ISSN 2278-3091

Volume 9, No.5, September - October 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse325952020.pdf

https://doi.org/10.30534/ijatcse/2020/325952020

Overlapping Triangle by using Modification of Advancing Front Mesh Method

Noorehan Awang[1,2,], Rahmita Wirza Rahmat[1], Ng Seng Beng[1], Azmi Jaafar[1], Puteri Suhaiza Sulaiman[1] , Mohd Rijal Ilias[3]

[1]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Selangor, Malaysia. gs43550@student.upm.edu.my, rahmita@fsktm.upm.edu.my, ngsengbeng@upm.edu.my,azmij@upm.edu.my, psuhaiza@upm.edu.my

[2]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM),Seremban, Negeri Sembilan, Malaysia, noorehan@uitm.edu.my

[3]Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM),Shah Alam, Selangor, Malaysia,mrijal@uitm.edu.my

ABSTRACT

In reverse engineering, often there is a chance of having missing entities while using hardware-based data procurement methods, which is perceived as a hole in the prototype. So as to develop an error-free prototype, the holes need to be eliminated.Therefore this paper proposes an enhanced algorithm of filling holes for 3D objects based on Advancing Front Mesh (AFM) method. The problem was solved for overlapping triangle occur in a triangle mesh where during triangle creation some triangle overlaps other triangle cause the triangle creation stop. Thus, in the algorithm, object was chosen and the procedure namely hole detection and hole filling was involved. The experiment demonstrates the result of enhanced algorithm devised for solving the overlapping triangles by the three objects.

Key words : advancing front mesh, hole filling, overlapping triangle, triangular mesh.

1. INTRODUCTION

In many modelling applications and computer-aided engineering, mesh-based or triangulated prototypes are extensively used to depict an object. In certain instances - wherein there is a simulation of natural processes - such as geo-technical or bio-medical applications or reverse engineering, a triangulated prototype is the only depiction possible. Furthermore, often there is a chance of having missing entities while using hardware-based data procurement methods, which is perceived as a hole in the prototypes.So as to develop an error-free triangulated prototype, the holes need to be eliminated. When there is a hole over a non-planar area, the intrinsic geometry must be approximated in such a way that now the filled hole fits in the calculated shape. Sadly, the hole filling operation, when conducted manually, takes a lot of time. Therefore, it is preferable to design an automatic algorithm which accurately fills the hole and approximates the geometry that is missing.

There was a lot of research recently conducted on filling holes [1-14]. Several prototypes comprise geometric factors that are not compatible with processes and programs that utilise CAD geometry. Silver faces and cracks, loss of correlation between surfaces and edges and inadequate accuracy could be some of the factors leading to this incompatibility.

Many research workers handle the issue by directly fixing the geometry[15-17]. Steinbrenner et. al [18] devised an edge merging formula for pairing up common edges. This way, it becomes possible to have computationally water-tight adjoining meshes through the edge curves that they share. However, there is a limitation to this formula. It is not applicable on two edge curves which are adjacent on one end and deviated on the other, thus not being able to repair the errors caused by irregularity in tolerances. A spatial partitioning algorithm was developed by Murali et. al [19] to establish an error-free surface limit of a prototype. Nevertheless, this algorithm could not tackle geometric intersections.

User-interactive tools were made by Peterson et. al [20] to efficiently prepare CAD geometries for mesh formation. Likewise, Morvan et. al [21-22] explicated a user-controlled virtual environment which provided tolls through which models could be manually corrected. However, these user-controlled processes were quite inefficient and cumbersome for large prototypes. Users would easily miss errors, thus introducing new errors and the programme was not robust enough to handle this.

Turk et. al [23] concentrated on building a topological methodology to remove polygon overlaps wherein they are clipped against each other so as to create polygonal prototypes using range data. Sadly, this method failed to take into account uneven topology and geometric intersections.

Yau et. al [24] proposed a surface reconstruction process for the comprehensive stitching of STL prototypes. This too was not quite efficient because the process took into consideration all data points and it would take a lot of time for an intricate prototype to get ‘stitched’. Moreover, this algorithm could not handle complex or minute components such as a fine comb shaped prototype.

Hu et. al [25] devised a process that employed an overlay grid methodology to fill in the gaps and holes and eliminate overlapping regions. But, in this case, the resulting geometries could be different from the actually expected ones. Dolenc and Makela [26] tried using a local method to fill in these cracks and gaps on the surface prototypes, but when there were numerous such cracks and gaps, this method caused a tremendous rise in the amount of polygons which defined the surface. Meier and Sheng [27] tried to zip small holes between surfaces. This was quite like the method described in this proposal. However, Meir and Sheng’s method concentrated only on topological construction. It did not cater to geometric intersections. This research focuses on the problem of an overlapping triangle that is created while constructing a triangle inside a gap. The new enhanced algorithm was created in order to tackle the issue so as to avoid repeating the process for creating a triangle.

