Springer

KSME

Journal of Mechanical Science and Technology 29 (12) (2015) 5289~5298
www.springerlink.com/content/1738-494x(Print)/1976-3824(Online)
DOI 10.1007/s12206-015-1131-9

Enhanced volume decomposition minimizing overlapping volumes for the recognition of design features †

Byung Chul Kim[1] and Duhwan Mun[2,*]

[1]*Department of Mechanical Engineering, Dong-A University, Busan, Korea*
[2]*Department of Precision Mechanical Engineering, Kyungpook National University, Gyeongsangbuk-do, Korea*

(Manuscript Received February 23, 2015; Revised August 12, 2015; Accepted August 19, 2015)

Abstract

Volume decomposition methods are one type of feature recognition method. They decompose a solid model into simple volumes which are mostly overlapped by each other. However, the overlap of decomposed volumes leads to unnatural results in the recognition of design features. In order to address this problem, we suggest a novel method called non-overlapping volume decomposition in which the overlap of volumes decomposed from a solid model is minimized; the overlap is only allowed when it is desirable from the viewpoint of feature-based 3D modeling practice. After introducing the concept of non-overlapping volume decomposition, we discuss technical issues and their solutions. Non-overlapping volume decomposition was also verified by experiments using a prototype system.

Keywords: Design feature recognition; Maximal volume decomposition; Non-overlapping volume decomposition; Sequential iterative volume decomposition

1. Introduction

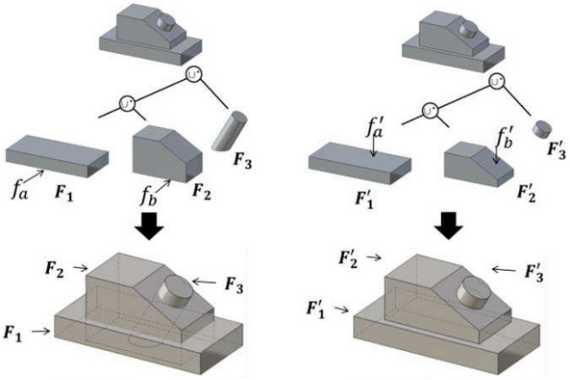
Volume decomposition is a method for decomposing a solid model into simple volumes. Volume decomposition has been applied in feature recognition [1-4], CAD model simplification [5, 6], mesh generation [7], and so on. Among them, feature recognition is the most representative application of volume decomposition. In particular, volume decomposition can recognize intersecting features that cannot be treated by graph-based methods [8-11] or hint-based methods [12, 13]. The literature on feature recognition methods is thoroughly summarized in Refs. [14, 15].

According to the approach used to decompose the volumes, volume decomposition methods can be divided into convex [16, 17] and cell-based [18, 19] decompositions. Convex decomposition methods recursively decompose an original shape into its convex hull and the delta volume between the convex hull and original shape. In these methods, the original shape is represented by a binary tree comprising Boolean unions and subtractions of the decomposed volumes. However, convex decomposition methods cannot be applied to shapes with curved surfaces because it is difficult to define the convex hull of a curved surface. In addition, fictitious faces that do not exist in an original shape could be generated after decomposition. These faces can distort the characteristics of the original shape.

Cell-based decomposition methods decompose an original shape into cells with a simple shape. The cells are then recombined to find larger volumes. Based on the cell recombination method, various volume decomposition methods are possible. In the maximal volume decomposition method, the most representative method among cell-based decomposition methods, the cells are recombined to the maximal volumes that do not contain concave edges. Cell-based volume decomposition methods can be applied to shapes with quadric surfaces and do not generate fictitious faces. However, the time needed to find the maximal volumes by cell recombination increases exponentially with the number of cells. To overcome this disadvantage, Woo proposed a fast cell-based decomposition approach [20]. However, neither method generates good results from the viewpoint of mechanical design because volume decomposition is represented only by Boolean unions.

In all cell-based decomposition methods, decomposed volumes overlap each other. Contrary to machining feature recognition, the overlapping volumes are not natural from the viewpoint of feature-based 3D modeling practice when recognizing design features since designers would usually create design features in such a way that they do not overlap each other. A design feature is typically created by referencing topological entities of an intermediate solid model, and the design feature and the intermediate solid model contact each other at the referenced topological entities. For example, in Fig. 1(b), $F_3^{'}$ is typically created by referring to the face $f_b^{'}$

*Corresponding author. Tel.: +82 54 530 1271, Fax.: +82 54 530 1278
E-mail address: dhmun@knu.ac.kr
† Recommended by Associate Editor Ki-Hoon Shin



(a) (b)

Fig. 1. (a) Maximal volume decomposition; (b) desirable volume decomposition.

of feature $F_2^{'}$. As a result, $F_3^{'}$ contacts with $F_2^{'}$ at the face $f_b^{'}$. However, existing cell-based decomposition methods result in overlapping volumes as shown in Fig. 1(a), and thus design features get recognized in the wrong way. For example, in Fig. 1(a), F_2 is created by referring to the face f_a of feature F_2, which causes F_1 and F_2 to overlap each other. In addition, in Fig. 1(a), F_3 is created by referring to the face f_b of feature F_2, which causes F_2 and F_3 to overlap each other. These are very unnatural from the viewpoint of feature-based 3D modeling practice.

To solve this problem, we suggest a novel method called non-overlapping volume decomposition in which the overlap of volumes decomposed from a solid is minimized; the overlap is only allowed when it is desirable from the view point of feature-based 3D modeling practice. The non-overlapping volume decomposition method decomposes a solid model by sequentially and iteratively applying four volume decomposition methods (fillet-round-chamfer decomposition, wrap-around decomposition, volume split decomposition, and non-overlapping maximal volume decomposition).

