

Euler Characteristic Scheme of Globally Searching for Flaws in Surface Modeling

Yukun Liu¹, Yong Yue², Dongwen Zhang¹, Chunhua Li¹

¹School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, China

²Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China

Email: lyklucky@sina.com, lyklucky@hebust.edu.cn, yong.yue@xjtlu.edu.cn

How to cite this paper: Liu, Y.K., Yue, Y., Zhang, D.W. and Li, C.H. (2020) Euler Characteristic Scheme of Globally Searching for Flaws in Surface Modeling. *Advances in Pure Mathematics*, **10**, 57-85.
<https://doi.org/10.4236/apm.2020.102005>

Received: January 8, 2020

Accepted: February 25, 2020

Published: February 28, 2020

Copyright © 2020 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper presents a new scheme of flaw searching in surface modeling based on Euler Characteristic. This scheme can be applied to surface construction or reconstruction in computer. It is referred to as Euler Accompanying Test (EAT) algorithm in this paper. Two propositions in algebraic topology are presented, which are the foundation of the EAT algorithm. As the modeling is the first step for rendering in the animation and visualization, or computer-aided design (CAD) in related applications, the flaws can bring some serious problems in the final image or product, such as an artificial sense in animation rendering or a mistaken product in industry. To verify the EAT progressive procedure, a three-dimensional (3D) stamp model is constructed. The modeling process is accompanied by the EAT procedure. The EAT scheme is verified as the flaws in the stamp model are found and modified.

Keywords

Algebraic Topology, Euler Characteristic, Flaw Searching, Surface Modeling, Topologic Consistence

1. Introduction

Along with the development of surface construction and reconstruction, many researchers have realized that mesh models may be constructed initially with some flaws because of the complex geometrical and topological structures of surfaces, and the limitations of scanning technologies and modeling schemes [1]-[10]. They have put a lot of efforts into repairing the flaws and filling holes. Although tens of researchers have been paying quite much attention to the hole filling quite a while [1]-[8], only several researchers [9] [10] focusing on the hole detecting are known, which cannot provide a method to find all the flaws in a

model.

Since modeling is the first step for rendering in the animation and visualization [11]-[16], or design and even manufacturing in industry [17]-[25], flaws produced in modeling can bring some serious problems in the final image or product. For example, a character in an animation has a tiny gap in his or her armpit under right arm, which should be fitted together. This gap is often invisible. It can, however, be seen when the character raises the right arm.

Another example is a case in fashion design, in which two adjoining cloth patches are not defined to fit together mistakenly. This problem may result in the problem that the edges between the two adjoining patches will not be stitched together in the following process.

The third example is that during interacting with computer, a viewer is interested in a mesh model and tries to take a close look at some parts of the model where there is a minute hole. The hole may be invisible when it is occluded or at a low resolution, but it may be seen when it is zoomed in and turned to be illuminated. The third example will be displayed in Section 4 of this paper.

For the above reasons, this paper presents a new scheme to globally search a mesh model for and locate the flaws. This new scheme is based on Euler characteristic, and named Euler Accompanying Test (shortened as EAT).

To give the theory foundation of algebraic topology, two propositions are presented for the EAT scheme in this paper. They can bridge the gap between the surface construction in computer and the theory of algebraic topology.

With the trivial amount of computing, the EAT scheme can assist in repairing and generating a topologically-correct model that has the same algebraic topological structure as that of its fitting target surface.

In this research, we also construct a new model of 3D PAMA stamp shown in **Figure 1** to display the progressive procedure of EAT scheme. The stamp model is programmed with OpenGL and VC++. There are two reasons for using a new model rather than the popular 3D models of Stanford Bunny, Skeleton Hand and Happy Buddha [26].

The first reason is that it is more efficient and tractable to search for, locate and correct a flaw during modeling than post-processing after the whole mesh model has been constructed. It is also why the new scheme is named Euler

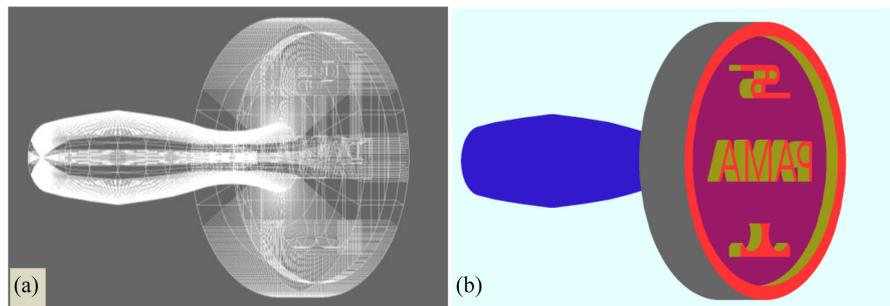


Figure 1. Two images of the model of the PAMA stamp: (a) The wire-frame image; (b) The fill-area image.

Accompanying Test (EAT), which means the modeling process accompanied by flaw-locating with the EAT. We will detail it in Section 3. The second reason is that this stamp mesh model consists of not only triangle facets but also polygons with more than three edges. The popular 3D models of Stanford Bunny, Skeleton Hand, and Happy Buddha [26] are composed of only triangle facets. We will apply these popular models gradually to the testing cases in the future.

The rest of this paper is arranged as follows: Section 2 introduces related studies; Section 3 details the EAT algorithm and the related propositions; Section 4 presents the searching cases of EAT on the PAMA stamp model; Section 5 is for the final conclusions.

2. Related Studies

In studies of surface construction and reconstruction in animation [13], visualization [11] [12] [14] [15] [16], and CAD [19], many modeling schemes are created to construct a composite surface with a mesh model to be approximate to a targeted object surface, which is referred to as a target surface in the rest part of this paper. The input of the construction process is usually a set of points sampled on the target surface or generated with reference to the target surface.

Among the studies, [14] presented the results of an approach for surface reconstruction from arbitrarily large point clouds of robotic maps. The authors in [15] proposed an approach to improve existing techniques of polynomial surface models. The research of [16] focused on the design of a modeling platform for a heterogeneous object (HEO) part based on a material distribution control function and hierarchical contour loop. These studies can give a clue to the broad applications of surface construction and reconstruction.

The composite surfaces are usually polygon patches glued together [21]-[25] [27]-[35]. Triangles and quads are mostly used as patches.

The work of [21] introduced an array-based algorithm for adaptive mesh refinement, which could generate millions of triangles. Their research of [22] proposed a local optimization method to construct triangular surface patches of high accuracy. In [24] and [25], methods of modeling surfaces composed of polygons were discussed. In [27], the authors presented a method that took an unorganized set of points to produce a surface of triangular facets. The study of [28] was the optimization of an initial triangular mesh. The research of [29] focused on automatic reconstruction of piecewise smooth surface models of triangular patches. In [30], the authors presented a method for combining a collection of range images into a single triangular mesh. The paper of [31] described a recursive bicubic B-spline patch subdivision algorithm which could generate rectangular control-point meshes. On the foundation of the study [31], the authors of [32] focused on the behavior of recursive division surfaces near extraordinary points. In [34], the authors proposed a method for refinement on B-spline surfaces with hierarchical subdivisions. The research of [35] developed the works of [31] and [32] to non-uniform recursive subdivision surfaces. It can be

summarized that triangular patches were used by [27] [28] [29] [30] and quad facets were applied to subdivision surfaces [31]-[35].

Few pentagons, hexagons and others are found in surface construction and reconstruction [36]-[41]. In [37], to generate features like sharp edges and corners, the authors merged Catmull Clark [31] and NURBS (Non-Uniform Rational B-spline) patches together to produce extended subdivision surfaces. In [38] [39], the authors presented a subdivision method that added extraordinary vertices to NURBS of arbitrarily high degree. Like [39], two methods [40] [41] could hold the geometrical details of neighboring polygons of extraordinary points.

Because of the complex geometrical and topological structures of a target surface, the limitations of scanning technologies, and the deficiency of a modeling scheme, a composite surface may be constructed with tiny holes or gaps that the target surface does not have [1]-[8] [17] [19] [20]. That is, the composite surface does not have the same topology as that of the target surface and cannot really represent the target surface in the mathematics and applications sense.

The papers [1] [3] made a survey on existing well-known hole-filling algorithms in 3D surface reconstruction and classified them. Researchers have proposed different methods to fill the holes in surface construction. In [2], a piecewise hole filling algorithm was presented. The study [4] used the method of positive definite Radial Basis Functions in progressively filling the holes in surface reconstruction. The authors of [5] presented a scattered point set hole-filling method to fill the holes of an incomplete surface. The paper of [6] introduced a technique to fill large holes of both texture and structure in a LiDAR scan. In the research of [7], the authors proposed a depth hole filling method to fill holes in images obtained from the Microsoft Kinect sensor. In the research of [8], the authors' argument was that for accessibility reasons or reflection problems, some parts of the object might not be scanned and could be formed as digitizing holes. The authors of [17] mentioned that many photos contained close-up cracks in the material of the sculpture that needed to reconstruct. In the research of [19], authors mentioned that in trimming models in CAD might cause unavoidable gaps in the models. In [20], to modify a gap where neighboring B-spline surface did not match exactly, the authors presented a locally refineable subdivision scheme called T-NURCCs (Non-Uniform Rational Catmull-Clark surface with T-junctions).

With its complex geometrical and topological structures, a great number of the 3D data of points on the target surface are required to input for modeling. If the 3D data are acquired by laser-range scanning, they are unstructured point clouds and missing some information by under-sampling sometimes [9]. For these reasons, the modeling is susceptible to unexpected holes. And the knowledge of holes in the model is vital for many applications dealing with point set surfaces [9].

On the other hand, to make a composite surface hold fine local features, re-

searchers of mesh model may add more sub-patches to some part of the composite surface with local refinement [32] [37] [38] [40] [41]. Along with the number of patches and sub-patches further increasing, local refinement may bring new unexpected invisible flaws into the composite surface [17] [19] [20]. As known, the consistence of global topological structure is as important as the local features for surface modeling. Holding the consistence of the global topological structure can make the model correct in the topological sense while having local features can make the model delicate. Neither of them should be ignored. Therefore, it is necessary to detect the new defects of the composite surface after local refinement.

