Journal of Mechanical Science and Technology 35 (7) 2021

Original Article
DOI 10.1007/s12206-021-0635-8

Keywords:
· 3D printer
· Delta volume generation
· Mesh offset
· Part maintenance
· Partially damaged part

Correspondence to:
Duhwan Mun
dhmun@korea.ac.kr

Citation:
Kim, Y., Kwon, K., Mun, D. (2021). Mesh-offset-based method to generate a delta volume to support the maintenance of partially damaged parts through 3D printing. Journal of Mechanical Science and Technology 35 (7) (2021) 3131~3143. http://doi.org/10.1007/s12206-021-0635-8

Received January 21st, 2021
Revised March 29th, 2021
Accepted April 5th, 2021

† Recommended by Editor
Hyung Wook Park

Mesh-offset-based method to generate a delta volume to support the maintenance of partially damaged parts through 3D printing

Youngki Kim[1], Kiyoun Kwon[2] and Duhwan Mun[3]

[1]Korea Institute of Machinery and Materials, 156 Gajeongbuk-ro, Yuseong-gu, Daejeon 34103, Korea, [2]School of Industrial Engineering, Kumoh National Institute of Technology, 61 Daehak-ro, Gumi-si, Kyeongsangbuk-do 39177, Korea, [3]School of Mechanical Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, Korea

Abstract Three-dimensional (3D) printing technology is an excellent tool for implementing multi item, small scale production or for manufacturing objects of complex shape, and has been utilized in many areas of daily life. One typical application is parts maintenance. For a partially damaged part to be repaired using a 3D printer, it is essential to generate a delta volume for the damaged area. A typical method of delta volume generation is to create a mesh using Delaunay triangulation or Poisson surface reconstruction from the point cloud of a laser scan of the damaged part and to perform a boolean subtraction operation with the mesh of the original part. However, when generating the delta volume, this method is prone to error due to noise, non-uniform sampling, and missing data in the point cloud. To address this problem, we propose a mesh offset based method capable of robust delta volume generation despite point cloud noise. This method consists of four steps: preprocessing, point cloud extraction, mesh extraction, and delta volume extraction. To experimentally validate the proposed method, a prototype system was developed and a numerical implementation was performed for a partially damaged ball valve.

1. Introduction

Three-dimensional (3D) printing technology, which has recently gained attention in the manufacturing industry, has greatly improved productivity in the product development process. 3D printers, unlike conventional machining processes, produce a 3D object by successively adding material layer by layer-a process that is interchangeably known as “rapid prototyping” or “additive manufacturing” (AM) in academic circles [1].

The most salient features of 3D printing technology are multi-item, small scale production and fabrication of objects of complex shape, which are difficult to implement with a conventional manufacturing method. Driven by this advantage over conventional manufacturing processes, 3D printing has been expanding its field of application across various industrial sectors, reaching far beyond the manufacturing industry [2]. Apart from its primary function of pre-production prototype manufacturing, 3D printing technology has evolved into a direct means of business involving individual design, manufacturing, and online marketing [3]. In addition to manufacturing applications, 3D printing technology also supports parts maintenance for repairing the damaged components of any part using a 3D printer [4]. In particular, maintenance of a partially damaged part by 3D printing is effective if the parts to be replaced are discontinued or expensive. For example, a partially damaged part in the high-pressure turbine cover of an F-15K fighter engine was maintained by 3D printing, thereby reducing costs by 93 % and improving procurement time by 78 % compared to purchasing new parts [5].

The first step in repairing a partially damaged part using a metal 3D printer is to generate the delta volume for the damaged area. The delta volume generation process usually comprises the following steps: point cloud generation for the partially damaged area using a laser scanner; mesh generation from the point cloud using Delaunay triangulation or Poisson surface reconstruction; delta volume generation via a boolean operation to determine the difference between the 3D computer aided design (CAD) model (mesh) of the original part and the mesh of the partially damaged one. However, this method is prone to error. When creating a mesh model from a point cloud using Delaunay triangulation, the reconstructed mesh model fails to accurately represent the target surface if there is noise or if the number of points in the cloud is insufficient. Poisson surface reconstruction requires the normal vector of each point to reconstruct a mesh model; if there are no normal vectors in the point cloud, normal estimation can still be conducted, but the calculated normal vectors will not be accurate. For example, when using the point cloud library (PCL), the normal is calculated using the nearest points. The normal estimation result varies depending on the number and distribution of the nearest points. The direction (+/-) of the normal is affected by the viewpoint. In addition, it is difficult to reconstruct a watertight mesh model in either method if points are insufficient in the point cloud. This error is due to noise, non-uniform sampling, and missing data in the point cloud [6].

To address this problem, we propose a mesh offset based method to generate the delta volume for the partially damaged area from the input data of the mesh model of an original part and the point cloud of a partially damaged part. The proposed method consists of preprocessing, extracting the point cloud of the damaged area, extracting the mesh segment corresponding to the damaged area, offsetting the mesh, and flattening the mesh boundary. The novelty of the proposed method lies in the algorithm that generates a volume by offsetting internal nodes of the mesh corresponding to the partially damaged area, to the points of the partially damaged area while keeping the mesh boundary at its original position. Contrary to the conventional methods in which the deviations in a point would result in mesh generation failure or the inaccurate generation of a mesh, the proposed method has the advantage of robustly generating a volume in mesh format. The features of the proposed method are verified through a delta volume generation experiment for the partially damaged part.

The rest of this paper is organized as follows: Sec. 2 analyzes previous studies on 3D printing technologies and mesh reconstruction using point cloud; Sec. 3 presents the mesh offset based method for generating the delta volume for the partially damaged area; Sec. 4 provides detailed explanations for each delta volume generation step; Sec. 5 discusses the experimental validation results of the proposed method; and Sec. 6 presents the conclusion.

2. Review of related studies

2.1 3D printing technology

The layer-based AM characteristic of 3D printing has advantages over conventional systems such as computerized numerical control (CNC) or machining center tools when it comes to fabricating complex shapes. Moreover, with continual advances in 3D printing technology, it offers an increasingly high level of precision [7, 8]. Fused deposition modeling (FDM), the most popular 3D printing method, manufactures models by extruding filaments, a material for AM of layers upon layers through a heated nozzle [9]. In FDM, the quality of a model depends on the hot end design and the characteristics of the material, because the product is produced by melting the material through hot end, which includes a nozzle and a heating device. Anitha et al. [10] defined layer thickness, road width, and speed deposition as variables that affect the results produced by FDM and determined that the layer thickness has the most effect through the Taguchi method. Another advantage of 3D printing is that the industry achieves cost reduction by using less material than CNC, which involves material removal in production [11].

As per the definitions set by the American Society for Testing and Materials (ASTM), there are currently seven categories of 3D printing technology: binder jetting, direct energy deposition (DED), material extrusion, material jetting, powder bed fusion (PBF), sheet lamination, and vat photopolymerization [12].

In binder jetting technology, a powder adhesive is selectively joined with powder particles to form a product [13]. This technique can use various materials such as plastics, ceramics, sand casting materials, and metals [14, 15]. The binder jetting method does not require a heat source such as a laser or electron beam for printing, and has the advantages of low cost and high manufacturing time compared to the PBF method [16]. However, this method also has a disadvantage in that it requires post processing to deal with the uneven surfaces and weakness of the product [17]. DED uses focused thermal energy to form an object by melting the material; it lends itself well to producing large parts due to its advantages of high deposition speed, high material efficiency, and easy repair, but has the limitations of being unsuitable for producing objects of complex shape and having poor surface treatment and low resolution [18].