The non-overlapping volume decomposition method adopts the sequential iterative volume decomposition procedure proposed in Ref. [21]. However, sequential iterative volume decomposition leads to the overlap of decomposed volumes because this decomposition uses maximal volume decomposition. On the other hand, the non-overlapping volume decomposition method uses non-overlapping maximal volume decomposition instead of maximal volume decomposition.

Non-overlapping maximal volume decomposition is a new volume decomposition method developed in this study. In this decomposition, the cells and maximal volumes of the original shape are first determined using maximal volume decomposition and then overlapping cells are recombined to remove the overlap. However, there are some exceptions where overlapping volumes are preferable. In these cases, the proposed method creates overlapping volumes instead of non-overlapping volumes.

The remainder of this paper is structured as follows. In Sec. 2, the decomposition procedure for generating non-overlapping volumes is explained. In Sec. 3, the technical issues to be considered for non-overlapping maximal volume decomposition are discussed. In Sec. 4, the proposed decomposition method is described. In Sec. 5, the implementation of a prototype system and experimental results with the test cases are presented. The summary and future work are given in Sec. 6.

2. Non-overlapping volume decomposition

2.1 Basic decomposition operations

The basic operations used in the non-overlapping volume decomposition method are fillet- round-chamfer decomposition, wrap-around decomposition, volume split decomposition, and non-overlapping maximal volume decomposition.

Fillets and rounds make volume decomposition difficult. In addition, filleted or rounded edges remove the concave edges that are used as clues for the decomposition. Consequently, it is important to remove fillets and rounds before applying other decomposition methods. For similar reasons, it is necessary to remove chamfers. 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. Fillet-round-chamfer decomposition generates only non-overlapping volumes.

When applying additive features, convex inner loops are often generated at feature intersections. Using such a characteristic, the wrap-around operation finds and fills holes or concave spaces in a shape. The wrap-around operation was first proposed by Koo and Lee [25]. Based on the wrap-around operation, Kim and Mun proposed wrap-around decomposition that is suitable for volume decomposition in Ref. [21]. Wrap-around decomposition generates only non-overlapping volumes.

When applying subtractive features, concave inner loops are often generated at the feature intersections. Volume split decomposition splits a shape into two parts based on these concave inner loops [21]. This decomposition generates only non-overlapping volumes.

Non-overlapping maximal volume decomposition is proposed in this paper. This decomposition recombines the cells and maximal volumes generated by maximal volume decomposition in order to remove the overlap, and it is described in Sects. 3 and 4.

2.2 Non-overlapping volume decomposition procedure

The volume decomposition procedure for generating non-overlapping volumes is shown in Fig. 2. Because rounds, fillets, and chamfers make volume decomposition difficult, the round-fillet-chamfer decomposition method is applied first. Because the non-overlapping maximal volume decomposition can be applied to various shapes, it is applied last to decompose the shapes that could not be decomposed by other methods. In addition, because wrap-around and volume split decompositions can reduce the number of cells, it is better to

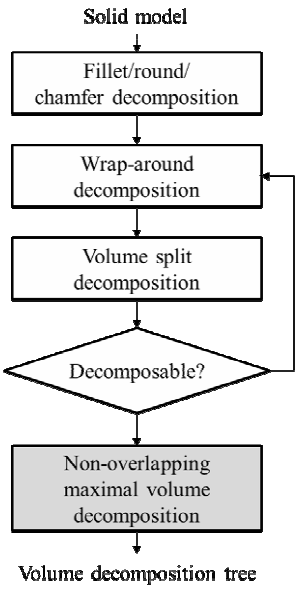


Fig. 2. Non-overlapping volume decomposition procedure.

apply these two decomposition methods before the non-overlapping maximal volume decomposition. The wrap-around and volume split decompositions are applied in between the round-fillet-chamfer decomposition and non-overlapping maximal volume decomposition.

The order of the wrap-around and volume split decompositions affects the quality of the result because they could be incorrectly applied when several convex and concave inner loops exist. That is, the wrap-around decomposition would be incorrectly applied to a concave inner loop and the volume split decomposition would be incorrectly applied to a convex inner loop when a concave inner loop exists inside a convex inner loop or vice versa.

The solution to this problem is 1) to iteratively apply the wrap-around and volume split decompositions and 2) to apply the two decomposition methods to corresponding convex or concave inner loops only [21]. That is, if the wrap-around decomposition is executed, the decomposition of opposite type inner loops (concave inner loops) is delayed to the following step (volume split decomposition).

3. Non-overlapping maximal volume decomposition

3.1 Maximal volume decomposition

Maximal volume decomposition decomposes a solid S into a set of simple volumes called the maximal volume. Volume V is called a maximal volume of S if V satisfies the following conditions [22]:

(1) $V \subseteq S$.

(2) V does not have a concave edge.

(3) Every halfspace of V is a halfspace of S.

(4) $A \not\subset V$, where A is a volume that satisfies the above conditions.

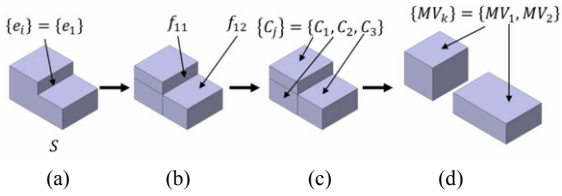


Fig. 3. Maximal volume decomposition.

The algorithm to find the maximal volumes of S is given as follows [21]:

(1) Search all concave edges $\{e_i\}$ of S (Fig. 3(a)).

(2) For each $\{e_i\}$, find two faces f_{i1} and f_{i2} sharing $\{e_i\}$ and extend them to infinity (Fig. 3(b)).

(3) Decompose S into cells $\{C_j\}$ using the non-regularized Boolean union of the extended faces (f_{i1} and f_{i2}) and S (Fig. 3(c)).

