

Mesh Editing with an Intuitive User Interface

Jinsuk Yang, Kyoungsu Oh and Hyung-Il Choi

Department of Media, Soongsil University

Abstract: As 3D scanning equipment has been developed recently, reverse engineering, which scans and models an object, is also evolving. However, the mesh generated by reverse engineering can have some errors, such as unclear edges and rough surfaces, because of omitted data from the 3D scanning equipment or measurement errors. To remove these errors, various mesh editing methods have been introduced. In general, however, the quality of the mesh varies according to the editing ability of a user and it requires a long editing time, because there are usually specific areas to be edited in detail. In this paper, an editing method based on an intuitive user interface to overcome the limitations of existing methods is proposed. First, an input mesh is divided into several mesh segments according to their shapes. Then, a primitive type is designated for a mesh segment and the segment is edited by modifying parameters in accordance with the intended shape. The proposed method supports an intuitive interface using the primitive and accelerates editing conforming to the intended shape.

Keywords: mesh editing, reverse engineering, surface fitting.

1. Introduction

As 3D scanning equipment, such as laser scanners and Computed Tomography (CT) scanners, have been developed recently, reverse engineering, which scans and models an object, such as machine components, without human hands is also evolving [1]. The acquired data from reverse engineering is usually presented as a triangle mesh – a triangle generated by connecting each vertex composed of mesh, and the triangle mesh is a modelling method to describe complicated and sophisticated parts of an object easily and swiftly. However, the acquired data from 3D scanning equipment can have problems, such as data omission, measuring errors, noise occurrence, and low resolution; also, the massive size of the data can decelerate processing. Accordingly, the mesh generated by reverse engineering can have various errors, such as unclear edges and rough surfaces [2].

To remove errors from the mesh, mesh editing, such as vertex modification [3] and surface smoothing [4], is necessary. For vertex modification, the vertex can be manually selected and moved, or the mesh shape can be transformed by control points. The former method can accurately edit the user requires, but it also requires a long time. Though the latter method requires shorter time, it is difficult to edit accurately as the user intends. Surface smoothing recursively moves a selected vertex according to the difference with neighboring vertices. This method is fast and simple, but causes serious shape changes sometimes. As explained, existing mesh editing methods make the user edit specific areas in detail, so the quality of the mesh differs according to the varying editing ability of users, and it also requires a long editing time.

In this paper, a new mesh editing method based on an intuitive user interface that overcomes the limitations of existing methods is proposed. Users can edit the mesh as they intend with an intuitive interface using the primitive, which enables faster editing. Figure 1 shows the result of the proposed method, and its key points are as follows.

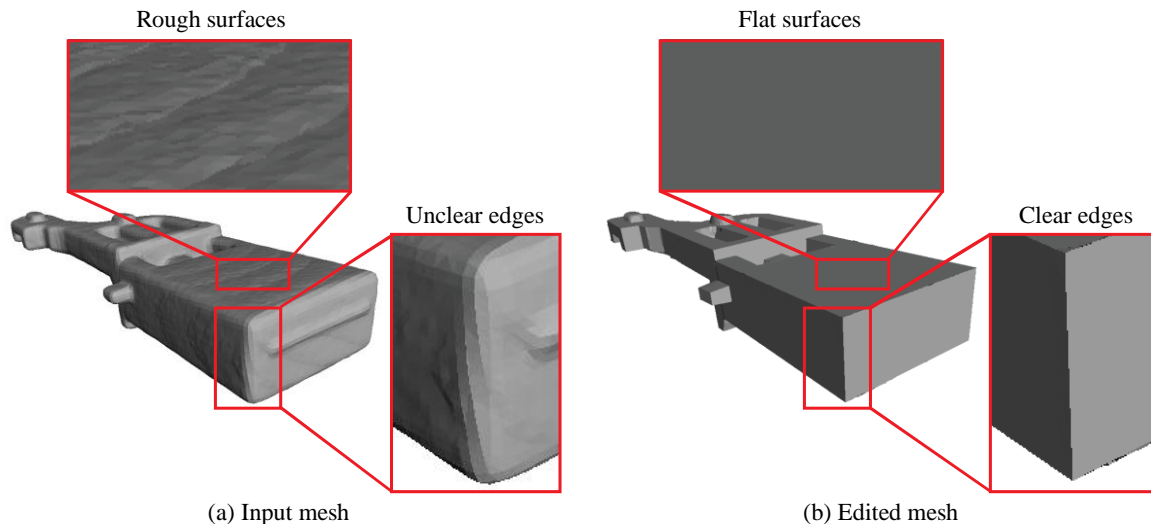


Fig. 1: Comparing input mesh with edited mesh: (a) The input mesh extracted from 3D-scanned data has errors, such as rough surfaces and unclear edges. (b) Errors in the input mesh have been removed by the proposed method.

