Int J Adv Manuf Technol (2014) 75:1393–1403
DOI 10.1007/s00170-014-6210-z

ORIGINAL ARTICLE

Stepwise volume decomposition for the modification of B-rep models

Byung Chul Kim · Duhwan Mun

Received: 2 July 2013 / Accepted: 28 July 2014 / Published online: 22 August 2014

Abstract During product design, the modification of computer-aided design (CAD) models is frequently required. However, modifying a boundary representation (B-rep) model, which is a typical type of CAD model, is difficult and time consuming. To address this problem, we propose a method for modifying B-rep models. In the proposed method, a B-rep model is decomposed into simple volumes by stepwise volume decomposition. A pseudo feature tree called the composition tree is generated with the decomposed simple volumes. A user can modify the B-rep model by transforming or deleting constituents of the composition tree and then replaying the updated composition tree. To demonstrate the proposed method, a prototype system was implemented, and experiments with test cases are performed. Based on the experimental results, we verified that B-rep models could be easily modified using the proposed method.

Keywords B-rep model modification · Cell-based decomposition · Composition tree · Stepwise volume decomposition · Volume split · Wrap-around decomposition

1 Introduction

Nowadays, everything from small parts to large assemblies is designed on a computer-aided design (CAD) system and is represented by 3D CAD models. Moreover, CAD models are used throughout the product life cycle including design, engineering analysis, manufacturing planning, testing, design modification, maintenance, and so on. Specifically, during the design phase, the best design is reached through many iterations [1–4]. During the iterations, exchange of CAD models with design or downstream applications and modifications of CAD models are frequent. However, the fact that design or downstream applications do not use a unique CAD model makes the modifications of CAD modes difficult.

To represent the shape of a CAD model, commercial mechanical CAD systems adopt a hybrid modeling approach that uses both a boundary representation (B-rep) model and a procedural model. In a B-rep model, the shape is explicitly represented by geometric elements such as points, curves, and surfaces and topological elements such as vertices, edges, and faces. B-rep models are used for shape visualization, geometric calculations, and user interactions. Because B-rep models do not contain the modeling history to construct the shape, it is very difficult to modify the shape. On the other hand, the shape in procedural models is represented by the sequence of modeling operations that generates the primitive shapes. Procedural models can be easily modified by changing the arguments or the modeling sequence of the modeling operations [5–7].

For the exchange of CAD models between design or downstream applications, indirect translation methods based on neutral CAD model formats such as the initial graphics exchange specification (IGES) and the standard for the exchange of product model data (STEP) are used. In these methods, a CAD model from the sending CAD system is translated into a neutral CAD model, and then, the translated neutral CAD model is translated into a CAD model for a downstream application. However, most of the neutral CAD models do not provide the information structure needed to represent procedural models. For this reason, after a CAD model is exchanged using neutral CAD models, the

B. C. Kim
Department of Mechanical Engineering, Dong-A University, 37, Nakdong-daero 550 beon-gilSaha-gu Busan 604-714, Republic of Korea
e-mail: mir7942@dau.ac.kr

D. Mun (✉)
Department of Precision Mechanical Engineering, Kyungpook National University, 2559, Gyeongsang-daero, Sangju, Gyeongsangbuk-do 742-711, Republic of Korea
e-mail: dhmun@knu.ac.kr

information in a procedural model becomes lost. Therefore, it is difficult to modify CAD models in downstream applications.

To solve this problem, we propose a method for modifying B-rep models. In the proposed method, a B-rep model is decomposed into simple volumes, and a pseudo feature tree consisting of the decomposed simple volumes is generated. The tree is called the "composition tree." After a user transforms or deletes simple volumes of the composition tree, a modified shape is generated by replaying the updated composition tree. The shape modification using the composition tree is not as easy as that using a procedural model. However, simple modifications such as changing the diameter and depth of holes, and moving the position and orientation of the partial volume, are possible. In particular, the proposed method could be very useful when downstream applications need to modify the exchanged B-rep models.

Rapid volume decomposition of a B-rep model is important for the realization of the proposed method. For that purpose, we developed stepwise volume decomposition. In stepwise volume decomposition, simple volumes are decomposed by sequentially applying cell-based decomposition [19] and other methods. The main characteristics of stepwise volume decomposition are that a B-rep model is decomposed into subtractive volumes as well as additive volumes and that the total decomposition time is reduced.

The remainder of this paper is organized as follows. A discussion of related works is given in Section 2. The complete procedure and details of the proposed method are presented in Sections 3 and 4, respectively. Experiments with a prototype system and suggestions for the enhancement of the proposed method are discussed in Section 5. Our closing remarks and a summary are given in Section 6.

2 Related works

In this section, related works are examined from three points of view: CAD model exchange, volume decomposition, and the modification of B-rep models. This study was motivated by the need to exchange data between CAD models. Volume decomposition is closely related to stepwise volume decomposition. Finally, our goal is the modification of B-rep models.

2.1 Exchange of procedural CAD models

The International Organization for Standardization (ISO) TC184/SC4 published STEP AP203 edition2. This standard specification includes the information structure for representing procedural models in order to overcome the limitation that STEP AP203 edition1 supports only B-rep models [5]. Nonetheless, commercial CAD systems still do not support STEP AP203 edition2. As alternatives for the exchange of procedural CAD models, the macro-parametric approach [8–10] and a method based on universal product representation (UPR) [11] have been proposed. Although these methods enable us to easily modify CAD models in receiving CAD systems, this is possible only when sending CAD system transfer procedural information of a CAD model. For that reason, these methods have a limited range of applications, and they cannot be applied to B-rep models.

2.2 Volume decomposition methods

Volume decomposition methods are used to decompose a complex shape into simple volumes. They are mainly used to recognize machining features [12–14] and are often used in other engineering areas [15, 16]. Volume decomposition is divided into convex decomposition [17, 18] and cell-based decomposition [19, 20].

Convex decomposition recursively decomposes an original shape into its convex hull and the delta volume between the convex hull and the original shape. In this method, an original shape is represented by a constructive solid geometry (CSG) tree which consists of Boolean unions and subtractions of the decomposed volumes. However, convex decomposition cannot be applied to shapes with curved surfaces because it is difficult to define the convex hull of a curved surface.