In [9], the authors present a method of detecting holes in point set surfaces. Their method is based on boundary probability with three criteria: angle, half-disc, and shape criteria. It is effective to extract boundary loops by repeating the process of searching for each vertex and identifying boundary points with the criteria. Their method, however, does not guarantee to find all the unexpected holes in a surface model since it is probable.

In [10], the authors present an empty disk approach for detecting the inner hole boundary of a planar point set based on Delaunay triangulation. They focus on the problem of detecting the boundary of individual inner holes in a planar point set. The limitation of their method is to detect only individual holes in a two-dimensional space, which is not appropriate for finding all the holes of a 3D model that is the issue studied in this paper.

In this paper, the presented EAT algorithm is applied to testing the topological structure of a mesh model gradually and globally to find all the flaws of the model and to modify them before passing a verified model to the succeeding process. Even better, it can be used to detect new defects of the model after local refinement.

3. EAT Scheme

A complicated surface model usually has a great deal of geometric details, and thus may contain a large number of vertices, edges and facets. The EAT has to scan and traverse all the vertices, edges and facets of the model to compute the Euler characteristic of the model. And then it can be compared with the Euler characteristic of its target surface. The larger number of edges and facets the model contains, the more time the EAT computing requires.

In order that the code of the EAT is not left in the modeling program after verified, we adopt the strategy that a copy of the modeling program is prepared for flaw-searching, which is called the modeling companion, shown in **Figure 2(a)**. The EAT algorithm code is inserted in the modeling companion rather than the modeling program itself. The EAT code is running only in the modeling companion.

Each time a flaw is found in the model companion, the modeling companion is revised, and the modeling program should be revised and updated as well.

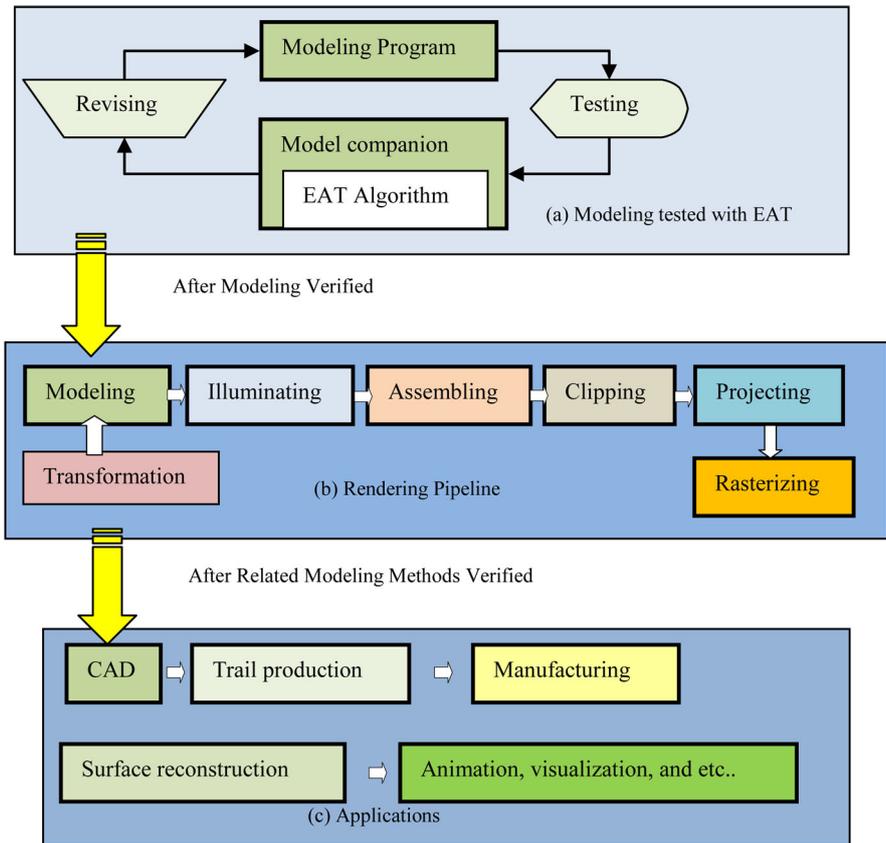


Figure 2. A simplified frame diagram with three layers: (a) At the top is the modeling tested with EAT; (b) In the middle is the computer graphics processing for rendering after modeling verified; (c) At the bottom are the applications of design and manufacturing in industry, and surface reconstruction in animation, visualization and others after related schemes verified.

Since the EAT algorithm code is running only in the model companion, the modeling program is without the EAT code in itself and thus the verified modeling program will not be slowed down by the EAT at run time.

After the model has been verified completely with the EAT scheme and all the existing flaws of model have been corrected, the verified modeling program is assembled into the rendering pipeline, shown in **Figure 2(b)**. If some flaws are related to a modeling method, the method should be modified.

After the related modeling method has been validated, the method is integrated into CAD environments in industry or the surface reconstruction systems of animation, visualization and others, shown in **Figure 2(c)**. And the succeeding procedures can be performed, as well.

Since fault-testing with the EAT is at the modeling stage, we will focus on the layer at the top of **Figure 2(a)** in this section.

3.1. Propositions in Algebraic Topology

The EAT algorithm is the core of the EAT scheme. It is based on the following Definition 1, Proposition 1 and Proposition 2.

Definition 1 The Euler characteristic of a triangulated surface M is

$$\chi(M) = n_v - n_e + n_f, \quad (1)$$

where n_v is the total number of vertices of M , n_e is the total number of edges of M , and n_f is the total number of triangle facets of M [42].

It is obvious that the invariant of Euler characteristic is independent of the meshing of a surface. It can be shown in **Figure 3**. If two triangle facets with a common edge are replaced with a quad facet and n_f is referred to as the total number of facets in identity (1), as shown in **Figure 3(a)** and **Figure 3(b)**, the identity (1) is still held. This deduction is based on the fact that the latter facet (no matter planar or curve) is homotopic to the former two triangle facets glued together. If each pair of three triangle facets has a common edge, these three triangle facets are replaced with a pentagon facet, and the identity (1) is also held, shown in **Figure 3(c)** and **Figure 3(d)**. Let us take the idea a step further. If each pair of four triangle facets has a common edge, these four triangle facets are replaced with a hexagon facet, and the identity (1) is held, too, shown in **Figure 3(e)** and **Figure 3(f)**.

With the algebraic topologies of surfaces [42] [43], the following two propositions are obvious.

Proposition 1 A compact surface has a numerical invariant called the Euler characteristic.

Proposition 2 Suppose that there are two compact surfaces, S_1 and S_2 . If their Euler characteristics are not identical, the topological structures of S_1 and

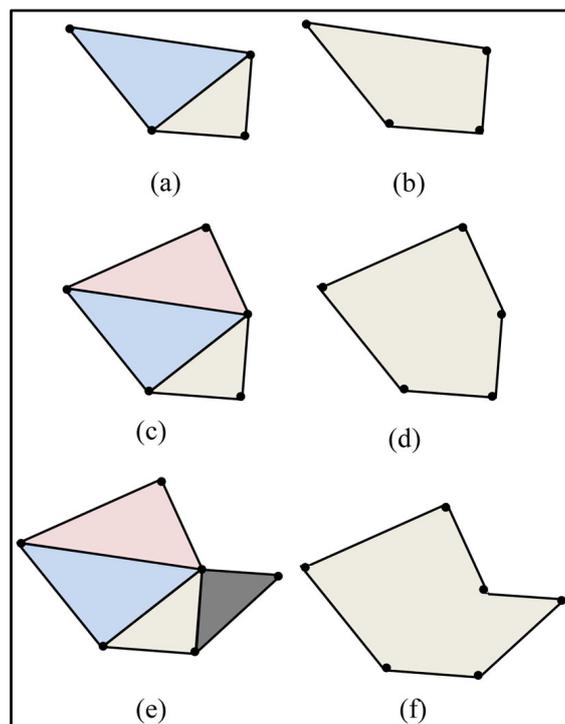


Figure 3. Illuminating the invariant of Euler characteristic independent of the meshing.

S_2 are not consistent with each other.

As the rigorous proofs of Proposition 1 and Proposition 2 are lengthy and require the homotopy theory, we do not give the proofs of Proposition 1 and Proposition 2, and readers can refer to the homotopy theory [42] [43].

Before the exploration of EAT algorithm, we have to define three sets of vertices, edges and facets of a tested model, which most of modeling methods in surface construction and reconstruction apply to surface modeling. Here we re-organize them in a three-layer data structure, and register every vertex, edge and facet in Vertex list, Edge list and Facet list, respectively, in order to compute the Euler characteristic of the model at different modeling stages.

3.2. Three-Layer Data Structure

In surface modeling, the input is a set of points, which may be sampled on a target surface by sensors or scanned the outward surface of an object by a scanner. The set of points can be organized as the set of original vertices, V_0 . Each element of V_0 has three fields (x_i, y_i, z_i) , which are three coordinates in an orthogonal coordinate system, respectively.

According to the spatial relationships of vertices and the order of the facets' being drawn, we construct a three-layer data structure for the modeling companion accompanied with the EAT.