2.2 Advantages of maintaining partially damaged parts by 3D printing

Kenney [19], who performed 3D printing based ship maintenance and assessed its cost reduction effects, reported that it achieved a 221.24 % increase in return on investment in comparison with conventional methods and reduced maintenance costs by USD 6.8 million. Antony and Goward [20] divided the lifecycle of an aircraft gas turbine blade into four stages and statistically analyzed the ratio of repair to replacement time at each stage to be 70 %, 93 %, 94 %, and 97 %, respectively. They argued that repair occupied the greatest proportion of maintenance time throughout the entire life cycle of a gas turbine blade and that defective parts could be repaired using 3D printing. Kim et al. [4] proposed a system that supported 3D printing based maintenance of parts, whereby a partially damaged part was replaced by a new part fabricated using a 3D printer after inspection. They also developed a database of 3D printable parts (parts library), a shape-based information retrieval system, and a shape error inspection technology that supported this system.

Related studies show that maintenance of parts using a 3D printer is effective. However, maintenance is not always possible for all damaged parts. If a part is severely damaged from cuts or huge cracks, it is difficult to repair the damaged part using a 3D printer. In this case, it is more effective to order a new part than to repair the part. The proposed method deals with repairable damages in operation, such as volume changes caused by wear and local damage to the surface.

2.3 Mesh generation based on point cloud data

A point cloud representing positions in $\mathbb{R}^3$, normal vectors, and colors is utilized in a wide variety of fields such as engineering, mesh generation, and computer graphics. In particular, reverse engineering using point cloud generated via 3D laser scanning of products or parts is a representative application in the field of mechanical engineering [21, 22].

Studies related to mesh reconstruction from point clouds such as radial basis functions (RBFs) [23, 24], Delaunay triangulation [25], and Poisson surface reconstruction [26] are still active research fields. Lee et al. [27] generated stereolithography (STL) files for 3D printing using the point cloud data obtained with laser scanning. After the initial point sampling and noise removal, they generated the mesh using Delaunay triangulation. However, this method has the problem of inconsistent results from Delaunay triangulation that depend on the values input by the user. And also, there is a limitation in that a large amount of computing power is used to generate a mesh model via Delaunay triangulation [28]. Zhou et al. [29] generated a mesh using Delaunay triangulation. This was done after generating a convex hull using point cloud data and a watertight mesh by removing low quality mesh points from the mesh previously generated and filling in the voids growing the neighboring mesh point. Kazhdan et al. [30] performed space segmentation of point cloud data into an octree structure and defined the vector field. They also performed Poisson surface reconstruction to generate the mesh by extracting the iso surface by solving Poisson's equation for the vector field. This method requires a lower computational time than Delaunay triangulation, which generates a mesh using a point cloud. However, it has a disadvantage in that the quality of the resulting mesh model varies according to the normal vector of the point cloud and the calculation time increases along with octree depth.

Piya et al. [31] proposed a method for generating a model for a partially damaged area based on a point cloud. After generating a defective region model using the point cloud data of the partially damaged area and an original area model based on the prominent cross sections (PCS) of the non-defective area point cloud, they generated a model for the partially damaged area using boolean operations. The drawback of this method was the necessity of manually post processing the PCS data when calculating the PCS for original model generation.

Tootooni et al. [32] obtained the point cloud by scanning a part manufactured by a 3D printer with a laser scanner, comparing the CAD model for that part with the point cloud, and computing the root mean square of the dimensional deviation. Thus, dimensional variation classification of the part generated by a 3D printer was performed via supervised machine learning. Khanzadeh et al. [33] measured the geometric deviation by comparing CAD models with the point cloud obtained by scanning a 3D printed part with a self-organizing map, an unsupervised machine learning algorithm.

The mesh offset based delta volume generation method proposed in the present study calculates the distances between points of the segmented cloud and triangles of the segmented mesh to create a delta volume for the damaged area, which requires less computing power than Delaunay triangulation or Poisson surface reconstruction. Furthermore, the proposed method is capable of robust delta volume generation despite point cloud noise. The noise can be categorized into two types: a set of points in a point cloud scattered away from the target area and deviations in points comprising a point cloud due to limits in scanner precision. The former can be removed by an outlier removal process. But the latter remains even after the preprocessing, which makes it difficult to generate a delta volume using conventional methods. The proposed method has the capability of robust generation of a delta volume even when there are deviations in points comprising a point cloud.

3. Delta volume generation method for a partially damaged area

3.1 3D printing-based maintenance process for partially damaged parts

As presented in Fig. 1, 3D printing based maintenance of partially damaged parts includes the following processing steps: 1) acquisition of the point cloud of a partially damaged part and retrieval of the 3D CAD model (mesh) of the original part; 2) generation of the delta volume for the damaged area and 3D printing using this volume for part maintenance; 3) inspection and application of the repaired part [34]. Shape error will occur between the surfaces of a partially damaged part and those of the generated delta volume due to limits in the precision of a 3D printer if the shape of the partially damaged area is complex and irregular. Therefore, surface smoothing processes such as machining, buffing, and blasting were conducted to reduce the shape error prior to the maintenance process. In addition, this study targets maintenance of the slightly damaged part.

First, when abrasion or cracking is observed on a part during operation, the point cloud of the damaged part is acquired by laser scanning. Then, the 3D CAD model (mesh) of the original part is retrieved from the part library based on part classification or shape comparison. Next, the shape difference between

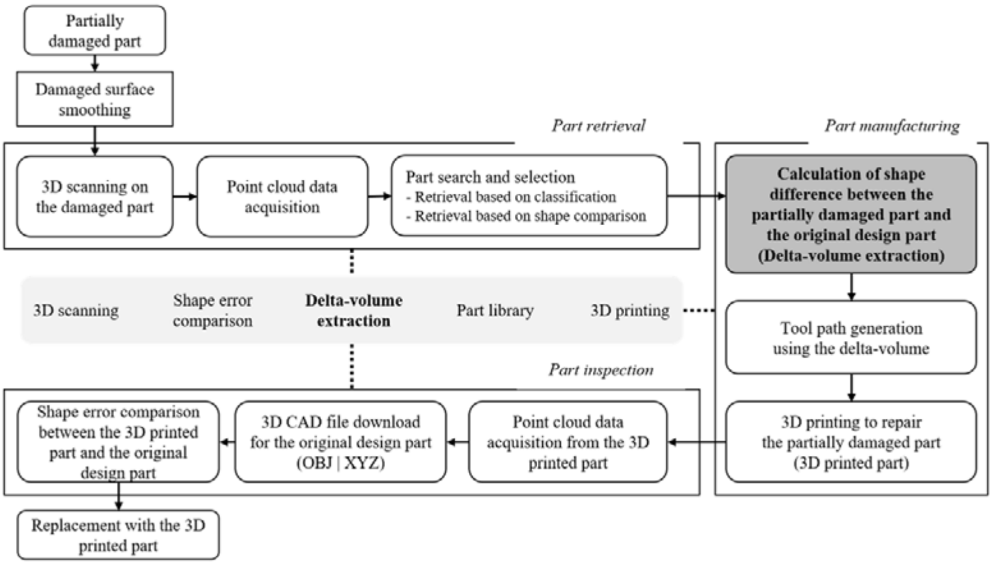

Partially damaged part
Damaged surface smoothing
Part retrieval
Part manufacturing
3D scanning on the damaged part
Point cloud data acquisition
Part search and selection
- Retrieval based on classification
- Retrieval based on shape comparison
Calculation of shape difference between the partially damaged part and the original design part (Delta-volume extraction)
3D scanning
Shape error comparison
Delta-volume extraction
Part library
3D printing
Tool path generation using the delta-volume
Part inspection
Shape error comparison between the 3D printed part and the original design part
3D CAD file download for the original design part (OBJ | XYZ)
Point cloud data acquisition from the 3D printed part
3D printing to repair the partially damaged part (3D printed part)
Replacement with the 3D printed part

Fig. 1. 3D printing based maintenance process for partially damaged parts.

the point cloud of the partially damaged part and the mesh of the original part is calculated for delta volume generation of the damaged area. Based on the delta volume thus generated, the tool path necessary for 3D printing is generated, followed by part maintenance. Maintenance is performed by filling in the damaged area using a 3D printer instead of fabricating the whole part. In the third step, once the repair is completed, shape error is inspected by comparing the difference in shape between the repaired (i.e., 3D printed) part and the original design. The repaired part is made available for use if the shape error falls within the tolerable range.