Cell-based decomposition decomposes an original shape into maximum volumes that do not contain concave edges within them. To find the maximum volumes, an original shape is first decomposed into cells with simple shapes. Then, the cells are combined to find the maximum volumes. In this method, the decomposition of an original shape is represented by Boolean unions of the maximum volumes and can be applied to the shapes with quadric surfaces. However, the amount of time required to find maximum volumes by combining the cells exponentially increases with the number of cells. To overcome this disadvantage, Woo proposed the fast cell-based decomposition approach [21]. Although his method can find the maximum volumes in a relatively short time, it sometimes finds incorrect maximum volumes.

2.3 Modification of B-rep models

A simple and direct method for modifying B-rep models is to replace the surfaces of faces of the model with other surfaces. However, this method is likely to cause errors because related topological elements, such as the adjacent edges and vertices, should be modified when the surfaces of faces are modified. If the topological elements need to be changed to modify B-rep models, applying this method becomes more difficult.

Kim and Han [23] proposed a method for modifying B-rep models by using cell-based decomposition. In their method, a model is decomposed into maximum volumes using cell-based decomposition. The maximum volumes are recognized as design features using feature recognition techniques, and a procedural model including design features is generated by using a macro-parametric approach [8–10]. Therefore, it is possible to modify a shape in a commercial CAD system. However, their method is only applicable to simple shapes since the cell-based decomposition is very slow for complex shapes.

Woo and Lee [22] proposed a method to modify B-rep models by using selective cell-based decomposition. In their method, a model is decomposed into only two parts based on the selected face: the maximum volume that contains the selected face and the other volume. This enhances the decomposition time. After the maximum volume including the selected face is modified, it is reunified with the other volume, and the modified shape is generated. However, negative volumes such as holes cannot be modified in this method because the maximum volumes are only represented with a Boolean union in cell-based decomposition.

Table 1 shows a comparison between previous works and our work. The major difference is that our stepwise volume decomposition is faster even though a whole model is decomposed, and the decomposition makes the modification of negative volumes possible.

3 Method for the modification of B-rep models

The procedure for modifying B-rep models is shown in Fig. 1. First, to modify a shape, it is decomposed into simple volumes (Fig. 1a). For the decomposition, stepwise volume decomposition is used. The volumes are then modified (Fig. 1b). After completing the modification, the decomposed volumes are reunited into one volume (Fig. 1c).

3.1 Stepwise volume decomposition

3.1.1 Volume decomposition procedure

Volumes decomposed by using various methods are shown in Fig. 2. As shown in Fig. 2a, cell-based decomposition decomposes the shape into only the additive volumes. The rectangular hole cannot be decomposed. To modify the dimensions of the hole, the four additive volumes around the hole should be modified. Cell-based decomposition has the disadvantage that it takes considerable time to decompose volumes. Convex decomposition can decompose the additive and negative volumes, as shown in Fig. 2b. However, convex decomposition generates fictitious faces that do not exist in the original shape. Furthermore, convex decomposition cannot be applied to shapes with curved surfaces.

Stepwise volume decomposition decomposes the shape as shown in Fig. 2c. This method has the following distinct characteristics compared to cell-based decomposition and convex decomposition. First, an original shape is decomposed into both additive volumes and subtractive volumes. Second, this method can be applied to shapes with quadratic surfaces. Finally, it is possible to decompose a complex shape within a reasonable amount of time.

Using stepwise volume decomposition, the procedure for decomposing a B-rep model is shown in Fig. 3. The steps of the procedure are explained in detail in Section 4. The procedure is summarized as follows:

- Decompose an original shape by removing fillets, rounds, and chamfers.
- Decompose the volume by applying the wrap-around operation to remove concave spaces such as holes.
- Decompose the volume by applying the volume-split operation to split the volume into several parts.
- Apply cell-based decomposition to the volumes.

The wrap-around operation of stepwise volume decomposition finds and removes concave spaces. Using the wrap-around operation, a shape can be decomposed into subtractive volumes.

Every operation used in stepwise volume decomposition can process quadratic surfaces. For free-form surfaces, the fillet, round, chamfer removal, wrap-around operation, and volume-split operation can be applied, but cell-based decomposition can be applied only to quadratic surfaces. To apply cell-based decomposition to free-form surfaces, the surfaces should be able to be extended. However, it is ambiguous to define extensions for free-form surfaces. It may be possible to extend a free-form surface to the direction tangent to the surface at the boundary, but this does not give meaningful results.

The computation time of cell-based decomposition exponentially increases with the number of cells, and the number of the cells is closely related to the number of concave edges of a shape. Both the wrap-around operation and the volume-split

Table 1 Comparison table between previous works and our work

	Our method	Woo and Lee [22]	Kim and Han [23]
Region of decomposition	Whole model	Selected region	Whole model
Decomposition method	Stepwise volume decomposition	Selective cell-based decomposition	Cell-based decomposition
Decomposition time	Fast	Fast	Slow
Editable volume type	Positive and negative volumes	Positive volumes	Positive volumes

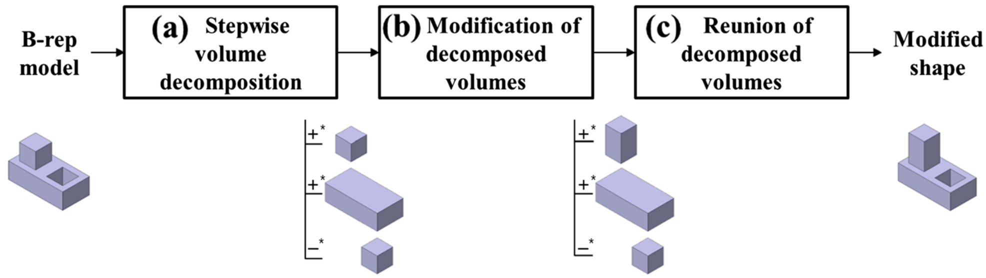

Fig. 1 Procedure for modifying B-rep models

operation remove some of the concave edges; thus, the number of the cells for cell-based decomposition is reduced. As a result, the total volume decomposition time decreases.