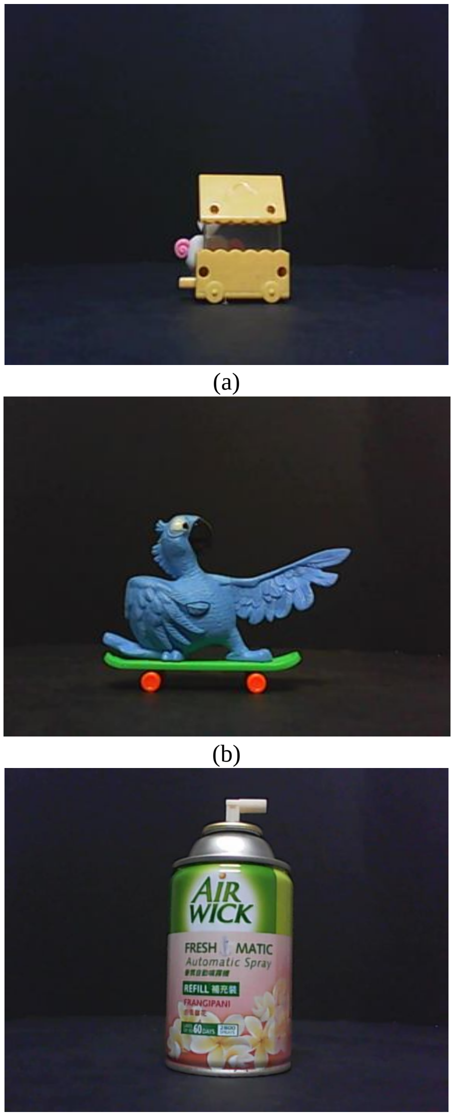
The problems focused for the research was overlapping triangle which happen during construction of triangle inside a hole. The new enhanced method was introduced to handle the problem so that the process doesn’t need to be redo for triangle creation.The remainder of this paper is organized as follows : Section 2 show the classification of data for all Subject. Section 3 shows boundary identification for Subject 1 which was not been explained in previous chapter for pro-processing process. Section 4 shows the holes filling using enhanced method for AFM handling the problem of overlapping triangle for all Subject. Section 5 is the conclusion of this research underline.

2. . CLASSIFICATION OF DATA

Three subjects are used in this research as in Figure 1. Table 1 enlists these three subjects which are used to tackle overlapping of triangle. Subject 1 is the model of a yellow board that has a transparent centre. The figure depicts only the left ear of a white cartoon figure. Subject 2 is a blue coloured bird. It stands on a green coloured skateboard that has orange wheels, and its left wing is stretched out. Subject 3 depicts a cylindrical shape body that has an L-shaped top. It is very colourful and has high contrast letters that read “air wick”.

Table 1: Classification of Subject

Subject Number	Subject
1	Popcorn (4th view)
2	Rio (1st view)
3	Mentos(1st view)


(a)

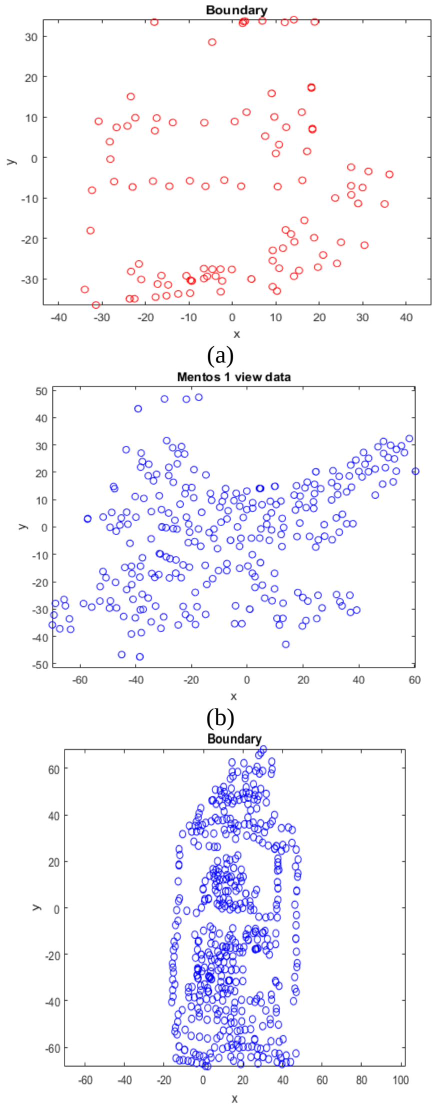
(b)

(c)

Figure 1: Subject in 3D
(a) Subject 1 (b) Subject 2 (c) Subject 3

We need to clear all control points before using them for surface reconstruction. We also need to identify repeated or similar control points and eliminate them as we want to avoid redundant control points. Initially, there were 146 control points for Subject 1 and after eliminating similar control points, it went down to 132(Figure 2(a)). Likewise, for the front view of Subject 2, there were 608 control points, and after deleting similar control points, the number came down to 570. It is portrayed in Figure 2(b). A control point is missing at Subject 2’s mouth.