The mesh offset based method for delta volume generation proposed in this study is employed when calculating the shape difference between the point cloud of the partially damaged part and the mesh of the original part. A brief explanation of the process of generating the delta volume is given in Sec. 3.2.

3.2 Mesh offset based delta volume generation

A traditional method for delta volume generation involves mesh generation from the point cloud of a partially damaged part using Delaunay triangulation or Poisson surface reconstruction, followed by delta volume generation using the mesh of the original part with Boolean operations. The main limitation of such a traditional method is its dependence upon point cloud quality. The Delaunay triangulation has mesh related limitations such as an error prone target surface representation due to noise and missing data in the point cloud, and requires exponentially increasing time for mesh generation with respect to the number of points in the cloud [35]. Poisson surface reconstruction requires a separate step for computing normal vector of the point cloud, because it has a great effect upon the resultant mesh. Even after the removal of outliers from the point cloud of the partially damaged part, the mesh generation process is prone to error because of the point position error (noise) inherent in laser scanning. Fig. 2(a) illustrates an inaccurate target surface representation caused by noise, and Fig. 2(b) illustrates a shape error caused by an insufficient number of scan data points.

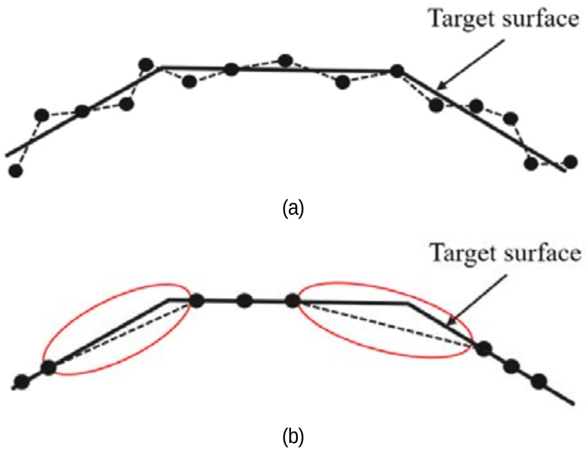


Fig. 2. Problems with mesh generation using a point cloud: (a) low shape quality due to noise in the point cloud; (b) shape error due to lack of a point cloud.

In contrast, the proposed mesh offset based delta volume generation method aims for robustness by using the laser scanned point cloud of the partially damaged part and the mesh of the original part in the maintenance process. Fig. 3 shows a block diagram of this process. The input values for this process are the point cloud of the partially damaged part (P_d) and the mesh of the original part (M_o), and the output value is the delta volume for the damaged area ($DV_{damaged}$). This process consists of preprocessing, point cloud extraction, mesh extraction, and delta volume extraction.

As shown in Fig. 4, P_d is pre-processed in the first step to ob-

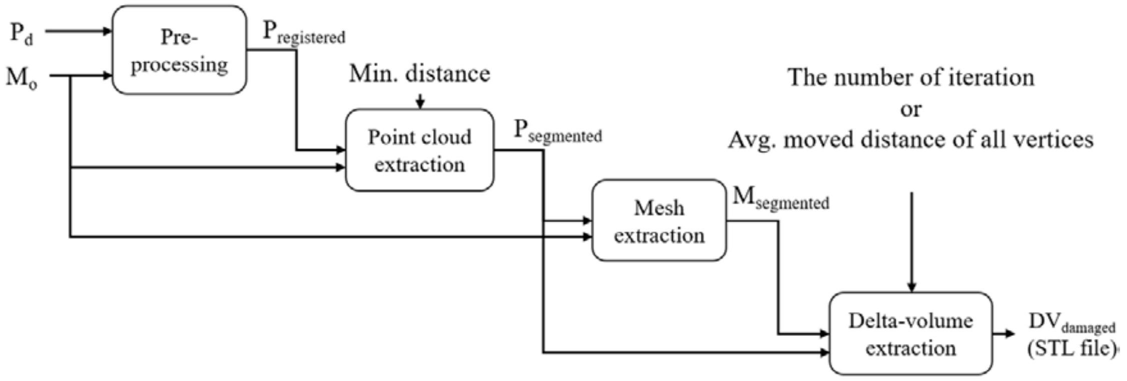


Fig. 3. Overall process for generating the delta volume of a partially damaged part.

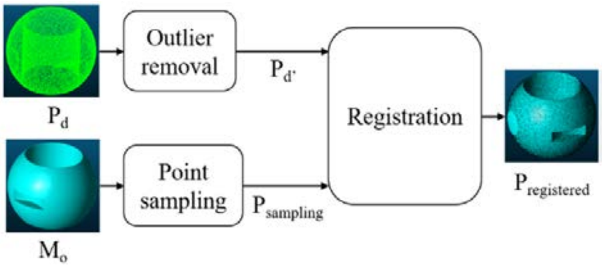


Fig. 4. First step: preprocessing.

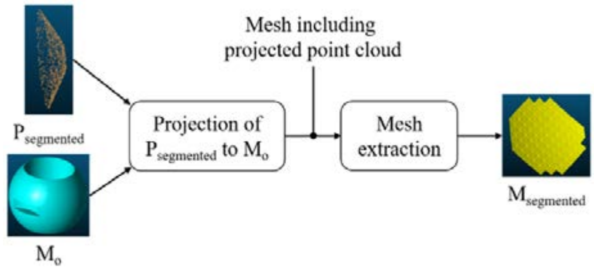

Mesh including projected point cloud
$P_{segmented}$
Projection of $P_{segmented}$ to M_o
Mesh extraction
$M_{segmented}$
M_o

Fig. 6. Third step: mesh extraction.

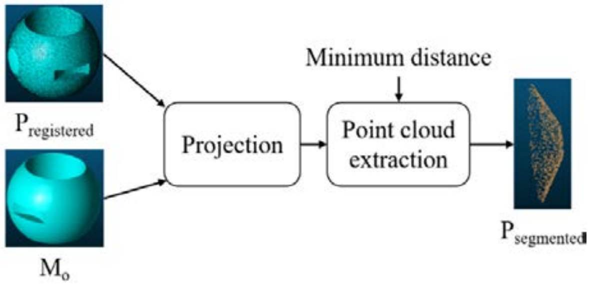


Fig. 5. Second step: point cloud extraction.

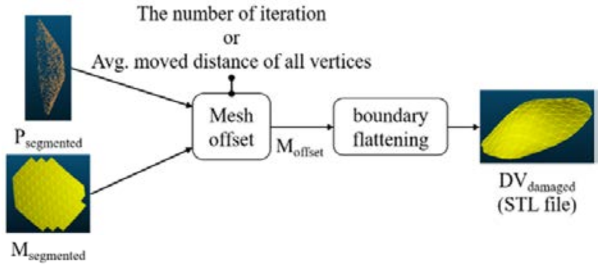


Fig. 7. Fourth step: delta volume generation.

tain a point cloud without outliers ($P_{d'}$), and to register to M_o. First, outliers are removed from P_d. Points are then sampled ($P_{sampling}$) from M_o for registration. Finally, $P_{d'}$ is registered to $P_{sampling}$, and the registered point cloud of the partially damaged part ($P_{registered}$) is output. Details on the preprocessing of an input point cloud are described in Sec. 4.1.

