User Interaction-Based Mesh Simplification for Preserving Geometric Shapes

Jinsuk Yang, Kangweon Jeong, Kyoungsu Oh, and Youngjo Chun

***Abstract*—**As 3D scanner technologies have developed, the reverse engineering process that scans and models an object has improved. Because these reverse engineered models are represented as triangle meshes, which contain many faces, they do provide difficulties including large storage capacity for digital memory, slow rendering speeds during visualizations and awkward manual user editing. To improve on these challenges, techniques have been studied to try to simplify the mesh structures while maintaining adequate geometry, such as edge detail, to properly represent the scanned object. However, previous methods focus on the automatic simplification of an entire mesh shape, which has the drawback that the mesh shape significantly changes. This paper presents a new mesh simplification based on user interaction where the user may manually divide a mesh into many different segments. Each of these segments can then be simplified by a reduction ratio depending on the user's criteria. The proposed method simplifies a mesh by taking into consideration the user's view and thus can more effectively maintain its necessary geometric shape.

***Keywords*—**Mesh decimation, Mesh segmentation, Mesh simplification, User-guided simplification.

I. Introduction

WITH current 3D scanner technologies, we are able to scan an object, digitally model it, and reverse engineer without manually making the object [1]. The reversed engineered models are typically represented as a structure of triangle meshes (with connected vertices), so it is possible to generate models with increased sophistication as the increases in precision. However, denser mesh models require simplification to improve rendering speed, reduce storage requirements, allow for easier editing by the user, and reduce printing costs. The process of mesh simplification is to reduce the number of faces while sufficiently preserving its original geometry [2], as is demonstrated with an example in Fig. 1.

Because of these important benefits, mesh simplification has been thoroughly studied [3]-[5] with the primary goal being to preserve the original mesh shape. For example, with the technique of decimation of triangle meshes [6], the characteristics of the vertices of the mesh were analyzed and those which had a curvature lower than a threshold level were removed and re-triangulated. Mesh optimization [7] calculations determined the curvature of the edges connecting each vertex and identified the edges that had a particular curvature. Then, the simplified mesh is created by connecting two vertices of the identified edges. However, this mesh simplification technique is limited to how accurate it can preserve the original mesh shape because it runs automatically. This paper proposes a new method for simplifying meshes by utilizing user interaction to overcome the limits of previous automated methods, in particular with preserving the original mesh shape. To accomplish this, a mesh is first automatically divided into several segments for the user to consider and compare to that of the original geometry. Each segment may then be combined or further divided manually by the user. Through this process, segments having a more precise shape may be obtained and the user may simplify the segments differently according to the characteristics of the faces of the segment shape. For example, plane segments might be treated with significant simplification while curved segments might have less. Finally, all of the simplified segments as determined by the user are combined into one mesh. The advantages of this proposed method include:

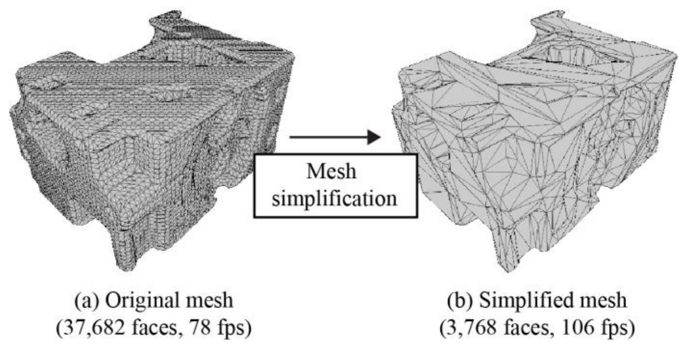

Fig. 1 Result of Mesh Simplification: The simplified mesh has improved rendering speed, reduced storage needs, more convenient for editing by the user, and has a lesser cost of printing compared to the original mesh all while maintaining a reasonably accurate representation of the initial model.

- A mesh can be first automatically divided into many segments, each of which may then be combined or further

Jinsuk Yang is with the Department of Media, Soongsil University, Seoul, South Korea (e-mail: ispio@ssu.ac.kr).