(4) Find the maximal volumes $\{MV_k\}$ by combining $\{C_j\}$ until they satisfy the conditions of a maximal volume (Fig. 3(d)).

In Fig. 3(a), the concave edge e_1 is found, and in Fig. 3(b), the faces f_{11} and f_{12} that share it are extended. As a result, three cells C_1, C_2 and C_3 are generated, as shown in Fig. 3(c). By combining C_1, C_2 and C_3, the maximal volumes MV_1 and MV_2 are found, as shown in Fig. 3(d).

3.2 Basic concept of non-overlapping maximal volume decomposition

In feature-based modeling, a design feature is typically created by referencing topological entities such as the faces and edges of an existing intermediate solid model. The design feature and the existing intermediate solid model contact each other at the referenced topological entity; they do not overlap volumetrically. Because of this, for the recognition of design features, maximal volumes must not overlap each other. However, because of the nature of maximal volume decomposition, the maximal volumes overlap each other. For example, MV_1 and MV_2 in Fig. 3(d) overlap with each other because cell C_2 is included in both maximal volumes. The overlap problem is resolved in the proposed method by allocating the common cell to only one maximal volume. Maximal volumes after the common cells have been rearranged are called non-overlapped maximal volumes.

For the solid shape in Fig. 3, there are two possible solutions for the non-overlapping volume decomposition, as shown in Fig. 4. When a designer models the solid shape in Fig. 3(a), a set of design features could be either the set depicted in Fig. 4(a) or Fig. 4(b) according to the designer's intent. From this example, it can be concluded that the result of non-overlapping volume decomposition is not unique. Therefore, maximal volumes allocated to the common cells should be determined by considering the design intent.

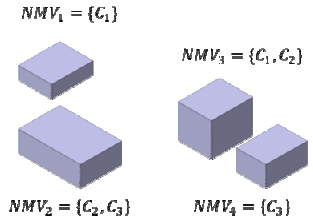


Fig. 4. Non-overlapped maximal volumes [23].

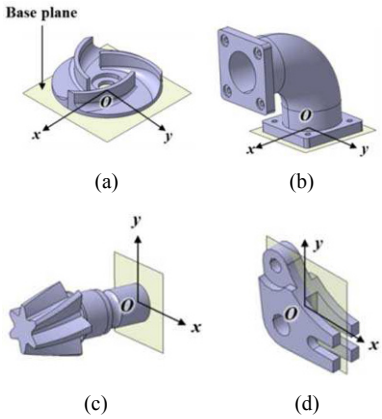


Fig. 5. Base plane examples.

3.3 Determination of the priority of maximal volumes

The priority by which maximal volumes are allocated common cells is decided heuristically. To decide this priority, we assume that a designer sets a base plane for feature-based modeling. The base plane is the plane that is the base for drawing a shape. A designer starts modeling by generating a shape on the base plane and proceeds modeling by adding or removing additional volumes to the shape.

Since the decomposition result in the proposed method depends on the position and orientation of the base plane, the base plane is a key measure for the priority judgement in the proposed heuristic approach. The base plane may be determined by some algorithms. However, designers would have different design intents from each other. It means that any base plane selection algorithm cannot satisfy every designer unless additional semantic and context information is provided in addition to a solid model. Due to this reason, the present study assumes that a designer directly sets a base plane.

For example, in Fig. 5(a), a designer selects a bottom face as a base plane, and in Fig. 5(d), a symmetry plane is selected as a base plane. A base plane is defined by one point on a plane and two perpendicular axes: the origin O, x-axis, and y-axis, respectively. The z-axis is computed by evaluating the cross product of the x- and y-axes. Fig. 5 shows several examples of a base plane.

After a base plane is selected, the priority of the maximal volumes is decided according to the following heuristic rules (in Figs. 6(a)-(e), MV_1 has a higher priority than MV_2):

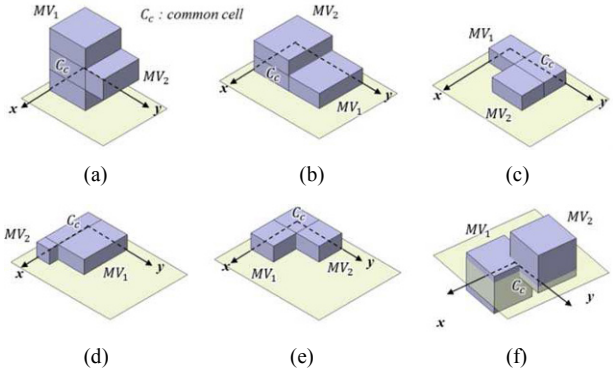


Fig. 6. Heuristics used for deciding the priority of maximal volumes.

(1) The maximal volume nearer to the base plane has a higher priority (Fig. 6(a)).

(2) The maximal volume for which a farthest point from the base plane along the z-axis is nearer to the base plane has a higher priority (Fig. 6(b)).

(3) The maximal volume nearer to the origin of the base plane has a higher priority (Fig. 6(c)).

(4) The maximal volume with the larger volume has a higher priority (Fig. 6(d)).

(5) The longer maximal volume along the x-axis has a higher priority (Fig. 6(e)).

(6) The longer maximal volume along the y-axis has a higher priority.

A volume has a higher priority when it is nearer to the base plane, broader, and flatter because a designer generally draws a broader and larger volume first. If the priority cannot be decided from the above criteria, the priority is arbitrary; this case occurs when a shape is origin symmetric, as shown in Fig. 6(f).

After the priority of the maximal volumes sharing a common cell is determined, the maximal volume with the highest priority is allocated the common cell.