- The user divides the input mesh into several mesh segments and edits them by mesh segment. Editing by mesh segment is faster and simpler than editing by points.
- The user sets a primitive type of the mesh segment and then edits it by modifying the parameters of the primitive. This enables intuitive editing by mesh segment.

This paper is composed as follows. The mesh segmentation method and mesh editing method using the primitive are explained in Chapter 2. Chapter 3 explains the results of the experiments, and the study is concluded in Chapter 4.

2. Mesh Editing with an Intuitive User Interface

In this chapter, the mesh editing method based on an intuitive user interface is explained. The proposed method is divided into a step for dividing the mesh (Section 2.1) and a step for editing the mesh by segment (Section 2.2). Figure 2 shows an overview of the process.

First, the user automatically divides the input mesh into several mesh segments. The mesh segments that do not conform to the intended shape can be manually merged or split by the user. Then, the mesh segment to be edited is selected and the primitive in the proper shape is also designated. After the quadric surface is estimated from the selected mesh segment and primitive type, the major parameters of the primitive are calculated. The user can edit the mesh segment to achieve the intended shape by modifying parameters.

2.1. Mesh Segmentation

To edit the mesh by segment, an input mesh is automatically or manually divided into mesh segments. The mesh segment means a fragment from a mesh that is divided into several parts, and it is marked with different colours, as Figure 3 shows, so that they can be distinguished from one another.

The automatic mesh segmentation is explained first. The user should select “Automatic Mesh Segmentation” on the menu after entering the value. In this step, the input value means the threshold to decide how finely the mesh is divided and also the difference of normal vector angles between two adjacent surfaces. For example, if the difference of normal vector angles between two adjacent surfaces is smaller than the input value, the two surfaces will be classified into the same segment. Otherwise, the mesh will be separated into other segments. This step is begun from a surface and recursively executed to many neighboring surfaces, like the flood-fill algorithm [5]. Figure 3 shows the result of automatic mesh segmentation. When a smaller input value is entered, the mesh is divided more finely.