In some cases, it is reasonable to exchange the sequence between the wrap-around operation and the volume-split operation. Therefore, we implemented a prototype system to allow a user to change the order of these two operations.

3.1.2 Representation of the volume decomposition results

After stepwise volume decomposition is applied, a composition tree is generated as shown in Fig. 2. The composition tree is the result of the decomposition and has the following characteristics:

- The nodes represent decomposed volumes, and the links represent the type of volume (additive or subtractive).
- The root node represents the original shape.
- The leaf nodes represent the volumes that are not to be decomposed further.
- The volume of a parent node represents the result of Boolean operations of the volumes in the child nodes. If the child node represents an additive volume, the volume is added using a regularized Boolean union. If the child node represents a subtractive volume, the volume is subtracted using regularized Boolean subtraction.
- There is an order among the sibling nodes. That is, the operation of the node that appears first has the highest priority.

When a shape is sequentially decomposed using stepwise volume decomposition, the child nodes are appended at each step. Volume decomposition is completed after the cell-based decomposition is applied. At this time, the leaf nodes represent the final volumes obtained from stepwise volume decomposition.

3.2 Modification and reunion of the volumes

After a shape is decomposed into simple volumes by using stepwise volume decomposition, a composition tree is obtained. A user can modify the decomposed volumes by translating, rotating, scaling, copying, and deleting specific nodes of the composition tree.

Consider the volume decomposition case shown in Fig. 4a. When scaling is applied to volumes ① and ②, the size of some of the shapes can be changed, as shown in Fig. 4b. When rotation or translation is applied to volumes ③ and ④, the position of some of the shapes can be changed, as shown in Fig. 4c. The number of holes can be changed by copying volumes ③ and ④, as shown in Fig. 4d. The shape can also be modified by deleting volumes ③ and ④, as shown in Fig. 4e.

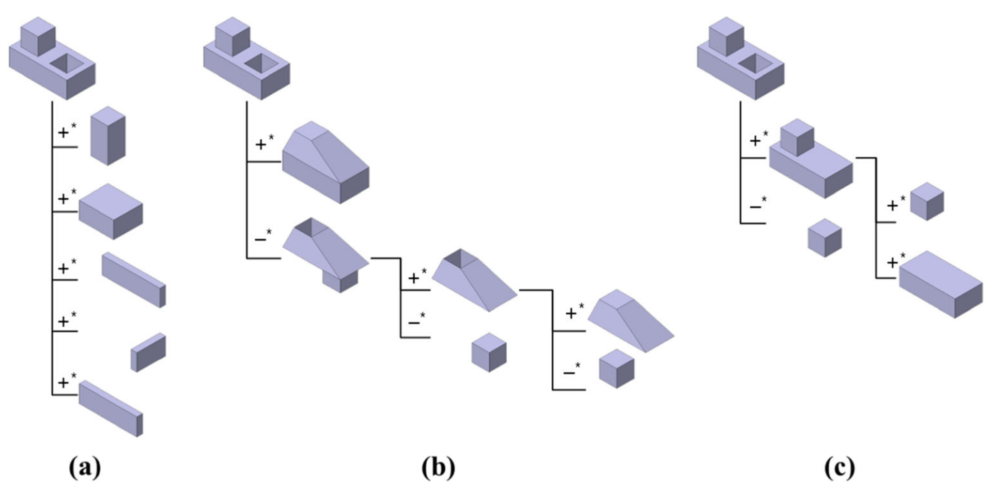

Fig. 2 Comparison of volume decomposition methods: **a** cell-based decomposition, **b** convex decomposition, and **c** stepwise volume decomposition

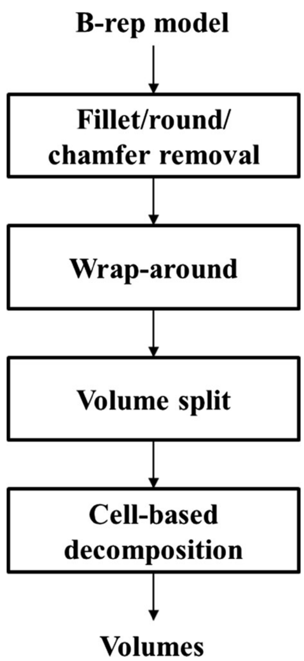

Fig. 3 Stepwise volume decomposition procedure

After modifying the decomposed volumes, every volume of the composition tree is reunited. This generates the modified shape. Because the reunion is performed using only Boolean operations, it is possible to modify the shape without considering the problem arising from topology change.

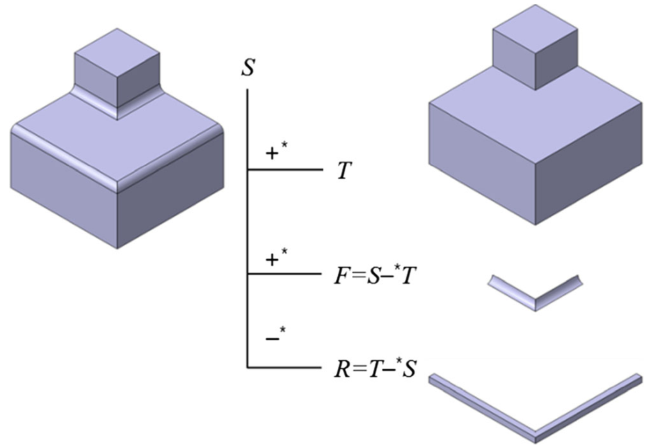

Fig. 5 Round and fillet removal and its composition tree representation

4 Details of stepwise volume decomposition

4.1 Removal of fillets, rounds, and chamfers

Fillets and rounds are features used for smoothing sharp corners and edges, improving handling safety, enhancing strength by reducing stress concentrations, and improving aesthetic appearance. However, fillets and rounds make volume decomposition difficult because they make shapes more complicated [24]. When cell-based decomposition is used, concave edges are used as clues for the decomposition. However, filleted or rounded edges remove concave edges. Consequently, it is important to remove fillets and rounds before the decomposition. For similar reasons, chamfers should be removed.

A representative method to remove fillets and rounds was proposed by Zhu and Menq [24]. However, we used the functions provided by the ACIS geometric modeling kernel.