3.4 Exceptional cases where overlapping volumes are preferable

The overlapping volumes are not always unnatural from the viewpoint of feature-based 3D modeling practice. There are some exceptions where overlapping volumes are preferable. They are categorized into three cases as follows; in these cases, overlapping volumes are created instead of non-overlapping volumes.

- Generation of concave edges: MV_2 becomes concave when MV_1 is allocated the common cell C_c in Fig. 7(b). If MV_2 owns C_c, the situation is reversed, that is, MV_1 becomes concave. In this case, both MV_1 and MV_2 should own C_c, as shown in Fig. 8(b).
- Increase in the number of maximal volumes: the number of the maximal volumes is two in Fig. 7(a), but the number of the non-overlapping maximal volumes becomes three. This is because MV_2 is split into two volumes

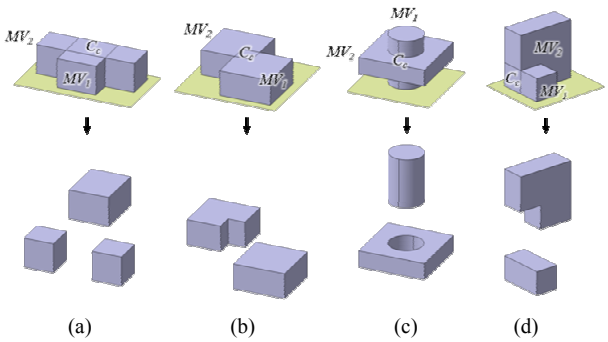


Fig. 7. Exceptional cases of the heuristic rules.

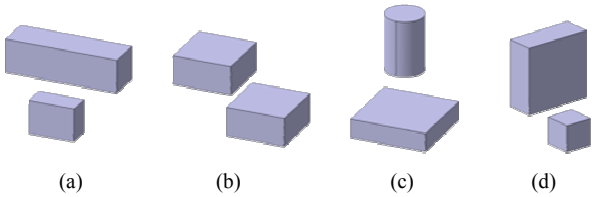


Fig. 8. Corrected non-overlapped maximal volumes for the exceptional cases.

when common cell C_c is owned by MV_1. In this case, MV_2 should own C_c, as shown in Fig. 8(a).

- Increase in the number of holes: the number of holes in MV_2 increases when MV_2 is allocated the common cell C_c in Fig. 7(c). In this case, generally MV_2 should own C_c. However, MV_1 is split into two volumes if MV_2 is allocated C_c. Therefore, both MV_1 and MV_2 should own C_c, as shown in Fig. 8(c).

The reason why overlapping volumes are allowed in the exceptional cases is not that the proposed method cannot decompose a solid model into non-overlapping volumes in the cases but that overlapping volumes are natural from the viewpoint of feature-based 3D modeling practice in the cases.

Designers would have different design intents from each other. Thus, unless additional semantic and context information is provided in addition to a solid model, it is impossible to decompose a solid model into simple volumes that satisfy every designer. Due to this reason, the heuristic approach is adopted in the proposed method and the non-overlapping volume decomposition was developed such that the overlap of volumes decomposed from a solid is minimized and the overlap is only allowed in the three exceptional cases.

4. Algorithm for finding non-overlapped maximal volumes

Given a solid S and a base plane BP, the non-overlapped maximal volumes are found as follows:

(1) The maximal volume decomposition is performed first for the solid S. Using the maximal volume decomposition, the cells $\boldsymbol{C}=\{C_0,\ldots,C_{n-1}\}$ and maximal volumes $\boldsymbol{MV}=\{MV_0,\ldots,MV_{m-1}\}$ of S are found such that each $MV_j \in \boldsymbol{MV}$ consists of a subset of $\boldsymbol{C}$.

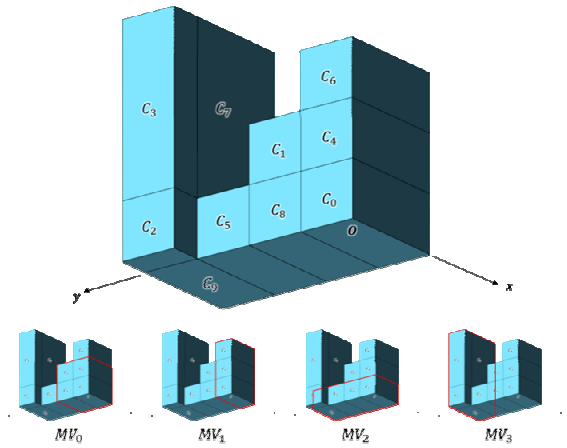


Fig. 9. Example model for non-overlapping maximal volume decomposition.

(2) For each cell $C_i \in \boldsymbol{C}(i=0,\ldots,n-1)$, all maximal volumes containing C_i are found. These maximal volumes are denoted by CMV_i, and an ordered set consisting of CMV_i is denoted by $\boldsymbol{CMV}$.

(3) The elements of $\boldsymbol{CMV}$ are sorted in ascending order by $n(CMV_i)$.

(4) For each $CMV_i \in \boldsymbol{CMV}(i=0,\ldots,n-1)$, the elements of CMV_i are sorted according to the heuristic rules of Sec. 3.3. This step determines the priority of the maximal volumes containing C_i. After the sorting, maximal volumes with higher priority come first.

(5) For each $CMV_i \in \boldsymbol{CMV}(i=0,\ldots,n-1)$, the cells to be processed together with cell C_i are found and denoted by GC_i. When two maximal volumes share more than two cells, the shared cells are processed together.

(6) For each cell $C_i \in \boldsymbol{C}(i=0,\ldots,n-1)$, the following steps are carried out if $n(CMV_i)>1$.

(a) Using the elements of CMV_i, every combination of the maximal volumes containing C_i are determined. In this case, the priority order of the combinations follows the heuristic rules of Sec. 3.3.