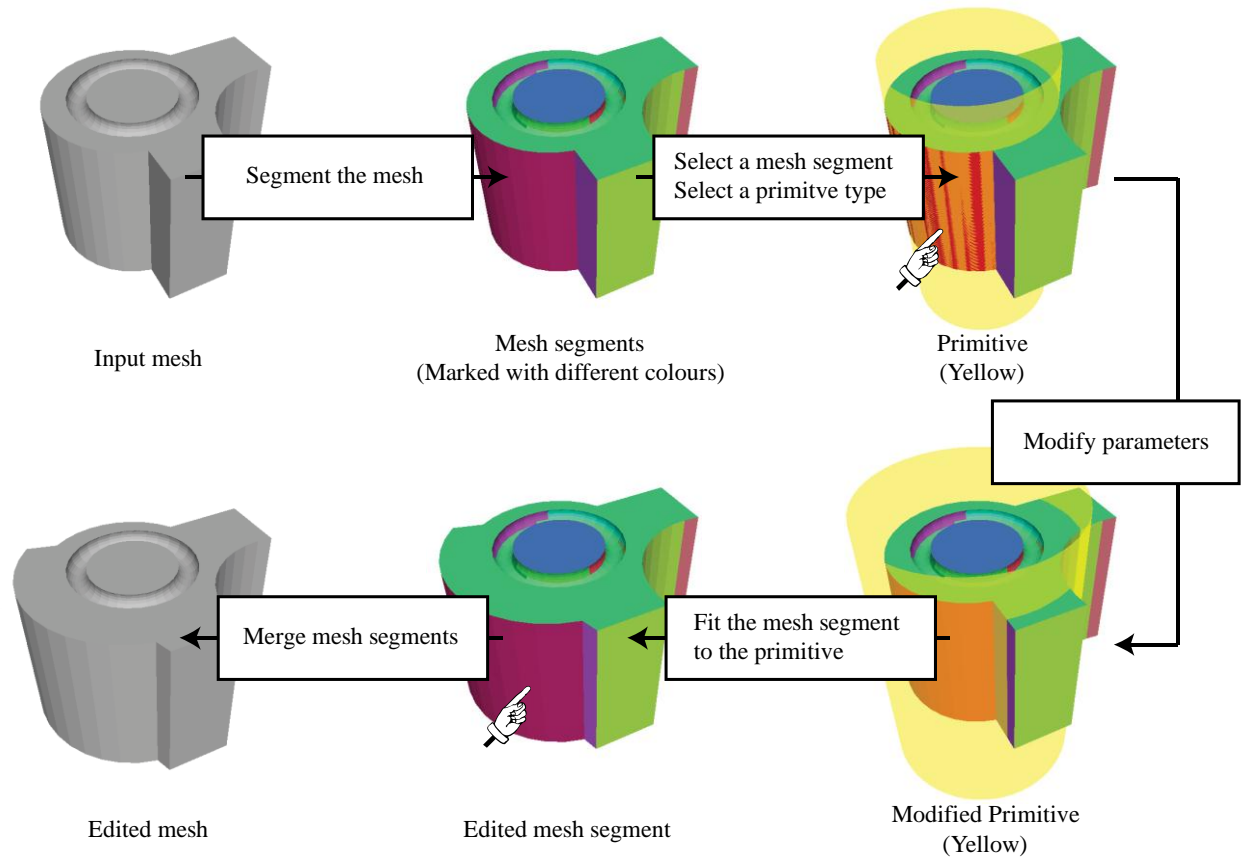


Fig. 2: Overview of the proposed method

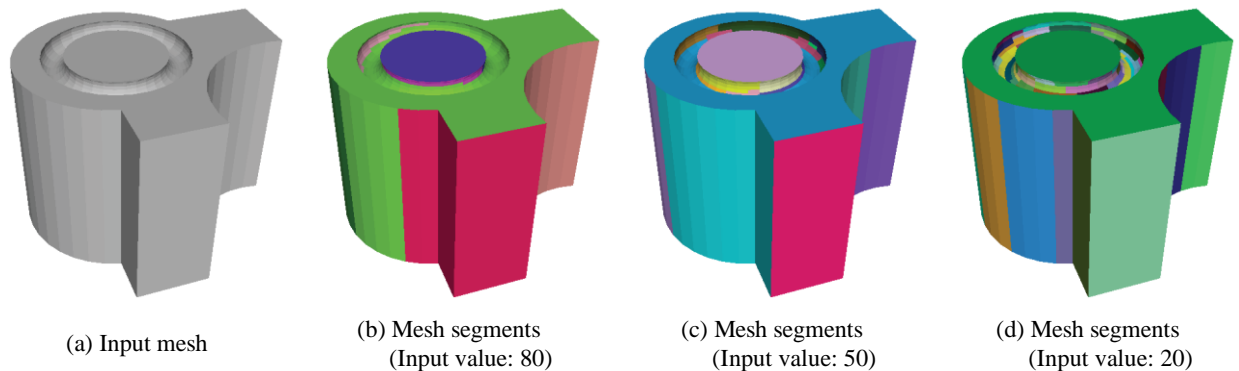


Fig. 3: Result of the automatic mesh segmentation in accordance with input value: The input value is the threshold value to determine the segmentation of the mesh and also means the difference of normal vector angles between two adjacent surfaces. When a smaller input value is entered, the mesh is divided more finely.

The mesh segment may be divided too finely by automatic mesh segmentation, or the shape may be reflected abnormally against the intentions of the user. This can be improved by manual mesh segmentation, as Figure 4 shows. Manual mesh segmentation is composed of the “mesh segments merge” and “split mesh segment” functions. The mesh segments merge function merges several segments into a mesh segment. When the user executes the mesh segments merge function on the menu after selecting several segments, the selected segments are unified as a mesh segment. On the other hand, the mesh segment split function separates a segment into several mesh segments. When the user draws a line on the intended part after executing the mesh segment split function on the menu, the segment is separated based on the line.

