) are selected by user as depicted in Figure 4. Thisis to provide some guidance to the algorithm that the top sphere belongs to the source sets, S while the bottom spherebelongs to the sink sets, T.
2. Count the number of vertices, *count_nodes (excluding the terminal vertices).*
3. Count the number of edges, *count_edges(excluding the edge adjacent to terminal vertices).*
4. Find all the connected pair of adjacent vertices that form the edges *(excluding those that are adjacent to the terminal vertices)* into a vector array, *tmpNormalEdges.*
5. Sort *tmpNormalEdges* in sequence, remove the duplicate vertices and store in *normalEdges*.
6. Construct a weighted directional graph, G. The connectivity between the intermediate vertices *(add_node)* and edges*(add_edge)*are formed(without the terminal vertices). At the same time, the weight values are assigned to each respective edges *(add_tweights)* based on the defined weight functions (explained in Section 2).

The constructed weighted directional graph, G is then applied to the next modules. The term nodes and paths are used torepresent vertices and edges respectively in the next section. The algorithm maintains two non-overlapping search trees S and$T$ with the roots at the source, s and sink, t throughout the process. Therefore, four types of vertices are used during theformation of search trees:



(a) (b)

Figure 4. (a) 3D duo-spheres (facets). (b) Top: The source vertex (in green), s selected by user. Bottom: The sink vertex (in blue), t selected by user.

- Free: Nodes that are neither in search tree S nor search tree T (nodes that have no parents).
- Active: Nodes that at the outer border of the search trees.
- Passive: Nodes that at the outer border of the search trees.
- Orphan: Nodes that are adjacent to the parents becomes invalid due to saturated edges.

The next three modules: growth, augmentation and adoption are repeated iteratively until an active node in one of the trees detects a neighbouring node that belongs to the other tree (details can be found in [17]). Consequently, an augmenting path is found. The overall structure of the algorithm is as follows:

Algorithm 1 The overall structure of the improved max-flow min-cut algorithm.

Require: Initialization: source node, $\boldsymbol{S} = \{\boldsymbol{s}\}$, sink node, $\boldsymbol{T} = \{\boldsymbol{t}\}$, active nodes, $A = \{s,t\}$, orphan nodes, $O = \{\emptyset\}$

1:
2: **while** true **do**
3: **grow** $\boldsymbol{S}$ or $\boldsymbol{T}$ to find an augmenting path $\boldsymbol{P}$ from $\boldsymbol{s}$ to $\boldsymbol{t}$
4:
5: **if** $\boldsymbol{P} = \{\emptyset\}$ **then** terminate
6: **augment** on $\boldsymbol{P}$
7: **adopt** orphans
8: **end while**

[1]A path for flow, f in a graph, G from source node, s to sink node, t consisting of edges in the support of G_f (edges with positive capacity in G_f)

Table 2. Comparison Results Among Various Weight Functions including Ours (Highlighted) on the 3D Duo-spheres

Tested Weight Functions	Acronym	Results
Stan Melax	M	×
Reciprocal Melax	RM	√
Kims-Levin	KL	×
Reciprocal Kims-Levin	RKL	△
Lindstrom-Turk	LT	×
Reciprocal Lindstrom-Turk	RLT	×

√:Successfully partition the 3D mesh into two segments.
△: Inconsistent partition- sometimes successful, and Sometimes failed.
×: Unsuccessfully partition the 3D mesh into two segments.

IV. Results & Discussion

We have conducted some experiments over the 3D duo-spheres using our proposed algorithm.

Soft Constraints Definition. Six various weight functions are tested and the results are summarised in Table 2. It has proved that by using RM as the weight function produces the best influence on the partition among others.

Partitioning. Two terminal nodes as hard constraints for segmentation are imposed by user as shown in Figure 4. The source node is denoted in green colour while the sink node is denoted by blue colour. These hard constraints provide some indications to certain nodes definitely have to be part of the source and certain nodes have to be part of the sink. These selected terminal nodes are preferred to be salient, peak, saddle or centroid vertices in order to maintain the consistency in our experiment.

As presented in Table 2, the existing weight functions of M, KL and LT are insufficient for augmentation stage due to the weight values for the terminal edges are equal to zero. In other words, these edges are saturated. Thus, the algorithm is unable to grow both search trees and consequently, all the nodes are assigned to source, s by default as illustrated in Figure 5(a), (b) and (c).

Nevertheless, our proposed weight functions of RM is able to produce two segments (source, s and sink, t) due to the calculated weight values are large enough to saturate the potential paths from source to sink, including the terminal edges. Thus, the maximum flow is achieved and corresponding minimum cut at the cutting edges of the 3D duo-spheres.

However, the weight function of RLT produces only one segment which is the sink set, T. This is because the weight values for the edges are more than zero, thus paths can be saturated from the source, s to the sink, t. The order of processing (First-In-First-Out) the active nodes and orphans may also have the significant effects on this result.