Originally, there were 608 control points. After deleting similar points, 572 control points were left as are shown in Figure 2 (c). We can see that a control point is missing on Subject 3’s handle. That is because the handle was white and while scanning it, the point was lost. Therefore, in the next step, so as to construct the boundary accurately, we will have to add a control point to the handle.



(a)

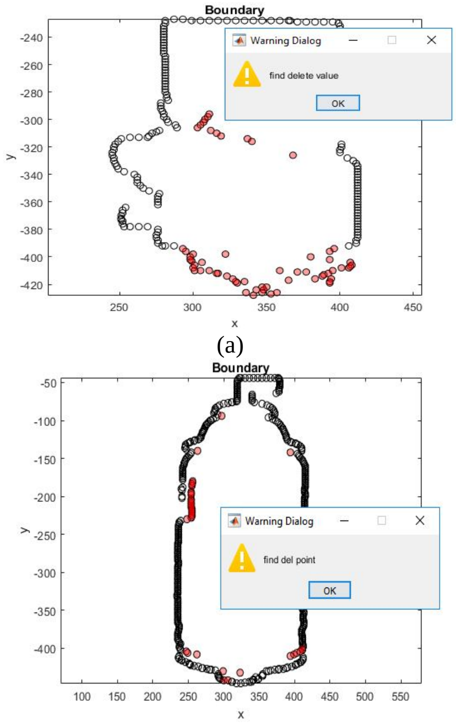
(b)

(c)

Figure 2: Control Point for Subject
(a) Subject 1 (b) Subject 2 (c) Subject 3

3. BOUNDARY IDENTIFICATION

The boundary identification procedure only for Subjects 1 and 3 will be explained in this subtopic because we have already discussed all other subjects in the previous chapters. We need to do a few processes for identifying each Subject's boundary. First of all, we will use edge detection to identify sharp qualities of the Subject. The Canny edge detection method is one of the typical methods used in the industry. The Canny edge detection method is a very efficient way of finding edges. It separates the noise from the Subject before identifying its edges. It's a standard method that uses threshold value. Figure 3 depicts the outcome of Canny edge detection technique. As per this figure, some control points had to be deleted because the data did not represent Subject's boundary and maybe the control point was located inside the boundary or it was noisy data. Total 65 control points were deleted for Subject 1 as shown in Figure 3(a). The evaluation method of Canny edge detection uses the high threshold (T_High = 0.0099) and low threshold (T_Low = 0.099). The outer boundary is detected by the scan line algorithm and some control points which are not required have been eliminated by the user. Figure 3(b) depicts the 33 data points that were eliminated.

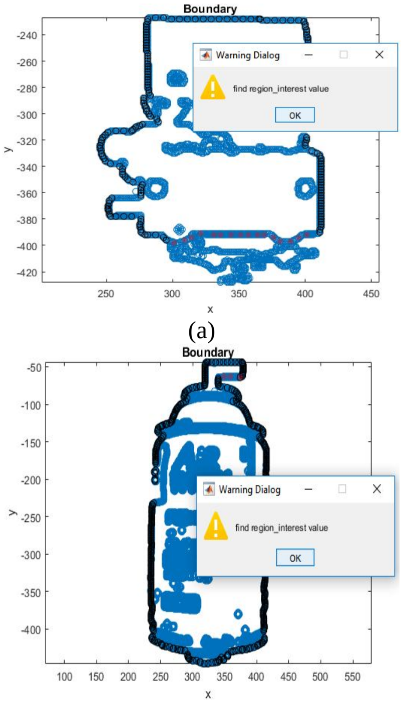


(a)

(b)

Figure 3 : Delete Outlier
(a) Subject 1 (b) Subject 3

We determined the boundary of all the Subjects by eliminating the outlines. Despite this, some features may be missing in a Subject. The following figure depicts how, by applying the threshold technique, the object in the foreground can be extracted from the original image. At the same time, some missing features can be extracted too. In the next step, the point at the missing area can be selected and added to the boundary. In this way, we obtain the full boundary of the Subjects. Figure 4(a) depicts the manually added control points, where 16 control points were added to the original boundary of Subject 1. In front view control points, it is seen that Subject 3's handle is missing a control point because the handle was white and some points were lost while scanning. Figure 4(b) depicts the two control points that were manually added in order to obtain a better boundary for Subject 3.