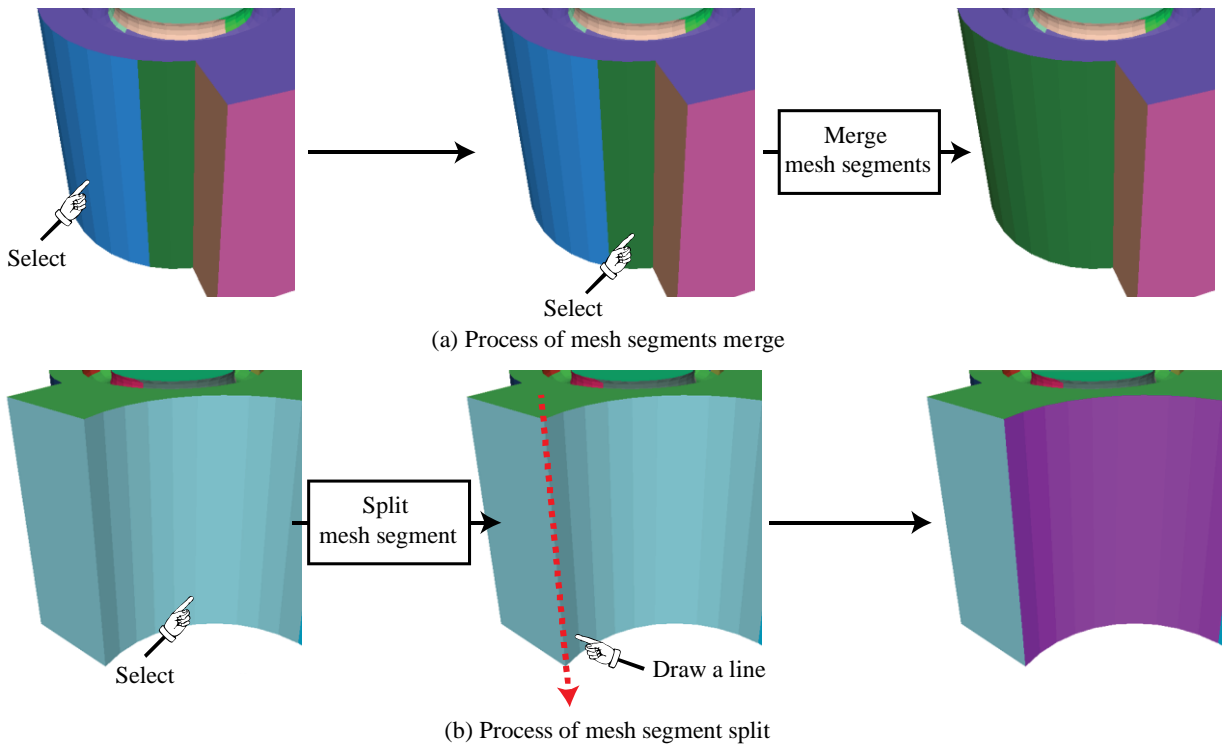


Fig. 4: Manual mesh segmentation process: (a) When the merge menu is executed after selecting mesh segments, the selected segments are unified as a segment. (b) When a line (red) is drawn on the intended part after selecting a segment and executing the split menu, the segment is divided by the line.

2.2. Mesh Segment Editing using Primitive

In this section, the mesh segment editing method using the primitive is explained. The process is executed by a mesh segment. As Figure 5 shows, the primitive is selected for the mesh segment, and then it can be edited quickly and intuitively by modifying parameters.