Kangweon Jeong is with the Department of Media, Soongsil University, Seoul, South Korea (e-mail: geekallthings@gmail.com).

Kyoungsu Oh is with the Department of Media, Soongsil University, Seoul, South Korea (corresponding author's phone: +8228223924; e-mail: oks@ssu.ac.kr).

Youngjo Chun is with the Department of Media, Soongsil University, Seoul, South Korea (e-mail: cyj@ssu.ac.kr).

divided manually at the user's discretion so that the finalized segments may more precisely represent the original shape.

- The user may now differentiate the level of simplification of the segments according to the characteristics (*e.g.,* planar or curved) of the faces of each segment by assigning a simplification value. Simplifying each segment individually is more effective in preserving the original shape of a mesh than doing so automatically over the entire mesh.

II. Mesh Simplification Based On User Interaction

Since previous mesh simplification methods use automatic processes uniformly covering the whole of the mesh shape, they typically cannot adequately maintain the original shape of mesh with the resulting edges, as shown Fig. 2. To better maintain the original shape of a mesh, we demonstrate an improved simplification process that first automatically segments a mesh

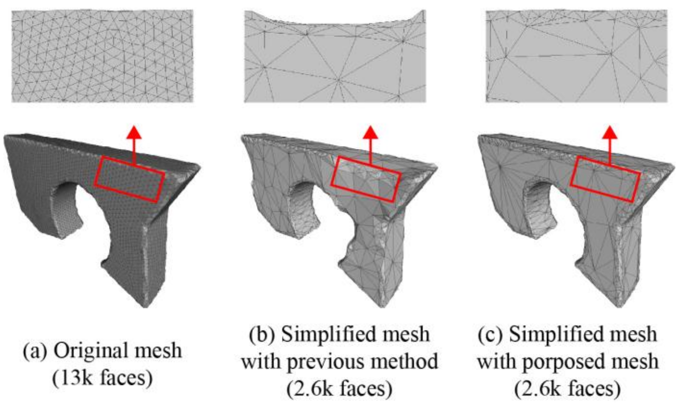

Fig. 2 Comparison between different mesh simplifications: (a) Original mesh. (b) Simplified mesh with automated simplification [9]. (c) Simplified mesh with proposed simplification. The proposed simplification can preserve the original mesh shapes.

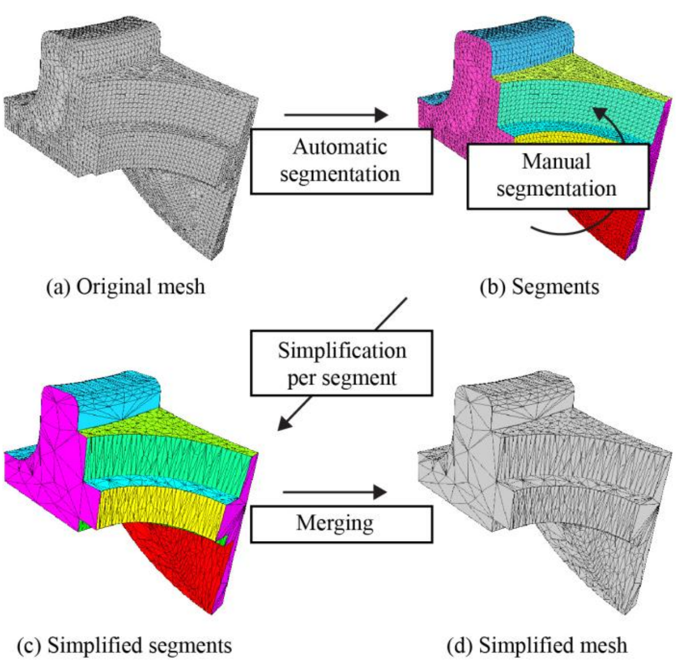

Fig. 3 Visual overview of the proposed method utilizing user interaction

(Section II-*A*) and then is additionally simplified and segmented manually by the user (Section II-*B*). These manually determined segmented meshes are categorized by their individual shape and then a degree of simplification is assigned by the user to further simplify each segment (Section II-*C*). Finally, the simplified segments are automatically merged into a single mesh, which forms the completed simplified mesh. The proposed method is illustrated in Fig. 3.