The data structure is shown in Figure 4. At the top abstract layer is Body

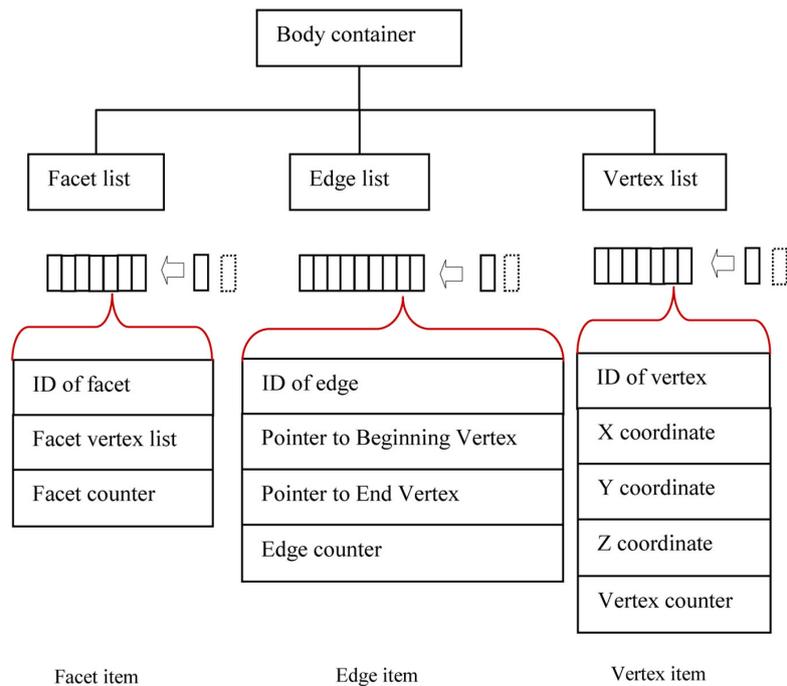


Figure 4. The data structure of EAT algorithm in a model companion has three layers. The Body container is at the top abstract data layer, which consists of three parts: Facet list, Edge list and Vertex list at the middle abstract layer. At the lowest abstract layer are Facet items, Edge items and Vertex items linked by Facet list, Edge list and Vertex list, respectively.

container, which consists of three parts, these being Facet list F_g , Edge list E_g , and Vertex list V_g .

- F_g represents the Facet list. It is used to add Facet item to it when a new facet is registered.
- E_g represents Edge list. It is used to add Edge item to it when a new edge is registered.
- V_g represents Vertex list. It is used to add Vertex item to it when a new vertex is registered.

F_g , E_g , and V_g are at the middle abstract layer of the three-layer data structure. At the lowest layer are three types of items: Facet item

$f(fID, fvList, fc)$, Edge item $e(eID, bvPointer, evPointer, ec)$, and Vertex item $v(vID, xCoord, yCoord, zCoord, vc)$.

- $f(fID, fvList, fc)$ is Facet item, which consists of three fields: identification of facet fID , facet vertex list $fvList$, and facet counter fc . Each distinct facet has a unique fID . Facet vertex list $fvList$ registers the vertex sequence of the facet item in the order of vertices' being drawn. Facet counter fc is used to count the times of being drawn of the facet.
- $e(eID, bvPointer, evPointer, ec)$ is Edge item, which is composed of four fields: identification of edge eID , a pointer to the beginning vertex $bvPointer$, a pointer to the end vertex $evPointer$, and edge counter ec . Each distinct edge has a unique eID . Each edge is connected to two vertices: the beginning vertex to which is pointed by the pointer of $bvPointer$, and the end vertex towards which is pointed by the pointer of $evPointer$. The number of facets connected to the edge is counted by the counter of ec during modeling.
- $v(vID, xCoord, yCoord, zCoord, vc)$ is Vertex item with five fields: identification of vertex vID , X coordinate $xCoord$, Y coordinate $yCoord$, Z coordinate $zCoord$, and vertex counter vc . Every individual vertex has a unique vID and three coordinate values $(xCoord, yCoord, zCoord)$ in the modeling space. The number of edges connected to the vertex is counted by the counter of vc during modeling.

3.3. EAT Algorithm

The EAT algorithm includes two different function parts. One function part is named the EAT testing algorithm (EAT-TA), shown in **Figure 5**. It is used to register a new facet, and its new vertices and new edges in Facet list, Vertex list and Edge list, respectively, and to increment each of their counters. Its code is embedded just after every facet modeling code in the model companion.

The other function part is called the EAT statistic algorithm (EAT-SA), also shown in **Figure 5**. The EAT-SA collects the statistics of the total numbers of vertices, edges and facets of the surface model, computes the Euler characteristic of the surface model, and outputs the results at different modeling stages.

- 1) EAT Testing Algorithm (EAT-TA)

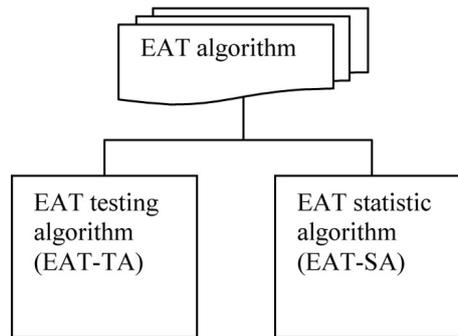


Figure 5. The EAT algorithm including two function-wise parts: the EAT testing algorithm (EAT-TA) and EAT statistic algorithm (EAT-SA).

To do exactly counting of the numbers of facets, edges and vertices of a tested surface model, the EAT-TA has to scan and traverse the model. As the modeling in surface construction and reconstruction is a dynamic and progressive process, the EAT-TA can take advantage of this process through each facet modeling accompanied by the EAT testing.

Figure 6 shows the calling site of EAT-TA and related operations in the model companion. As shown, a call of the EAT-TA is usually made just after the tested facet modeling. **Figure 7** shows the pseudocode of EAT-TA, which can be easily translated into VC++ language or other computer languages.

During a facet modeling, some vertices and edges may have not been registered while other vertices and edges may have been registered in the previous facet modeling. At this moment, the vertices and edges that have not been registered are referred to as new vertices, v_{new} , and new edges, e_{new} , respectively, which have not been added to Vertex list and Edge list. The vertices and edges that have been registered are called old vertices, v_{old} , and old edges, e_{old} , which already exist in Vertex list and Edge list, respectively, shown in **Figure 7**.

That is, a new vertex or new edge has to be registered first and then their counters increment. For an old vertex or old edge, it is not registered again, but its counter increments.

2) EAT Statistic Algorithm (EAT-SA)

The EAT-SA is for statistic computing of the numbers of vertices, edges, and facets. **Figure 8** shows the pseudocode of EAT-SA, which can be also translated easily into VC++ or other computer languages.

The EAT-SA can be put at the end of the modeling companion for the fault-detecting of the whole model. It can be also inserted at any intermediate site in the modeling companion. In this paper, these intermediate sites are called *test points*.

In fact, it is more effective to set a test point at a site where we want to detect whether or not the modeling before this site holds any defect.

A intermediate test point can help search for and locate the defects in the

```

The beginning of modeling and EAT-TA of a facet.
...
The code of modeling.
...
Acquire the vertex sequence of the facet.
Set the geometrical arguments.
Set the argument of outputting the information of
  vertex items and edge items of the facet.
Call the EAT-TA with the set arguments.
....
The end of modeling and EAT-TA of a facet.

```

Figure 6. The calling site of the EAT-TA and the related operations of a facet in the model companion.

```

The beginning of EAT-TA pseudocode.

Do register a facet item by adding it to  $F_g$ .
For (all the vertices of the facet) do the operations
in the following pair of braces.
{
  Check whether the vertex is  $v_{new}$  or  $v_{old}$ .
  If the vertex is a  $v_{new}$ , do register it by adding it
  to  $V_g$ , and increment its  $vc$ .
  If the vertex is a  $v_{old}$ , do increment its  $vc$ .
}
For (all the edges of the facet) do the operations in
the following pair of braces.
{
  Check whether the edge is  $e_{new}$  or  $e_{old}$ .
  If the edge is a  $e_{new}$ , do register it by adding it to
   $E_g$ , and increment its  $ec$ .
  If the edge is a  $e_{old}$ , do increment its  $ec$ .
}
If the outputting argument is 1, output the infor-
mation of vertex items and edge items of the fac-
et item.

The end of EAT-TA pseudocode.

```

Figure 7. The EAT testing algorithm (EAT-TA) pseudocode.

```

The beginning of EAT-SA pseudocode.
Assign the total number of items in  $F_g$  to the vari-
able  $idf$ .
Assign the total number of items in  $E_g$  to the vari-
able  $ide$ .
Assign the total number of items in  $V_g$  to the vari-
able  $idv$ .
Check Euler characteristic with identity (1):
EulerCharacteristic =  $idv - ide + idf$ .
Output  $idv$ ,  $ide$ ,  $idf$ , and EulerCharacteristic.
The end of EAT-SA of pseudocode.

```

Figure 8. The EAT statistic algorithm (EAT-SA) pseudocode.

partial modeling just before this test point and correct them in the model companion and in the modeling program as well, shown in **Figure 2(a)**.

When considering that there are numerous facets in a complicated surface model, there may exist a large number of defects in the whole unverified model [9]. In the progressive test way, a large problem of numerous defects can be divided into smaller ones with several defects. It is easier to conquer each of smaller sub-problems by correcting one of them at a time than the larger repair problem of the complex surface model as a whole in one step.

3.4. Progressive EAT Procedure

As known before, a surface modeling is a progressive process, and the EAT-SA can be done at different stages as well. It means that it can be used to test an unfinished model in which some facets have been constructed and others have not been.

We can set a test point and make an EAT-SA calling just after the drawing of the unfinished model. The EAT algorithm can help check whether or not the partial model has the Euler characteristic equal to that of its corresponding part of the target surface. If yes, the partial model has been verified. If not, the EAT-TA can locate the faults that are indicated to correct.

After the previous partial model has been corrected and verified, the modeling is resumed. We call the test point that has just been verified the *last verified point*.

For newly modeling, a new test point can be set at a proper distance after the last verified point, and a new EAT-SA calling is inserted at this new test point. We call this new point the *current test point*.

As the facets before the last verified point have been verified without faults, new faults detected by the EAT-SA at the current test point exist in the facets between the last verified point and the current test point. For the model companion, these newly-detected facets are the new facets items that have been just added to Facet list. In this way, it is easy to locate the new faults.

Accompanied by the progressive EAT procedure, the newly modeling is always constructed on the foundation of the verified partial model.