As shown in Fig. 5, the point cloud is extracted in the second step. First, the point cloud of the damaged area ($P_{segmented}$) is extracted using $P_{registered}$ and M_o; then, the distance values are calculated by projecting $P_{registered}$ onto M_o and the points with distance values below the user's input values are removed in preparation for the extraction of $P_{segmented}$. Details on the extraction of a point cloud of the damaged area are described in Sec. 4.2.

As shown in Fig. 6, the mesh segment ($M_{segmented}$) corresponding to the damaged area is extracted from M_o in the third step. First, $P_{registered}$ is projected onto M_o to identify the mesh segment that corresponds to the damaged area; then, the segmented mesh is extracted from the entire mesh. Details on the extraction of a mesh corresponding to the damaged area are described in Sec. 4.3.

As shown in Fig. 7, $DV_{damaged}$ is generated in the fourth step. First, each point of $P_{segmented}$ is projected onto $M_{segmented}$ to generate the distance map. Then, individual vertices of the mesh segment corresponding to the damaged area ($M_{segmented}$) are offset by the respective points in the cloud using the distance map, thus generating the mesh of the damaged area (M_{offset}). This process is iterated until the number of inputs or the average distance moved by all vertices falls below the input value in order to improve the accuracy of the mesh of the damaged area. Thus, M_{offset} and $M_{segmented}$ are merged, and the initial delta volume is generated. Finally, delta volume boundary flattening is performed to reduce the boundary error between the initial delta volume and the damaged area, and $DV_{damaged}$ is generated. After delta volume boundary flattening, $DV_{damaged}$ is stored as an STL file. Details on the mesh offset and boundary

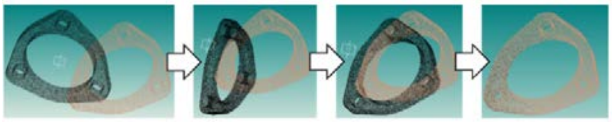


Fig. 8. Registration result using the iterative closest point algorithm.

flattening are described in Secs. 4.4 and 4.5.

The proposed mesh offset based delta volume generation method consists of the following technological elements: point cloud preprocessing, damaged area point cloud extraction, extraction of the mesh segment corresponding to the damaged area, mesh offset, and delta volume generation including mesh boundary flattening. These processes are explained in detail in Sec. 4.

4. Technologies involved in partial damage model generation for part maintenance

4.1 Point cloud preprocessing

Point cloud filtering is a technique for removing outliers to reduce error in the process of registering P_d to M_o. Outliers are unnecessary datapoints that occur far from the surface during scanning. As an outlier removal algorithm, we used the radius-based method available in the PCL [36]. This method requires that the user input the radius of sphere used for searching points and the number of points. Computation is performed on all points in the cloud, and the selected points having a smaller number of points than the input value within the input radius are removed.

Because P_d has a different reference coordinate system from the one used in the original design model, the coordinate systems should be synchronized in the registration process. A translation and rotation matrix is computed to minimize the distance error *d* between the points of $P_{d'}$ ($X = \{x_i \in R^3\}$) and the sampling points from M_o ($P = \{p_i \in R^3\}$). For the registration, we used the iterative closest point (ICP) algorithm. The distance error d was calculated by Eq. (1) as [37]

$$d(R,t) = \frac{1}{N_P}\sum_{i=1}^{N_P}\left\|x_i - R \cdot p_i - t\right\|^2 \quad (1)$$

where t is the translation matrix, R is the rotation matrix, and N_p is the number of sampled points. ICP computes the distance error *d* between two pointsets using singular vector decomposition. The calculated translation and rotation matrices are then applied to X and the distance error d is calculated. This process is iterated until d is smaller than the threshold. Fig. 8 illustrates the process of ICP based registration of two point sets.

4.2 Extraction of the point cloud of the damaged area

Extraction of the point cloud of the damaged area is the process by which $P_{segmented}$ is extracted from $P_{d'}$. In order to extract $P_{segmented}$, the distances between points of $P_{d'}$ and M_o are found by projecting $P_{d'}$ onto M_o. If the distance is smaller than the threshold, the point is removed by judging it not to be in the damaged area.

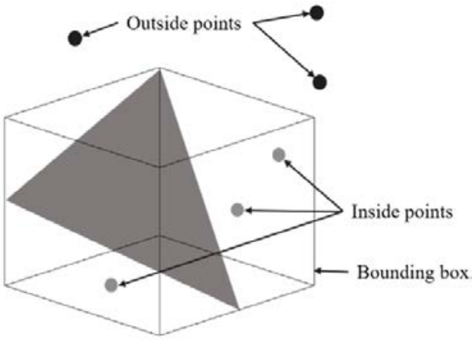


Fig. 9. Point separation using the bounding box.

The distance between a point belonging to $P_{d'}$ and M_o is calculated with an algorithm for calculating the minimum distance between a point and a mesh in 3D space [38]. The minimum distance Q between any given point P in a 3D space and a point in T within a triangle (V_1, V_2, V_3) constituting the mesh is expressed by Eqs. (2) and (3):

$$Q(s,t) = \left|T(s,t) - P\right|^2, \quad (2)$$

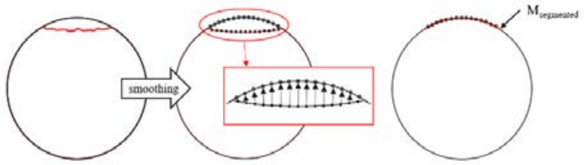
$$T(s,t) = B + sE_0 + tE_1 \ \text{for}\ (s,t) \in \{s \geq 0, t \geq 0, s+t \leq 1\} \quad (3)$$

where B is the vector from point P to vertex V_1 of the triangle, E_0 is that from V_1 to V_2, and E_1 is that from V_1 to V_3.

However, the computational time increases with the increase in $P_{d'}$ and the number of triangles constituting M_o. To solve this problem and reduce computational time, we compute the bounding box around each triangle constituting the mesh, as shown in Fig. 9, and calculate only the distance between points within the bounding box. Moreover, to expand the bounding box enclosing the mesh, the expansion coefficient α is added to each dimension of the box as follows: width = width * (1+α), length = length * (1+α), and height = height * (1+α). The expanded bounding box contains all significant points belonging to the damaged area.

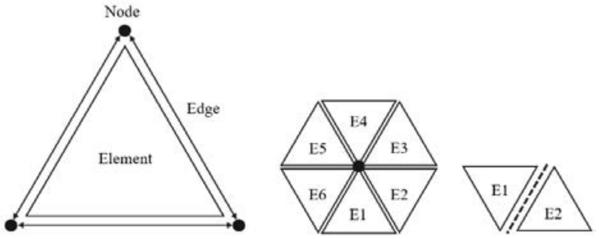
4.3 Extraction of the mesh corresponding to the damaged area

Extraction of the mesh corresponding to the damaged area is a process by which the non-damaged area, i.e., the mesh except for $M_{segmented}$, is removed from M_o. First, the $P_{segmented}$ obtained by the previous process is projected onto M_o, as shown in Fig. 10(a). The direction and position of projection of $P_{segmented}$ are the minimum distance between $P_{segmented}$ and M_o obtained by calculation, and information about the mesh containing the projected point is stored. Once the projection of $P_{segmented}$ is completed, we observe whether each mesh contain the projected point. If there are no projected points in the mesh,



(a) (b)

Fig. 10. Extraction of the mesh corresponding to partially damaged area: (a) $P_{segmented}$ projection onto M_o; (b) $M_{segmented}$ including projected $P_{segmented}$.



(a) (b) (c)

Fig. 11. Mesh data structure: (a) element; (b) node with adjacent elements; (c) edge with adjacent elements.

the mesh is removed from M_o, thus yielding $M_{segmented}$, which is a set of meshes containing projected points, as shown in Fig. 10(b).