A. Automatic Segmentation

The initial automatic segmentation process divides a mesh into several large segments depending on its shape. These divided segments are separated according to the difference of the normal vector between adjacent faces. For example, when the normal vector of two adjacent sides is smaller than a threshold value, the two faces are categorized as being a part of the same segment, otherwise they are labelled as different segments. Here, the set threshold value is input from the user to initiate the automatic segmentation.

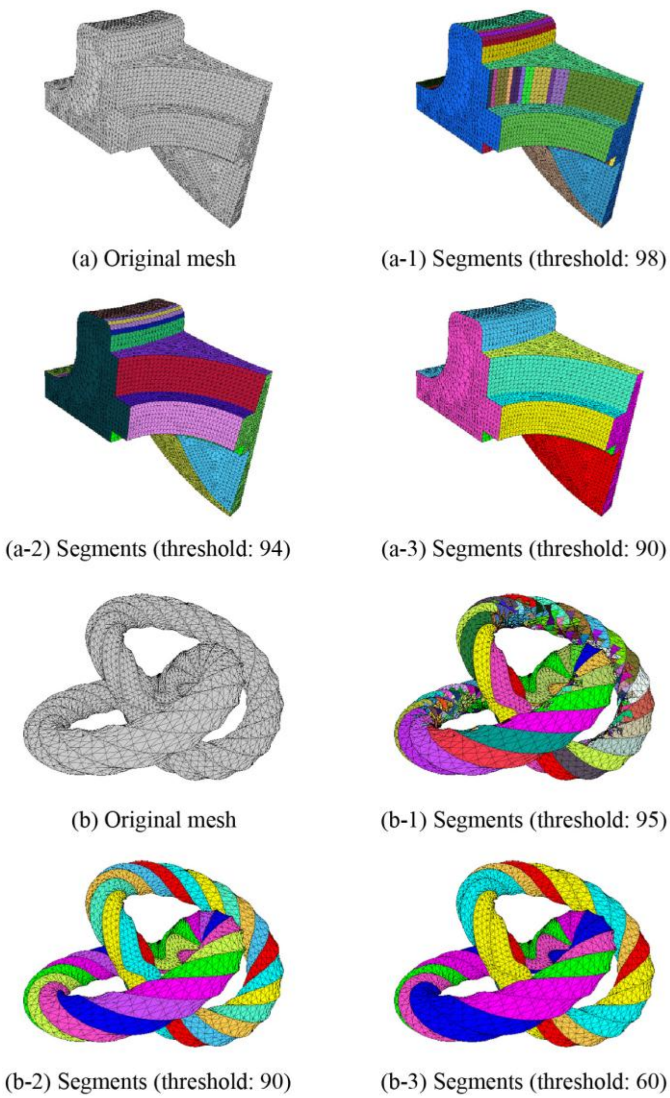

Fig. 4 The result of automatic segmentation. The higher the threshold, the more a mesh is segmented.

The method utilized here to automatically divide the mesh into several segments is similar to flood-fill algorithm [8]. First, one face *f1* of the mesh is selected at random and identified as the first segment. Next, the inner product of the normal vector for *f1* and its neighboring face, *f2,* is computed. If the result is smaller than the preset threshold, assign *f2* into the first segment. Continue by computing the inner product of the normal vector for another face *f3* adjacent to *f1* to determine if it should be placed in the first segment. Likewise, compute the inner product of the normal vectors of all the faces close to *f1* to combine into the first segment as necessary. Then, compute the inner product of the normal vector for *f4,* which is the first segment containing all of the new added segments, and the faces adjacent to *f4*. The same procedure is repeated until there are no remaining faces to calculate for consideration into the first segment. The remaining segments are then processed through this same routine until all faces in the mesh have been assigned to a segment.