If the results of EAT-SA at the current test point show that the Euler characteristic of the partial model is not identical to that of the corresponding part of the target surface, we have to know which facet modeling may cause the problem. In this situation, we can set 1 to the argument of outputting the information of vertex items and edge items of facets in the newly modeling after the last verified point, and set 0 to the arguments of the facet items before the last verified point. The setting of the arguments has been indicated in **Figure 6**. In this way, we can focus just on the newly-added facet items. It is not necessary to examine the facet items in the partial model before the last verified point.

At the modeling end of the target surface, we put the final EAT-SA at the end of model companion. The resulting numbers of total facets, edges and vertices of

the model, and its Euler characteristic are outputted and displayed on the computer screen. We can compare the outputted Euler characteristic to that of the target surface. If they are not equal to each other, try to find the problem locations in the newly modeling after the last verified point and correct the related facets in the model companion and the modeling program as well. And perform the EAT algorithm again until all the faults are corrected. If two Euler characteristics are identical, the modeling has been verified. And the model can be used to represent the target surface upon the Euler characteristic invariant according to Proposition 1 and Proposition 2.

In the next section, some EAT applications will be presented, which can illustrate the progressive EAT procedure clearly.

4. EAT Applications and Results

In this section, the EAT algorithm is applied to test the stamp model, shown in **Figure 1**. In Sections 4.1, 4.2, and 4.3, we present three application cases that show the models with some faults tested by the EAT scheme. In Section 4.4, we will present the model companion accompanied by the progressive EAT procedure at several stages.

4.1. The First Case with a Defect Tested by EAT

In this section, we present a partial stamp model that has a tiny triangle without filling. The triangle is so small that is almost invisible.

In this case, the modeling before the last verified point has been verified, shown in **Figure 9**. To make the views distinct, facets in different parts are depicted with different colors, seen in the front view and side view of fill-area images of **Figure 9(a)** and **Figure 9(b)**. In **Figure 10**, the tested results with the EAT scheme on the partial model in **Figure 9** show that the number of vertices, edges, and facets of the partial model are 1658, 2938, and 1279, respectively. Its Euler characteristic is -1 , which is the same as that of the corresponding compact surface with two inside holes.

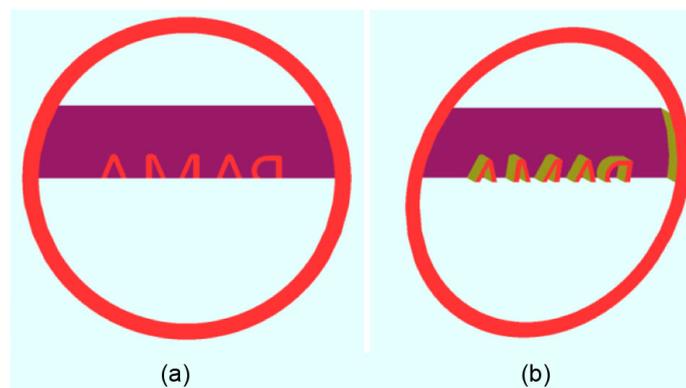


Figure 9. The modeling images of partial stamp model that has been verified. (a) The front view of fill-area image. (b) The side view of fill-area image.

```
vertex number ***1658***vertex number
edge number ***2938***edge number
facet number ***1279***facet number
Euler characteristic***-1***Euler characteristic
```

Figure 10. A screenshot of execution results of EAT algorithm of the partial stamp model in **Figure 9**. The first row is the vertex number, 1658; the second row is the edge number, 2938; the third row is the facet number, 1279; and the bottom row is the Euler characteristic, -1 .

At this time, we set a new current test point and test the part between the last verified point and the current test point.

Figure 11 shows the tested results with the EAT scheme at the current test point. The Euler characteristic is -2 , which is different from the Euler characteristic of the corresponding compact surface with two inside holes, -1 . For this reason, the modeling between the last verified point and the current test point has to be checked carefully. Through exploration and location, a tiny triangle in which is not filled is found.

Figure 12(a) is the front view of fill-area image, and **Figure 12(b)** is the side view of fill-area image of the partial model with the empty triangle. The empty triangle is too small to be identified in **Figure 12(a)** and **Figure 12(b)**.

After the image with the invisible empty triangle in **Figure 12** having been zoomed in and turned to be illuminated, a tiny white point appears at the joint between the green and the purple parts circled by the black rectangle of **Figure 13(a)**. **Figure 13(b)** is the image in the black rectangle in **Figure 13(a)** that is zoomed in even closer. In **Figure 13(b)**, we can see the empty part clearly, which looks like a narrow segment. The length of one edge of this triangle is almost 0.001 of lengths of the other two. It is why the triangle is difficult to be identified.

Let us modify the defect by filling in the empty triangle and test the model again. The results of this new test are shown in **Figure 14**. At the top first row is the information of three vertex items of the filled triangle, these being vertex items of vIDs 1698, 1699, and 6807. The vertex counter (vc) values of them are 3, 4, and 4, respectively. 3 means the vertex item of vID 1698 joins three edges, and 4 means the vertex item of vID 1699 or 6807 links four edges.

The top second, third and fourth rows of **Figure 14** are the information of three edge items of the filled triangle, which are edge items with eIDs 2858, 2991, and 2996, respectively. All of their edge counter (ec) values are 2. 2 means each of them connects two facets.

The bottom four rows of **Figure 14** show the results at the current test point. The number of vertices is 1738, the number of edges is 3090, and the number of facets is 1351. The Euler characteristic is -1 , which is the same as that of the corresponding compact surface with two inside holes, -1 . This partial stamp model has been modified and verified at this stage.

```

vertex number ***1658***vertex number
edge number ***2938***edge number
facet number ***1279***facet number
Euler characteristic***-1***Euler characteristic

vertex number ***1738***vertex number
edge number ***3090***edge number
facet number ***1350***facet number
Euler characteristic***-2***Euler characteristic
    
```

Figure 11. A screenshot of execution results of EAT algorithm. The top four rows are the results of the partial stamp model before the last verified point, which is shown in Figure 10. The bottom four rows are the results of the partial stamp model before the current test point. The bottom first row is the vertex number, 1738; the bottom second row is the edge number, 3090; the bottom third row is the facet number, 1350; the bottom fourth row is the Euler characteristic, -2.

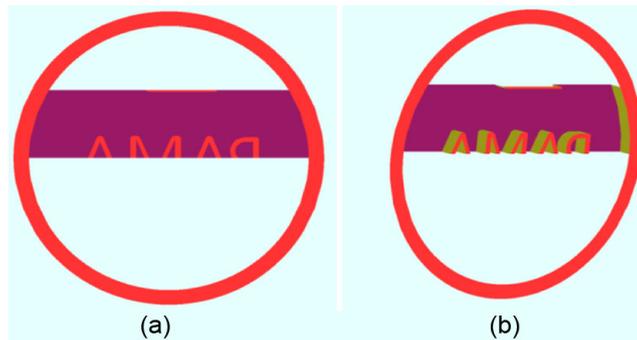


Figure 12. The modeling images of partial stamp model with an invisible empty triangle before the current test point. (a) The front view of fill-area image. (b) The side view of fill-area image.

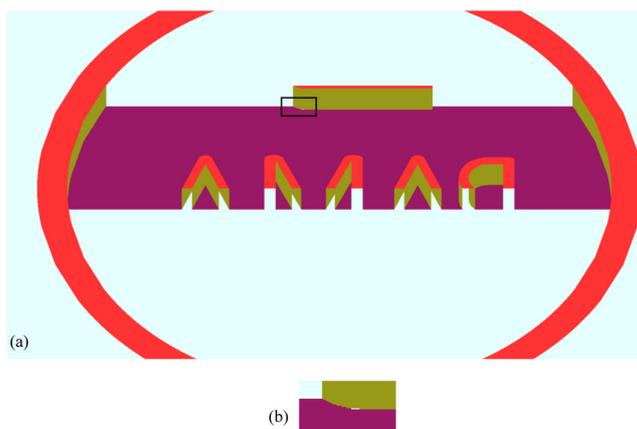


Figure 13. The images of the partial stamp model of Figure 12, which are enlarged. (a) The image of facing upward view of the partial stamp model with a tiny empty triangle facet that is a white point marked by the black rectangle. (b) The enlarged image in the black rectangle.

```

1698**3**1699**4**6807**4**
2858**2**
2991**2**
2996**2**

vertex number **1738**vertex number
edge number **3090**edge number
facet number **1351**facet number
Euler characteristic**-1**Euler characteristic

```

Figure 14. A screenshot of the execution results of the EAT algorithm after the defect of **Figure 13** has been modified. The top first row is vID and vertex counter (vc) value of each vertex item of the modified triangle. The top second, third and fourth rows are eID and edge counter (ec) value of each edge item of the modified triangle. The bottom half are the results of numbers of vertices, edges, and facets of the partial model in **Figure 13** having been corrected, and its Euler characteristic of -1 .

4.2. The Second Case with a Defect Tested by EAT

In this section, we present the second case with a quad that was ignored before and is tested by EAT.

The modeling before the last verified point has been verified, shown in **Figure 15**. **Figure 15(a)** is the front view of fill-area image; **Figure 15(b)** is the side view of fill-area image. In **Figure 16**, the results at the last verified point show that the number of vertices, edges, and facets of the partial model are 2189, 3963, and 1773, respectively. Its Euler characteristic is -1 , which is the same as that of the corresponding compact surface with two inside holes.

After the last verified point, some facets are added to the modeling. A new current test point is set and the EAT is executed. **Figure 17** shows the results at the current test point. The Euler characteristic is -2 , which is different from the Euler characteristic of the corresponding compact surface with two inside holes, -1 . **Figure 18(a)** is the front view of fill-area image of the partial model before this current test point, and **Figure 18(b)** is its side view of fill-area image. In **Figure 18**, the fault is invisible.

Let us take a closer look at the newly-added facets, shown in **Figure 19**. A narrow white quad appears at the joint between the green and the orange parts marked by the black rectangle of **Figure 19(a)**. **Figure 19(b)** is the image circled by the black rectangle in **Figure 19(a)**, which is zoomed in even closer. In **Figure 19(b)**, we can see the empty quad more clearly, which looks like a narrow belt. It is so small that it can be seen only in a proper direction. That is why the quad is difficult to be found.