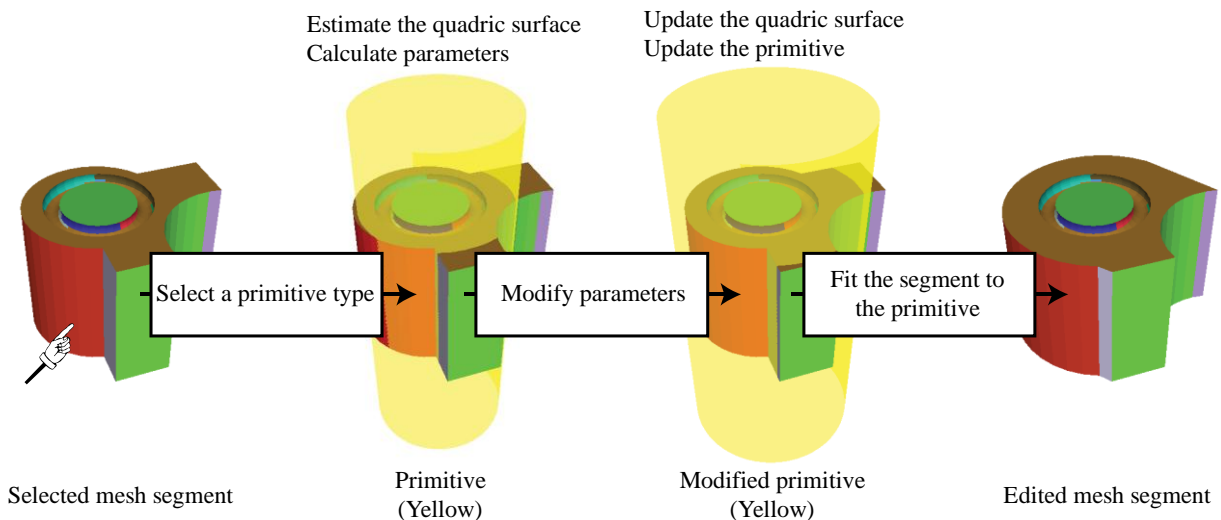


Fig. 5: Mesh editing process using primitive modification

First, the user selects the most similar primitive type (from among the plane, sphere, cylinder, and cone) with the shape of the mesh segment after selection of a mesh segment. For example, the plane will be selected if the mesh segment looks like a plane, and the cylinder will be selected if it looks like a cylinder. To estimate the quadric surface from the mesh segment and primitive type, the [6] is applied. Then, the major parameters of the

primitive are calculated using the estimated quadric surface. In this step, the kinds of parameters to be calculated are different from each primitive. For example, equation (1) explains the basic cylinder, and the major parameter is the radius of the cylinder.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (1)$$

If the selected primitive is the circular cylinder, equation (1) can be simplified to equation (2), and the radius of the cylinder is a^2 .

$$x^2 + y^2 = a^2 \quad (2)$$

The calculated radius of the cylinder can be shown to the user, and the user can modify the primitive by entering a desired radius. Finally, when the mesh segment is fitted to the modified primitive, the segment is transformed in accordance with the intended shape.

3. Experiment Results

To evaluate the quality of the edited mesh, the input mesh and edited mesh were compared by Computer-Aided Design (CAD) to measure the accuracy of the mesh. The input mesh was generated by the marching cubes method [7] after scanning the light metal machine components produced by die casting with 3D CT. The edited mesh resulted from the input mesh after editing it for an hour and a half, and the CAD model was produced by CAD software in advance (Figure 6).