The resulting mesh shape is then comprised of several segments containing all of the original faces. Fig. 4 shows the result of this automatic process, which may be optimized by the user through the adjustment of the threshold value to determine joining facing.

B. Manual Segmentation

The segments automatically divided following the procedure outline in Section II.*A* can be further improved by user feedback. For example, if the user determines that two segments have a similar shape, then these two may be merged by combining them into a single segment. Or, a segment that has been merged with faces that are too different can be re-divided into separate segments by drawing a line along the segment where the split should occur. The additional manual process of further merging and splitting segments can result in a final mesh structure that better reflects the original shape, as seen in Fig. 5 and 6.

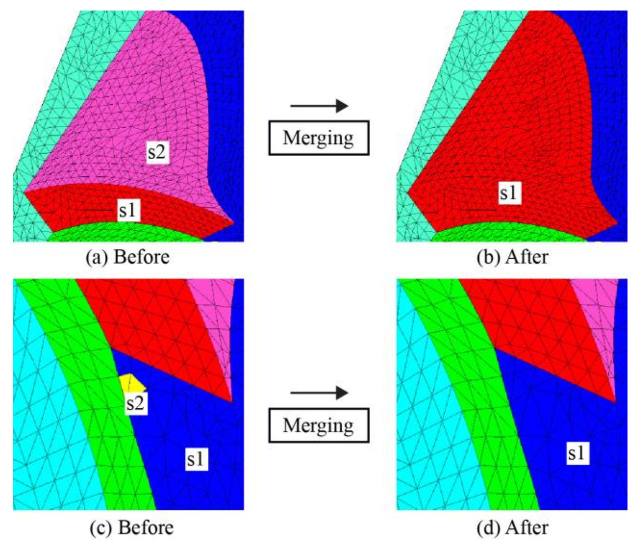

Fig. 5 The result of merging two segments *s1* and *s2* into one segment *s1*. The user manually selects two segments and executes a merging function to combine them into one segment.

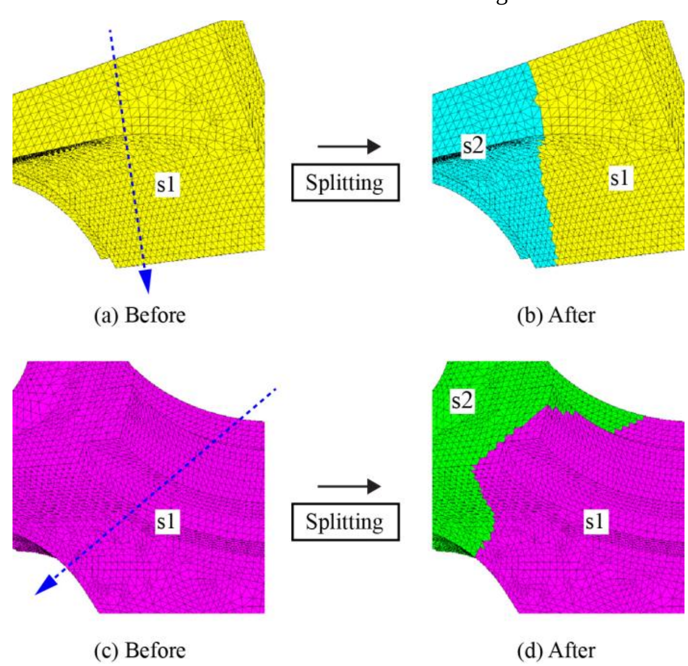

Fig. 6 The result of splitting one segment *s1* into two segments *s1* and *s2*. The user manually selects a segment and draws a line (denoted as the dot-line arrow) along the path to divide the segment.

C. Mesh Simplification per Segment

In this process, the segmented meshes are further simplified by the user manually entering different values for simplification (called a reduction ratio) to each segment to perform a mesh simplification. For example, a segment that appears planar requires a higher reduction ratio value for simplification while a curved segment requires less simplification and a correspondingly smaller reduction ratio.