find region_interest value

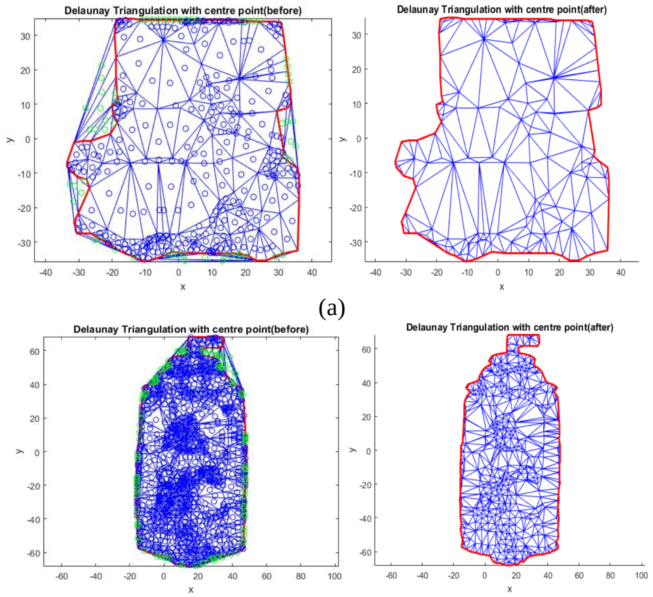
(a)

find region_interest value

(b)

Figure 4 : Threshold method
(a) Subject 1 (b) Subject 3

The boundary object thus obtained was then applied to the Delaunay triangulation, as shown in red. For Subject 1, the control points provide a final Delaunay triangulation connectivity list of 248x3, whereas, for Subject 3, we obtain a 1123x3 list. The centres of all triangles were calculated and are displayed as a point in Figure 5(a).The point shown in blue is that which lies inside the boundary. The point outside the control polygon is highlighted in green. The green data point outside the polygon was subsequently removed to get the results, as shown in Figure 5(b).



(a)

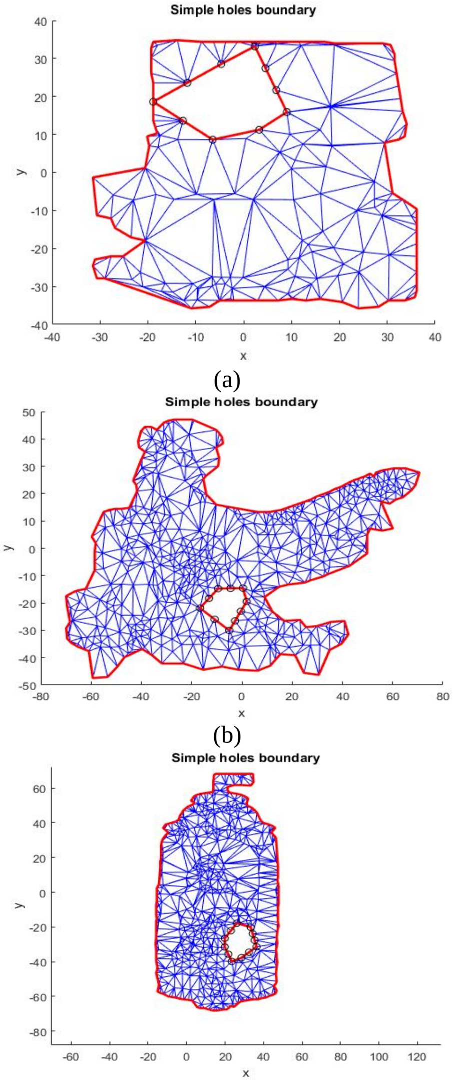
(b)

Figure 5: Constraints Delaunay Triangulation

(a) Subject 1 (b) Subject 3

4. HOLE IDENTIFICATION

The area of all the triangles was summed up and using the condition that the triangle with an area higher than the mean area was considered at the hole. As demonstrated in Figure 6, the user may choose the hole for all the subjects.



(a)

(b)

(c)

Figure 6 : Holes for Subject
(a) Subject 1 (b) Subject 2 (c) Subject 3

5. ADVANCING FRONT MESH (AFM) METHOD

Advancing front mesh (AFM) method was proposed by [28] and represented with six steps:

Step 1: Three points was chosen as a starting point label as A,B and C

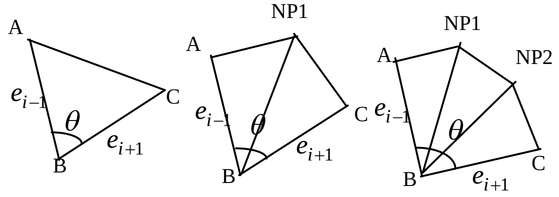
Step 2: Angle θ_i between the edges AB and BC was calculated

Step 3: Three different criteria was categorised accordingly as below (see Figure 7).

(a) If $\theta \leq 75^0$,point A and C was connected to create a triangle,no new point introduced (Figure 7(a)).

(b) If $75^0 < \theta < 135^0$, new point created label as NP1 and this created two triangle Figure 7(b).

(c) If $\theta \geq 135^0$, two new point NP1 and NP2 create a three triangle as shown in Figure 7(c).