Next, we modify the defect by filling in the empty quad and test the model again. The results of this test are shown in **Figure 20**. At the top first row is the information of four vertex items of the filled quad, these being vertex items of vIDs 7786, 7788, 7789 and 7790. The vertex counter (vc) values of them are 4, 3, 3, and 3, respectively. 4 of vc means that the vertex item of vID 7786 joins four edges, and 3 of vc means that the vertex item of vID 7788, 7789, or 7790 links

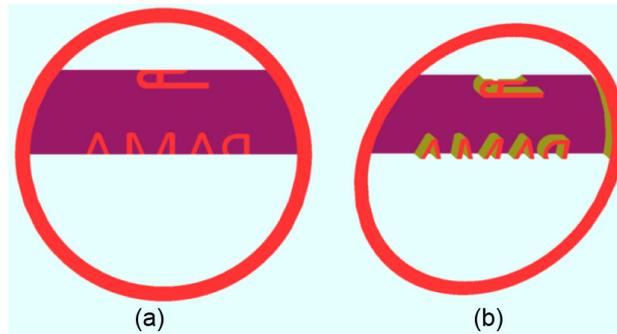


Figure 15. The modeling images of partial stamp model that has been verified for the second example in Section 4.2. (a) The front view of fill-area image. (b) The side view of fill-area image.

```
vertex number ***2189***vertex number
edge number ***3963***edge number
facet number ***1773***facet number
Euler characteristic***-1***Euler characteristic
```

Figure 16. A screenshot of execution results of EAT algorithm, displaying results of numbers of vertices, edges, and facets of the partial stamp model in Figure 15, and its Euler characteristic of -1 .

```
vertex number ***2189***vertex number
edge number ***3963***edge number
facet number ***1773***facet number
Euler characteristic***-1***Euler characteristic

vertex number ***2255***vertex number
edge number ***4082***edge number
facet number ***1825***facet number
Euler characteristic***-2***Euler characteristic
```

Figure 17. A screenshot of execution results of EAT algorithm. The top half is the results of numbers of vertices, edges, and facets of the partial stamp model before the last verified point, and its Euler characteristic of -1 . The bottom half is the results of the partial stamp model before the current test point, and the Euler characteristic of -2 .

three edges.

The top second, third, fourth and fifth rows of Figure 20 are the information of four edge items of the filled quad, which are edge items with eIDs 4007, 4010, 4013, and 4045, respectively. Three of their edge counter (ec) values are 2. 2 of ec means each of them connects two facets. One of them is 1, which means that when this quad facet is constructed, the edge item of eID 4045 is a new edge, shown in Figure 7.

The bottom four rows of Figure 20 show the results at the current test point. The number of vertices is 2255, the number of edges is 4082, and the number of

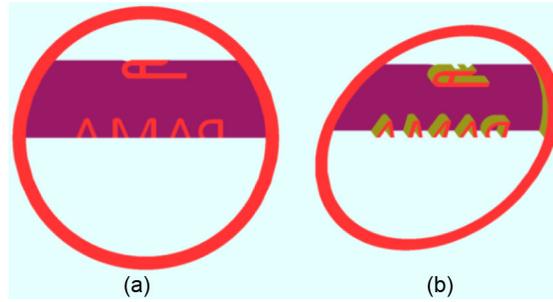


Figure 18. The images of partial stamp model with an invisible empty quad before the current test point. (a) The front view of fill-area image. (b) The side view of fill-area image.

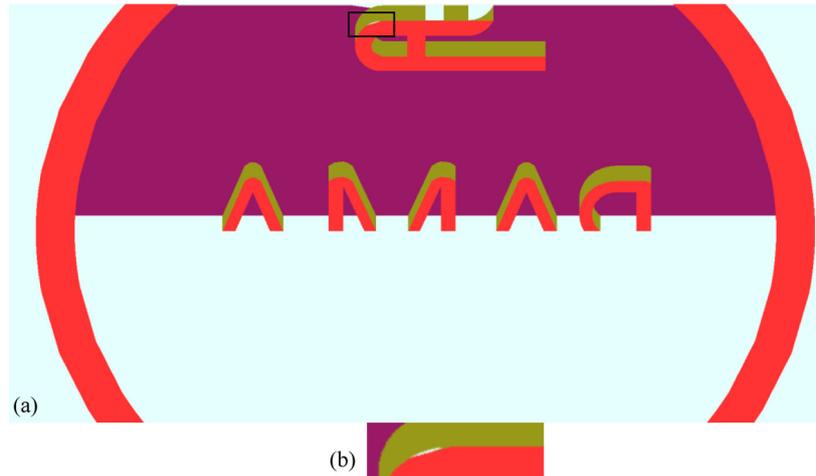


Figure 19. The images of the partial stamp model in **Figure 18**, which are enlarged. (a) The image of facing downward view of the partial stamp model with a tiny empty quad facet that is a white narrow quad marked by the black rectangle. (b) The enlarged image in the black rectangle.

```

7786**4**7788**3**7790**3**7789**3**
4007**2**
4010**2**
4013**2**
4045**1**

vertex number ***2255***vertex number
edge number ***4082***edge number
facet number ***1826***facet number
Euler characteristic***-1***Euler characteristic
    
```

Figure 20. A screenshot of execution results of EAT algorithm after the defect in **Figure 19** is modified. The top first row is vID and vertex counter (vc) value of each vertex item of the modified quad. The top second, third, fourth and fifth rows are eID and edge counter (ec) value of each edge item of the modified quad. The bottom half is the results of numbers of vertices, edges, and facets of the partial stamp model in **Figure 19** that has been corrected, and its Euler characteristic of -1 .

facets is 1826. The Euler characteristic is -1 , which is the same as that of the corresponding compact surface with two inside holes, -1 . This partial stamp model is verified at this stage.

4.3. The Third Case with a Defect by EAT

In this section, we present the third case with a fault that is counter-intuitive and tested by EAT.

In this case, we still use the partial model stage in **Figure 15** as the last verified point. In the newly-added facets, we set a new test point and search for a problem different from the previous two.

In **Figure 21**, the front and side views of fill-area and wire-frame images are shown. Let us zoom in the image of **Figure 21(c)** and take a closer look at the right top corner of the top character, circled by the blue rectangle in **Figure 22(a)**. **Figure 22(b)** and **Figure 22(c)** are amplified images in the blue rectangle of **Figure 22(a)**. In **Figure 22(c)**, five vertex items are marked with their vIDs 6891, 2650, 2567, 2566, and 6874, respectively.

In the geometrical sense, the pentagon bordered with five edges ($\{6891, 2650\}$, $\{2650, 2567\}$, $\{2567, 2566\}$, $\{2566, 6874\}$, and $\{6874, 6891\}$) is the same as the quad bordered with four edges ($\{6891, 2650\}$, $\{2650, 2567\}$, $\{2567, 6874\}$, and $\{6874, 6891\}$). In the topologic sense, they are, however, different. The former facet has five vertices, five edges and one facet; the latter facet consists of four vertices, four edges and one facet. And the edge of $\{2567, 6874\}$, which is the yellow one in **Figure 22(c)**, is different from the two edges, $\{2567, 2566\}$ and

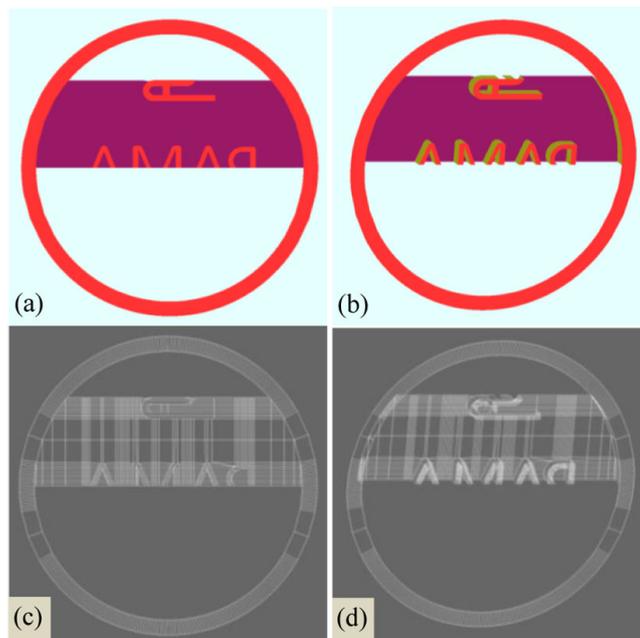


Figure 21. The images of partial stamp model for testing a new fault in Section 4.3. (a) The front view of fill-area image. (b) The side view of fill-area image. (c) The front view of wire-frame image. (d) The side view of wire-frame image.