Once the user selects a segment to be simplified and inputs an appropriate reduction ratio, and edge collapse method [9], [10] is utilized to perform the local mesh simplification. Edge collapse, as seen in Fig. 7, is a mesh simplification where vertex *v1* and *v2* are merged into *v'*. The calculation shifts *v1* and *v2* into a new position *v'* and adjusts the edges connected to *v1* and *v2* to match the edges of *v'*. Through this process, one or more faces are removed from the initial mesh. When considering the merging of segments, we avoided removing the edges along the boundary of segments. Fig.8 shows the result of this mesh simplification process per segment where the edge collapse of CGAL (Computational Geometry Algorithms Library) [11] was used for mesh simplification.

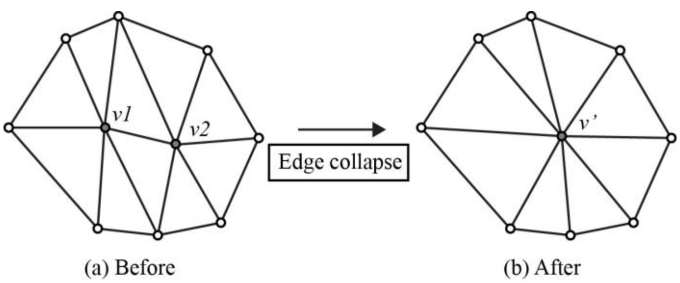

Fig. 7 Concept of edge collapse

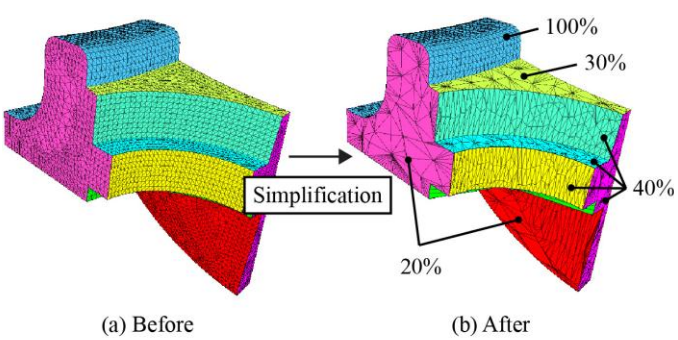

Fig. 8 The result of mesh simplification per segment. The % value labelled refers to the reduction ratio selected by the user for the corresponding segment

III. Experimental Results

Fig. 9 presents the results of the mesh simplification process where the same reduction ratio was applied to each segment. As is demonstrated, lower reduction ratios resulted in segments that were more simplified. Fig. 10 compares the results of mesh simplification between earlier methods and the proposed method with differences between the original and the simplified meshes being visualized by Hausdorff distance of MeshLab [12]. Mesh regions marked closer to the blue coloring represents a smaller difference, while larger differences between the original and simplified meshes tend toward the red. It is clear that the overall shape of the mesh was considerably changed by previous simplifications, but minimal change has resulted from the proposed method.

IV. Conclusions

This paper proposes a user interaction-based mesh simplification method to minimize the change in shape of the original mesh after simplification. This process first divides a mesh into several segments utilizing both automated and manual segmentation, and then further simplification is accomplished by assigning different degrees to each segment by the user. As demonstrated, utilizing previous methods to simplify the mesh automatically as a whole, the resulting mesh shape changed to considerably from the original, whereas the proposed method offered a significant improvement in the minimization of overall shape distortion. Of course, because the proposed method is a manual process based on user feedback, additional processing time might be required depending on the user's individual experience, when compared to previous automated methods, but this procedural time might be reduced by developing an efficient and user-friendly interface.

