AN EFFECTIVE HOLE DETECTION METHOD FOR 3D MODELS

Yuehong Wang, Rujie Liu, Fei Li

Fujitsu R&D Center, Beijing, China
{wangyh, rjliu, lifei}@cn.fujitsu.com

Susumu Endo, Takayuki Baba, Yusuke Uehara

Fujitsu Laboratories LTD. Kawasaki, Japan
{endou.susumu-02,baba-t,yuehara}@jp.fujitsu.com

ABSTRACT

An automatic hole detection approach is proposed for triangular mesh models in this paper. Different to some other existing methods, the proposed method here aims to find solid holes inside 3D models instead of detecting polygon missing regions for geometry incorrect 3D models. In the method, a number of planes are firstly detected by grouping interconnected coplanar triangles, then model contour is extracted from the boundaries of adjacent planes, and after that model vertices are grouped into several disjoint clusters based on the extracted contour. By analyzing the relationship between planes and vertex clusters, all the holes inside 3D models are finally identified. Experiments verify that our method is efficient and effective.

Index Terms— Three-dimensional models, planes, contour, hole, triangular mesh

1. INTRODUCTION

Three-dimensional (3D) techniques such as model retrieval, segmentation and matching, have attracted more and more attention. For these applications, various structure characteristics are popularly used, and among them hole is an important semantical feature. Taking model retrieval for example, if the query model has holes, hole feature could be used to filter many 3D models without holes, thereby the retrieval speed can be greatly improved and more meaningful results could be achieved. Additionally, the hole detection results can be directly used as input for hole filling [1, 2, 3, 4], and other fields.

There are some methods [5, 6, 7] related to hole identification or detection. In [5], ray-scanning is used to generate 3D point clouds of cavity region, and then point clouds are filtered and triangulated to construct irregular shaped models. In [6], X-ray inspection method is proposed to measure the internal geometry feature of 3D objects and can position the drilled holes. Both of the aforementioned methods require special equipment. For digitalized 3D models, there isn't much work published about hole detection as far as we know. For example to restore patch missing area, the method in [7] is proposed to identify the hole boundary of 3D models, where one curve is manually drawn firstly and then the hole boundaries of inner surface and outer surface are searched according to the points of smoothed curve. Besides, in our previous segmentation method [8], hole detection is one step to keep the original shape after segmentation, where inclusion relationship of divided parts need to be calculated and this step is not so efficient for complicated models.

Toward the above problems, one automatic hole detection method is proposed for triangular mesh models in this paper. In our method, the basic geometry elements, namely, the planes contained in model surface and model contour are extracted and utilized. There are three primary stages as follows for detecting holes inside 3D models.

1. **Plane Detection:** This stage aims to detect the planes contained in the model surface, where the adjacency information and normal direction of triangle faces are used. In this stage, the interconnected coplanar triangle faces are grouped to form a number of planes. Compared to the method in [8], the detected planes are more simple, which is helpful to simplify the subsequent stages for efficiency improvement.

2. **Contour Extraction:** In this stage, model contour is robustly extracted, where the adjacency information of detected planes is used. The contour lines are determined by checking the line segments in the boundaries of adjacent planes, and these contour lines form a contour graph. Compared to [8], the new contour extraction algorithm is more robust for imperfect models with triangle intersection or patch missing.