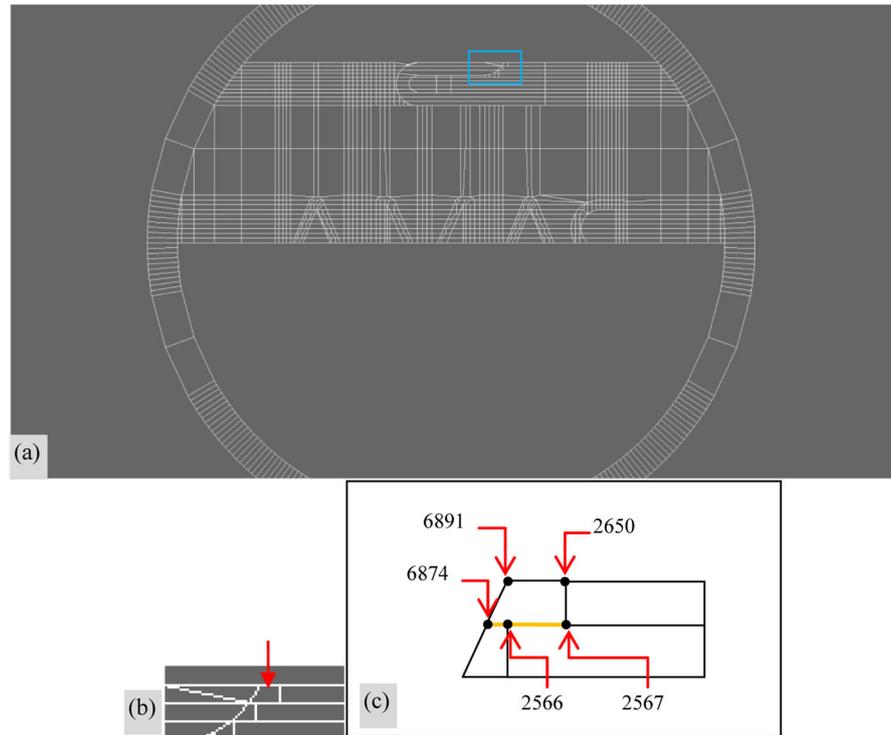


Figure 22. The images of the partial stamp model in **Figure 21(c)**, which are enlarged. (a) The image of the partial stamp model with a problem quad marked by the blue rectangle. (b) The amplified image in the blue rectangle of (a). (c) The quad pointed by the red arrow in (b) is redrawn and marked with vIDs of vertex items. The yellow edge is the problem one.

{2566, 6874}.

For one edge, {2567, 6874}, the results of the EAT are shown in **Figure 23**. It results in an Euler characteristic, -2 , which is different from -1 of the corresponding compact surface with two inside holes. The reason is that the edge of {2567, 6874} is treated as one edge different from as two edges, {2567, 2566} and {2566, 6874}.

For the modeling of the neighborhood of this problem facet, two edges, {2567, 2566} and {2566, 6874}, have been added to Edge list, E_g . In the topologic sense, it means that there is a crack between the new Edge {2567, 6874} and two old Edges, {2567, 2566} and {2566, 6874}. That is, this model is corresponding to a compact surface with three inside holes.

We modify this fault by dividing Edge {2567, 6874} into two Edges, {2567, 2566} and {2566, 6874}, and test it with the EAT algorithm again. The results are shown in **Figure 24**. Compared to the results in **Figure 23**, there are five vertex items in **Figure 24**, but four vertex items in **Figure 23**. The vertex item of vID 2566 is added to this pentagon facet. There are five edge items in **Figure 24**, but four edge items in **Figure 23**. Among these five edge items, three edge items of eIDs 3933, 3935, and 4053 are old edges, and their edge counter (ec) values are 2. The other two edge items of eIDs 4054 and 4055 are new edges, and their edge counter (ec) values are 1. But in **Figure 23**, there is one old edge item of eID

```
6874**4**2567**2**6891**2**2650**1**
4053**2**
4054**1**
4055**1**
4056**1**

vertex number ***2255***vertex number
edge number ***4083***edge number
facet number ***1826***facet number
Euler characteristic***-2***Euler characteristic
```

Figure 23. A screenshot of execution results of EAT algorithm for the partial model in **Figure 22**. The top first row is vID and vertex counter (vc) value of each vertex item of the problem quad. The top second, third, fourth and fifth rows are eID and edge counter (ec) value of each of four edge items of the quad. The bottom half is the results of numbers of vertices, edges, and facets of the partial stamp model, and its Euler characteristic of -2 .

```
6874**4**2566**3**2567**2**6891**2**2650**1**
3933**2**
3935**2**
4053**2**
4054**1**
4055**1**

vertex number ***2255***vertex number
edge number ***4082***edge number
facet number ***1826***facet number
Euler characteristic***-1***Euler characteristic
```

Figure 24. A screenshot of execution results of EAT algorithm for the partial model of **Figure 22** that has been modified by dividing Edge {2567, 6874} into two Edges, {2567, 2566} and {2566, 6874}. The top first row is vID and vertex counter (vc) value of each vertex item of the pentagon. The top second, third, fourth, fifth and sixth rows are eID and edge counter (ec) value of each of five edge items of the pentagon. The bottom half is the results of numbers of vertices, edges, and facets of the partial stamp model in **Figure 22** that has been corrected, and its Euler characteristic of -1 .

4053, and three new edge items of eIDs 4054, 4055, and 4056. It means one of new edge items, the edge item of eID 4056 (that is Edge {2567, 6874}) does not meet with the old ones, the edge items of eIDs 3933 and 3935 (Edges {2567, 2566} and {2566, 6874}). There is a crack between them.

In **Figure 23**, the number of vertices is the same as 2255 in **Figure 24**. Both numbers of facets in **Figure 23** and **Figure 24** are equal to 1826. But the number of edges, 4083, in **Figure 23**, is one more than 4082 in **Figure 24**. Therefore, the Euler characteristic, -2 , in **Figure 23**, is one less than that -1 in **Figure 24**.

In Section 4.4, we will present the stamp model companion accompanied by the progressive EAT procedure at several steps until the whole model construction is finished and verified.

4.4. Testing the Stamp Model at Several Stages with EAT

In this section, we present the construction process of the stamp model accompanied by the EAT procedure. Because of the limitation of this paper length, we only display four typical stages in this process. Of course, the real construction process of the stamp model has been tested and modified at far more than four stages because it has many details of the geometrical and topological structures.

1) Testing Fringe Ring and Middle Rows of Top Face

An unfinished model is shown in **Figure 25**. This unfinished model includes the fringe ring and several middle lines of top face of the stamp model. Most facets are quads. Several triangle facets are filled at corners. They are shown in the front view and side view of wire-frame images in **Figure 25(a)** and **Figure 25(c)**. The corresponding fill-area images are shown in **Figure 25(b)**, and **Figure 25(d)**. Facets in different parts are depicted in different colors.

In **Figure 26**, four rows are the results of numbers of vertices, edges, and facets of the unfinished model in **Figure 25**, and its Euler characteristic, respectively. We can see that the number of vertices is 1213, the number of edges is 2054, and the number of facets is 840. Its Euler characteristic is -1 . This Euler characteristic is equal to one of the corresponding compact surface with two inside holes. Thus, this partial model is verified.

2) Testing Half Top Face

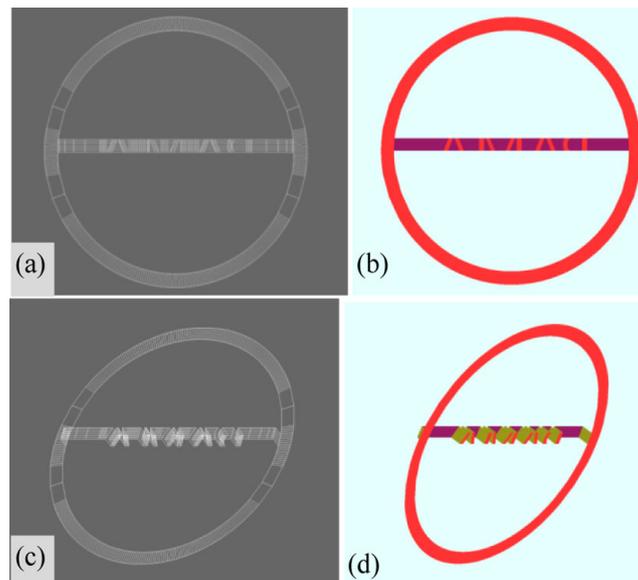


Figure 25. The modeling images of fringe ring and middle rows of top face of the stamp model. (a) The front view of wire-frame image. (b) The front view of fill-area image. (c) The side view of wire-frame image. (d) The side view of fill-area image.

The half top face of stamp model is shown in **Figure 27**. This unfinished model consists of the fringe ring and half top face of the stamp.

Most facets in this unfinished model are quads. Several triangle facets are filled at corners, and several pentagons and hexagons are formed in some facets with curved borders. It is shown in the front view and side view of wire-frame images of **Figure 27(a)** and **Figure 27(c)**. Facets of different parts are depicted in different colors in the front view and side view of fill-area images of **Figure 27(b)** and **Figure 27(d)**.

The tested results of the model in **Figure 27** are shown in **Figure 28**. It is seen that the number of vertices is 2809, the number of edges is 5219, and the number of facets is 2410. Its Euler characteristic is 0, which is equal to that of the corresponding compact surface with one inside hole. The partial model is verified at this stage.

3) Testing Top Face, Side Ring and Back Face

The top face, side ring and back face of stamp model are constructed at this stage, shown in **Figure 29**. All of facets of side ring and back face are quads.

Figure 29(a), **Figure 29(c)** and **Figure 29(e)** are the front view, side front

```
vertex number ***1213***vertex number
edge number ***2054***edge number
facet number ***840***facet number
Euler characteristic ***-1***Euler characteristic
```

Figure 26. A screenshot of execution results of EAT algorithm, displaying numbers of vertices, edges, and facets of the partial model in **Figure 25**, and its Euler characteristic of -1 .

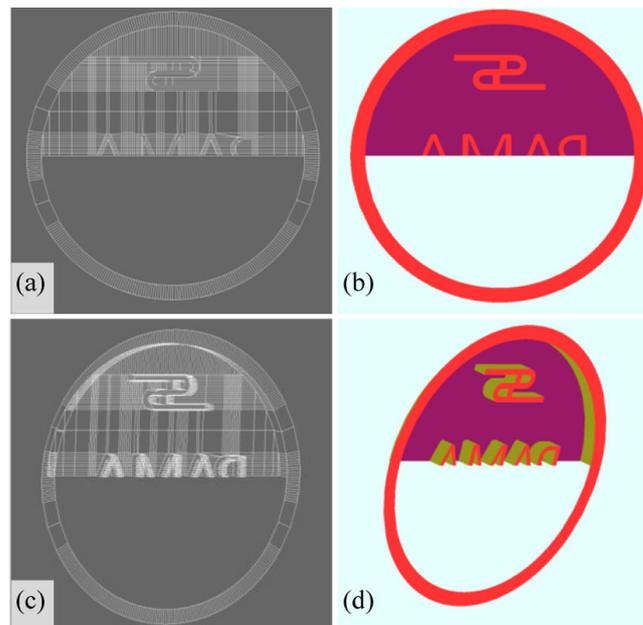


Figure 27. The modeling images of half top face of the stamp model. (a) The front view of wire-frame image. (b) The front view of fill-area image. (c) The side view of wire-frame image. (d) The side view of fill-area image.

```
vertex number ***2809***vertex number
edge number ***5219***edge number
facet number ***2410***facet number
Euler characteristic ***0***Euler characteristic
```

Figure 28. A screenshot of execution results of EAT algorithm, displaying numbers of vertices, edges, and facets of the partial model in **Figure 27**, and its Euler characteristic of 0.