When fillets are removed from a shape, a volume is removed, and when rounds are removed from a shape, a volume is added. Therefore, the added volume and the removed volume should be represented in the composition tree. Let S be a shape before fillets and

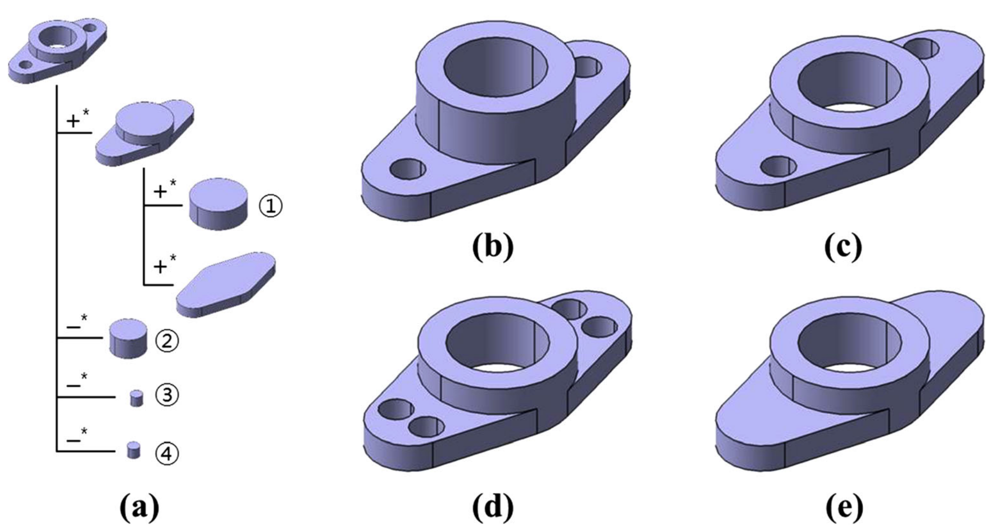

Fig. 4 Modification examples of decomposed shapes: **a** composition tree, **b** scaling of volumes ① and ②, **c** rotation of volumes ③ and ④, **d** copy of volumes ③ and ④, and **e** deletion of volumes ③ and ④

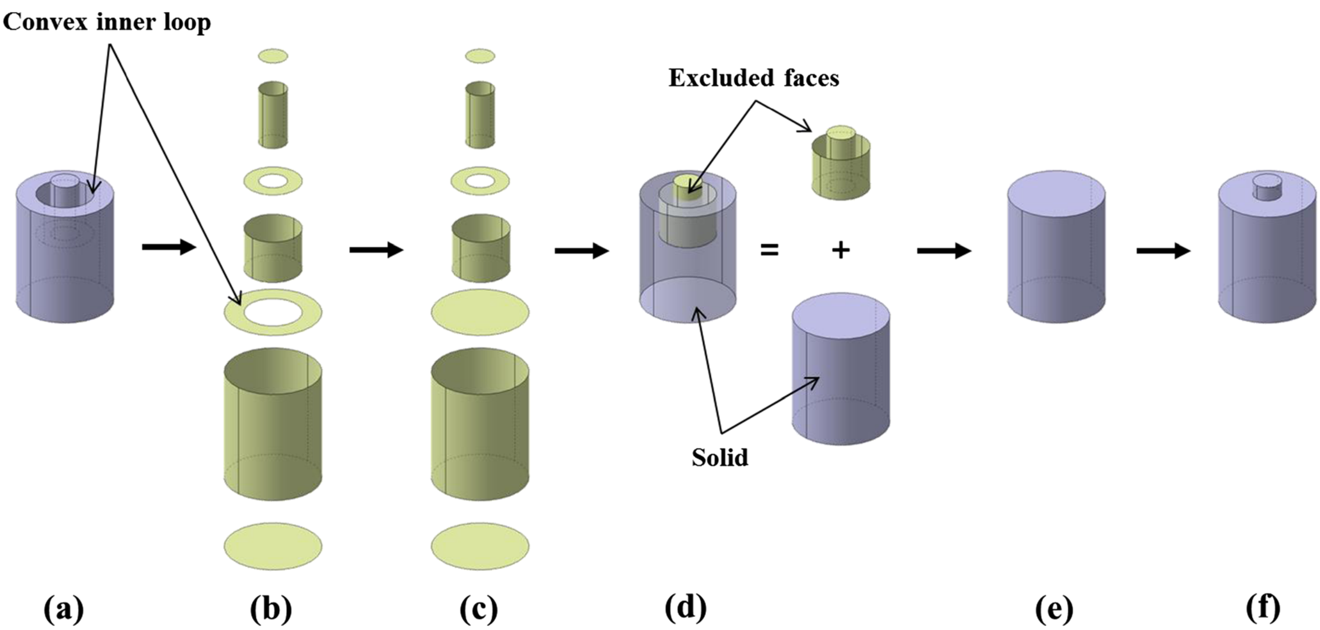

Fig. 6 Concept of wrap-around operation

rounds are removed and T be a shape after fillets and rounds are removed. Then, the added volume F by the fillets and the removed volume R by the rounds are represented as follows:

$$F = S{-}^{*}T$$
$$R = T{-}^{*}S$$

The composition tree of the decomposition of S due to fillet and round removal is represented as shown in Fig. 5.

The ACIS geometric modeling kernel also provides chamfer removal functions. Therefore, chamfers can be removed by a similar method with fillet and round removal.

4.2 Wrap-around operation

The wrap-around operation [25], which is similar to wrapping a shape with a plastic wrap, finds and fills holes or concave spaces in a shape. We used the wrap-around operation to decompose subtractive volumes. Because the wrap-around operation removes concave inner loops, it reduces the number of cells. As a result, the operation reduces the time required for cell-based decomposition. The wrap-around operation plays a role similar to calculating a convex hull in convex decomposition. In contrast to finding the convex hull, the wrap-around operation does not generate fictitious faces.