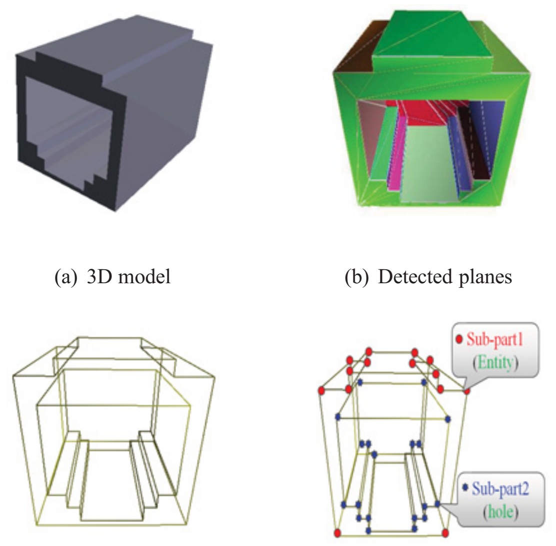
3. **Hole Identification:** From the contour graph, it is found that the vertices of each hole usually form one individual sub-graphs. In this stage, the connected vertices of contour graph are firstly clustered into several sets and form sub-graphs, and then the sub-graphs corresponding to holes are identified by analyzing the relationship between the sub-graphs and the detected planes. In this stage, some rules are designed.

There are two main merits of our method: 1) The contour extraction algorithm is robust; 2) The hole identification process is efficient due to simple processes such as convex hull calculation. Experiments also verify that our method is efficient and effective.

The paper is structured as follows. The following section introduces our hole detection method including the basic framework and the details of its three stages. The experimental results are presented in section 3, followed by the conclusion in section 4.

2. HOLE DETECTION METHOD

Figure 1 presents the overall framework of our proposed hole detection approach, where the input is a triangular mesh model and the output is a series of detected holes. Our method consists of three primary steps: plane detection, contour extraction, and hole identification based on planes and model contour. In following sections, the three steps will be presented in detail.



(a) 3D model (b) Detected planes

(c) Extracted contour graph (d) Hole identification result

Fig. 1. The framework of our proposed hole detection method.

2.1. Plane detection

This step aims to detect the planes contained in model surface. In this step, the normal directions of all the triangles are firstly calculated and the directions are assumed to point outwards of model surface; and then the interconnected coplanar triangles are merged to form a number of planes. For instance, for the model in Fig.1(a), the detected planes are shown in Fig.1(b) with different colors. The proposed plane detection algorithm is efficient, and furthermore the geometry structure of the detected planes are more simple compared to our previous work, which can simplify the subsequent stages.

2.2. Contour extraction

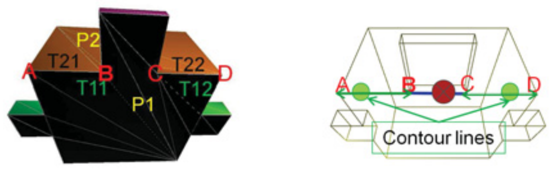
For some complicated models particulary for scanned models, the problems of patch missing and triangle intersection often occur, and these problems usually affect the performance of contour extraction. For example in this case, it is difficult or impossible to extract closed boundary of planes.

To decrease the impact of such problems, we present a new algorithm, where the contour lines are extracted from the boundaries of adjacent planes, and the extracted contour lines construct one contour graph (*e.g.* Fig.1(c)). In our method, two rules are designed to determine the contour lines: 1) The contour line must lie in the boundary of two adjacent planes; 2) In each of the two adjacent planes, there must exist at least one triangle that includes or intersects with the contour line. In this manner, most of contour lines can be found though perfect closed contour may be not extracted for quite complicated models or geometry incorrect models. Based on these two rules, the following primary steps are applied to extract all contour lines and Fig.2 gives one simple example for clear illustration.

1. For each detected plane, find and record all its adjacent planes, *e.g.* in Fig.2(a), the plane *p1* and *p2* are a pair of adjacent planes.
2. For every pair of adjacent planes (taking *p1* and *p2* in Fig.2(a) as an example), the contour lines are extracted from their boundaries as follows:
   - Find the points shared by them, *e.g.* four vertices are found for *p1* and *p2*, namely, *A*, *B*, *C*, *D*.
   - Calculate the non-overlapped line segments among the found points, and these line segments lie in their boundary definitely. For *p1* and *p2*, the calculated line segments are *A*-*B*, *B*-*C* and *C*-*D*, and the three line segments satisfy the first rule of contour lines.
   - Check all the line-segments to extract the contour lines based on the pre-designed second rule. For example in Fig.2(a), the line segment *A*-*B* is included in the triangle *T11* in plane *p1* and the triangle *T21* in plane *p2*, therefore it is judged as one contour line. In a similar way, the line segment *C*-*D* is considered as a contour line too. However, the line segment *B*-*C* is not a contour line because no triangle exists in plane *p2* that includes or intersects with *B*-*C*. The extracted contour is shown in Fig.2(b).