(a) (b) (c)

Figure 7 : Three rules of angle in AFM
(a) $\theta \leq 75^0$ (b) $75^0 < \theta < 135^0$ (c) $\theta \geq 135^0$

Step 4: Threshold was use where distance between new point and B was used $\overline{NP1B}$ (this was used as a reference to search nearest active point)

$$Threshold = \frac{1}{4} \times (\overline{NP1B}) \qquad (1)$$

Step 5: Update the control points

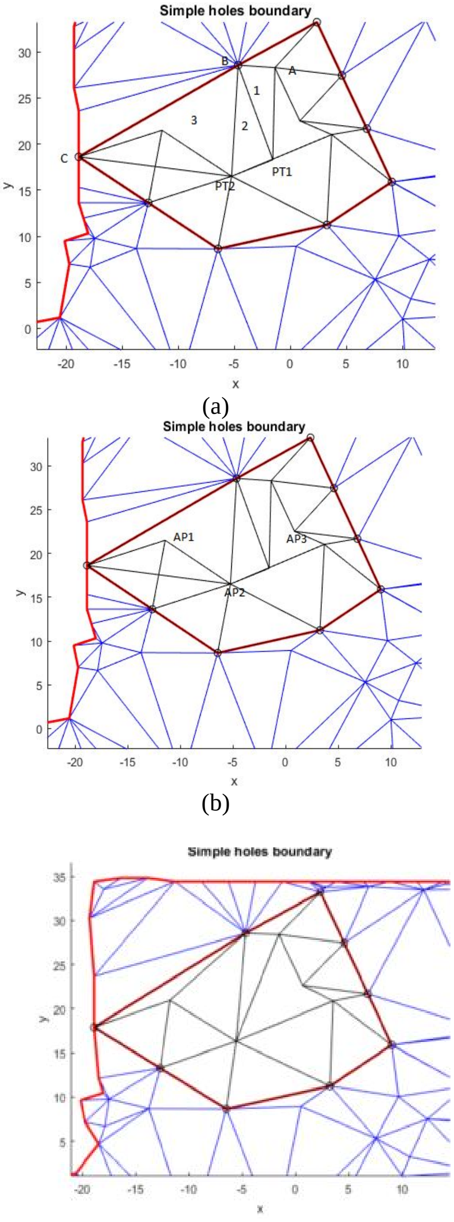
Step 6: Repeat step 1 through step 4 until the whole region of holes has been created and fill with triangle

There is a challenge when the triangle creation process creates an overlapping triangle. This causes the process to cease since triangle creation cannot proceed further, and the chances are that the system may need to repeat the process. An enhanced method has been devised to tackle this challenge.

Subject 1

In the seventh iteration for Subject 1, it is observed that the two triangles created require two new points PT1 and PT2 to be present near to the closest active point. The triangle-created overlap requires that NP1 and NP2 be accommodated at another location. This is highlighted in Figure 8(a).

EDG may be used to find the nearest active point for all the triangles, and the remaining data points are saved. Figure 8(b) shows the three active points. Next, the algorithm checks whether there is an active point located inside the triangle created by points PT1 and PT2. At this point, AP1 has not been associated with any triangle creation and hence is chosen as NP1, also because it lies closest to PT2. The next step is to determine the distance between NP1 and the active point. AP2 was identified as that point and selected as NP2. This enhanced version creates the results, as shown in Figure 8(c).



(a)

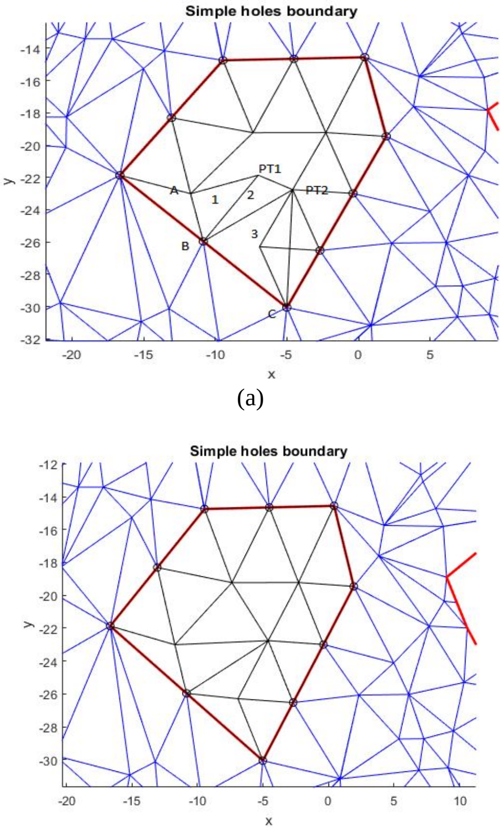
(b)

(c)

Figure 8 : Subject 1
(a) Problem (b) Active Point (c) Final Result

Subject 2

Triangle creation for Subject 2 has a point where $\theta > 135^0$, causing the edge to overlap with that of another triangle, thus generating an error and halting the process. Given this challenge, an active point needs to be located within the triangle shown in Figure 9(a). The required active point may be picked up from the EDG since the list contains the selection candidates for filling holes during the next iteration. Of the three triangles obtained, triangle 3 crosses to another triangle, and we find an active point within this triangle. However, in this case, PT1 is located to the left and hence cannot be a candidate to be considered as NP1 but maybe for NP2. To locate NP2, NP1 was set as the reference point and the distance from NP1 to all active points is calculated. NP2 is located, as shown in Figure 9(b).