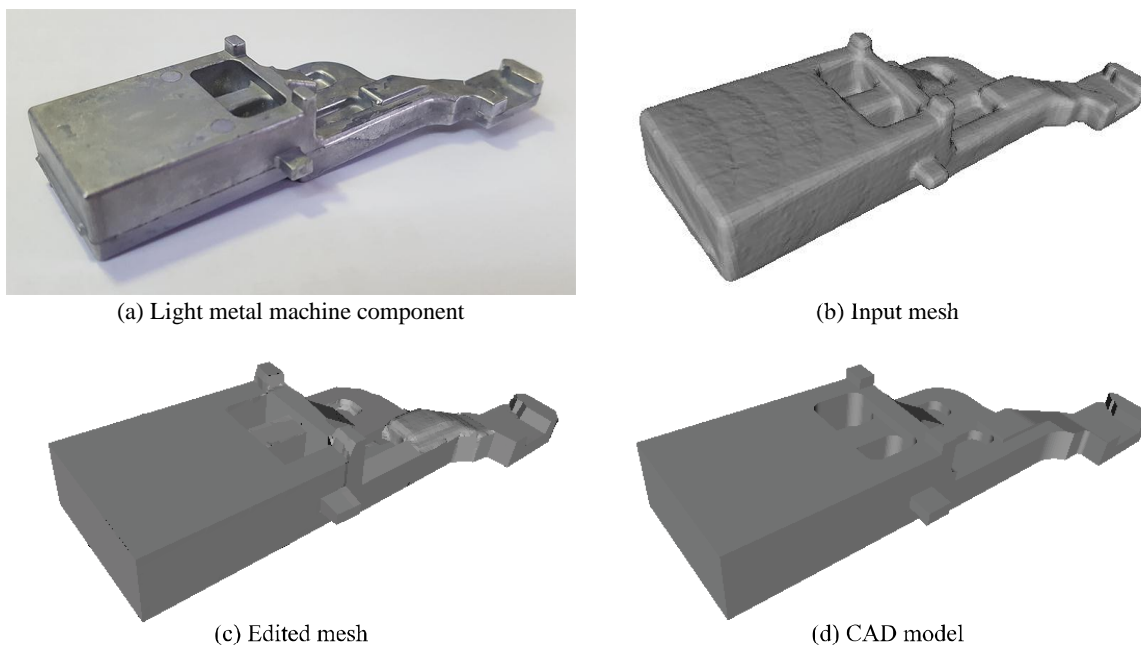


Fig. 6: Experiment data: (a) Light metal machine component produced by die casting¹ (90×25×12mm). (b) Generated mesh after scanning (a) with 3D CT. (c) Edited result of (b) with the proposed method. (d) Generated CAD model referring to the assembly drawing of (a).

The accuracy of the mesh can be measured as follows. A million samples are uniformly generated on the input mesh and edited mesh, and the shortest distance to the CAD model from each sample is calculated. Equation (3) is applied to calculate the accuracy of the mesh.

¹ Die casting is a metal casting process that is characterized by forcing molten metal under high pressure into a mold cavity.

$$Accuracy(\%) = \frac{N(Total_sample) - N(Error_sample)}{N(Total_sample)} \times 100 \quad (3)$$

The error sample has the shortest distance to the CAD model, which is more than the error permission, and the error permission level follows the Korean Industrial Standards [8].

Figure 7 shows the result of mesh accuracy. The error permission is set to 0.42mm, (a) is the accuracy of the input mesh, and (b) is the accuracy of the edited mesh. In the case of (a), the average distance was 0.43mm and the accuracy was 54.3%, so about half of the samples were different from the CAD model. On the other hand, (b) showed an average distance of 0.21mm and an accuracy of 81.4%, so it was improved by 27.1% compared with (a).