4.4 Mesh offset

M_{offset} is generated by offsetting the $M_{segmented}$ to $P_{segmented}$. Before offsetting $M_{segmented}$, a data structure is adopted for the mesh. A mesh contains coordinate information for the vertices, vertex normals, and indices of triangles; this is sufficient for visualization, but insufficient for generating, updating, and deleting the mesh. In this study, we adopted the data structure proposed in Ref. [39], which defined a single triangular face constituting the mesh as an element, and edges and vertices of the triangular face as element edges and nodes, as shown in Fig. 11(a). Each element has three edges and three nodes, which have information about adjacent elements. Based on this data structure, when a specific element, element edge, or node is selected, information about the adjacent element, element edge, and node can be efficiently obtained.

To offset $M_{segmented}$, $P_{segmented}$ is projected to $M_{segmented}$, as shown in Fig. 12(a), and each element stores the position of the projected point and projected distance. When a distance map is generated for all elements of $M_{segmented}$, the moving distance of each node is calculated. If a node is a boundary node of $M_{segmented}$, it is excluded from the calculation. The pseudocode for calculating the distance moved by each node is as follows.

The distance moved by each node is shown by Eq. (4):

$$d_{offset} = \frac{1}{n * m} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(d_j \right) . \tag{4}$$

Algorithm 1. Calculating the distance moved by each node.

```
Input: All nodes in $M_{segmented}$
Output: Distance
for i = 0 to # of nodes in $M_{segmented}$
    if node i is not boundary node then
        for j = 0 to # of adjacent elements
            total points += projected points onto j-th element
            for k = 0 to # of projected points onto j-th element
                distance += projected distance of k-th point
            end for
        end for
        distance /= total points
    end if
end for
```

Here, n represents the number of adjacent elements of a node and m represents the number of projected points on each adjacent element. For example, a single node has six adjacent elements in Fig. 11(b). Adjacent elements store the number of projected points, projected positions, and projected distances from $P_{segmented}$. The offset distance of each node is calculated by averaging the projected distances stored in adjacent elements. At this time, boundary nodes are determined to keep the boundary of $M_{segmented}$. If there is one adjacent element in an element edge, this element edge becomes a boundary edge and two nodes belonging to the element edge become boundary nodes. On the other hand, if an element edge has two adjacent elements, two nodes of the element edge are identified as internal nodes. A delta volume is generated in the way of keeping boundary nodes and offsetting internal nodes. This ensures keeping the boundary of $M_{segmented}$ in its original position.

After calculating the offset distance of all nodes included in $M_{segmented}$, M_{offset} is generated by moving nodes as shown in Fig. 12(b).

To reduce the error between the damaged area and M_{offset}, the mesh offset process is iterated until the number of iterations (user inputs) or d_{offset} is smaller than the maximum tolerable margin of error. In the second iteration of the mesh offset, $P_{segmented}$ is projected onto initial M_{offset}, as shown in Fig. 13(a), and the distance map is computed again. M_{offset} is updated based on the thus computed distance map, as shown in Fig. 13(b), by repeating the mesh offset process. After finishing the mesh offset process, $DV_{damaged}$ can be generated by merging boundary nodes at the same position in M_{offset} and $M_{segmented}$, because these meshes have the same boundary nodes there.

4.5 Mesh boundary flattening

The $DV_{damaged}$ generated in the mesh offset process has a

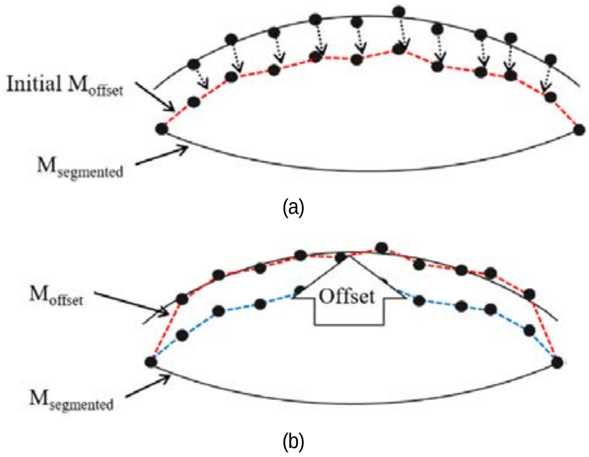


Fig. 12. Mesh offset process using the distance map between $P_{segmented}$ and $M_{segmented}$: (a) projection of $P_{segmented}$ onto $M_{segmented}$ to calculate distance; (b) mesh offset using the average distance.

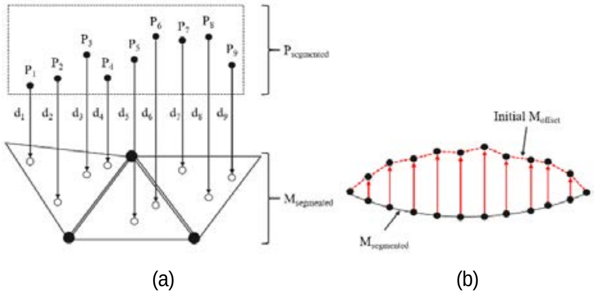

P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9
$P_{segmented}$
d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9
$M_{segmented}$
Initial M_{offset}
$M_{segmented}$
(a)
(b)

Fig. 13. Iterated update of M_{offset}: (a) projection of $P_{segmented}$ onto offset mesh; (b) updated M_{offset}.

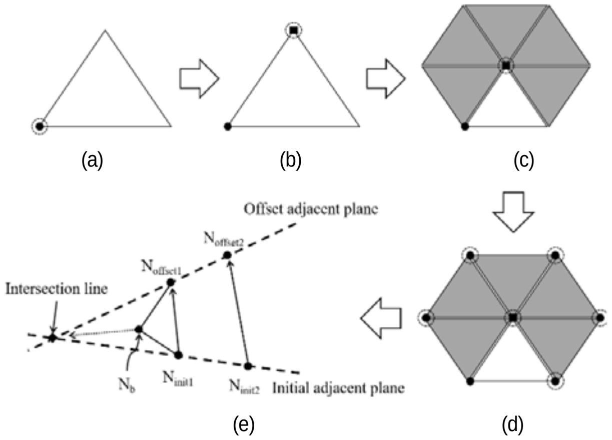


Fig. 14. Flattening the boundary of a mesh: (a) search for a boundary node (N_b); (b) search for a non-boundary node (N_{nb}); (c) search for adjacent elements; (d) search for nodes in adjacent elements, except the selected boundary node; (e) make adjacent planes using nodes and move the boundary.

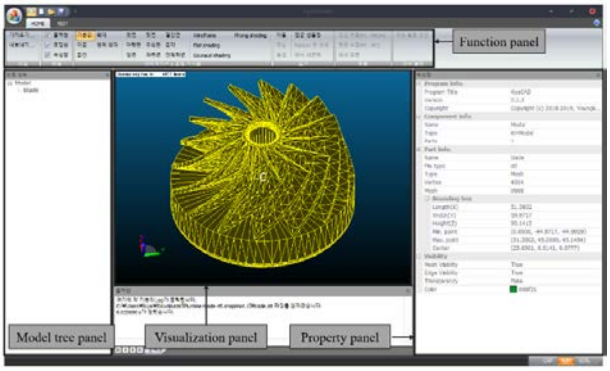


Fig. 15. Mesh offset based delta volume extraction system.