(b) Among these combinations, combinations are found that have the highest priority and where the exceptional cases in Sec. 3.4 do not occur. The maximal volumes included in these combinations own C_i.

In the above algorithm, the result of maximal volume decomposition in Step 1 is unique by the definition of the maximal volume in Sec. 2. In Steps 2-6, sorting by priority based on the heuristics ensures the result of the algorithm is unique. Therefore, the final result of the algorithm is unique.

To demonstrate the algorithm in detail, we apply it to the solid in Fig. 9.

Step 1: The cells and maximal volumes of the solid in Fig. 9 are as follows:

$$\boldsymbol{C}=\{C_0,C_1,C_2,C_3,C_4,C_5,C_6,C_7,C_8,C_9\}$$

$$\boldsymbol{MV} = \{MV_0, MV_1, MV_2, MV_3\}$$
$$MV_0 = \{C_0, C_1, C_4, C_8\}$$
$$MV_1 = \{C_0, C_4, C_6\}$$
$$MV_2 = \{C_0, C_5, C_8, C_9\}$$
$$MV_3 = \{C_2, C_3, C_7, C_9\}.$$

Step 2: For each cell, $\boldsymbol{CMV}$ is as follows:

$$\boldsymbol{CMV} = \{CMV_0, CMV_1, \ldots, CMV_9\}$$
$$CMV_0 = \{MV_0, MV_1, MV_2\}$$
$$CMV_1 = \{MV_0\}$$
$$CMV_2 = \{MV_3\}$$
$$CMV_3 = \{MV_3\}$$
$$CMV_4 = \{MV_0, MV_1\}$$
$$CMV_5 = \{MV_2\}$$
$$CMV_6 = \{MV_1\}$$
$$CMV_7 = \{MV_3\}$$
$$CMV_8 = \{MV_0, MV_2\}$$
$$CMV_9 = \{MV_2, MV_3\}$$

Step 3: After sorting, $\boldsymbol{CMV}$ becomes:

$$\boldsymbol{CMV} = \begin{Bmatrix} CMV_1, CMV_2, CMV_3, CMV_5, CMV_6, CMV_7, CMV_4, \\ CMV_8, CMV_9, CMV_0 \end{Bmatrix}$$

Step 4: For each $CMV_i (i = 0, \ldots, 9)$, the priority of the maximal volumes are evaluated. The base plane is selected as the plane defined by the x-axis, y-axis, and origin O, as shown in Fig. 9. According to the heuristic rules in Sec. 3.1, the priorities are as follows:

$$MV_2 > MV_3 > MV_0 > MV_1.$$

Therefore, the maximal volumes in each CMV_i are sorted as follows:

$$CMV_1 = \{MV_0\}$$
$$CMV_2 = \{MV_3\}$$
$$CMV_3 = \{MV_3\}$$
$$CMV_5 = \{MV_2\}$$
$$CMV_6 = \{MV_1\}$$
$$CMV_7 = \{MV_3\}$$
$$CMV_4 = \{MV_0, MV_1\}$$
$$CMV_8 = \{MV_2, MV_0\}$$
$$CMV_9 = \{MV_2, MV_3\}$$
$$CMV_0 = \{MV_2, MV_0, MV_1\}.$$

Step 5: For each $CMV_i (i = 0, \ldots, 9)$, the cells to be processed together are identified. As shown in Fig. 9, when processing MV_0 and MV_1, C_0 and C_4 should be processed together. In addition, when processing MV_0 and MV_2, C_0 and C_8 should be processed together. Therefore, GC_4 and GC_8 are defined as follows.

$$GC_4 = \{C_0\}$$
$$GC_8 = \{C_0\}.$$

Step 6: For each cell $C_i (i = 0, \ldots)$, the maximal volume that will own C_i is selected. Because CMV_1, CMV_2, CMV_3, CMV_5, CMV_6 and CMV_7 have one maximal volume, C_1 is allocated to MV_0, C_2 is allocated to MV_3, C_3 is allocated to MV_3, C_5 is allocated to MV_2, C_6 is allocated to MV_1, and C_7 is allocated to MV_3. At this phase, each maximal volume includes the following cells:

$$MV_0 = \{C_0, C_1, C_4, C_8\}$$
$$MV_1 = \{C_0, C_4, C_6\}$$
$$MV_2 = \{C_0, C_5, C_8, C_9\}$$
$$MV_3 = \{C_2, C_3, C_7, C_9\}.$$

At this stage, CMV_4, CMV_8, CMV_9 and CMV_0 have more than one maximal volume. Therefore, they need further investigation. For CMV_4, C_4 can be included in MV_0, MV_1, or both. In addition, C_0 should be processed together with C_4. The inclusion of C_4 in MV_0 does not incur any of the exceptional cases in Sec. 3.4. Therefore, C_0 and C_4 are allocated to MV_0. At this phase, each maximal volume includes the following cells.

$$MV_0 = \{C_0, C_1, C_4, C_8\}$$
$$MV_1 = \{C_6\}$$
$$MV_2 = \{C_0, C_5, C_8, C_9\}$$
$$MV_3 = \{C_2, C_3, C_7, C_9\}.$$

For CMV_8, C_8 can be included in MV_2, MV_0, or both. In addition, C_0 should be processed together with C_8. Inclusion of C_8 in MV_2 does not incur any of the exceptional cases in Sec 3.4. Therefore, C_0 and C_8 are allocated to MV_2. At this phase, each maximal volume includes the following cells:

$$MV_0 = \{C_1, C_4\}$$
$$MV_1 = \{C_6\}$$
$$MV_2 = \{C_0, C_5, C_8, C_9\}$$
$$MV_3 = \{C_2, C_3, C_7, C_9\}.$$

For CMV_9, C_9 can be included in MV_2, MV_3, or both. If MV_2 owns C_9, MV_3 becomes concave. Therefore, MV_2 should not include C_9 even if it has a higher priority than