2.3. Hole identification

By analyzing the characteristics of contour graph, it is found that hole vertices usually connect to themselves and form individual sub-graphs, *e.g.* the hole *cluster 2* in Fig.3(b). The



(a) (b)

Fig. 2. The illustration of contour extraction.

main reason is that the holes must be completely included by some entities and accordingly hole vertices are usually inside the region of the planes (*e.g.* the specified plane in Fig.3(a)) which connect the hole and its nearest entity, and inside lines of such planes are not considered as contour lines in contour extraction stage.

Based on this finding, the model vertices are firstly clustered to form several sub-graphs, and then the relationship between the planes and sub-graphs is analyzed to detect the holes. To explain our hole identification algorithm clearly, the model shown in Fig.3(a) is used as an example, and the detail of identification algorithm is presented as follows.

1. Group the interconnected vertices through contour lines to form several separate sub-graphs, *e.g.* in Fig.3(b), the model vertices are clustered into two sets.

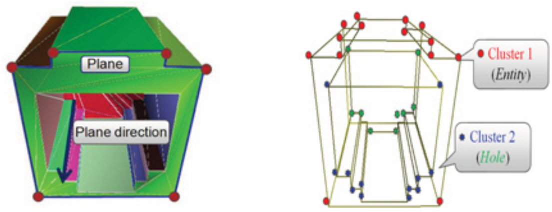
2. Calculate the correspondence relationship between vertex clusters and detected planes, and find all the planes which are shared by two or more clusters, *e.g.* the specified plane in Fig.3(a).

3. Process all the found planes in sequence as follows until all the sub-graphs corresponding to holes are identified, and the specified plane shown in Fig.3(a) is used as an illustration example.

    - Calculate the convex hull of the plane points, *e.g.* the points marked by red circle in the specified plane in Fig.3(a).
    - As for every hole, it must be completely included by other parts, so it's impossible to share points with convex hull. Therefore in this step, such sub-graphs that share some points with convex hull are identified as entities (not holes). As a result, *cluster 1* in Fig.3(b) is identified as an entity because it has six convex hull points.
    - Process the left vertex clusters to identify the holes as follows, using *cluster 2* in Fig.3(b) as an example.

        (a) Find such points from the cluster that satisfy two conditions: 1) not in the specified plane; 2) directly connect to the specified plane by contour lines. For *cluster 2*, the found points are marked in green in Fig.3(b).

        (b) If the found points lie in negative side of the specified plane, the cluster is identified as a hole. Otherwise, it is judged as an entity. For *cluster 2*, the found points in green are in negative side of the specified plane in Fig.3(a), so *cluster 2* is identified as a hole. This rule is set based on the finding that the vertices of holes should be inward instead of outward relative to the specified plane, because the holes must be included by its neighboring entities.



(a) (b)