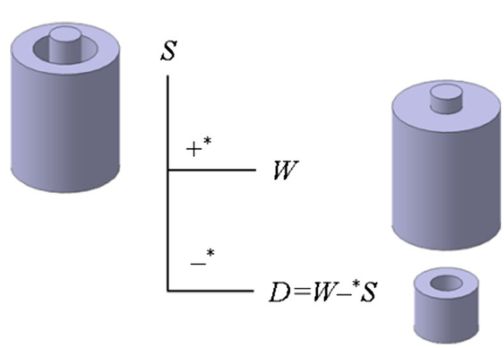

Fig. 7 Composition tree representation of wrap-around operation

The algorithm of the wrap-around operation is shown in Fig. 6 and is summarized as follows:

- Find and mark the convex inner loops in a shape. In Fig. 6a, the inner loop on the top face is a convex inner loop.
- Separate every face. In Fig. 6b, the original shape is separated into seven faces.
- Find and remove the marked inner loops in the separated faces. In Fig. 6c, the inner loop on the top face is removed.
- Sew all the separated faces. At this time, there are faces that cannot constitute solids. In Fig. 6d, there are one solid and four faces after sewing.
- As shown in Fig. 6e, delete the faces that cannot constitute solids.
- As shown in Fig. 6f, unite the sewed solids with the original shape by using a Boolean union.

The wrap-around operation described by Koo and Lee [25] resulted in the solid shown in Fig. 6e. However, we modified the wrap-around operation to produce the solid shown in Fig. 6f by applying a Boolean union to the solid shown in Fig. 6e and the original shape shown in Fig. 6a.

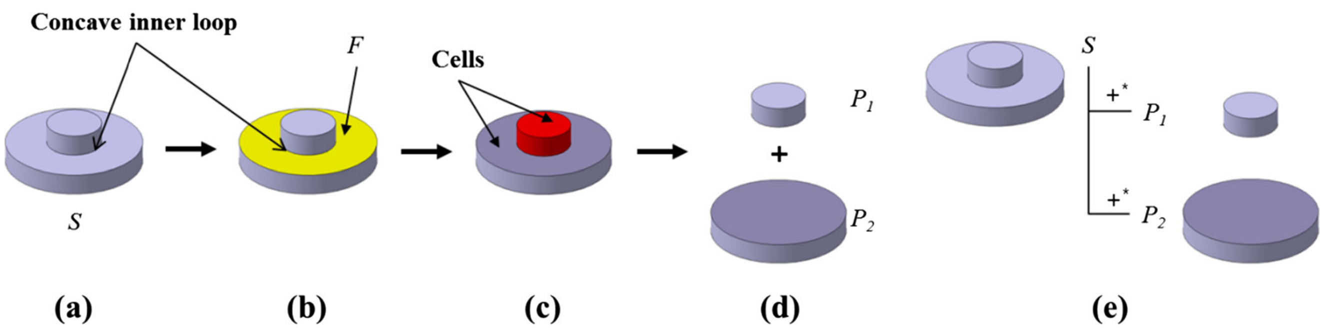

Fig. 8 Volume-split operation and its composition tree representation

Figure 6a and f show the results before and after applying the wrap-around operation, respectively. As shown in Fig. 6f, the concave space shaped like a hole disappeared. As a result, the number of cells generated by the cell-based decomposition was reduced.

Let *S* be a shape before the wrap-around operation is applied and *W* be a shape after the wrap-around operation is applied. Then, *S* can be decomposed into *W* and $W-^{*}S$, as shown in Fig. 7. Sometimes, inner loops remain such as *D* shown in Fig. 7. In these cases, the wrap-around operation is again applied to *D*.

4.3 Volume-split operation

When a shape has a concave inner loop, the shape can be split into two parts based on the loop. In this study, we call this the volume-split operation. It is similar to the cutting operation of the convex decomposition described by Kim [18]. However, the cutting operation cannot be applied to shapes with curved surfaces. The algorithm of the volume-split operation is shown in Fig. 8, and its details are as follows:

- Let *S* be an original shape. As shown in Fig. 8a, find the concave inner loop of *S*.
- Duplicate the face having the concave inner loop. Then, let *F* be the duplicated face.
- Remove the inner loop from *F*.
- Unite *S* and *F* by using a non-regularized Boolean union. Then, *S* has two distinguished cells, as shown in Fig. 8c.
- Separate the cells into the volumes, respectively, as shown in Fig. 8d. Then, the decomposition of *S* can be represented by the composition tree shown in Fig. 8e.

After the volume-split operation is applied, the number of concave edges is reduced. As a result, the number of the cells generated by the cell-based decomposition is also reduced, and application of the volume-split operation reduces the time needed for cell-based decomposition. In addition, face *F* shown in Fig. 8 can be interpreted as a reference face on the base cylinder, which is referred to by the small cylinder. By this interpretation, the volume-split operation decomposes shapes more meaningfully than call-based decomposition.

One reason to use the wrap-around operation and the volume-split operation is to remove concave edges from a shape. Other methods, such as graph-based methods, can be used as an alternative if they can achieve the same goal. However, it is difficult to decompose a shape having intersecting features by using graph-based methods. In contrast, the wrap-around operation and the volume-split operation can decompose shapes having intersecting features.

4.4 Cell-based decomposition

In stepwise volume decomposition, maximal volume decomposition [20] among the cell-based decomposition approaches is used. Maximal volume decomposition