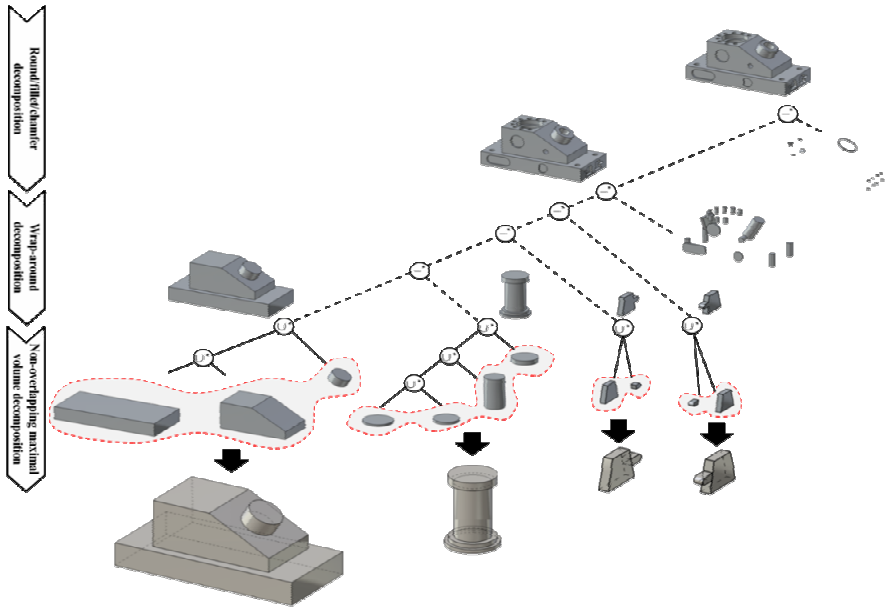


Fig. 10. Non-overlapping maximal volume decomposition of ANC101 part.

MV_3. If MV_3 includes C_9, no exceptional cases occur. Therefore, MV_3 becomes the owner of C_9. At this phase, each maximal volume includes the following cells:

$$MV_0 = \{C_1, C_4\}$$
$$MV_1 = \{C_6\}$$
$$MV_2 = \{C_0, C_5, C_8\}$$
$$MV_3 = \{C_2, C_3, C_7, C_9\}.$$

For CMV_0, C_0 has already been processed together with C_8. At this phase, each maximal volume includes the same cells at the previous phase.

$$MV_0 = \{C_1, C_4\}$$
$$MV_1 = \{C_6\}$$
$$MV_2 = \{C_0, C_5, C_8\}$$
$$MV_3 = \{C_2, C_3, C_7, C_9\}.$$

All CMV_i have been investigated. Therefore, the non-overlapping maximal volumes are determined as follows:

$$NMV_0 = \{C_1, C_4\}$$
$$NMV_1 = \{C_6\}$$
$$NMV_2 = \{C_0, C_5, C_8\}$$
$$NMV_3 = \{C_2, C_3, C_7, C_9\}.$$

5. Implementation and experiments

A prototype system was implemented in the C++ language with the ACIS geometric modeling kernel, Hoops3D visualization library, and Microsoft foundation classes (MFC). Our prototype system supports ISO 10303 STEP (standard for the exchange of product model data) and IGES (initial graphics exchange specification) formats. Input files in the experiments were modelled in CATIA V5 and exported into the STEP format. The prototype system imports input files in a STEP format and sequentially performs fillet-round-chamfer decomposition, wrap-around decomposition, volume split decomposition, and non-overlapping maximal volume decomposition. Non-overlapping maximal volume decomposition is applied to the maximal volumes that have been decomposed by maximal volume decomposition.

Fig. 10 shows non-overlapping volume decomposition of an ANC10 part. In this case, the bottom face of the part was used as the base plane and non-overlapping maximal volume decomposition is depicted as a solid line. As can be seen in this figure, overlapping volumes do not exist in the volume decomposition tree.

For comparison, the same part has also been decomposed into its maximal volume decomposition. The result is depicted in Fig. 11 as a solid line. As shown in Fig. 11, the volumes decomposed by maximum volume decomposition overlap. In contrast, the same volumes in Fig. 10 do not overlap.

Fig. 12 shows non-overlapping volume decomposition of a blade part. As shown in this figure, overlapping volumes do not exist in the volume decomposition tree.

For comparison, the part has also been decomposed into its maximal volume decomposition. The result is depicted in Fig. 13. The decomposed blades in Fig. 13 are axially longer (in the vertical direction) than the decomposed blades in Fig. 12. This is because the decomposed blades in Fig. 13 overlap with the circular bottom plate whereas the decomposed blades in Fig. 12 do not.

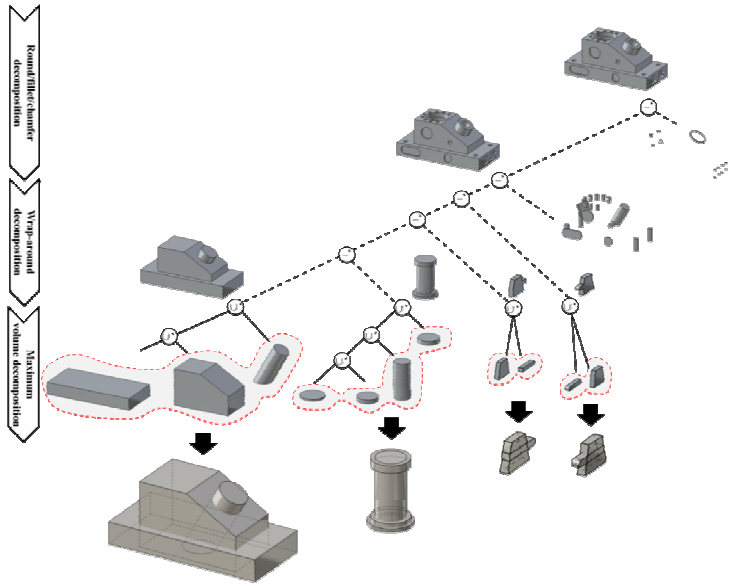


Fig. 11. Maximal volume decomposition of ANC101 part.