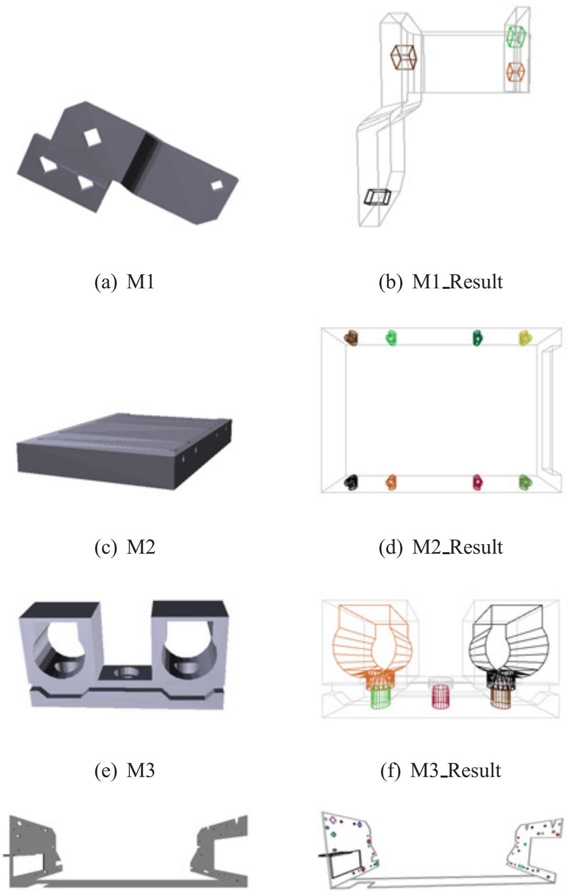
Fig. 3. The illustration of hole identification.

3. EXPERIMENTAL RESULTS

3.1. Performance evaluation

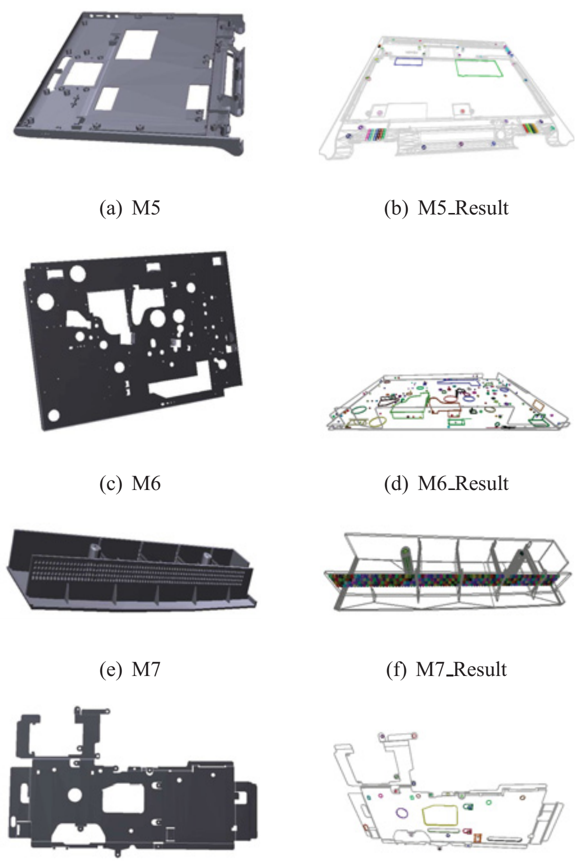
To evaluate the performance of our hole detection method, the experiments are conducted on simple models and complicated models. Here, the number of triangle faces is used to characterize the model complexity. The models with less than 5000 triangle faces are treated as simple models, and other models are considered as complicated models. As page limited, only several detection results are displayed in Fig.4 and Fig.5 respectively. In each row of two figures, the left one denotes the triangular mesh model and the right one represents the hole detection result. Additionally for some models, the viewpoints of two figures in a row are different in order to show more structure information of 3D models. To show detection result clearly, the contour lines of entities are displayed in gray, while the contour lines of holes are shown in different colors. From these pictures, it is found that the detection accuracy is satisfactory particularly for simple models.

Besides the visual results, the experiments about detection efficiency are also conducted, and the results are showed in Tab.1. Here, the experiments are conducted on a PC with Intel(R) Xeon(TM) CPU(3GHz) and 4GB memories. In the table, *NV*, *NT*, *NC* and *NH* represent the numbers of model vertices, model triangles, separated sub-graphs and identified holes, and *TC* denotes the time cost of hole detection. From



(a) M1 (b) M1_Result (c) M2 (d) M2_Result (e) M3 (f) M3_Result (g) M4 (h) M4_Result

Fig. 4. Hole detection results for some simple models.