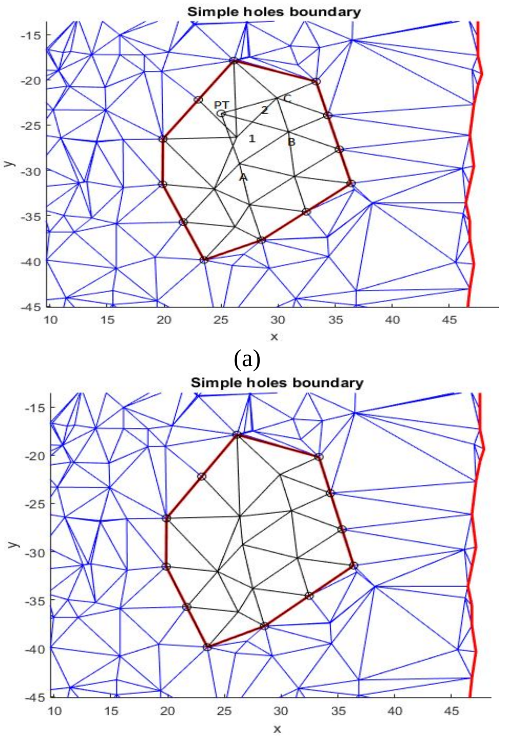
(a)

(b)

Figure 9 : Subject 2
(a) Problem ((b) Final Result

Subject 3

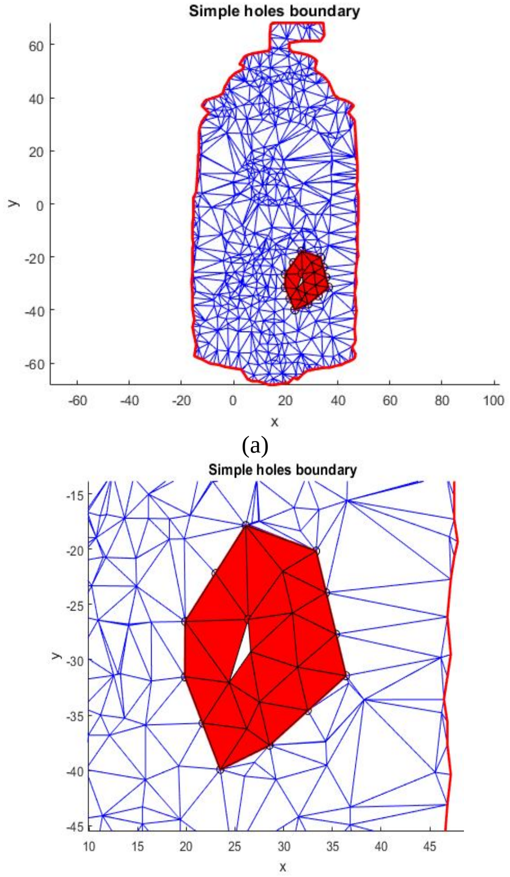
During the second cycle of the 14th iteration, it was found that PT was not connected to any closest data points; therefore, NP could not be found; this is highlighted in Figure 10(a). Adhering to the EAFM method, the three closest points were identified and marked as red data points 1, 2, and 3. Those three points were connected, forming a triangle. Now the requirement is to identify which of these three may be NP. The distance of each point was determined, and point 3 was set as NP because points 1 and 2 may not be used as active points given; they have already been used. The result thus obtained is stated in Figure 10(b). The extension of Subject 3 is a triangle that was generated automatically and is highlighted in white. The following figure 11 highlights the final triangle thus created.



(a)

(b)

Figure 10: Subject 3
(a) Problem (b) Final Result



(a)

(b)