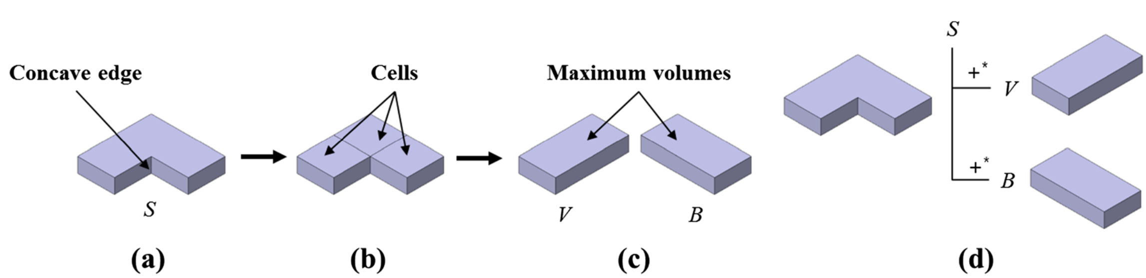

Fig. 9 Maximum volume decomposition and its composition tree representation

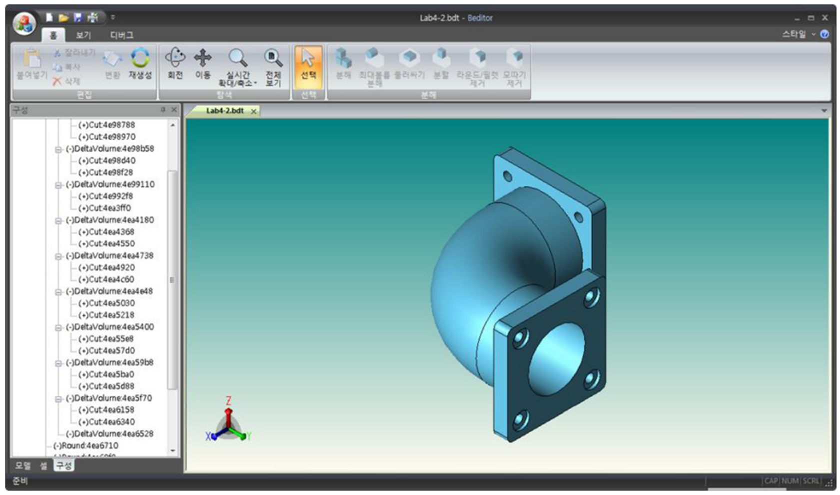

Fig. 10 Prototype system implementation

decomposes shapes into simple volumes called maximal volumes. If a volume V satisfies the following conditions, V is a maximal volume of a solid S [22]:

- $V \subseteq S$.
- V does not have a concave edge.
- Every half-space of V is a half-space of S.
- $B \not\subset V$, where B is a volume that satisfies the above conditions.

To decompose a shape into maximal volumes, every concave edge of the shape is searched, the faces sharing the concave edges are extended, and then, the shape is decomposed into cells. Then, the cells are combined until they satisfy the condition of the maximal volume. The concept of the maximal volume decomposition algorithm is shown in Fig. 9.

5 Implementation and experiments

A prototype system that modifies B-rep models was implemented by using the proposed method, as shown in Fig. 10.

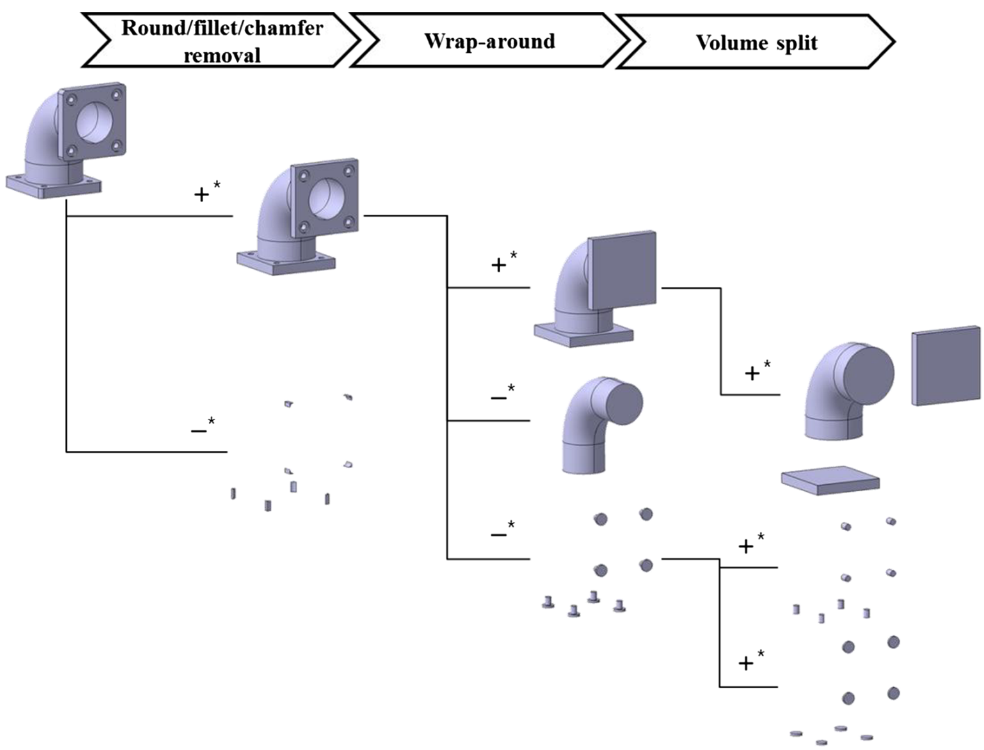

Fig. 11 Stepwise volume decomposition of the elbow part

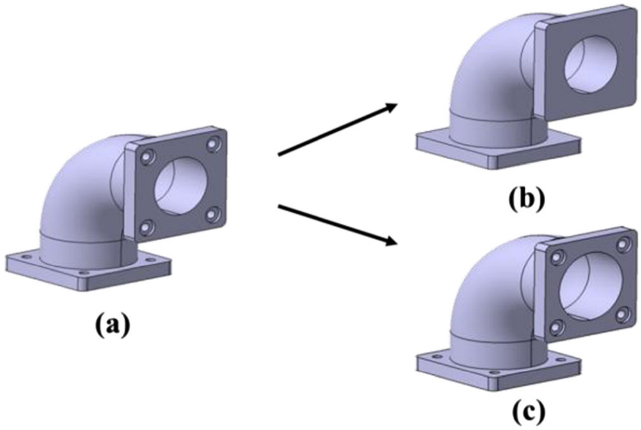

Fig. 12 Modification of the elbow part: **a** the original part, **b** step holes are removed, and **c** the inner diameter of the elbow is changed

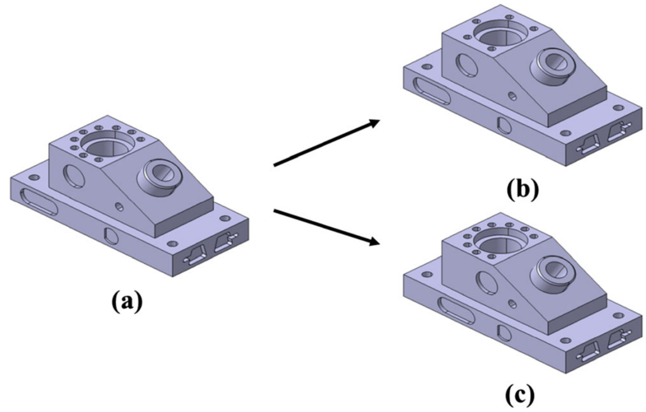

Fig. 14 Modification of the ANC101 part: **a** original part, **b** some holes on the top face are removed, and **c** two pocket features are moved

For the implementation, the C++ language, ACIS geometric modeling kernel, Hoops3D visualization library, and Microsoft foundation classes (MFCs) were used. The composition tree shown on the left-hand side of Fig. 10 shows the result of the stepwise volume decomposition, and the screen shown on the right-hand side of Fig. 10 shows the decomposed volumes.