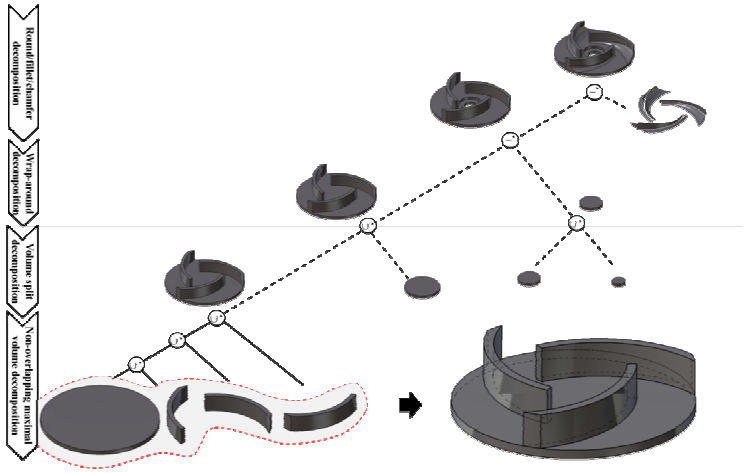


Fig. 12. Non-overlapping maximal volume decomposition of a blade part.

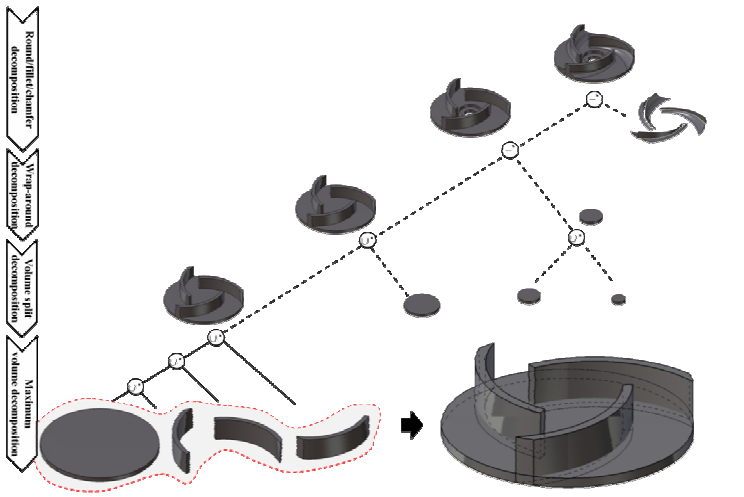


Fig. 13. Maximal volume decomposition of a blade part.

Table 1. Decomposition time comparison.

	Maximal volume decomposition (a)	Non-overlapping volume decomposition (b)	Net time ((b)-(a))
ANC101	436 *ms*	458 *ms*	22 *ms*
Blade	931 *ms*	1026 *ms*	95 *ms*

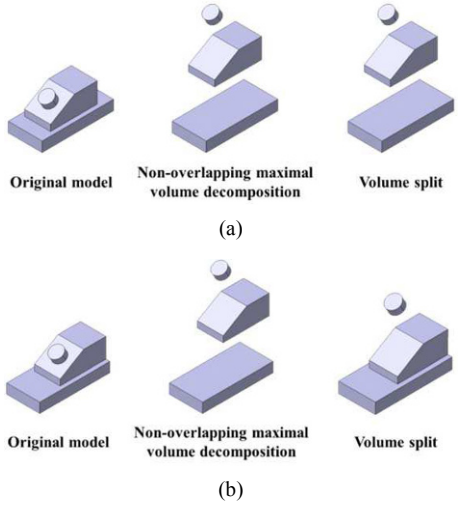


Fig. 14. Comparison of non-overlapping maximal volume and volume split decompositions.

Table 1 compares the calculation times of non-overlapping maximal volume and maximal volume decompositions. The time was measured on a PC with Intel Core i7 CPU and 8 GB RAM. Because the non-overlapping maximal volume decomposition is performed after the maximal volume decomposition, the difference between the two methods in Table 1 is the net time for the non-overlapping maximal volume decomposition. The net time is significantly smaller than the time for the maximal volume decomposition.

Fig. 14 shows the difference between the non-overlapping maximal volume and volume split decompositions. The volume split decomposition method decomposes a solid into non-overlapping volumes, but can be applied only to a solid with a concave inner loop.

6. Conclusion

The non-overlapping volume decomposition method was proposed for decomposing a solid into non-overlapping volumes for the recognition of design features. This method decomposes a solid model by sequentially and iteratively applying four volume decomposition methods (fillet-round-chamfer decomposition, wrap-around decomposition, volume split decomposition, and non-overlapping maximal volume decomposition). It was developed in this study in order to solve the limitations of maximal volume decomposition.

The non-overlapping maximal volume decomposition is based on the maximum volume decomposition. Due to this reason, the proposed method has the same limitations as the maximum volume decomposition; the amount of time required to find maximum volumes by combining the cells exponentially increases with the number of cells. To alleviate the limitation, fillet-round-chamfer decomposition, wrap-around decomposition, and volume split decomposition were performed in advance in the proposed decomposition method since these decompositions significantly reduce the number of cells.