On the other hand, the weight function of RKL sometimes produces binary segments, but mostly failed to do so. This scenario might be due to the unpredictably changes during the adoption stage.

Overall, the experiment has proved that our proposed weight function Reciprocal Melax (RM) is the optimal weight function to be used as the soft constraint together with the adapted max-flow min-cut algorithm for binary segmentation.

V. Conclusions & Future Work

We have tested our proposed binary segmentation algorithm on 3D duo-spheres. Six weight functions are used as the soft constraints including three of our newly defined weight functions. These weight functions are used to facilitate the partition process. We have adopted the improved maximum-flow minimum-cut algorithm as partitioning in the 3D domain. As a conclusion, our proposed weight function of Reciprocal Melax (RM) is effective and robust for binary segmentation on the 3D duo-spheres.

The major limitation of our proposed approach is the requirement of the 3D duo-spheres to be regular and consists of two separable geometric shapes. Hence, there are several possibilities in future directions for this work.

A straightforward extension to our approach would be applying the proposed weight function of RM together with the proposed maximum-flow minimum-cut on the complex 3D meshes. Apart from that, more analysis and studies are required in the partition algorithm in order to produce accurate segmentation.

Acknowledgment

Many thanks to the author's supervisor, Professor Ewe Hong-Tat and co-supervisor Professor Lee Byung-Gook for their guidance and patience. It would not have been possible without their strong support. This research was also supported by the internal seed fund of University Tunku Abdul Rahman Research Fund (UTARRF) with the project number:
IPSR/RMC/UTARRF/2014-C1/S02 (6200/S47).

References

[1] Takahashi, S., Wu, H.Y., Saw, S.H., Lin, C.C. and Yen, H.C. 2011, September. Optimized topological surgery for unfolding 3d meshes. In Computer graphics forum. 30(7), 2077-2086. Oxford, UK: Blackwell Publishing Ltd.

[2] Vanek, J., Galicia, J.G., Benes, B., Měch, R., Carr, N., Stava, O. and Miller, G.S. 2014, September. PackMerger: A 3D print volume optimizer. In Computer Graphics Forum. 33(6), 322-332.

[3] Apaza-Agüero, K., Silva, L. and Bellon, O.R. 2015, September. Mesh segmentation with connecting parts for 3D object prototyping. In 2015 IEEE International Conference on Image Processing (ICIP). 16-20. IEEE.

[4] Bücking, T.M., Hill, E.R., Robertson, J.L., Maneas, E., Plumb, A.A. and Nikitichev, D.I. 2017. From medical imaging data to 3D printed anatomical models. PloS one.12(5), p.e0178540.

[5] Katz, S. and Tal, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM transactions on graphics (TOG).*22*(3), 954-961.

[6] Sam, V., Kawata, H. and Kanai, T. 2012. A robust and centered curve skeleton extraction from 3D point cloud. Computer-Aided Design and Applications, 9(6), 869-879.

[7] Gao, L., Lai, Y.K., Huang, Q.X. and Hu, S.M. 2013, May. A data-driven approach to realistic shape morphing. In Computer graphics forum. 32(2pt4), 449-457. Oxford, UK: Blackwell Publishing Ltd.

[8] Jiang, L., Ye, J., Sun, L. and Li, J. 2019. Transferring and fitting fixed-sized garments onto bodies of various dimensions and postures. Computer-Aided Design.106, 30-42.

[9] Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S. and Dobkin, D. 2004. Modeling by example. ACM transactions on graphics (TOG). 23(3), 652-663.

[10] Dubrovina, A., Xia, F., Achlioptas, P., Shalah, M., Groscot, R. and Guibas, L.J. 2019. Composite shape modeling via latent space factorization. In Proceedings of the IEEE International Conference on Computer Vision. 8140-8149.

[11] Philip Shilane, M.K.T.F. and Min, P. 2005. Princeton shape retrieval and analysis group: Princeton shape benchmark.

[12] Thomas Ebner, L.B.-G. and Gumhold, S. 2006. Development of a surface simplification tool using the kmmcs data structure, Master's thesis, University of Technology Dresden.

[13] Shamir, A. 2008, September. A survey on mesh segmentation techniques. In Computer graphics forum. 27(6), 1539-1556. Oxford, UK: Blackwell Publishing Ltd.

[14] Cohen, J., Olano, M. and Manocha, D. 1998, July. Appearance-preserving simplification. In Proceedings of the 25th annual conference on Computer graphics and interactive techniques. 115-122.

[15] Luebke, D.P. 2001. A developer's survey of polygonal simplification algorithms. IEEE Computer Graphics and Applications.21(3), 24-35.

[16] Melax, S. 1998. A simple, fast, and effective polygon reduction algorithm. Game Developer, 11, 44-49.

[17] Boykov, Y. and Kolmogorov, V. 2004. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE transactions on pattern analysis and machine intelligence, 26(9), 1124-1137.

[18] Saw, S. H. Han, Y.-D. Lee, B.-G. and Ewe, H.-T.2015. A weighted min-cut max-flow approach for 3d mesh segmentation, China Academic Journal Electronic Publishing House, 574–577.