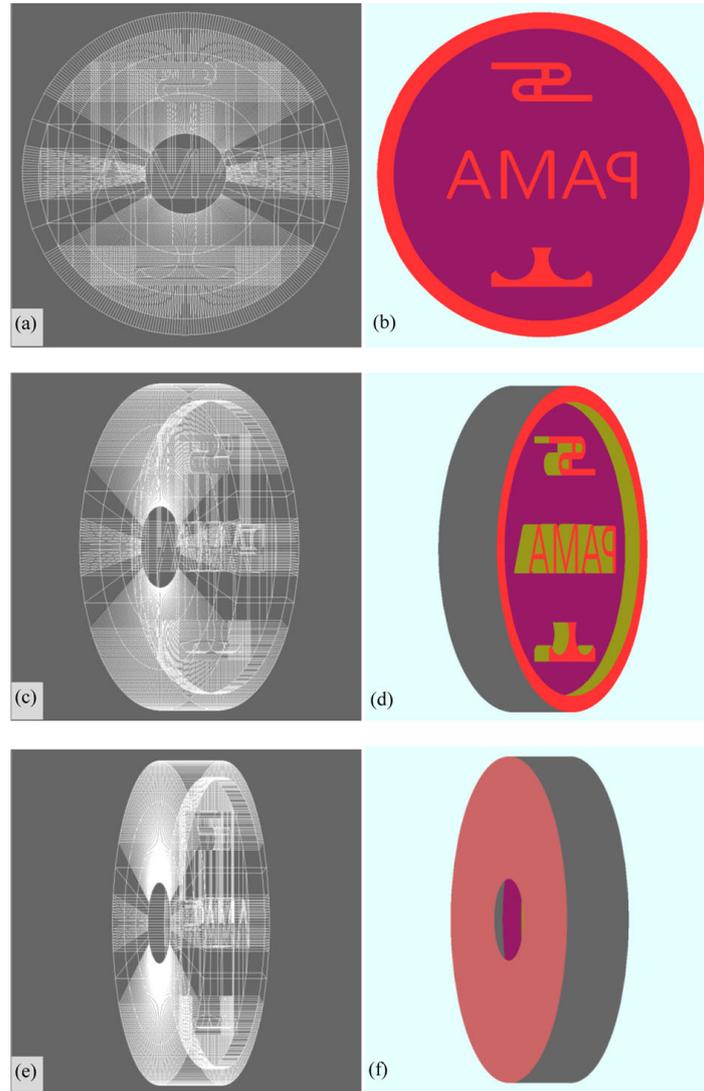


Figure 29. The modeling images of the top face, side ring and back face of the stamp model. (a) The front view of wire-frame image. (b) The front view of fill-area image. (c) The side front view of wire-frame image. (d) The side front view of fill-area image. (e) The side back view of wire-frame image. (f) The side back view of fill-area image.

view, and side back view of wire-frame images, respectively. **Figure 29(b)**, **Figure 29(d)** and **Figure 29(f)** are the front view, side front view, and side back view of fill-area images, respectively.

The tested results of the partial model in **Figure 29** are shown in **Figure 30**.

We can see that the number of vertices is 5769, the number of edges is 11,134, and the number of facets is 5366. Its Euler characteristic is 1, which is equal to that of the corresponding compact surface without any inside hole. The partial model is verified at this stage.

4) Testing the Whole Stamp Modeling

The whole stamp modeling is finished at this stage, shown in **Figure 31**. The finished model is composed of the top face, side ring, back face and handle of the stamp model. Most facets of the handle are quads, except the triangle facets at the tip of handle.

Figure 31(a) and **Figure 31(c)** are the front view and side back view of wire-frame images, respectively. **Figure 31(b)** and **Figure 31(d)** are the front view and side back view of fill-area images, respectively. **Figure 1** shows the front side views.

The tested results of the finished stamp model in **Figure 31** are shown in **Figure 32**. It is seen that the number of vertices is 9514, the number of edges is 18,910, and the number of facets is 9398. Its Euler characteristic is 2, which is consistent with that of the corresponding surface of a 3D ball. The whole modeling has been verified totally at this stage.

```
vertex number ***5769***vertex number
edge number ***11134***edge number
facet number ***5366***facet number
Euler characteristic***1***Euler characteristic
```

Figure 30. A screenshot of execution results of EAT algorithm, displaying numbers of vertices, edges, and facets of the partial model in **Figure 29**, and its Euler characteristic of 1.

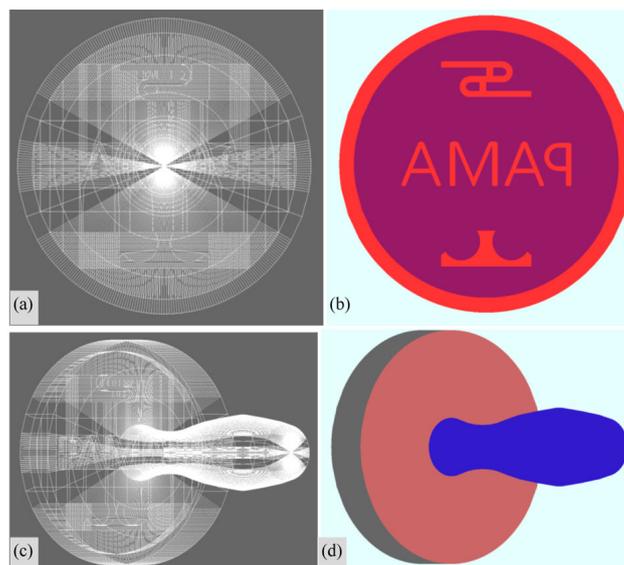


Figure 31. The modeling images of finished stamp model. The front view of (a) the wire-frame image and (b) fill-area image. The side back view of (c) wire-frame image and (d) fill-area image.

```

vertex number ***9514***vertex number
edge number ***18910***edge number
facet number ***9398***facet number
Euler characteristic***2***Euler characteristic

```

Figure 32. A screenshot of execution results of EAT algorithm, displaying numbers of vertices, edges, and facets of the finished model in **Figure 31**, and its Euler characteristic of 2.

For the EAT algorithm, there are more application cases than the above. Since we cannot present all of them, we only explain some more features here.

We first explain the meanings of three more indicators of the EAT algorithm in applications, these being the face counter fc , edge counter ec and vertex counter vc .

As known before, during testing, the EAT scans facets of a model one facet after another without any repeated facet. The facet counter fc value of the scanned facet is equal to 1. If its fv is greater than 1, the facet has a defect that must be modified. Because an edge in a simply connected surface model can be connected to two facets at most, the edge counter ec value of the edge is smaller than or equal to 2. If the ec value is greater than 2, the edge has a defect that must be modified or the tested surface model is not simply connected. The value of a vertex counter vc in a scanned facet can be a positive integer, which means there are vc number of edges connected to the vertex item.

In addition, for the model companion, one of the important features of the EAT algorithm is that the defect of a facet can be tested, found, and immediately modified at the stage of this facet modeling. In this way, a verified model can be always provided to the succeeding process, shown in **Figure 2**.

5. Conclusions

With the algebraic topology, the EAT algorithm opens a new test window for observing and examining surface modeling in surface construction and reconstruction. It computes the Euler characteristic of a model, and examines the topologic consistence of the model by comparing the Euler characteristic of the model with that of its target surface.

The EAT algorithm consists of two functions: EAT testing algorithm (EAT-TA) and statistic algorithm (EAT-SA). Both functions are short and flexible, so they are easily inserted in a model companion program. Even better, a clean and verified model can be provided without the EAT code left after the EAT-searching.

The EAT algorithm is adaptable and can be used to test different polygon facets of various surface models in surface construction and reconstruction. The EAT algorithm is independent of the metrics of facets. A facet much smaller than most of facets in a model can be identified.

In future, the EAT algorithm will be applied to test more forms of models that may have different values of Euler characteristics and different topologic structures. It can be sure that more potential of the EAT algorithm can be revealed.

Acknowledgements

This work has been supported by Hebei Project of Sponsoring High-Level Talents [grant number C2015003045], Doctoral Research Foundation of Hebei University of Science and Technology [grant number 201417], and Hebei Project of Sponsoring Introduced Overseas Researchers [grant number C201811].