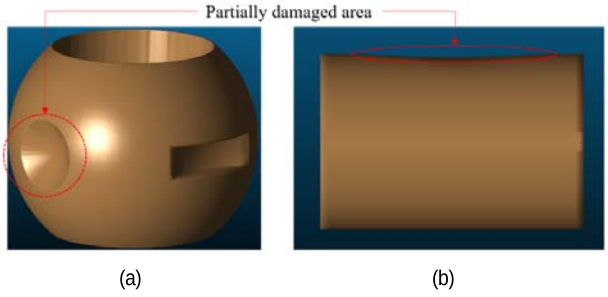


Fig. 16. Test case: (a) ball valve model; (b) cylinder model.

zigzag shape, and the boundary of the delta volume differs greatly from the real boundary of the damaged area. Therefore, mesh boundary flattening is conducted to reduce the error between the boundary of $DV_{damaged}$ and the real boundary of the damaged area. As shown in Fig. 14, such flattening comprises the following process steps:

1) Search for a boundary node (N_b);

2) Search for a non-boundary node (N_{nb}) in the element containing the boundary node (N_b);

3) Search for elements adjacent to the non-boundary node (N_{nb});

4) Search for nodes other than the non-boundary node (N_{nb}) selected from the adjacent elements;

5) Generate an initial adjacent plane using the initial node positions and an offset adjacent plane using the post offset positions, and then project the boundary node (N_b) along the intersection line of the two planes.

5. Implementation and experiment

We developed a prototype system (see Fig. 15) to validate the damaged area delta volume generation process. The proposed prototype system consists of a point cloud preprocessing module, point cloud and mesh extraction modules, and a delta volume extraction module.

The prototype system was developed with C++ and a Microsoft Foundation Class (MFC) was used to implement the graphic user interface. PCL and the visualization toolkit (VTK) [40] are used to process point cloud and mesh data, respectively. An ACIS modeling kernel [41] was used for 3D CAD data processing, and HOOPS [42] was used for 3D visualization of the 3D CAD, point cloud, and mesh data.

For the experiment to generate the delta volume of the partially damaged area, two test cases, a ball valve and a cylinder model, were used, as shown in Fig. 16. The entire experimen-

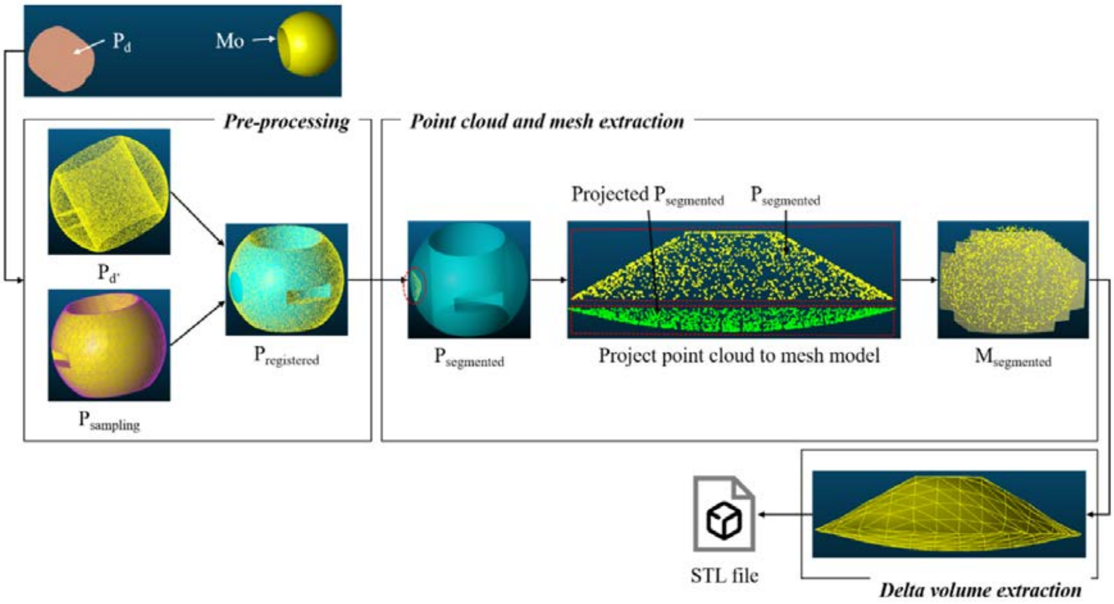


Fig. 17. Experiment on delta volume generation for the partially damaged ball valve model.

tal procedure and the result of using a ball valve model (Fig. 16(a)) are presented in Fig. 17.

The point cloud of the partially damaged ball valve part used in the experiment is composed of 100000 points, and the mesh of the original part is composed of 101220 elements and 303660 nodes. First, outliers were removed from the point cloud of the partially damaged ball valve model. Then, the initial point cloud was prepared for registration to the mesh of the original part using the preprocessing module. Using the radius based outlier removal method, 3027 outliers were removed under the configuration of a 1 mm search radius. From the mesh of the original part, 100000 points were sampled and registered to synchronize the coordinate systems of the point cloud of the partially damaged part with the mesh of the original part.

During the process of point cloud extraction, a segmented point cloud for the damaged area was generated using the registered point cloud of the partially damaged part and the mesh of the original part. If the threshold value is too high to remove points outside of the partially damaged area, the segmented point cloud of this area has not been adequately extracted. This adversely affects the shape of the delta volume. To address this, the threshold value was set to 1 mm and the bounding box expansion coefficient was applied. In consequence, a segmented point cloud of 1613 points was extracted for the partially damaged area.

In the process of extracting the mesh corresponding to the damaged area, the point cloud of the damaged area extracted was projected onto the mesh of the original part (as illustrated in Fig. 18(a)) to acquire the mesh corresponding to the damaged area. Fig. 18(b) represents the results of the extraction of the mesh reflecting only the projected points of the damaged area.

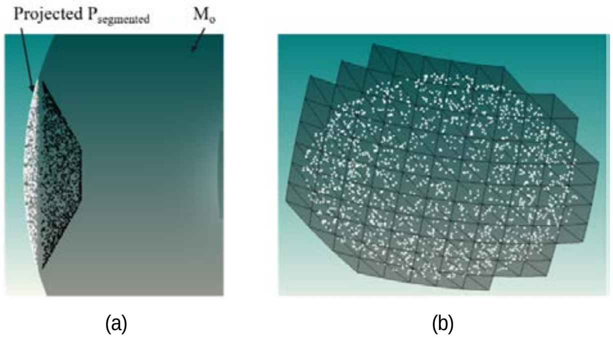


Fig. 18. The result of extracting the mesh corresponding to $P_{segment}$.

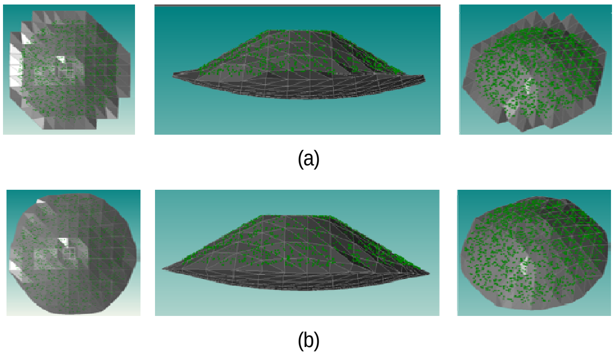


Fig. 19. The result of delta volume generation for a ball valve model: (a) without boundary flattening; (b) with boundary flattening.

The delta volume of the damaged area is generated using its segmented point cloud and the corresponding mesh segment. This volume is generated concurrently with a mesh offset and mesh boundary flattening, and the user can store the result as an STL file for 3D printing. Fig. 19 shows delta volume extractions with and without mesh boundary flattening.

Table 1. Comparison of the proposed method with other existing methods from the viewpoint of delta volume generation for the ball valve model.