Figure 11: Extension case
(a) Extension case (b) close up

Enhanced Algorithm

Case I: $75^0 < \theta < 135^0$

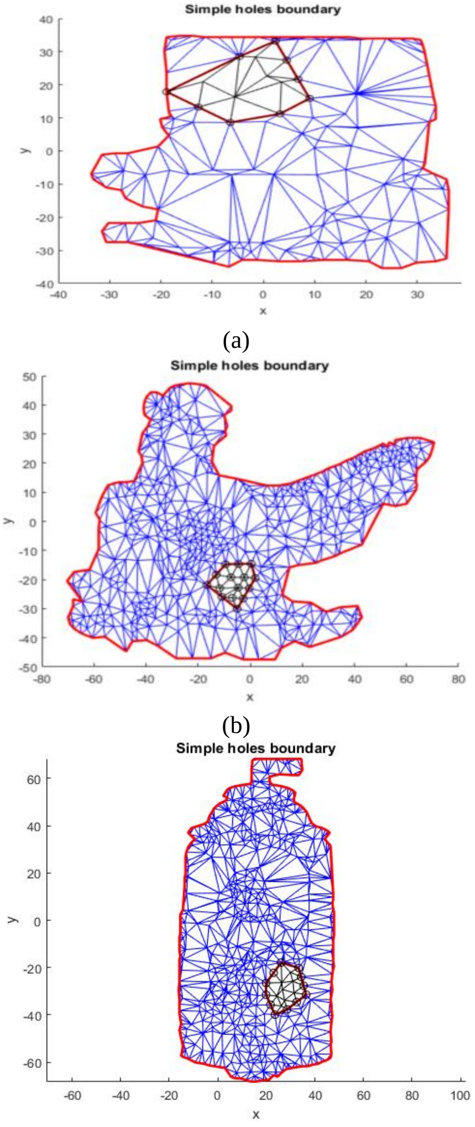
Check either there are any active point inside the triangle created by new created point (PT1). Use EDG which store all active point which are left.The active point inside the triangle was considered to be the new point (NP)

Case II: $\theta > 135^0$

Check either there are any active point inside the triangle created by first new created point (PT1) and second new created point (PT2).Use EDG which store all active point which are left.The active point inside the triangle was considered to be the first new point (NP1) Calculate the nearest active point NP1 and this point become the second new point (NP2).

Final Result

Final result obtained after the enhanced method was shown in Figure 12 below:



(a)

(b)

(c)

Figure 12: Final Subject
(a) Subject 1 (b) Subject 2 (c) Subject 3

Table 2: Number of vertices and triangles before and after for three subject

Model	Number of vertices		Number of triangles		Running Time
	before	after	before	after	
Subject 1	132	137	248	265	46.7969
Subject 2	570	576	1123	1141	82.4532
Subject 3	572	577	1123	1147	45.7344

6. CONCLUSION

So far, the conclusion of the study is that the enhanced algorithm devised for solving the overlapping triangles used three objects. During boundary identification, a missing point was discovered, which could have been lost during scanning. Therefore, the identification of the boundary required the deletion and addition of one point. The algorithm was continued until overlapping triangles required the adjustment of new point to another suitable position. This problem stops the triangle creation process, and it might be required to restart the process from the beginning. The automatically constructed white triangle may be understood as an extension of the enhanced algorithm. Once the process was completed, the triangles before and after were counted, and the comparison is presented in Table 2. The comparison also considers the running time for each subject. It was found that Subject 2 had the maximum number of vertices, while Subject 3 led to the formation of the highest number of triangles. That said, Subject 3 had the shortest running time since there were not many deletions and additions of vertices

ACKNOWLEDGEMENT

I would like to give my gratitude and thanks to Universiti Teknologi Mara (UiTM) for funding grant for the the journal publication

REFERENCES

1. G. Luo, Y. Zhu, and B. Guo. **Fast MRF-Based Hole Filling for View Synthesis**, *IEEE Signal Processing Letters*, *25*(1), 75-79, 2018.
2. N. Adhikary and B. Gurumoorthy. **A slice-based algorithm for automatic and feature-preserving hole-filling in a CAD mesh model**, *Computer-Aided Design and Applications*, 15:6, 780-795,2018.
3. M. A. Fortes, P. González, A. Palomares and M. Pasadas. **Filling holes with geometric and volumetric constraints**,*Computers & Mathematics with Applications*, *74*(4), 671-683, 2017.
4. C. Dinesh, I.V. Bajić, and G. Cheung. **Exemplar-based framework for 3D point cloud hole filling**, In *Visual*

Communications and Image Processing (VCIP), 2017 IEEE,pp. 1-4,IEEE, 2017, December.