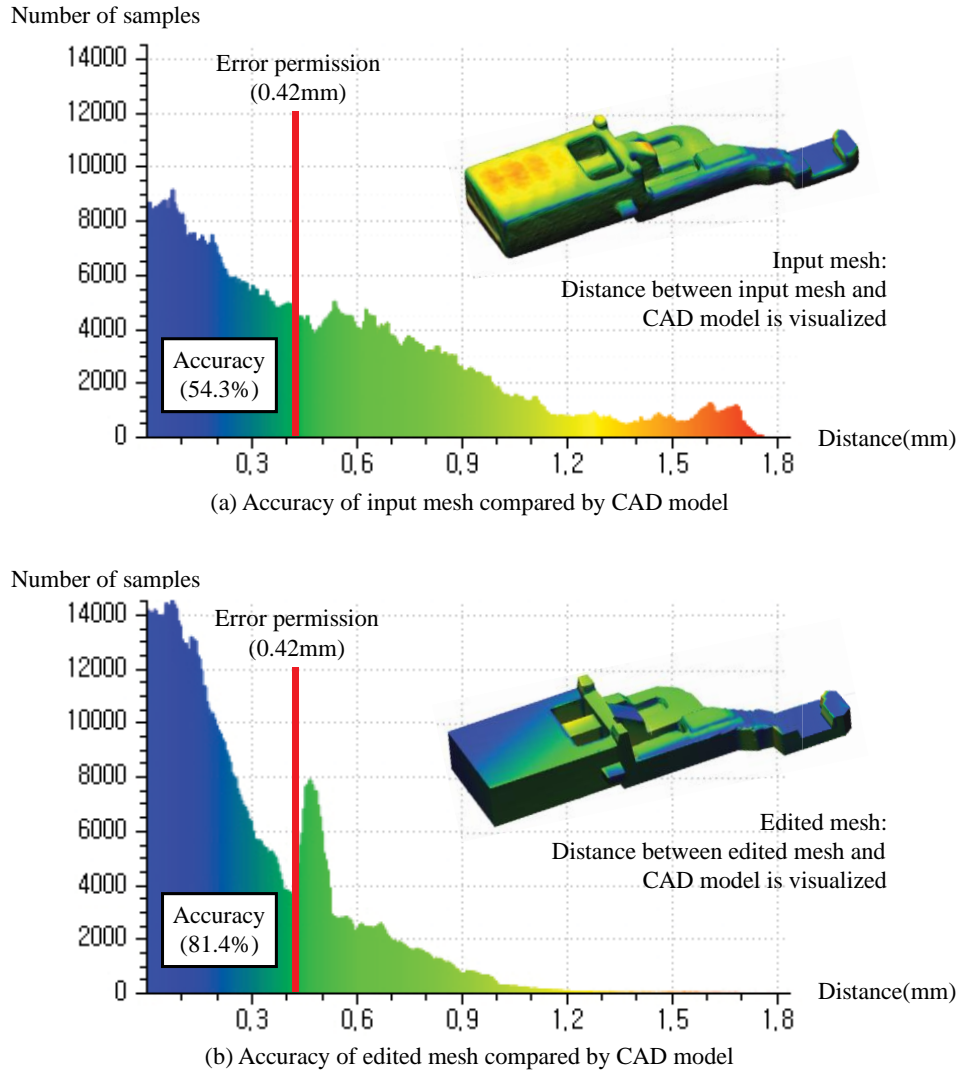


Fig. 7: Result of mesh accuracy: The edited mesh showed 27.1% higher accuracy than the input mesh, and most of the samples had a distance to the CAD model of hardly over 1.0mm.

4. Conclusion

In this paper, a new mesh editing method, which is based on an intuitive user interface and improves the limitations of existing methods, has been proposed. The user divides the input mesh into mesh segments using automatic mesh segmentation. The result of automatic mesh segmentation can be modified by manual mesh segmentation. Then, a primitive type for each mesh segment is designated, and the user edits the segment by modifying parameters.

The proposed method processes editing by mesh segment, which enables faster editing than editing by points, and it supports an intuitive editing method that modifies the parameters of the primitive. The experiment result has proved how accurately the input mesh can be edited compared with the CAD model. Although the proposed method lacks detailed editing functions, it has shown the potential to evolve into a more powerful and user-friendly mesh editor united with existing methods.

5. Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.10047039, Development of the realtime inspection SW and reverse engineering SW based on multiple 2D x-ray images and CT images)

6. References

- [1] Sokovic, M., and J. Kopac, "RE (reverse engineering) as necessary phase by rapid product development," *Materials Processing Technology*, 175.1, 2006, pp.398-403.
- [2] Pauly, M., Mitra, N. J., Giesen, J., Gross, M. H., and Guibas, L. J, "Example-based 3D scan completion," *Symposium on Geometry Processing*, No. EPFL-CONF-149337, 2005, pp. 23-32.
- [3] Hsu, W. M., Hughes, J. F., and Kaufman, H, "Direct manipulation of free-form deformations," *ACM Siggraph Computer Graphics*, ACM, 26.2, 1992, pp. 177-184.
- [4] Field, D. A, "Laplacian smoothing and Delaunay triangulations," *Communications in applied numerical methods*, 4.6, 1998, pp. 709-712.
- [5] Glassner, A. S, *graphics gems I*, Elsevier, 1993, ch. 4.
- [6] Andrews, J., and Séquin, C. H, "Type-constrained direct fitting of quadric surfaces," *Computer-Aided Design and Applications*, 11.1, 2014, pp. 107-119.
- [7] Lorensen, W. E., and Cline, H. E, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM siggraph computer graphics*, ACM, 21.4, 1987, pp. 163-169.
- [8] "Castings-System of dimensional tolerances and machining allowances," KS B 0250:2000, *Korean Industrial Standards*, 2005.