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Guo, X., Xiao, J. and Wang, Y. (2018) A Survey on Algorithms of Hole Filling in 3D Surface Reconstruction. *Visual Computer*, **34**, 93-103. <https://doi.org/10.1007/s00371-016-1316-y>
- [2] Jun, Y. (2005) A Piecewise Hole Filling Algorithm in Reverse Engineering. *Computer-Aided Design*, **37**, 263-270. <https://doi.org/10.1016/j.cad.2004.06.012>
- [3] Attene, M., Campen, M. and Kobbelt, L. (2013) Polygon Mesh Repairing: An Application Perspective. *ACM Computing Surveys*, **45**, Article No. 15. <https://doi.org/10.1145/2431211.2431214>
- [4] Casciola, G., Lazzaro, D., Montefusco, L.B. and Morigi, S. (2005) Fast Surface Reconstruction and Hole Filling Using Positive Definite Radial Basis Functions. *Numerical Algorithms*, **39**, 289-305. <https://doi.org/10.1007/s11075-004-3643-8>
- [5] Wu, X. and Chen, W. (2014) A Scattered Point Set Hole-Filling Method Based on Boundary Extension and Convergence. *Proceedings of the 11th World Congress on Intelligent Control and Automation*, Shenyang, 29 June-4 July 2014, 5329-5334. <https://doi.org/10.1109/WCICA.2014.7053624>
- [6] Doria, D. and Radke, R.J. (2012) Filling Large Holes in LiDAR Data by Inpainting Depth Gradients. *Proceedings of 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Providence, 16-21 June 2012, 65-72. <https://doi.org/10.1109/CVPRW.2012.6238916>
- [7] Yang, N.-E., Kim, Y.-G. and Park, R.-H. (2012) Depth Hole Filling Using the Depth Distribution of Neighboring Regions of Depth Holes in the Kinect Sensor. *Proceedings of 2012 IEEE International Conference on Signal Processing, Communication and Computing*, Hong Kong, 31 July-3 August 2012, 658-661. <https://doi.org/10.1109/ICSPCC.2012.6335696>
- [8] Quinsat, Y. and Iartigue, C. (2015) Filling Holes in Digitized Point Cloud Using a Morphing-Based Approach to Preserve Volume Characteristics. *The International Journal of Advanced Manufacturing Technology*, **81**, 411-421. <https://doi.org/10.1007/s00170-015-7185-0>
- [9] Bendels, G.H., Schnabel, R. and Klein, R. (2006) Detecting Holes in Point Set Surface. *Journal of WSCG*, **14**, 89-96.
- [10] Methirumangalath, S., Kannan, S.S., Parakkat, A.D. and Muthuganapathy, R. (2017) Hole Detection in a Planar Point Set: An Empty Disk Approach. *Computers & Graphics*, **66**, 124-134. <https://doi.org/10.1016/j.cag.2017.05.006>
- [11] Cunningham, S. (2008) *Computer Graphics: Programming in OpenGL for Visual Communication*. Prentice Hall Inc., Upper Saddle River.
- [12] Botsch, M. and Sorkine, O. (2008) On Linear Variational Surface Deformation Me-

- thods. *IEEE Transactions on Visualization and Computer Graphics*, **14**, 213-230. <https://doi.org/10.1109/TVCG.2007.1054>
- [13] Xu, Q., Wang, J. and An, X. (2014) A Pipeline for Surface Reconstruction of 3-Dimensional Point Cloud. *Proceedings of 2014 International Conference on Audio, Language and Image Processing*, Shanghai, 7-9 July 2014, 822-826. <https://doi.org/10.1109/ICALIP.2014.7009909>
- [14] Wiemann, T., Mitschke, I., Mock, A. and Hertzberg, J. (2018) Surface Reconstruction from Arbitrarily Large Point Clouds. *Proceedings of 2018 Second IEEE International Conference on Robotic Computing*, Laguna Hills, 31 January-2 February 2018, 278-281. <https://doi.org/10.1109/IRC.2018.00059>
- [15] Odesanya, O.S., Waggenspack, W.N. and Thompson, D.E. (1993) Construction of Biological Surface Models from Cross-Sections. *IEEE Transactions on Biomedical Engineering*, **40**, 329-334. <https://doi.org/10.1109/10.222325>
- [16] Shi, J., Zhu, L., Li, Z., Yang, J. and Wang, X. (2017) A Design and Fabrication Method for a Heterogeneous Model of 3D Bio-Printing. *IEEE Access*, **5**, 5347-5353. <https://doi.org/10.1109/ACCESS.2017.2692248>
- [17] Fuhrmann, M. and Goesele, M. (2014) Floating Scale Surface Reconstruction. *ACM Transactions on Graphics*, **33**, Article No. 46. <https://doi.org/10.1145/2601097.2601163>
- [18] Lávička, M., Šír, A. and Vršek, J. (2016) Smooth Surface Interpolation Using Patches with Rational Offset. *Computer Aided Geometric Design*, **48**, 75-85. <https://doi.org/10.1016/j.cagd.2016.09.002>
- [19] Shen, J., Kosinka, J., Sabin, M. and Dodgson, N. (2016) Converting a CAD Model into a Non-Uniform Subdivision Surface. *Computer Aided Geometric Design*, **48**, 17-35. <https://doi.org/10.1016/j.cagd.2016.07.003>
- [20] Sederberg, T.W., Zheng, J., Bakenov, A. and Nasri, A. (2003) T-Splines and T-NURCCs. *ACM Transactions on Graphics*, **22**, 477-484. <https://doi.org/10.1145/882262.882295>
- [21] Coêlho, J., Gattass, M. and Lopes, H. (2019) ARTMe: A New Array-Based Algorithm for Adaptive Refinement of Triangle Meshes. *Engineering with Computers*, **35**, 1-20. <https://doi.org/10.1007/s00366-018-0579-5>
- [22] Bock, K. and Stiller, J. (2018) Optimizing Triangular High-Order Surface Meshes by Energy-Minimization. *Engineering with Computers*, **34**, 659-670. <https://doi.org/10.1007/s00366-017-0565-3>
- [23] Chen, H. and Shen, J. (2018) Denoising of Point Cloud Data for Computer-Aided Design, Engineering and Manufacturing. *Engineering with Computers*, **34**, 523-541. <https://doi.org/10.1007/s00366-017-0556-4>
- [24] Hoschek, J. and Lasser, D. (1993) *Fundamentals of Computer Aided Geometric Design*. A K Peters, Wellesely.
- [25] Baker, H. (2004) *Computer Graphics with OpenGL*. Third Edition, Pearson Prentice Hall, Upper Saddle River.
- [26] Wang, Y., Zheng, J. and Wang, H. (2019) Fast Mesh Simplification Method for Three-Dimensional Geometric Models with Feature-Preserving Efficiency. *Scientific Programming*, **2019**, Article ID: 4926190. <https://doi.org/10.1155/2019/4926190>
- [27] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W. (1992) Surface Reconstruction from Unorganized Points. *Proceedings of SIGGRAPH'92*, Chicago, 26-31 July 1992, 71-78. <https://doi.org/10.1145/133994.134011>
- [28] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W. (1993) Mesh

- Optimization. *Proceedings of SIGGRAPH93*, Anaheim, 2-6 August 1993, 19-26. <https://doi.org/10.1145/166117.166119>
- [29] Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., Mcdonald, J., Schweitzer, J. and Stuetzle, W. (1994) Piecewise Smooth Surface Reconstruction. *Proceedings of SIGGRAPH94*, Orlando, 24-29 July 1994, 295-302. <https://doi.org/10.1145/192161.192233>
- [30] Turk, G. and Levoy, M. (1994) Zippered Polygon Meshes from Range Images. *Proceedings of SIGGRAPH94*, Orlando, 24-29 July 1994, 311-318. <https://doi.org/10.1145/192161.192241>
- [31] Catmull, E. and Clark, J. (1978) Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes. *Computer-Aided Design*, **10**, 350-355. [https://doi.org/10.1016/0010-4485\(78\)90110-0](https://doi.org/10.1016/0010-4485(78)90110-0)
- [32] Doo, D.W.H. and Sabin, M.A. (1978) Behaviour of Recursive Division Surfaces near Extraordinary Points. *Computer-Aided Design*, **10**, 356-360. [https://doi.org/10.1016/0010-4485\(78\)90111-2](https://doi.org/10.1016/0010-4485(78)90111-2)
- [33] Bolz, J. and Schröder, P. (2002) Rapid Evaluation of Catmull-Clark Subdivision Surfaces. *Proceedings of the Seventh International Conference on 3D Web Technology*, Tempe, 24-28 February 2002, 11-17. <https://doi.org/10.1145/504502.504505>
- [34] Forsey, D.R. and Bartels, R.H. (1988) Hierarchical B-Spline Refinement. *ACM SIGGRAPH Computer Graphics*, Vol. 22, 205-212. <https://doi.org/10.1145/54852.378512>
- [35] Sederberg, T.W., Zheng, J., Sewell, D. and Sabin, M. (1998) Non-Uniform Recursive Subdivision Surfaces. *Proceedings of ACM SIGGRAPH98*, Orlando, 19-24 July 1998, 387-394. <https://doi.org/10.1145/280814.280942>
- [36] Velho, L. and Zorin, D. (2001) 4-8 Subdivision. *Computer Aided Geometric Design*, **18**, 397-427. [https://doi.org/10.1016/S0167-8396\(01\)00039-5](https://doi.org/10.1016/S0167-8396(01)00039-5)
- [37] Müller, K., Reusche, L. and Fellner, D. (2006) Extended Subdivision Surfaces: Building a Bridge between NURBS and Catmull-Clark Surface. *ACM Transactions on Graphics*, **25**, 268-292. <https://doi.org/10.1145/1138450.1138455>
- [38] Cashman, T.J., Augsdörfer, U.H., Dodgson, N.A. and Sabin, M.A. (2009) NURBS with Extraordinary Points: High-Degree, Non-Uniform, Rational Subdivision Schemes. *ACM Transactions on Graphics*, **28**, Article No. 46. <https://doi.org/10.1145/1531326.1531352>
- [39] Cashman, T.J. (2010) NURBS-Compatible Subdivision Surfaces. Doctoral Thesis, University of Cambridge, Cambridge. <https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.608572>
- [40] Shen, J., Kosinka, J., Sabin, M.A. and Dodgson, N.A. (2014) Conversion of Trimmed NURBS Surfaces to Catmull-Clark Subdivision Surfaces. *Computer Aided Geometric Design*, **31**, 486-498. <https://doi.org/10.1016/j.cagd.2014.06.004>
- [41] Campen, M. and Zorin, D. (2017) Similarity Maps and Field-Guided T-Splines: A Perfect Couple. *ACM Transactions on Graphics*, **36**, Article No. 91. <https://doi.org/10.1145/3072959.3073647>
- [42] Massy, W.S. (1991) Algebraic Topology: An Introduction. Springer-Verlag, Berlin/Heidelberg/New York.
- [43] Hirsch, M.W. (2001) Differential Topology. Springer-Verlag, Berlin/Heidelberg/New York.