[19] Ebner, T. 2006. Development of a surface simplification tool using the kmmcs data structure), Master's thesis.

[20] Hoffman, D.D. and Singh, M. 1997. Salience of visual parts. Cognition. 63(1), 29-78.

[21] Hoffman, D.D. and Singh, M. 1997. Salience of visual parts. Cognition. 63(1), 29-78.

[22] Lindstrom, P. and Turk, G. 1999. Evaluation of memoryless simplification. IEEE Transactions on Visualization and Computer Graphics. 5(2), 98-115.

[23] Dyn, N., Hormann, K., Kim, S.J. and Levin, D. 2001. Optimizing 3D triangulations using discrete curvature analysis. Mathematical methods for curves and surfaces. 1, 135-146.

[24] Desbrun,M., Meyer, M.,Schröder, P.,Barr, A.H. 2000. Discrete differential-geometry operators in nD.The Caltech Multi-Res Modeling Group. preprint

[25] Ford Jr, L.R. and Fulkerson, D.R. 2015. Flows in networks. Princeton university press, London.

[26] William, J. C., William, H. C., William, R. P. and Alexander, S. 1997. Combinatorial Optimization. A Wiley-Interscience, New York. 1 ed.

[27] Abdullah, N. and Hua T.K.2017. The Application of the Shortest Path and Maximum flow with bottleneck in Traffic Flow of Kota Kinabalu. Journal of Computer Science & Computational Mathematics. 7(2), 37-43.

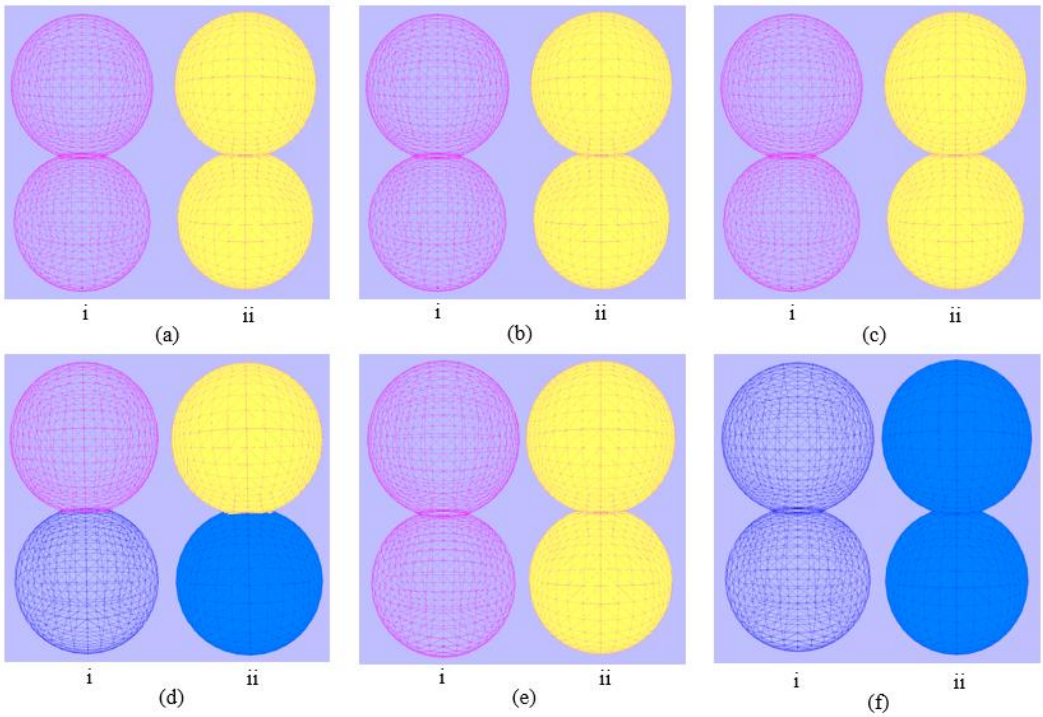


Figure 5. Binary segmentation results on the 3D duo-spheres. (a)i, (b)i, (c)i, (d)i, (e)i and (f)i Wireframe segmented 3D mesh. (a)ii, (b)ii,(c)ii, (d)ii, (e)ii and (f)ii Facets segmented 3D mesh. (a), (b), (c), (e) Unsuccessful binary segmentation using the weight functions of M, KL,LT and RKL (consists of only one source segment, S which coloured in red for wireframe 3D segmented mesh but yellow for facets 3Dsegmented mesh). (f) Unsuccessful binary segmentation using the weight function of RLT (consists of only one sink segment, T whichcoloured in blue for wireframe 3D segmented mesh and facets 3D segmented mesh). (d) Successful binary segmentation using the weightfunction of RM (consists of two segments: source, S and sink, T which coloured in pink and blue respectively for wireframe 3D segmentedmesh, while yellow and blue respectively for facets 3D segmented mesh).