5.1 Experiments with test cases

Figure 11 shows the result of stepwise volume decomposition for the elbow part. It took 1.81 s to decompose the part on a PC with an Intel Core i7 CPU and 8 GB of RAM. Additive volumes and subtractive volumes occurred together as a result of the volume decomposition. If the cell-based decomposition is applied, such subtractive volumes cannot be decomposed.

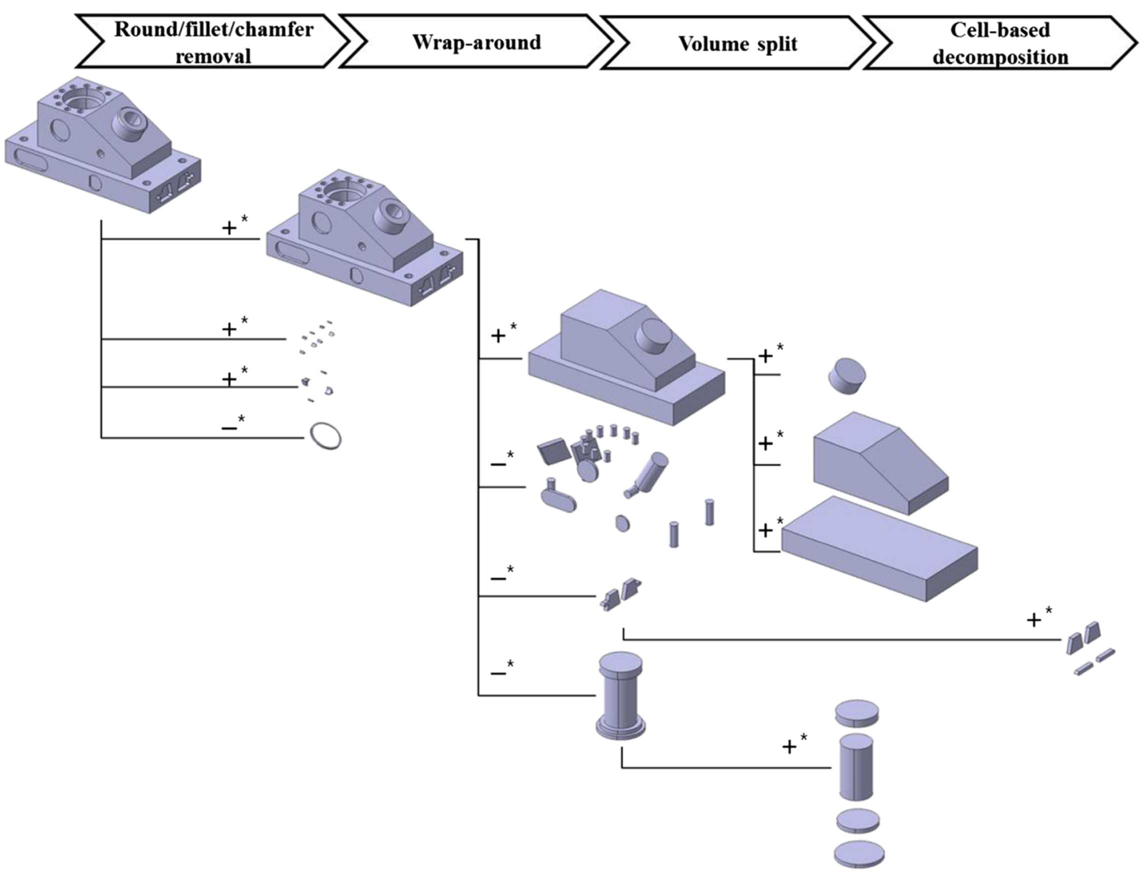

Fig. 13 Stepwise volume decomposition of the ANC101 part

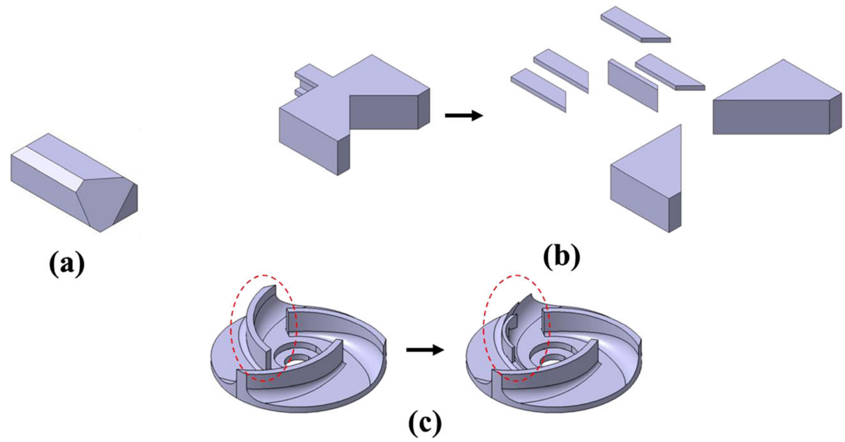

Fig. 15 Limitations of the proposed method

In the example shown in Fig. 11, the volume decomposition was completed at the volume-split step, and cell-based decomposition was not performed.

Figure 12 shows the result of the modification of the elbow part using the decomposed volumes shown in Fig. 11. Figure 12b shows the removal of the step holes in the flange, and Fig. 12c shows a change in the inner radius of the elbow part. Modification of such subtractive volumes is not possible when cell-based decomposition or convex decomposition is used.