A detailed algorithm for finding the non-overlapping maximal volumes was also presented. The non-overlapping volume decomposition method was verified by experiments with a prototype system. From the experiments, it was found that non-overlapping volume decomposition generates better decompositions than maximal volume decomposition from the viewpoint of design feature recognition.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014R1A1A1006181) and by Plant Research Program funded by the Ministry of Land, Infrastructure and Transport (14IFIP-B091004-01).

References

[1] Y. S. Kim, Recognition of form features using convex decomposition, *Computer-Aided Design*, 24 (9) (1992) 461-476.
[2] S. Kailash, Y. Zhang and J. Fuh, A Volume decomposition approach to machining feature extraction of casting and forging component, *Computer-Aided Design*, 33 (8) (2001) 605-617.
[3] Y. S. Kim and E. Wang, Recognition of machining features for cast then machined parts, *Computer-Aided Design*, 34 (1) (2002) 71-87.
[4] Y. Woo and H. Sakurai, Recognition of maximal features by volume decomposition, *Computer-Aided Design*, 34 (3) (2002) 195-207.
[5] S. Kwon, B. C. Kim, D. Mun and S. Han, Simplification of feature-based 3D CAD assembly data of ship and offshore plant equipment using quantitative metrics, *Computer-Aided Design*, 59 (2015) 140-154.
[6] Y. Woo, Automatic simplification of solid models for engineering analysis independent of modeling sequences, *Journal of Mechanical Science and Technology*, 23 (7) (2009) 1939-1948.
[7] Y. Lu, R. Gadh and T. Tautges, Feature based hex meshing methodology: feature recognition and volume decomposition, *Computer-Aided Design*, 33 (3) (2001) 221-232.
[8] S. Joshi and T. C. Chang, Graph based heuristics for recognition of machined features from a 3-d solid model, *Computer-Aided Design*, 20 (2) (1988) 58-66.
[9] S. H. Chuang and M. R. Henderson, Three-dimensional

shape pattern recognition using vertex classification and the vertex-edge graph, *Computer-Aided Design*, 22 (6) (1990) 377-387.
[10] P. Gavankar and M. R. Henderson, Graph-based extraction of protrusions and depressions from boundary representations, *Computer-Aided Design*, 22 (7) (1990) 442-450.
[11] S. Gao and J. Shah, Automatic recognition of interacting machining features based on minimal condition subgraph, *Computer-Aided Design*, 30 (9) (1998) 727-739.
[12] J. H. Vadenbrande and A. A. G. Requicha, Spatial reasoning for the automatic recognition of machinable features in solid models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15 (12) (1993) 1269-1285.
[13] W. C. Regli, S. K. Gupta and D. S. Nau, Extracting alternative machining features: an algorithmic approach, *Research in Engineering Design*, 7 (3) (1995) 173-192.
[14] J. H. Han, M. Pratt and W. C. Regli, Manufacturing feature recognition from solid models: a status report, *IEEE Transactions on Robotics and Automation*, 16 (6) (2000) 782-796.
[15] J. J. Shah, D. Anderson, Y. S. Kim and S. Joshi, A discourse on geometric feature recognition from CAD models, *Journal of Computing and Information Science in Engineering*, 1 (1) (2001) 41-51.
[16] K. Tang and T. Woo, Algorithmic aspects of alternating sum of volumes. part 1: data structure and difference operation, *Computer-Aided Design*, 23 (5) (1991) 357-366.
[17] Y. S. Kim and D. J. Wilde, A convergent convex decomposition of polyhedral objects, *Journal of Mechanical Design*, 114 (3) (1992) 468-477.
[18] H. Sakurai, Volume decomposition and feature recognition, part i: polyhedral objects, *Computer-Aided Design*, 27 (11) (1995) 833-843.
[19] H. Sakurai and P. Dave, Volume decomposition and feature recognition, part ii: curved objects, *Computer-Aided Design*, 28 (6-7) (1996) 519-537.
[20] Y. Woo, Fast cell-based decomposition and applications to solid modeling, *Computer-Aided Design*, 35 (11) (2003) 969-977.
[21] B. C. Kim and D. Mun, Feature-based simplification of boundary representation models using sequential iterative volume decomposition, *Computers & Graphics*, 38 (2014) 97-107.
[22] Y. Woo and S. H. Lee, Volumetric modification of solid cad models independent of design features, *Advances in Engineering Software*, 37 (12) (2006) 826-835.
[23] B. C. Kim and D. Mun, Non-overlapping volume decomposition using maximum volumes (in Korean), *Transactions of the Society of CAD/CAM Engineers*, 19 (1) (2014) 50-60.
[24] H. Zhu and C. H. Menq, B-Rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition, *Computer-Aided Design*, 34 (2) (2002) 109-123.
[25] S. Koo and K. Lee, Wrap-around operation to make multi-resolution model of part and assembly, *Computers & Graphics*, 26 (5) (2002) 687-700.

Byung Chul Kim is an Assistant Professor at the Department of Mechanical Engineering at Dong-A University, Korea. He received his PhD and MS in Mechanical Engineering from KAIST, and a BS in Mechanical Engineering from Korea University. Before joining Dong-A University, he was a senior research engineer at Samsung Heavy Industries. His research interests are in the areas of Geometric Modeling, Plant IT and Bio-CAD.

Duhwan Mun is an Associate Professor at the Department of Precision Mechanical Engineering of Kyungpook National University. Prior to that, he was a Senior Research Fellow at the Maritime & Ocean Engineering Research Institute (MOERI), a branch of Korea Ocean Research & Development Institute (KORDI). His research interests include computer-aided design, industrial data standards for product data exchange, product lifecycle management, knowledge-based engineering, and virtual reality for engineering applications. He received a BS in Mechanical Engineering from Korea University; a MS and PhD in Mechanical Engineering from Korea Advanced Institute of Science and Technology (KAIST).