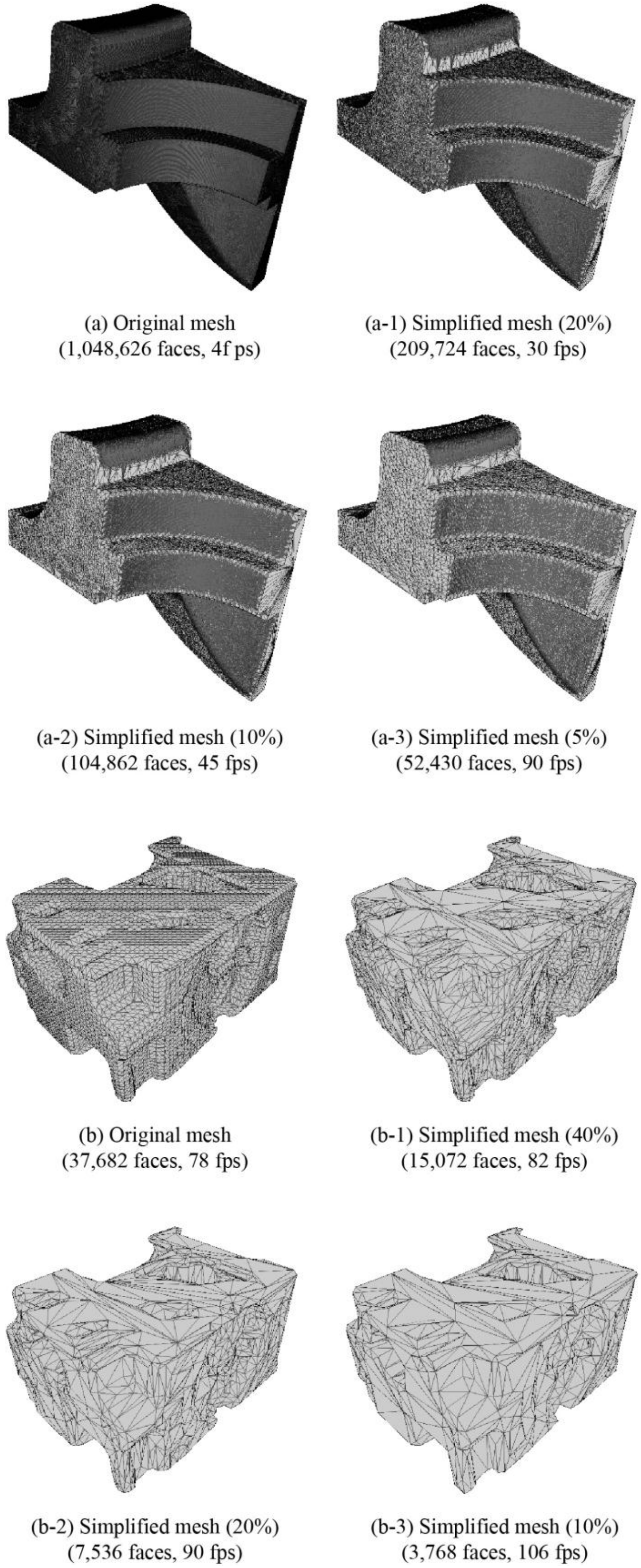

Fig. 9 The results of mesh simplification. The % value is the selected reduction ratio resulting in greater simplification with lower ratios.