Figure 13 shows the result of stepwise volume decomposition of the ANC101 part. 1.78 s was needed to decompose it using the same PC. Additive volumes and subtractive volumes are shown together in Fig. 13.

Figure 14 shows the modification result of the ANC101 part based on the volume decomposition shown in Fig. 13. The removal of some holes on the top face is shown in Fig. 14b. Figure 14c shows two pocket features translating on the side face.

5.2 Suggestions for the enhancement of the proposed method

Stepwise volume decomposition has several limitations. First, because stepwise volume decomposition decomposes a shape into convex volumes, the shape cannot be decomposed further even though the shape is complex, as shown in Fig. 15a. However, both cell-based decomposition and convex decomposition also have this limitation.

Second, if inner loops do not exist in a shape, the wrap-around operation and the volume-split operation cannot be applied. As a result, the stepwise decomposition becomes cell-based decomposition. Consequently, the time needed for volume decomposition increases. Because the decomposed volumes overlap each other, it is difficult to modify the shape. In the case shown in Fig. 15b, the shape is decomposed into seven overlapped volumes, which makes it difficult to modify the shape.

Finally, when the shape next to rounds and fillets is modified, the rounds and fillets should also be changed. However, it is not easy to modify the shape in such a way by using stepwise volume decomposition. In Fig. 15c, one of the blades is modified, but the fillets next to the blade are not modified. This can be partially addressed by using the method proposed by Zhu and Menq [24]. Their method finds and stores rounded or filleted edges and applies the rounds or fillets to the stored edges after modification. However, if the topology changes after modification, this method cannot be applied.

6 Conclusion

We propose a method for modifying B-rep models. This method is based on stepwise volume decomposition. When B-rep models are modified using the proposed method, it is easy to modify the shapes without procedural model information, and it is not necessary to consider the topology changes. Using stepwise volume decomposition, it is possible to modify subtractive volumes, whereas using cell-based decomposition, it is not. In addition, while cell-based decomposition takes considerable time to decompose, stepwise volume decomposition reduces the time needed to decompose the shapes by applying the wrap-around operation and volume-split operation. Convex decomposition cannot be applied to shapes with curved surfaces. However, stepwise volume decomposition can be applied to shapes with quadric surfaces.

Acknowledgments This work was supported by the Dong-A University Research Fund. The authors gratefully acknowledge this support.

References

1. Aziz H, Gao J, Maropoulos P, Cheung WM (2005) Open standard, open source and peer-to-peer tools and methods for collaborative product development. Comput Ind 56(3):260–271
2. Vezzetti E (2009) Product lifecycle data sharing and visualisation: web-based approaches. Int J Adv Manuf Technol 41(5–6): 613–630
3. Alemanni M, Destefanis F, Vezzetti M (2011) Model-based definition design in the product lifecycle management scenario. Int J Adv Manuf Technol 52(1–4):1–14
4. Tran TA, Park JY (2014) Development of integrated design methodology for various types of product–service systems. J Comput Des Eng 1(1):37–47
5. Pratt MJ, Anderson BD, Ranger T (2005) Towards the standardized exchange of parameterized feature-based CAD models. Comput Aided Des 37(12):1251–1265
6. Kim J, Pratt MJ, Iyer RG, Sriram RD (2008) Standardized data exchange of CAD models with design intent. Comput Aided Des 40(7):760–777
7. Cheon S-U, Kim BC, Mun D, Han S (2012) A procedural method to exchange editable 3D data from a free-hand 2D sketch modeling system into 3D mechanical CAD systems. Comput Aided Des 44(2): 154–161
8. Choi G, Mun D, Han S (2002) Exchange of CAD part models based on the macro-parametric approach. Int J CAD/CAM 2(1):13–21
9. Mun D, Han S, Kim J, Oh Y (2003) A set of standard modeling commands for the history-based parametric approach. Comput Aided Des 35(13):1171–1179
10. Kim B, Han S (2007) Integration of history-based parametric translators using the automation APIs. Int J Prod Lifecycle Manag 2(1): 18–29
11. Rappoport A (2003) An architecture for universal CAD data exchange. Proceedings of ACM Symposium on Solid Modeling and Applications 2003, Seattle, pp 266–269
12. Kim YS (1992) Recognition of form features using convex decomposition. Comput Aided Des 24(9):461–476
13. Kailash S, Zhang Y, Fuh J (2001) A volume decomposition approach to machining feature extraction of casting and forging component. Comput Aided Des 33(8):605–617
14. Woo Y, Sakurai H (2002) Recognition of maximal features by volume decomposition. Comput Aided Des 34(3):195–207
15. Litte G, Clark D, Corney J, Tuttle J (1998) Delta-volume decomposition for multi-sided components. Comput Aided Des 30(9):695–705
16. Lu Y, Gadh R, Tautges T (2001) Feature based hex meshing methodology: feature recognition and volume decomposition. Comput Aided Des 33(3):221–232
17. Tang K, Woo T (1991) Algorithmic aspects of alternating sum of volumes. Part 1: data structure and difference operation. Comput Aided Des 23(5):357–366
18. Kim YS, Wilde DJ (1992) A convergent convex decomposition of polyhedral objects. J Mech Des 114(3):468–477
19. Sakurai H (1995) Volume decomposition and feature recognition, part I: polyhedral objects. Comput Aided Des 27(11):833–843
20. Sakurai H, Dave P (1996) Volume decomposition and feature recognition, part II: curved objects. Comput Aided Des 28(6–7):519–537
21. Woo Y (2003) Fast cell-based decomposition and applications to solid modeling. Comput Aided Des 35(11):969–977
22. Woo Y, Lee SH (2006) Volumetric modification of solid CAD models independent of design features. Adv Eng Softw 37(12): 826–835
23. Kim I, Han S (2012) Generation of design history from B-rep models by mapping and sequencing of design features. Adv Sci Lett 10(1): 528–532
24. Zhu H, Menq CH (2001) B-rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition. Comput Aided Des 34(2):109–123
25. Koo S, Lee K (2002) Wrap-around operation to make multi-resolution model of part and assembly. Comput Graph 26(5): 687–700