5. C.W. Wang, T.C. Yang, S.H. Chiang, and T. Wang. **Identifying and filling occlusion holes on planar surfaces for 3-D scene editing**, In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017*,pp. 251-254, IEEE, 2017, December.
6. C. Xia and H.Zhang. **A fast and automatic hole-filling method based on feature line recovery**, *Computer-Aided Design and Applications*, *14*(6), 751-759,2017.
7. Y.Fu and T.Wu. **Research on depth hole filling algorithm based on Kinect**, In *Signal Processing, Communications and Computing (ICSPCC), 2017 IEEE International Conference on*, pp. 1-5, IEEE, 2017, October.
8. S. C. Lin, T. K. Shih and H. H. Hsu. **Filling holes in 3D scanned model base on 2D image inpainting**, In *Ubi-media Computing and Workshops (Ubi-Media), 2017 10th International Conference on*,pp. 1-6, IEEE, 2017, August.
9. L. Yang, Q. Yan & C. Xiao. **Shape-controllable geometry completion for point cloud models**, *The Visual Computer*, 33(3), pp. 385-398,(2017).
10. Hsi-An Chiao Tung Ta Hsueh. **Rapid Mesh Hole Filling in Body Model**, *Journal of Xi'an Jiaotong University*,52(8), pp.37-42, (2018).
11. X. Wang,J. Hu, D. Zhang, L.Guo,H. Qin and A.Hao. **Multi-scale geometry detail recovery on surfaces via empirical mode decomposition**,*Computers & Graphics*, 70, pp. 118-127,(2018).
12. N.Luo, & Q.Wang. **Fast self-repairing region growing surface reconstruction algorithm for unorganised point cloud data**, *International Journal of Computer Applications in Technology*, 56(2), pp. 121-131,(2017).
13. K. M.Tran and M. H. Tran. **Filling Hole on the Surface of 3D Point Clouds Based on Reverse Computation of Bezier Curves**, In *Information Systems Design and Intelligent Applications*, Springer, Singapore, pp. 334-34,(2018).
14. S.Zhao, W. Zhang, W.Sheng and X. Zhao. **A Frame of 3D Printing Data Generation Method Extracted from CT Data**, *Sensing and Imaging*, 19(1), 12,(2018).
15. L. Praneetha S.K , A.K , G. Viharika R, D. Sree M and V. P.**Automated Leaf Disease Detection in Corn Species through Image Analysis,** *International Journal of Advanced Trends in Computer Science and Engineering*,Volume 8, No.6, pp. 2893-2899,2019
16. Dr.A.Nagarajan and J. V. Wason. **Machine Learning Approach to Predict Lung Cancer using CT scan Images,** *International Journal of Advanced Trends in Computer Science and Engineering*,Volume 8, No.6, pp. 2974-2976,2019.
17. J.Ade , A. Ali ,S. ,A. K. Rayza, E.Latifah, N.Endang and W.Teguh .**Representation Learning for Dental Image Identification,** *International Journal of Advanced Trends in Computer Science and Engineering*,Volume 8, No.6, pp. 3476-3480,2019.
18. J.P. Steinbreiner, N.J Wyman and J.R Chawner. **Procedural CAD Model Edge Tolerance Negotiation for Surface Meshing**, *Engineering with Computers*, Vol. 17, pp. 315-325,(2001).
19. T.M Murali and T.A Funkhouser. **Consistent Solid and Boundary Representations from Arbitrary Polygonal Data**, Proc. 1997 Simposium on *Interactive 3D Graphics*,Providence, Rhode Island (A. Van Dam, Ed.),pp 155-162, Asscoc. Comput. Mach,Press,New York,(1997).
20. N. Anders Peterson and Kyle K.Chand. **Detecting Translation Errors in CAD Surfaces and preparing geometries for Mesh Generation,***Center for Applied Scientific Computing*, Lawrence Livermore National Labs, Livermore, CA 94551,(Aug 2001), UCRL-JC-144019.
21. S.M Morvan and G.M Fadel. **IVES:An Interactive Virtual Environment for the Correction of STL Files**, Proc. *Conf. Virtual Design.*, Univ. Of California,Irvine, (Aug,1996).
22. S.M Morvan and G.M Fadel. **An Interactive Correction of STL Files in a Virtual Environment**, *Proc. Solid Freeform Fabrication Symposium*, U.Texas, Austin,Texas USA, (Aug,1996).
23. G. Turk and M. Levoy. **Zippered Polygon Meshes from Range Images**, Prof. of *ACM SIGGRAPH*, pp. 311-318,(1994).
24. H.T. Yau, C.C. Kuo and C.H. Yeh. **Extension of Surface Reconstruction Algorithm to the Global Stitching and Repairing of STL Models**, *Computer Aided Design*, 35,pp. 477-486,(2003).
25. J.T. Hu, Y. K. Lee, T. Blacker and J. Zhu. **Overlay Grid Based Geometry Cleanup,** Proc.11th International ; *Meshing Roundtable,* http://www.imr.sandia.gov/11imr/main.html,Ithaca, New York, USA, (Sep. 2002).
26. I. Makela and Dolenc. **Some efficient Procedure for Correcting Triangulated Models**, Proc, *Solid Free Form Fabrication Sump*. H.L. Marcus et. al., pp. 126-134, Univ. Of Texas, Austin, (Aug. 1993).
27. X. Sheng and I.R. Meier. **Generating Topological Structures for Surface Models**, *IEEE Computer Graphics and Applications*, vol. 15, no.6, pp. 35-42,(Nov1995).
28. W. Zhao,S. Ghao,H.Lin. **A robust Hole filling algorithm for triangular mesh**,*Journal of the Visual Computer,International Journal of Computer Graphics*,23(12),987-997,(2007).