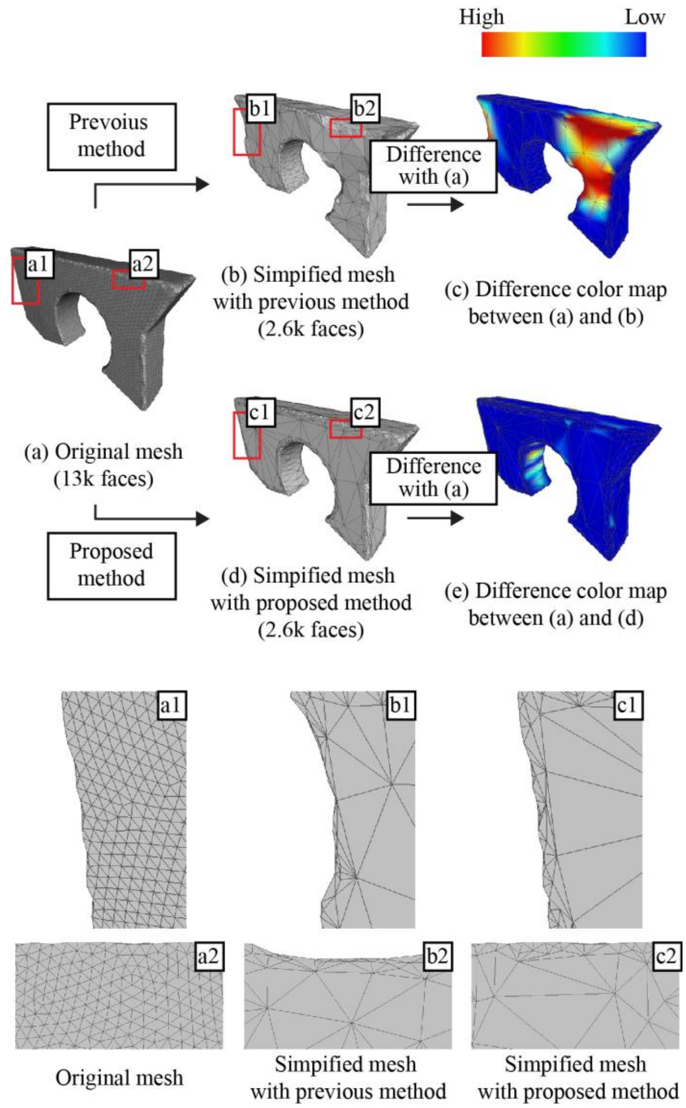

Fig. 10 A comparison of results from the mesh simplification process from the previous method [9] and the proposed method. The shape of the mesh resulting from the proposed simplification method has been changed significantly less as measured by the Hausdorff distance analysis.

ACKNOWLEDGMENT

This work was supported by the ICT R&D program of MSIP/IITP. [2014(10047039), Development of the realtime inspection SW and reverse engineering SW based on multiple 2D x-ray images and CT images]

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No.2011-0012214)

REFERENCES

[1] Varady, T., Martin, R. R., and Cox, J., "Reverse engineering of geometric models—an introduction", *Computer-Aided Design*, 29(4), 1997, pp. 255-268.
http://dx.doi.org/10.1016/S0010-4485(96)00054-1

[2] Garland, M., and Heckbert, P. S., "Surface simplification using quadric error metrics", *In Proceedings of the 24th annual conference on Computer*

graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co.., 1997, pp. 209-216. http://dx.doi.org/10.1145/258734.258849

[3] DeCoro, C., and Tatarchuk, N., "Real-time mesh simplification using the GPU", *In Proceedings of the 2007 symposium on Interactive 3D graphics and games*, 2007, pp. 161-166.

[4] Isenburg, M., Lindstrom, P., Gumhold, S., and Snoeyink, J., "Large mesh simplification using processing sequences", *In Proceedings of the 14th IEEE Visualization 2003*, 2003, pp. 61. http://dx.doi.org/10.1109/VISUAL.2003.1250408

[5] Dey, T. K., Giesen, J., and Hudson, J., "Decimating samples for mesh simplification", *In Proceedings of the 13th Canadian Conf. Comput. Geom*, 2001, pp. 85-88.

[6] Schroeder, W. J., Zarge, J. A., and Lorensen, W. E., "Decimation of triangle meshes", *In ACM Siggraph Computer Graphics*, Vol. 26, No. 2, 1992, pp. 65-70. http://dx.doi.org/10.1145/142920.134010

[7] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W., "Mesh optimization", *In Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993, pp. 19-26. http://dx.doi.org/10.1145/166117.166119

[8] Glassner, A. S., *graphics gems I*. Elsevier, 1993, ch. 4.

[9] Chen, B, "LOD-sprite technique for accelerated terrain rendering", *Visualization'99*, 1999, pp. 291-536. http://dx.doi.org/10.1109/VISUAL.1999.809900

[10] Wünsche, B., "A survey and evaluation of mesh reduction techniques", *In Proceedings of the IVCNZ 98*, 1998, pp. 393-398.

[11] CGAL (Computational Geometry Algorithms Library),s http://www.cgal.org

[12] MeshLab, http://meshlab.sourceforge.net