	Delaunay triangulation (MatLab)	Poisson surface reconstruction (PCL)	This research (in-house)
Input data	Point cloud (points: 1559)	Point cloud (points: 1559)	Point cloud + mesh (points: 1559 / triangles: 472)
Necessity of vertex normal	X	O	X
Robustness against point cloud noise	X	O	O
Calculated volume [mm^3]	4315.6	4314.49	4484.05
Error rate [%]	3.9	3.93	0.15

Ground truth volume: 4491 mm^3

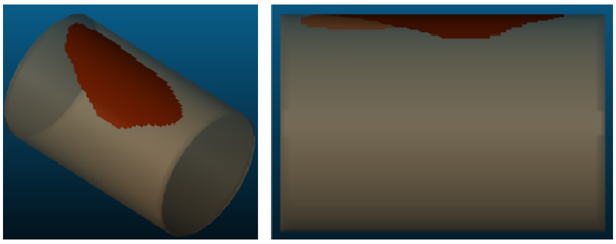


Fig. 20. The result of delta volume generation for the cylindrical model: (a) ISO view; (b) side view.

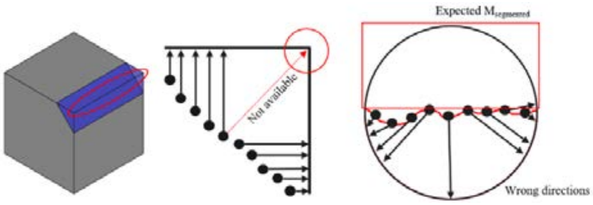


Fig. 21. The limitation shapes: (a) partially damaged area containing a sharp edge; (b) part with large damaged area.

We validated the efficacy of the proposed mesh offset based delta volume extraction system for robust generation of the delta volume for a point cloud with noise by performing a module based experiment using a prototype system and a partially damaged ball valve specimen. Table 1 compares the results of the proposed method to those of conventional methods for reconstructing mesh models from a point cloud. For comparison, MatLab was used to calculate the volume of the Delaunay triangulation after generating a tetra mesh, and Poisson surface reconstruction was calculated by the PCL library. The ground truth volume was calculated from the original 3D CAD model. Compared to Delaunay triangulation and Poisson surface reconstruction, the advantages of the proposed method include high robustness against noise in the point cloud and no need for normal estimation of the point cloud. As shown in Table 1, the delta volume of the partially damaged area reconstructed by the proposed method has an error rate of 0.15 %, which is better than that of conventional methods.

The results of the experiment for the cylindrical model (Fig. 16(b)) are shown in Fig. 20. This model is a cloud of 267496 points. The time required for delta volume construction in the cylindrical model on a computer equipped with an i9-9900k CPU was 0.143 seconds. When we reduced the number of points for the cylindrical model to 133748, the reconstruction time was 0.079 seconds. As we can see from this result, the reconstruction time is proportional to the number of points in the proposed method.

The proposed method has a shape limitation in that, if there are sharp edges in the extracted segmented mesh that is mapped to the partially damaged area (as shown in Fig. 21(a)), it is difficult to generate a delta volume for this area. This is because the points comprising a segmented point cloud are difficult to uniformly projected to triangles near the edges. In addition, this study targets the maintenance of slightly damaged parts; the points are projected in the wrong direction when the damaged area is relatively large compared to the size of a part, as shown in Fig. 21(b).

6. Conclusion

3D printing technology can be employed not only to produce parts, but also to maintain them. Damaged area delta volume generation is essential for a metal 3D printer to be used for the maintenance and repair of a partially damaged part. However, the process of generating the mesh from the point cloud of such a part is prone to error due to the noise, non-uniform sampling, and missing data in the point cloud.

To resolve this problem, we proposed a mesh offset based delta volume generation method. This method aims at robust delta volume generation for a partially damaged area using its point cloud and the mesh of the original part in the maintenance and repair process. The main technological processes employed in the proposed method are point cloud preprocessing, damaged area point cloud extraction, extraction of the mesh segment corresponding to the damaged area, mesh offset, and delta volume generation, including mesh boundary flattening. We developed a prototype system and processing modules, constructed a scenario using a partially damaged ball valve specimen, and performed an experiment to validate the efficacy of our proposed method. The experimental results confirmed that our method was capable of robust delta volume generation for a point cloud with noise.

The mesh model generated by the mesh offset based method has an error in final shape due to two approximation processes. The error in the first approximation occurs when converting a boundary representation (B-rep) 3D CAD model into a mesh model. An error in the second approximation occurs when offsetting the mesh repeatedly. The shape error due to conversion from the B-rep model to the mesh model can be reduced by generating a finer mesh. The error in the offset process can be minimized by increasing the number of iterations.

In the future, we plan to upgrade the main technological processes comprising the proposed method. Since each node of the segmented mesh is offset by averaging distances between the node and the points in a segmented cloud that are projected to triangles adjacent to the node, there is an error in the offset result even after several iterations. To solve this problem, the offset algorithm needs to be improved so that a delta volume with greater accuracy can be generated by optimizing the offset distance. And improvements in algorithms are needed to reduce runtime. To accomplish this, we will apply spatial partitioning techniques such as binary space partitioning and octree, as well as parallel-computing techniques such as OpenMP and CUDA.

Acknowledgments

This research was supported by the Industrial Core Technology Development Program (Project ID: 20009324) funded by the Korea government (MOTIE), by the AI-based Gasoil Plant O&M Core Technology Development Program (Project ID: 21ATOG-C161932-01) funded by the Korea government (MOLIT), and by the Basic Science Research Program (Project ID: NRF-2019R1F1A1053542) through the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT).

Nomenclature

P_d	: Point cloud of the partially damaged part
M_o	: Original mesh model
$DV_{damaged}$	: Delta volume of the partially damaged area
$P_{d'}$	: Point cloud with outliers removed
$P_{sampling}$	: Sampled points from original mesh
$P_{registered}$	: Registered point cloud
$P_{segmented}$	: Point cloud of the partially damaged area
$M_{segmented}$	: Mesh of the partially damaged area
M_{offset}	: Offset mesh

References

[1] J. H. Shim, W. S. Yun and T. J. Ko, Successful examples of 3D printing technology-based start-up enterprises, *Journal of the Korean Society of Manufacturing Process Engineers*, 15 (2) (2016) 104-110.

[2] S. Ford and M. Despeisse, Additive manufacturing and sustainability: an exploratory study of the advantages and challenges, *Journal of Cleaner Production*, 137 (2016) 1573-1587.

[3] T. Rayna and L. Striukova, From rapid prototyping to home fabrication: how 3D printing is changing business model innovation, *Technological Forecasting and Social Change*, 102 (2016) 214-224.

[4] H. Kim, M. Cha, B. C. Kim, T. Kim and D. Mun, Part library-based information retrieval and inspection framework to support part maintenance using 3D printing technology, *Rapid Prototyping Journal*, 25 (3) (2019) 630-644.

[5] InssTek, *Repairing Jet Engine Part and Extending Life Cycle for Korean Air Force*, http://www.insstek.com/content/customer_success/cs_4, InssTek, Inc., Daejeon (2019).

[6] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine and C. T. Silva, A survey of surface reconstruction from point clouds, *Computer Graphics Forum*, 36 (1) (2017) 301-329.

[7] M. B. Bauza, S. P. Moylan, R. M. Panas, S. C. Burke, H. E. Martz, J. S. Taylor, P. Alexander, R. H. Knebel, R. Bhogaraju, M. T. O'Connell and J. D. Smokovitz, Study of accuracy of parts produced using additive manufacturing, *Proc. of the 2014 Spring Topical Meeting*, 57 (2014) 86-91.

[8] M. Salmi, K. S. Paloheimo, J. Tuomi, J. Wolff and A. Mäkitie, Accuracy of medical models made by additive manufacturing (rapid manufacturing), *Journal of Cranio-Maxillofacial Surgery*, 41 (7) (2013) 603-609.