(a) M5 (b) M5_Result (c) M6 (d) M6_Result (e) M7 (f) M7_Result (g) M8 (h) M8_Result

Fig. 5. Hole detection results for some complicated models.

this table, it is evident that our method can detect holes for most 3D models in real time, while several minutes may be required for complicated models with the method [8].

In our hole detection method, there are two main factors that affect the detection speed. 1) The time cost generally increases with the numbers of vertices and triangle faces as the two numbers characterize the model complexity to some degree. 2) The number of separated sub-graphs also influences the detection speed because most of vertex clusters need be checked in hole identification process.

3.2. Further analysis

As shown in Fig.4 and Fig.5, most of holes can be detected very well. However, the detection result may need improvement for some complicated models, *e.g.* the result of the zoomed area (marked with red lines in Fig.6(a)) is shown in Fig.6(b). In this region, the left hole is correctly detected, while the right one cannot be detected because it connects to the bottom plate by contour lines. In this case, segmentation module like our previous method [8] can be adopted to divide the holes from entities firstly, and then holes can be identified by analyzing the divide parts based on the information of contour and planes.

Table 1. Some data about experiments of detection efficiency.

	NV	*NT*	*NC*	*NH*	*TC(ms)*
M1	84	180	5	4	16
M2	384	796	9	8	31
M3	404	824	7	5	31
M4	551	1246	51	42	63
M5	3431	7006	81	54	343
M6	7879	16364	2054	149	687
M7	9340	19980	826	291	1031
M8	11388	22172	769	30	2437

(a) (b)

Fig. 6. Further analysis example.

4. CONCLUSION

In this paper, a novel hole detection method is proposed to find all the 3D solid holes inside triangular mesh models. In the method, such geometry characteristics of model contour and plane are extracted and utilized. The method consists of three primary processes: plane detection for detecting the planes contained in model surface, robust contour extraction for extracting contour lines from the boundaries of adjacent planes, and efficient hole identification for detecting holes by analyzing the contour graph and designing some rules. The experimental results show that our method can detect most of holes inside 3D models efficiently even for complicated models.

5. REFERENCES

[1] Aktouf Z., Bertrand G., and Perroton L., "A 3D-hole closing algorithm," in *6th International Workshop on Discrete Geometry for Computer Imagery*, 1996, pp. 36–47.

[2] Davis J., Marschne S.R., and Garr M., "Filling holes in complex surfaces using volumetric diffusion," in *IEEE First International Symposium on 3D Data Processing, Visualization, and Transmission*, 2001, pp. 428–438.

[3] Y. Jun, "A piecewise hole filling algorithm in reverse engineering," *Computer-Aided Design*, vol. 37, pp. 263–270, February 2005.

[4] R. Sagawa and K. Ikeuchi, "Hole filling of a 3D model by flipping signs of a signed distance field in adaptive resolution," *IEEE Transcations on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 686–699, April 2008.

[5] Liu M., Luo Z., and Yuan W., "3D information acquisition of disaster cavity based on CMS," *Science and Technology Review*, vol. 29, pp. 32–36, April 2011.

[6] J.A. Noble, R. Gupta, and J. Mundy, "High precision X-Ray stereo for automated 3D CAD-based inspection," *IEEE Transcations on Robotics and Automation*, vol. 14, pp. 292–302, April 1998.

[7] Kong L., Yao Y., and Hu Q, "Hole boundary identification algorithm for 3D closed triangle mesh," *Computer Engineering*, vol. 36, pp. 177–180, September 2010.

[8] Wang Y., Liu R., and Endo S., "A novel 3D model segmentation approach," in *International Conference on Computer Graphics and Imaging*, 2010, pp. 172–178.