[9] S. Mahmood, A. J. Qureshi and D. Talamona, Taguchi based process optimization for dimension and tolerance control for fused deposition modelling, *Additive Manufacturing*, 21 (2018) 183-190.

[10] R. Anitha, S. Arunachalam and P. Radhakrishnan, Critical parameters influencing the quality of prototypes in fused deposition modelling, *Journal of Materials Processing Technology*, 118 (1-3) (2001) 385-388.

[11] M. Martorelli, S. Gerbino, M. Giudice and P. Ausiello, A comparison between customized clear and removable orthodontic appliances manufactured using RP and CNC techniques, *Dental Materials*, 29 (2) (2013) e1-e10.

[12] ASTM F2792-12a, *Standard Terminology for Additive Manufacturing Technologies*, ASTM International, West Conshohocken, PA (2012).

[13] J. A. Gonzalez, J. Mireles, Y. Lin and R. B. Wicker, Characterization of ceramic components fabricated using binder jetting additive manufacturing technology, *Ceramics International*, 42 (9) (2016) 10559-10564.

[14] Z. Chen, Z. Li, J. Li, C. Liu, C. Liu, Y. Li and F. Yuelong, 3D printing of ceramics: a review, *Journal of the European Ceramic Society,* 39 (4) (2019) 661-687.

[15] I. Gibson, D. Rosen and B. Stucker, Binder jetting, *Additive Manufacturing Technologies,* Springer, New York, NY (2015) 205-218.

[16] M. Zenou and L. Grainger, Additive manufacturing of metallic materials, *Additive Manufacturing,* Butterworth-Heinemann (2018) 53-103.

[17] H. Miyanaji, S. Zhang, A. Lassell, A. Zandinejad and L. Yang, Process development of porcelain ceramic material with binder jetting process for dental applications, *JOM*, 68 (3) (2016) 831-841.

[18] J. Deckers, J. Vleugels and J. P. Kruth, Additive manufacturing of ceramics: a review, *Journal of Ceramic Science and Technology*, 5 (4) (2014) 245-260.

[19] M. E. Kenney, Cost reduction through the use of additive manufacturing (3D printing) and collaborative product lifecycle management technologies to enhance the Navy's maintenance programs, *Master's Thesis*, Naval Postgraduate School, Monterey, United States (2013).

[20] K. C. Antony and G. W. Goward, Aircraft gas turbine blade and vane repair, *Superalloys*, 745 (1988) 44827638.

[21] J. Huang and C. H. Menq, Automatic CAD model reconstruction from multiple point clouds for reverse engineering, *Journal of Computing and Information Science in Engineering*, 2 (3) (2002) 160-170.

[22] G. Barill, N. G. Dickson, R. Schmidt, D. I. Levin and A. Jacobson, Fast winding numbers for soups and clouds, *ACM Transactions on Graphics (TOG)*, 37 (4) (2018) 43.

[23] J. C. Carr et al., Surface interpolation with radial basis functions for medical imaging, *IEEE Transactions on Medical Imaging*, 16 (1) (1997) 96-107.

[24] J. C. Carr et al., Reconstruction and representation of 3D objects with radial basis functions*, ACM SIGGRAPH 01* (2001) 67-76.

[25] D. T. Lee and B. J. Schachter, Two algorithms for constructing a Delaunay triangulation, *International Journal of Computer and Information Sciences*, 9 (3) (1980) 219-242.

[26] M. Kazhdan, M. Bolitho and H. Hoppe, Poisson surface reconstruction, *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, 7 (2006).

[27] S. H. Lee, H. C. Kim, S. M. Hur and D. Y. Yang, STL file generation from measured point data by segmentation and Delaunay triangulation, *Computer-Aided Design*, 34 (10) (2002) 691-704.

[28] S. Lee, C. I. Park and C. M. Park, An improved parallel algorithm for Delaunay triangulation on distributed memory parallel computers, *Parallel Processing Letters*, 11 (02n03) (2001) 341-352.

[29] D. Zhou, Y. Xu, Q. Zhang and X. Wei, A new triangulation algorithm from 3D unorganized dense point cloud, *TENCON 2015-2015 IEEE Region 10 Conference*, IEEE (2015) 1-6.

[30] M. Kazhdan and H. Hoppe, Screened poisson surface reconstruction, *ACM Transactions on Graphics (ToG)*, 32 (3) (2013) 29.

[31] C. Piya, J. M. Wilson, S. Murugappan, Y. Shin and K. Ramani, Virtual repair: geometric reconstruction for remanufacturing gas turbine blades, *Proc. of the 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (2011) DETC2011-48652.

[32] M. S. Tootooni, A. Dsouza, R. Donovan, P. K. Rao, Z. J. Kong and P. Borgesen, Classifying the dimensional variation in additive manufactured parts from laser-scanned three-dimensional point cloud data using machine learning approaches, *Journal of Manufacturing Science and Engineering*, 139 (9) (2017) 091005.

[33] M. Khanzadeh, P. Rao, R. Jafari-Marandi, B. K. Smith, M. A. Tschopp and L. Bian, Quantifying geometric accuracy with unsupervised machine learning: using self-organizing map on fused filament fabrication additive manufacturing parts, *Journal of Manufacturing Science and Engineering*, 140 (3) (2018) 031011.

[34] H. Kim, M. Cha, B. C. Kim, I. Lee and D. Mun, Maintenance framework for repairing partially damaged parts using 3D printing, *International Journal of Precision Engineering and Manufacturing* (2019) 1-14.

[35] S. W. Sloan, A fast algorithm for generating constrained Delaunay triangulations, *Computers and Structures*, 47 (3) (1993) 441-450.

[36] W. Garage, *PCL*, The Point Cloud Library, https://pointclouds.org/ (2019).

[37] P. J. Besl and N. D. McKay, Method for registration of 3-D shapes, *Sensor Fusion IV: Control Paradigms and Data Structures*, International Society for Optics and Photonics, 1611 (1992) 586-606.

[38] *Geometric Tools*, https://www.geometrictools.com/ (2019).

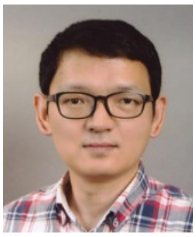
[39] K. Y. Kwon, Design point generation method from a lightweight model for dimensional quality management in shipbuilding, *Journal of Ship Production and Design,* 35 (4) (2019) 353-363.

[40] Kitware, *The Visual Toolkit (VTK)*, https://vtk.org/, Kitware, Inc., Clifton Park (2019).

[41] Spatial, *3D ACIS Modeler*, https://www.spatial.com/products/3d-acis-modeling, Spatial Corporation, Broomfield (2019).

[42] Tech Soft 3D, *HOOPS Visualize*, https://www.techsoft3d.com/products/hoops/visualize/, Tech Soft 3D, Inc., Bend (2019).

Youngki Kim is a Post Doctor of the Korea Institute of Machinery and Materials, Daejeon, South Korea. He received his Ph.D. in Mechanical Engineering from Korea Advanced Institute of Science and Technology. His research interests are in CAD data translation, 3D CAD model reconstruction and 3D printing.



Ki-Youn Kwon is an Assistant Professor at the School of Industrial Engineering, the Kumoh National Institute of Technology. He has received his Ph.D. from KAIST, and a M.S. and B.S. from Korea University. His research interests are computer-aided design, computer-aided manufacturing, mesh generation, digital collaboration, cad model exchange, dimensional quality management. His application domains are all industries related to mechanical engineering such as automobile, shipbuilding and plant.

Duhwan Mun is a Professor at the School of Mechanical Engineering at Korea University. He received a B.S. in Mechanical Engineering from Korea University; an M.S. and Ph.D. in Mechanical Engineering from KAIST. His research interests include CAD, industrial data standards, PLM, knowledge-based engineering, virtual reality for engineering applications, and 3D printing.